

**פרויקט סיום - קורס למידה عمוקה**

**גילוי דלקת ריאות בצלומי רנטגן**

שם הסטודנט, תעודה זהות: סער גוזלן 204188403 ,

שם הסטודנט, תעודה זהות: אורן דנילוב 203824545 ,

שם הסטודנט, תעודה זהות: שמואל איליב, 318450293

מרצה: ד"ר אמר אדר

תאריך הגשה: 31.07.2022

# Contents

- 2 -	.....	Contents
- 4 -	.....	מבוא
- 4 -	.....	מה זה דלקת ריאות?
- 4 -	.....	מטרת הפרויקט
- 5 -	.....	משימה 1 – תכנון רשותות عمוקות
- 5 -	.....	דגשים כליליים על הקוד
- 5 -	.....	הכנות ה- dataset
- 6 -	.....	מבנה רשות א':
- 6 -	.....	הסביר על שכבות הרשות:
- 8 -	.....	מבנה רשות ב':
- 12 -	.....	משימה 2 – אימון הרשותות והערכת ביצועים
- 12 -	.....	עבור רשות א':
- 12 -	.....	2.1. גרפ התוכניות של תהליך האימון
- 12 -	.....	2.2+2.3 F-SCROE Precision Recall ו חישוב
- 12 -	.....	עבור רשות ב':
- 12 -	.....	2.1. גרפ התוכניות של תהליך האימון
- 13 -	.....	2.2+2.3 F-SCROE Precision Recall ו חישוב
- 15 -	.....	משימה 3.1
- 15 -	.....	סעיף א' – רשות א' – הוספה שכבה
- 16 -	.....	מסקנה לגבי הוספה שכבה בראשת הראשונה
- 16 -	.....	סעיף ב' – רשות א' – שינוי מספר קרנלי הקונבולוציה
- 17 -	.....	סיכום התוצאות:
- 17 -	.....	פירוט התוצאות:
- 23 -	.....	סעיף א – רשות ב' – הוספה של שכבה
- 25 -	.....	סעיף ב – רשות ב' – הוספה שכבות נסתרות
- 25 -	.....	סיכום התוצאות:
- 25 -	.....	פירוט התוצאות:
- 35 -	.....	משימה 3.2
- 35 -	.....	רשות א'
- 35 -	.....	סיכום התוצאות
- 36 -	.....	פירוט תוצאות
- 49 -	.....	רשות ב'
- 49 -	.....	סיכום התוצאות ומסקנות:
- 50 -	.....	פירוט התוצאות:
- 74 -	.....	משימה 3.3
- 74 -	.....	רשות א' סעיף א – שינוי change dropout probability.
- 76 -	.....	רשות א' סעיף ב' – add early stop and change patience
- 76 -	.....	סיכום התוצאות
- 76 -	.....	פירוט התוצאות
- 78 -	.....	רשות ב' סעיף א' – שינוי drop
- 78 -	.....	סיכום התוצאות

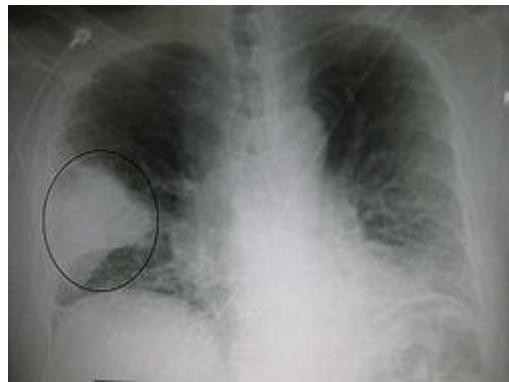
## חזר לתוכן עניינים

- 78 -	פירוט התוצאות
- 81 -	רשות ב' סעיף ב' - add early stop and change patience
- 84 -	בדיקה היתכנות
- 84 -	רשות א'
- 86 -	רשות ב'
- 88 -	סיכום התוצאות :
- 89 -	מסקנות כלליות
- 90 -	סיכום

## מבוא

### מה זה דלקת ריאות?

דלקת ריאות היא דלקת של רקמת הריאה, אשר לרוב מקורה זיהומי. נהוג לבצע צילום רנטגן של בית החזה על מנת לאבחן דלקת ריאות. צילום חזה מבוצע לרוב כאשר מצבו של החולים חמור, כאשר אין שיפור במצבו לאחר 48 שעות של טיפול או כאשר יש צורך לשולג גורמים אחרים למצבו של החולים. צילום החזה יכול להראות את האзор בריאות שנפגע על ידי דלקת ריאות. הוא גם יכול לעזור להבדיל בין דלקת ריאות לבין אחרים כגון ברונכיטיס (דלקת של הסימפונות).  
לדוגמא :



בצילום, דלקת ריאות באונה הימנית שצורתה טרייז (מוקף בעיגול).

### מטרת הפרויקט

בפרויקט זה נמשך פתרונות מבוססי רשתות נוירוניים מסוג CNN לגילוי דלקת ריאות מכ-5863 צילומים אמיתיים.

את התמונות לאיימון ובדיקה קיבלנו מהמרצה בקישור הבא :

<https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>

נראה דוגמא לצילום רנטגן של ריאות והבחנה ביןיהם :



בצילום A ניתן לראות ריאות במצב נורמלי (ללא מחלת).  
בצילום B ניתן להבחין ריאות של חוליה הלוואה בדלקת ריאות.

רשת הנוירוניים שנטכנן בפרויקט זה תוכל לבצע הבחנה בין ריאות של אדם חוליה בדלקת ריאות ובין צילום חזה של אדם שאינו חוליה בדלקת ריאות.  
כלומר, בקבלת תמונה כטקסטו הרשות תוציא פלט האם הבדיקה בדלקת ריאות או לא.

## משימה 1 – תכנון רשותות عمוקות

### דגשים כלליים על הקוד:

הקוד מחולק למספר פונקציות על מנת לאפשר שליטה ובדיקה הקוד, וכן ביצירת סדר. ככלומר הפכנו את הקוד ליותר אפליקטיבי ונגיש למשתמש.

כמו כן, הקוד מכיל ספירות רבות ומגוונות למשל:  
- matplotlib שמאפשר לנו את היכולת לבצע ניתוחים גרפיים.  
- tensorflow ו Keras, שאיתם בונים את הרשות ומאמנים אותה.

### הבנת ה-dataset

כתבנו מחלקה dataset שמקלה בגישה ובשליפת הנתונים הדורושים מהdrive ללמידה הרשותות. drive עצמו חילקנו את התמונות ל 3 סטים נפרדים :

Train set .1  
Valid set .2  
Test set .3

```
> class dataset:  
>     """Import data from google drive and split it into lists of:  
>     x_train, y_train, x_val, y_val, x_test, y_test.  
>     ImegeSize is a number that define the square size of the image  
>     to be resize in order to uniform the training dataset input.  
>     ImageType defines how many colors the image has - gray (1 dimension)  
>     or RGB (3 dimensions)."""  
>     def __init__(self, ImegeSize, ImageType=cv2.IMREAD_GRAYSCALE):  
>         # paths of the image:  
>         training_dir='/content/drive/MyDrive/Colab Notebooks/finalproject/train'  
>         test_dir='/content/drive/MyDrive/Colab Notebooks/finalproject/test'  
>         validation_dir='/content/drive/MyDrive/Colab Notebooks/finalproject/val'  
>         # creating array of images and array of labels: x=images, y=labels  
>         self.x_test, self.y_test = self.make_dataset(test_dir, ImegeSize, ImageType)  
>         self.x_val, self.y_val = self.make_dataset(validation_dir, ImegeSize, ImageType)  
>         self.x_train, self.y_train = self.make_dataset(training_dir, ImegeSize, ImageType)  
  
>         def make_dataset(self, dir, ImegeSize, ImageType): ...
```

הסטים נשלפו כך שהם מאוחסנים בזיכרון לרשימות X,Y בעבור כל סט, ככלומר הם מחולקים  
ל-image and label.  
במחלקה ניתן לראות שקדום כל אנחנו מבצעים גישה אל ה `drive` וכנסים דרך `path` אל המיקום  
שבו נמצא `dataset` שלו.  
אנחנו מבצעים נרמול בעבור כל התמונות ע"י חלוקת ערך של כל פיקסל בתמונה ב 1/255 , מה  
ש מעביר לנו את `data set` שmagiu בגוני אפור להיות בטווח בין 0 ל 1.

בנוסף, `ImageDataGenerator` מאפשר לנו לבצע שינויים בתמונות `dataset` שלנו.  
ככלומר, עשינו שימוש `Augmentation` על מנת לבצע העשרה של הדאטה :

```
CreateDataGeneratorAugmentation(DataSet.x_train,zoom_range,shift_range,rotation_range)
```

בהמשך התוכנית אנחנו מייצרים אובייקט מסווג `ImageDataGenerator` אשר מבצע augmentation אשר מאפשר לנו לבצע מספר סוגים של טרנספורמציות שמאפשר לנו להגדיל את `.datasetn`.

Rotation\_range פרמטר שגדיר את התחום מאפס עד הפרמטר שבו יבוצעו רוטציות  
rndomליות של התמונות בתחום.

## חזר לתוכן עניינים

פרמטר שאיתו ניתן לבצע resize לתמונות. כי הרি התמונות המקוריות סביר להניח מוגעות **בגדלים שונים** וקביעת resize (250X250) שווה לכל התמונות מאפשר לנו להפוך את Inputה של המערכת להיות אחיד).

32 עבור ה training ועבור ה validation נבחר להיות 32, כלומר בכל Batch\_Size מקבצים. והגנרטור של התמונות ייצור 32 מקבצים.

## מבנה רשת א':

```
def CreateNetworkWithoutTransferLearning(ImageResize,drop,KERNAL, MoreLayer=0):
    #creating the network:
    model = Sequential()

    model.add(Conv2D(32*KERNAL , (3,3) , strides = 1 , padding = 'same' , activation = 'relu' ,
    | | | | | input_shape = (ImageResize,ImageResize,1)))
    model.add(BatchNormalization())
    model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))

    model.add(Conv2D(64*KERNAL , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
    model.add(Dropout(drop))
    model.add(BatchNormalization())

    model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
    model.add(Conv2D(64*KERNAL , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
    model.add(BatchNormalization())

    model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
    model.add(Conv2D(128*KERNAL , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
    model.add(Dropout(drop))
    model.add(BatchNormalization())

    model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
    model.add(Conv2D(128*KERNAL , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
    model.add(Dropout(drop))
    model.add(BatchNormalization())

    model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
    model.add(Conv2D(256*KERNAL , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
    model.add(Dropout(drop))
    model.add(BatchNormalization())

    model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
    model.add(Flatten())
    model.add(Dense(units = 128 , activation = 'relu'))
    model.add(Dropout(drop))

    if(MoreLayer): # part of mission 3.1
        for i in range(MoreLayer):
            model.add(Dense(128, activation = 'relu'))
            model.add(Dropout(drop))

    model.add(Dense(units = 1 , activation = 'sigmoid'))

    model.summary()
    return model
```

## הסבר על שכבות הרשת:

1. **שכבה הconvולוציה :**  
(שכבות הconvולוציה נבדלות במספר הkerneils שלהם)

- היא מקבץ של פילטרים ذو ממדים.
- מספר הפילטרים שלו הוא 32 והוא ייחסית לפרמטר KERNAL Kernel size והוא (3,3) כלומר ממד הפילטרים הם 3X3.
- הקלט בשכבה הראשונה של הרשות הוא התמונה עצמה, הגדרנו קודם לכן "נរמול" gray scale level 250x250x1, כלומר התמונה בוגודל של 1x250x250.
- פונקציית האקטיבציה מוגדרת להיות `relu`.
- כל התמונות בסט להיות בגודל של 1x250x250, כלומר התמונה בpadding `Stride=1` כלומר הקונולוציה זהה צעד אחד כל פעם.
- כלומר ה ה באותם ממדים כמו ה.

2. שכבת Batch normalization :  
בחרנו לעשות שימוש בשכבה זו כדי לנרמל כל batch

תפקידו לנרמל את התמונות שנמצאות בbatch לתחולת 0 ומשנות 1 במטרה לגרום לרשות להיות מהירה יותר ויציבה יותר ועשו את אימון הרשות מהיר יותר.

3. Max pooling :  
אנחנו עושים שימוש בשכבה זאת על מנת להקטין את ממד התמונה לאורך הדורך שהוא עבורת ברשת העמוקה, בחרנו בMAX כדי לחת את החלק הדומיננטי בתמונה לצורך האבחנה, שכן AVG יממן את ההבדלים וזאת תופעה לא רצואה.  
Dropout.4

4. Dropout :  
אננו משתמשים בשכבה זו כדי למנוע over fitting, לtraining data

5. Flatten :

אננו עושים שימוש בשכבה זו על מנת לשדרר את כל המידע שהיתקבל עד כה לוקטור אחד. כדי שנוכל להעביר את המידע לשכבה הבאה MLP.

6. Dense :

השתמשנו בפונקציה זו כדי לבנות בלוק `mlp` שמכילה 128 פרס פטרוניים שמחוברים לказה הרשות. כל פרס פטרון מכיל בתוכו פונקציית אקטיבציה בשם RELU.

7. Dense נוסף :

השתמשנו בפונקציה זה כדי להוסיף פרספיטרונו שמכיל פונקציית אקטיבציה של Sigmoid שיבצע לנו סיווג ביןארי בצורה היסטבורותית.

חלק זה על ידי פקודה יכול להויסף שכבות נוספות ל `mlp` במשימה 3.1 אנחנו עושים במנגנון זה שימוש כדי לשנות את עומק הרשות.

```
if(MoreLayer): # part of mission 3.1
    for i in range(MoreLayer):
        model.add(Dense(128, activation = 'relu'))
        model.add(Dropout(drop))
```

## חזר לתוכן עניינים

ניתן לראות שהחלק המשמעותי ביותר מבחןת כמות הפרמטרים הדרושים למידה נמצא בחלק MLP – סה"כ 6,554,113 פרמטרים  
בעוד שבחלק הקונבולוציוני יש סה"כ 387,840 פרמטרים.  
אם היינו עושים את כל המערכת רק MLP אז היינו מכנים בכניסה לרשת אם כן את התמונה בגודל 200X200 ולאחר מכן מכך עושים flatting, היינו מקבלים וקטור בגודל 40,000.  
ואם היינו בוחרים מספר זהה של perceptron (256) אז מספר הפרמטרים יהיה דרוש לנו ברשת היה :

$$parameters\ to\ learn = (40,000 + 1) * 256 + 257 = 10,240,513$$

אם כך רשת CNN חסכה לנו

$$10,240,513 - 6,554,113 - 387,840 = 3,298,560 parameters\ to\ learn$$

מספר משמעותי מאוד.

## מבנה רשת ב'

### בחירת הרשת

על מנת לבחור רשת מתאימה כרשת בסיס, חיפשו ומיצאו מחקר לגבי רשותות עמוקות שנבדקו מול דלקת ריאות וגם קורונה.  
המחקר שמצאו בודק 3 רשותות CNN שונות.

להלן תוצאות המחקר :

M. Rahimzadeh and A. Attar

Informatics in Medicine Unlocked 19 (2020) 100360

Table 3

This table reports the number of true and false positives and false negatives for each class.

Fold	Network	COVID-19 Correct detected	COVID-19 Not detected	COVID-19 Wrong detected	PNEUMONIA Correct detected	PNEUMONIA Not detected	PNEUMONIA Wrong detected	NORMAL Correct detected	NORMAL Not detected	NORMAL Wrong detected
1	Xception	26	5	101	3983	437	569	6245	606	378
	ResNet50V2	27	4	96	3858	562	480	6334	517	507
	Concatenated	26	5	68	3745	675	309	6526	325	628
2	Xception	23	8	42	3874	546	409	6426	425	528
	ResNet50V2	22	9	67	3659	761	501	6340	511	713
	Concatenated	23	8	27	3913	507	434	6413	438	492
3	Xception	21	9	28	3942	478	436	6411	440	463
	ResNet50V2	22	8	97	3770	650	392	6433	418	587
	Concatenated	25	5	35	3847	573	342	6502	349	550
4	Xception	22	9	42	3818	602	433	6411	440	576
	ResNet50V2	22	9	78	4015	405	758	6065	786	364
	Concatenated	26	5	77	3860	560	480	6340	511	519
5	Xception	21	10	41	4041	379	502	6335	516	362
	ResNet50V2	21	10	42	3604	816	284	6549	302	802
	Concatenated	24	7	43	3941	479	390	6442	409	462

מتوוך התוצאות לקהנו fold 1 במסום וחשבנו עבור שלושת הרשותות את ביצועיהם על פי הנוסחאות שלמדו בcliffeה :

$$pres = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN},$$

$$F\ score = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \quad accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

Network	acc	F Score	precision	recall	TP	FP	TN	FN
xception	0.89696	0.87146	0.875	0.86795	3983	569	6245	606
ResNet50v2	0.91171	0.88659	0.88935	0.88385	3858	480	6334	507
concatenated	0.94186	0.92196	0.92378	0.92015	3745	309	6526	325

ולכן בחרנו את רשת ResNet50V2 להיות רשת הבסיס שלנו מטווך ראייה שבפרויקט זה אינו מעונייננו לשפר את ביצועי המודל.

## חזר לתוכן עניינים

```
def CreateNetworkWithTransferLearning(ImageResize, drop, MoreLayer=0, Percptorn_factor=7):
    input_shape = (ImageResize, ImageResize, 3)
    base_model = tf.keras.applications.ResNet50V2(weights='imagenet', input_shape=input_shape, include_top=False)

    for layer in base_model.layers:
        layer.trainable = False # freez the base model layers

    model = Sequential()
    model.add(base_model)
    model.add(GlobalAveragePooling2D())
    model.add(Dense(2**Percptorn_factor, activation = 'relu'))
    model.add(Dropout(drop))
    if(MoreLayer): # part of mission
        for i in range(MoreLayer):
            model.add(Dense(2**Percptorn_factor, activation = 'relu'))
            model.add(Dropout(drop))
    model.add(Dense(1, activation='sigmoid'))

    model.summary(expand_nested=False,show_trainable=True)
    return model,base_model
```

מבנה הרשת בהתאם לשיטה  
בשיטת ה - transfer learning אנו משתמשים במודל בסיס שכבר מגע מאומן ועליו מרכיבים עוד רשת MLP עם מספר שכבות כרצוננו על מנת לבצע סיווג לדאטה הנוסף שלנו.

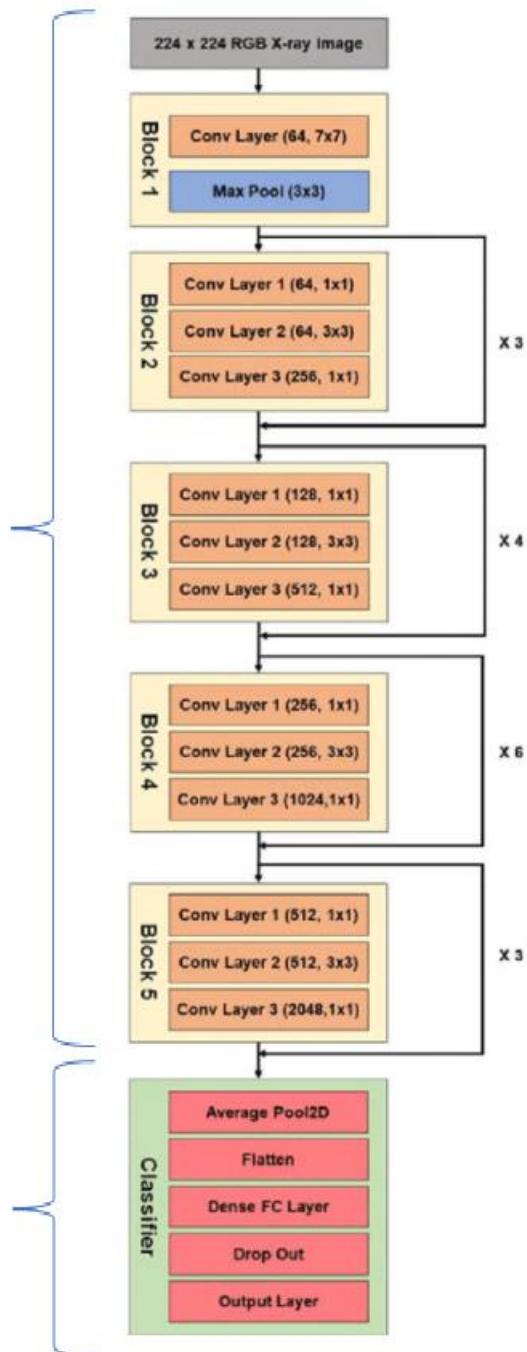
בפתרון שלנו **מודל הבסיס** הוא ResNet50V2

בדיאגרמה מטה ניתן לראות את הארכיטקטורה של מודל הבסיס.  
ניתן לראות שלמודל ישנים 50 שכבות.  
וכן בכל הבלוקים מלבד הראשון קיים מרכיב ה – Residual שמקוצר חיבורים (mphiquit אט השגיאה כתוצאה מהעומקת הרשת, ככל הנראה מוצמצם משמעותית את תופעת ה vanishing gradient).

## ResNet-50V2

בלוק הבסיסי.  
 $Layers\ number =$   
 $= 2 + 3 \cdot 3 + 3 \cdot 4 + 3 \cdot 6 + 3 \cdot 3$   
 $= 50$

הבלוק המורכב.  
 אנחנו מוסיפים בצורה עצמאית מחוץ  
 למודול `base`.



ניתן להסביר מהמבנה, שאפשר להשתמש בו ליישום שבו נלקחת תמונה כלשהי והיא עובר צמצום של ממדיה ומחולצים ממנו דפוסים ותבניות.  
 נאמר קודם שרשימת הבסיסית ישם סך הכל 50 שכבות, לשכבות אלו יש מוטיב שחזור על עצמו לאורך הרשות, יוצא מן הכלל הוא הבלוק הראשון.

הבלוק הראשון :

## חזר לתוכן עניינים

	input_1 (InputLayer)	[ (None, 220, 220, 3) ]	0	N
	conv1_pad (ZeroPadding2D)	(None, 226, 226, 3)	0	N
	conv1_conv (Conv2D)	(None, 110, 110, 64)	9472	N
	pool1_pad (ZeroPadding2D)	(None, 112, 112, 64)	0	N
	pool1_pool (MaxPooling2D)	(None, 55, 55, 64)	0	N

הבלוק הראשון מקבל את התמונה המקורית (עד כדי Resize להכנסת Input אחד)

	conv5_blocks_2_relu (Activation)	(None, 7, 7, 512)	0	N
	conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	N
	conv5_block3_out (Add)	(None, 7, 7, 2048)	0	N
	post_bn (BatchNormalization)	(None, 7, 7, 2048)	8192	N
	post_relu (Activation)	(None, 7, 7, 2048)	0	N

אלו השכבות האחרונות במודל base, ניתן לראות שבמוצא הרשות יוצאים 2048 feature map בגודל 7X7.

לאחר מכן אנחנו מגדרים את השכבות הבאות:

	global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0	Y
	dense (Dense)	(None, 128)	262272	Y
	dropout (Dropout)	(None, 128)	0	Y
	dense_1 (Dense)	(None, 1)	129	Y

השכבות הללו מגדרות ייחדיו את בלוק MLP.

נשים לב לשכבה הglobal average pooling , שכבה זאת לוקחת בעצם מכל feature map מוצע ושם אותו בוקטור. כך הכל יישם 2048feature map או סך הכל מתקבל וקטור בגודל 2048, זאת שיטה שבעצם מכינה את dataן לרשות MLP.

וחתמת השיטה של transfer learning בדרך כלל נאמן רק את השכבות הללו. **להעביר לשכב**

**האימון**

למעט בfine – tuning שנבחר שכבות נוספות מ-ResNet שאוותם נאמנו יחד עם בלוק MLP.

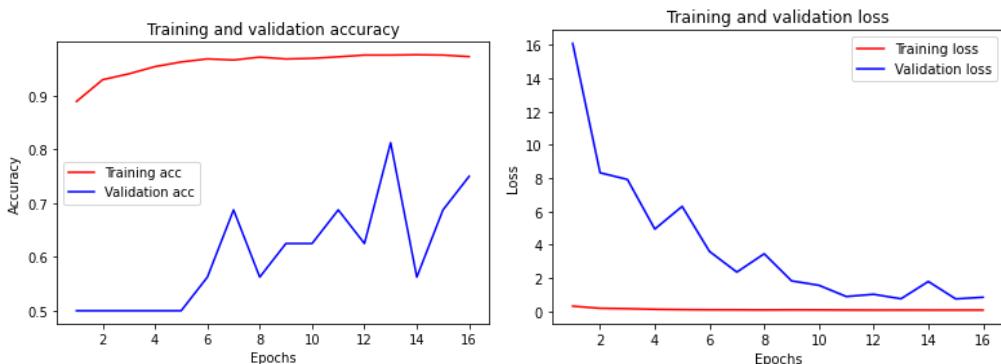
## משימה 2 – אימון הרשות והערכת ביצועים

\* ReduceLROnPlateau זהה פונקציית שmbצעת הפחתה בקצב הלמידה慢速 כדי הריצה, היא מבצעת ניטור לפרמטר מסוים ובאופן דומה למנגנון early stopping בפרמטר patience.

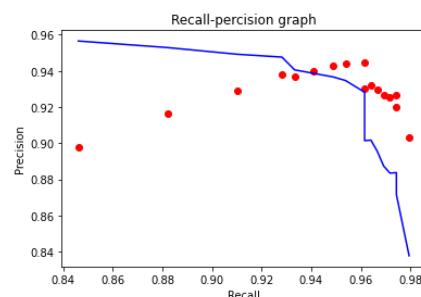
### עבור רשות א'

- אלגוריתם האימון – ReduceLROnPlateau+ Adam
- EPOHCS = 16
- Learning rate = 0.001
- Batch size = 32

#### 2.1. גրף התוכניות של תהליכי האימון



#### 2.2. גրף הביצועים – Precision Recall – 2.2+2.3



The f\_score max is: 0.9445843828715366

The corresponding threshold for the maximum f\_score is: 0.5

Model Accuray = 92. 948%

Model Loss = 0.23

מנקודה זו ואילך ביצועים אלו של הרשות יישמו בעברונו רפרנס ומכאן ננסה לשפר את ביצועי הרשות במשימות הבאות.

### עבור רשות ב'

#### 2.1. גראף התוכניות של תהליכי האימון

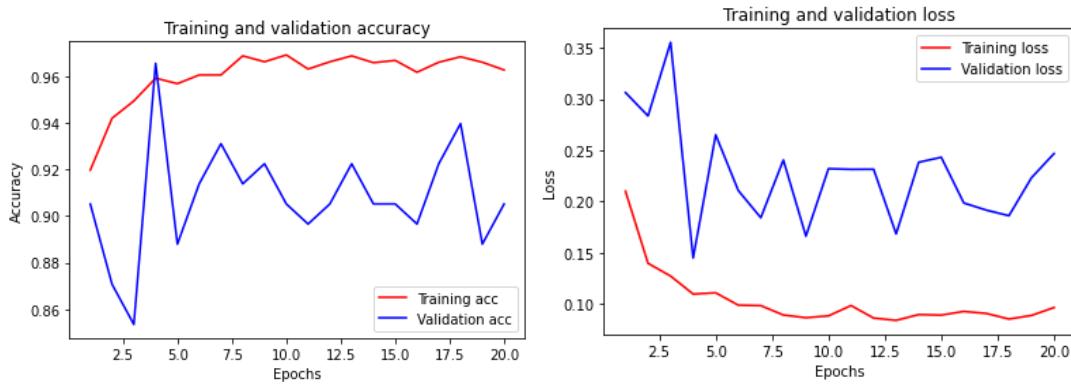
##### שלב א

- אלגוריתם האימון – Adam + ReduceLROnPlateau
- EPOHCS 20
- Learning rate = 0.001
- Batch size = 32

Total params: 23,827,201

Trainable params: 262,401

Non-trainable params: 23,564,800



**Estimated Model Accuray = 91.025%**  
**Loss of the model = 0.24**

#### שלב ב - fine tune

- אלגוריתם האימון – Adam + ReduceLROnPlateau

- EPOHCS 10

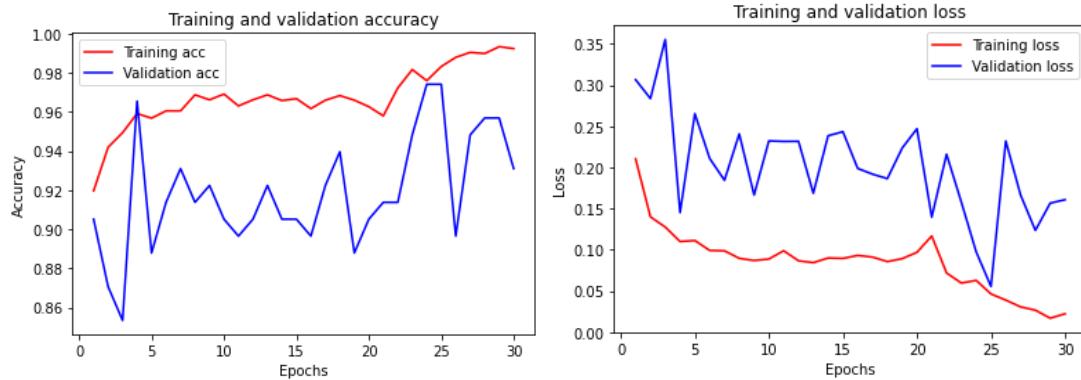
- Learning rate = 0.0001

- Batch size = 32

Total params: 23,827,201

Trainable params: 15,496,449

Non-trainable params: 8,330,752



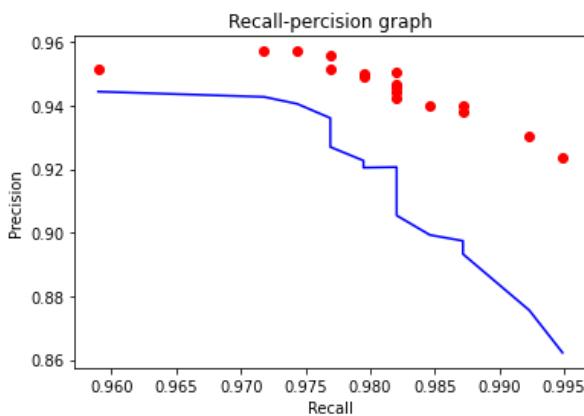
**Estimated Model Accuray = 93.1%**  
**Loss of the model = 0.23**

#### F-SCROE Precision Recall – 2.2+2.3

כפי שניתן היה לראות קודם, באמצעות שיטת fine tuning הצלחנו לשפר את רמת הדיקוק של המודל.

להלן גרפ הביצועים של הרשת :

## חזר לתוכן עניינים



The f\_score max is: 0.9571788413098237

The corresponding **threshold** for the maximum f\_score is: 0.8

כניל' לגבי הרשות השנייה, מנוקדיה זו ואילך ביצועים אלו של הרשות יישמו בעבורנו רפרנס ומכאן  
נכשה לשפר את ביצועי הרשות במשימות הבאות.

## 3.1 מנגנון

### סעיף א' - רשות א' - הוספה שכבה

האימונו התבצע לפי הפרמטרים הבאים :

- אלגוריתם האימון – ReduceLROnPlateau+ Adam
- EPOHCS = 16
- Learning rate = 0.001
- Batch size = 32

שכבות הקונבולוציה הקיימות לאחר חילוץ הנתונים מהתמונה מניב 256 תמונות קטנות בגודל 4X4. לדעטנו אין טעם בהוספה של שכבה נוספת לקונבולוציה כי הגודל של התמונה כבר קטן מספיק. אז במקום זאת החלפנו להוסיפה שכבה נסורתה לחץ MLP שאולי תשפר את הביצועים במהלך הסיווג. מבנה השכבה : 128 פרטיטוונים עם פונקציית הפעלה Relu עם הסתברות של 0.1.

בפועל רמת הדיק היא 91.5 אחוז שזה נמוכה כמעט במעט מותzáה ללא השינוי (92.9 אחוז). וכך עבור רשות זאת בסעיף הבא נבצע שינוי אחר.

Optimizer: Adam , Ir: 0.001 , EPOHCS: 16 , KERNAL: 1 , MoreLayer: yes

#### Optimizer: Adam , Ir: 0.001 , EPOHCS: 16 , KERNAL: 1 , MoreLayer: yes ####

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
=====		
conv2d_6 (Conv2D)	(None, 250, 250, 32)	320
batch_normalization_6 (Batch Normalization)	(None, 250, 250, 32)	128
max_pooling2d_6 (MaxPooling 2D)	(None, 125, 125, 32)	0
conv2d_7 (Conv2D)	(None, 125, 125, 64)	18496
dropout_5 (Dropout)	(None, 125, 125, 64)	0
batch_normalization_7 (Batch Normalization)	(None, 125, 125, 64)	256
max_pooling2d_7 (MaxPooling 2D)	(None, 63, 63, 64)	0
conv2d_8 (Conv2D)	(None, 63, 63, 64)	36928
batch_normalization_8 (Batch Normalization)	(None, 63, 63, 64)	256
max_pooling2d_8 (MaxPooling 2D)	(None, 32, 32, 64)	0
conv2d_9 (Conv2D)	(None, 32, 32, 128)	73856
dropout_6 (Dropout)	(None, 32, 32, 128)	0
batch_normalization_9 (Batch Normalization)	(None, 32, 32, 128)	512
max_pooling2d_9 (MaxPooling 2D)	(None, 16, 16, 128)	0
conv2d_10 (Conv2D)	(None, 16, 16, 128)	147584
dropout_7 (Dropout)	(None, 16, 16, 128)	0
batch_normalization_10 (Batch Normalization)	(None, 16, 16, 128)	512
max_pooling2d_10 (MaxPooling 2D)	(None, 8, 8, 128)	0
conv2d_11 (Conv2D)	(None, 8, 8, 256)	295168
dropout_8 (Dropout)	(None, 8, 8, 256)	0

## חזר לתוכן עניינים

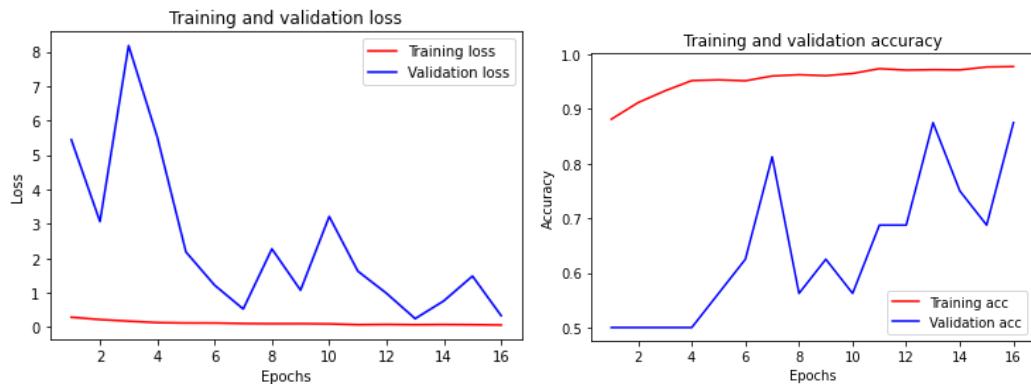
```

batch_normalization_11 (Batch Normalization)      1024
max_pooling2d_11 (MaxPooling2D)                 0
flatten_1 (Flatten)                            0
dense_2 (Dense)                               524416
dropout_9 (Dropout)                           0
dense_3 (Dense)                               16512
dropout_10 (Dropout)                          0
dense_4 (Dense)                               129
=====
Total params: 1,116,097
Trainable params: 1,114,753
Non-trainable params: 1,344

```

Accuracy of the model is - 91.50640964508057 %

זאת השכבה  
שהוספנו



### מסקנה לגבי הוספת השכבה בראש הרשות

הוספה השכבה לא העיליה בשיפור רמת הדיקוק ואף למשה הורידה אותו ולכן אין טעם להוסיף שכבה זאת לבנייה הרשות הזאת.

### סעיף ב' - רשות א' - שינוי מספר קרנלי הקונבולוציה

בעקבות התוצאות הללו משוחחות בסעיף הקודם החלנו שבסעיף זה ננקוט בשיטה שונה וננסה לשנות את מבנה חלק הקונבולוציה של הרשות.

בסעיף זה נבדוק חמייה שינויים של מספר קרנלי הקונבולוציה את השינוי עשינו באופן ייחסי בכל הרשות בצד לשמור על פרופורציות זהות. הגדרנו את הParmeter KERNAL שקובע את החלק היחסית של הפילטרים שנלקחים ביחס לרשות המקורית.

נבדוק את השינויים הבאים :

$$KERNEL = [0.25, 0.5, 0.75, 1.25, 1.5]$$

כאשר 1 KERNEL המצביע שקול לבנייה של הרשות המקורית.

## חזר לתוכן עניינים

```

def CreateNetworkWithoutTransferLearning(ImgaeResize,drop,KERNAL):
    #creating the network:
    model = Sequential()

    model.add(Conv2D(32*KERNAL , (3,3) , strides = 1 , padding = 'same' , activation = 'relu' , input_shape = (ImgaeResize,ImgaeResize,1)))
    model.add(BatchNormalization())
    model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))

    model.add(Conv2D(64*KERNAL , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
    model.add(Dropout(drop))
    model.add(BatchNormalization())

    model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
    model.add(Conv2D(128*KERNAL , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
    model.add(Dropout(drop))
    model.add(BatchNormalization())

    model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
    model.add(Conv2D(128*KERNAL , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
    model.add(Dropout(drop))
    model.add(BatchNormalization())

    model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
    model.add(Conv2D(256*KERNAL , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
    model.add(Dropout(drop))
    model.add(BatchNormalization())

    model.add(Flatten())
    model.add(Dense(units = 128 , activation = 'relu'))
    model.add(Dropout(drop))
    model.add(Dense(units = 1 , activation = 'sigmoid'))

    model.summary()
    return model

```

### סיכום התוצאות:

Accuracy	Loss	KERNAL
87.82%	0.44	0.25
91.82%	0.27	0.5
91.82%	0.25	0.75
92.95%	0.23	1
91.02%	0.32	1.25
91.02%	0.23	1.5

אחרי כל הניסויונות ניתן לראות שüberו יחס הקרנלים שבחרנו, הרשות המקורית הניבתה את התוצאות הטובות ביותר.  
וכן ניתן לראות שüberו יחס 0.25 קיימת שונות גבוהה יחסית.  
בדיעבד אולי היינו צריכים במקומות לבצע שינוי יחסילם, אולי היינו צריכים לשנות רק בשכבה הכניסה שכן אם בכניסה היו יותר קרנלים ייתכן כי יתר מידע יהיה מוחלץ מהתמונה המקורית.

### פירוט התוצאות:

Optimizer: Adam , lr: 0.001 , EPOHCS: 16 , KERNAL: 0.25

נראה כאן איך לדוגמא התבצע השינוי של מספר הפילטרים בכל שכבה קונבולוציה.  
עבור המקרה שבו :

**KERNAL = 0.25**

ניתן לראות שנלקחו **רביע** מכמות הkernelים של הרשות המקורית.  
למשל עבור השכבה הקונבולוציונית הראשונה יש 8 פילטרים.  
**שזה בדיק רביע 32 שהוא המקורי.**

### Optimizer: Adam , lr: 0.001 , EPOHCS: 16 , KERNAL: 0.25 ###  
Model: "sequential"

Layer (type)	Output Shape	Param #
<hr/>		
conv2d (Conv2D)	(None, 250, 250, 8)	80
batch_normalization (BatchN)	(None, 250, 250, 8)	32
ormalization		
max_pooling2d (MaxPooling2D)	(None, 125, 125, 8)	0
conv2d_1 (Conv2D)	(None, 125, 125, 16)	1168
dropout (Dropout)	(None, 125, 125, 16)	0
batch_normalization_1 (Batac)	(None, 125, 125, 16)	64
hNormalization		
max_pooling2d_1 (MaxPooling)	(None, 63, 63, 16)	0
2D)		
conv2d_2 (Conv2D)	(None, 63, 63, 16)	2320
batch_normalization_2 (Batac)	(None, 63, 63, 16)	64
hNormalization)		
max_pooling2d_2 (MaxPooling)	(None, 32, 32, 16)	0
2D)		
conv2d_3 (Conv2D)	(None, 32, 32, 32)	4640

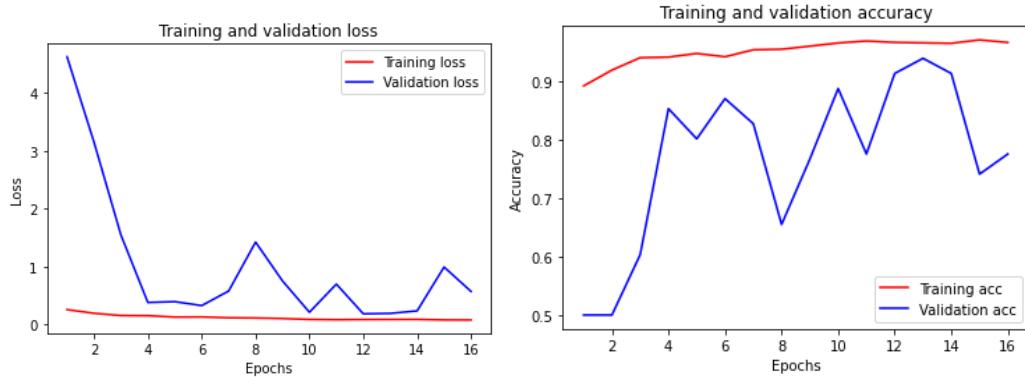
## חומר לתוכן עניינים

```

dropout_1 (Dropout)      (None, 32, 32, 32)      0
batch_normalization_3 (Batch Normalization) (None, 32, 32, 32)    128
max_pooling2d_3 (MaxPooling 2D) (None, 16, 16, 32)      0
conv2d_4 (Conv2D)        (None, 16, 16, 32)    9248
dropout_2 (Dropout)      (None, 16, 16, 32)      0
batch_normalization_4 (Batch Normalization) (None, 16, 16, 32)    128
max_pooling2d_4 (MaxPooling 2D) (None, 8, 8, 32)      0
conv2d_5 (Conv2D)        (None, 8, 8, 64)     18496
dropout_3 (Dropout)      (None, 8, 8, 64)      0
batch_normalization_5 (Batch Normalization) (None, 8, 8, 64)    256
max_pooling2d_5 (MaxPooling 2D) (None, 4, 4, 64)      0
flatten (Flatten)        (None, 1024)        0
dense (Dense)            (None, 128)        131200
dropout_4 (Dropout)      (None, 128)        0
dense_1 (Dense)          (None, 1)         129
=====
Total params: 167,953
Trainable params: 167,617
Non-trainable params: 336

```

Loss of the model is - 0.4463822841644287  
 Accuracy of the model is - 87.82051205635071 %



### Optimizer: Adam , lr: 0.001 , EPOCHS: 16 , KERNAL: 0.5

## Optimizer: Adam , lr: 0.001 , EPOCHS: 16 , KERNAL: 0.5 ##  
 Model: "sequential\_1"

```

Layer (type)           Output Shape        Param #
===== 
conv2d_6 (Conv2D)      (None, 250, 250, 16)   160
batch_normalization_6 (Batch Normalization) (None, 250, 250, 16)    64
max_pooling2d_6 (MaxPooling 2D) (None, 125, 125, 16)      0
conv2d_7 (Conv2D)        (None, 125, 125, 32)    4640
dropout_5 (Dropout)      (None, 125, 125, 32)      0
batch_normalization_7 (Batch Normalization) (None, 125, 125, 32)    128
max_pooling2d_7 (MaxPooling 2D) (None, 63, 63, 32)      0
conv2d_8 (Conv2D)        (None, 63, 63, 32)    9248
batch_normalization_8 (Batch Normalization) (None, 63, 63, 32)    128
max_pooling2d_8 (MaxPooling 2D) (None, 32, 32, 32)      0
conv2d_9 (Conv2D)        (None, 32, 32, 64)    18496
dropout_6 (Dropout)      (None, 32, 32, 64)      0
batch_normalization_9 (Batch Normalization) (None, 32, 32, 64)    256

```

## חומר לתוכן עניינים

```

max_pooling2d_9 (MaxPooling (None, 16, 16, 64)      0
2D)

conv2d_10 (Conv2D)      (None, 16, 16, 64)    36928

dropout_7 (Dropout)    (None, 16, 16, 64)      0

batch_normalization_10 (Batch Normalization (None, 16, 16, 64)    256
chNormalization)

max_pooling2d_10 (MaxPooling (None, 8, 8, 64)      0
g2D)

conv2d_11 (Conv2D)      (None, 8, 8, 128)     73856

dropout_8 (Dropout)    (None, 8, 8, 128)      0

batch_normalization_11 (Batch Normalization (None, 8, 8, 128)    512
chNormalization)

max_pooling2d_11 (MaxPooling (None, 4, 4, 128)      0
g2D)

flatten_1 (Flatten)    (None, 2048)        0

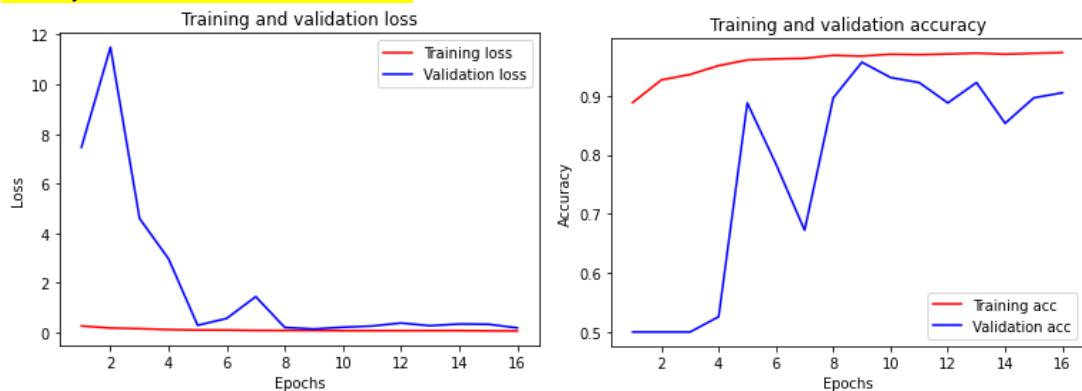
dense_2 (Dense)        (None, 128)         262272

dropout_9 (Dropout)    (None, 128)         0

dense_3 (Dense)        (None, 1)           129

=====
Total params: 407,073
Trainable params: 406,401
Non-trainable params: 672
Loss of the model is - 0.2721377909183502
Accuracy of the model is - 91.82692170143127 %

```



Optimizer: Adam , lr: 0.001 , EPOCHS: 16 , KERNAL: 0.75  
 ### Optimizer: Adam , lr: 0.001 , EPOCHS: 16 , KERNAL: 0.75 ###

Model: "sequential\_2"

Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 250, 250, 24)	240
batch_normalization_12 (Batch Normalization)	(None, 250, 250, 24)	96
max_pooling2d_12 (MaxPooling)	(None, 125, 125, 24)	0
conv2d_13 (Conv2D)	(None, 125, 125, 48)	10416
dropout_10 (Dropout)	(None, 125, 125, 48)	0
batch_normalization_13 (Batch Normalization)	(None, 125, 125, 48)	192
max_pooling2d_13 (MaxPooling)	(None, 63, 63, 48)	0
conv2d_14 (Conv2D)	(None, 63, 63, 48)	20784
batch_normalization_14 (Batch Normalization)	(None, 63, 63, 48)	192

## חזר לתוכן עניינים

```

max_pooling2d_14 (MaxPoolin (None, 32, 32, 48)      0
g2D)

conv2d_15 (Conv2D)      (None, 32, 32, 96)    41568

dropout_11 (Dropout)   (None, 32, 32, 96)    0

batch_normalization_15 (Bat (None, 32, 32, 96)    384
chNormalization)

max_pooling2d_15 (MaxPoolin (None, 16, 16, 96)    0
g2D)

conv2d_16 (Conv2D)      (None, 16, 16, 96)    83040

dropout_12 (Dropout)   (None, 16, 16, 96)    0

batch_normalization_16 (Bat (None, 16, 16, 96)    384
chNormalization)

max_pooling2d_16 (MaxPoolin (None, 8, 8, 96)     0
g2D)

conv2d_17 (Conv2D)      (None, 8, 8, 192)    166080

dropout_13 (Dropout)   (None, 8, 8, 192)    0

batch_normalization_17 (Bat (None, 8, 8, 192)    768
chNormalization)

max_pooling2d_17 (MaxPoolin (None, 4, 4, 192)    0
g2D)

flatten_2 (Flatten)    (None, 3072)        0

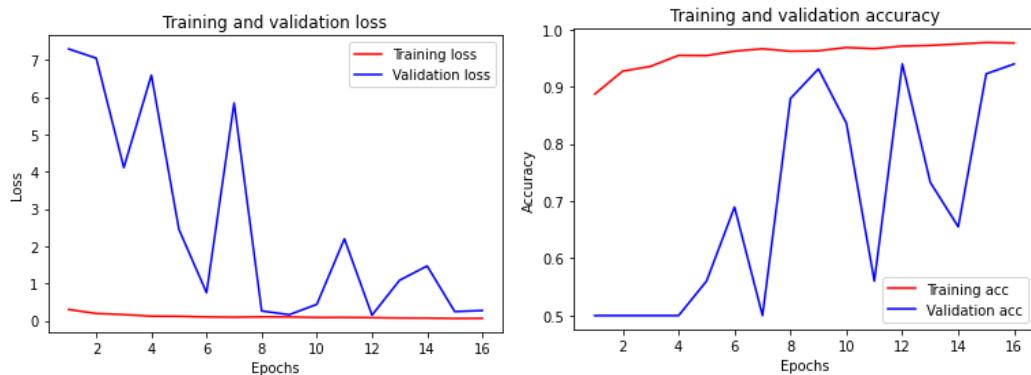
dense_4 (Dense)        (None, 128)         393344

dropout_14 (Dropout)   (None, 128)         0

dense_5 (Dense)        (None, 1)           129

=====
Total params: 717,617
Trainable params: 716,609
Non-trainable params: 1,008
Loss of the model is - 0.25438135862350464
Accuracy of the model is - 91.82692170143127 %

```



Optimizer: Adam , lr: 0.001 , EPOCHCS: 16 , KERNAL: 1.25  
### Optimizer: Adam , lr: 0.001 , EPOCHCS: 16 , KERNAL: 1.25 ###

Model: "sequential\_3"

Layer (type)	Output Shape	Param #
conv2d_18 (Conv2D)	(None, 250, 250, 40)	400
batch_normalization_18 (Batch Normalization)	(None, 250, 250, 40)	160
max_pooling2d_18 (MaxPooling2D)	(None, 125, 125, 40)	0
conv2d_19 (Conv2D)	(None, 125, 125, 80)	28880
dropout_15 (Dropout)	(None, 125, 125, 80)	0

## חזר לתוכן עניינים

```

batch_normalization_19 (Batch Normalization (None, 125, 125, 80) 320
chNormalization)

max_pooling2d_19 (MaxPooling2D (None, 63, 63, 80) 0
g2D)

conv2d_20 (Conv2D) (None, 63, 63, 80) 57680

batch_normalization_20 (Batch Normalization (None, 63, 63, 80) 320
chNormalization)

max_pooling2d_20 (MaxPooling2D (None, 32, 32, 80) 0
g2D)

conv2d_21 (Conv2D) (None, 32, 32, 160) 115360

dropout_16 (Dropout) (None, 32, 32, 160) 0

batch_normalization_21 (Batch Normalization (None, 32, 32, 160) 640
chNormalization)

max_pooling2d_21 (MaxPooling2D (None, 16, 16, 160) 0
g2D)

conv2d_22 (Conv2D) (None, 16, 16, 160) 230560

dropout_17 (Dropout) (None, 16, 16, 160) 0

batch_normalization_22 (Batch Normalization (None, 16, 16, 160) 640
chNormalization)

max_pooling2d_22 (MaxPooling2D (None, 8, 8, 160) 0
g2D)

conv2d_23 (Conv2D) (None, 8, 8, 320) 461120

dropout_18 (Dropout) (None, 8, 8, 320) 0

batch_normalization_23 (Batch Normalization (None, 8, 8, 320) 1280
chNormalization)

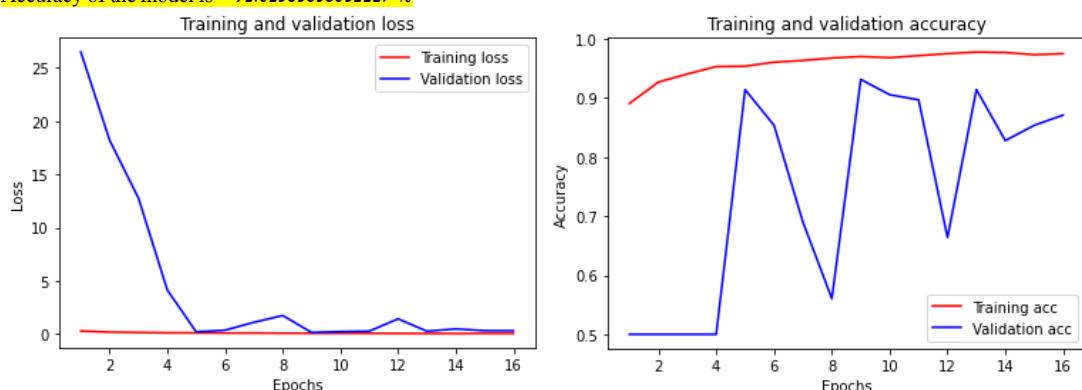
max_pooling2d_23 (MaxPooling2D (None, 4, 4, 320) 0
g2D)

flatten_3 (Flatten) (None, 5120) 0

dense_6 (Dense) (None, 128) 655488

dropout_19 (Dropout) (None, 128) 0
dense_7 (Dense) (None, 1) 129
=====
Total params: 1,552,977
Trainable params: 1,551,297
Non-trainable params: 1,680
Loss of the model is - 0.31647154688835144
Accuracy of the model is - 91.02563858032227 %

```



Optimizer: Adam , lr: 0.001 , EPOCHS: 16 , KERNAL: 1.5  
 ### Optimizer: Adam , lr: 0.001 , EPOCHS: 16 , KERNAL: 1.5 ###

Model: "sequential\_4"

Layer (type)	Output Shape	Param #
conv2d_24 (Conv2D)	(None, 250, 250, 48)	480
batch_normalization_24 (Batch Normalization (None, 250, 250, 48) 192         chNormalization)		
max_pooling2d_24 (MaxPooling2D (None, 125, 125, 48) 0		

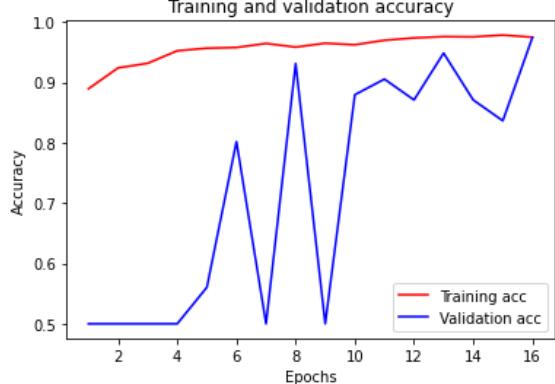
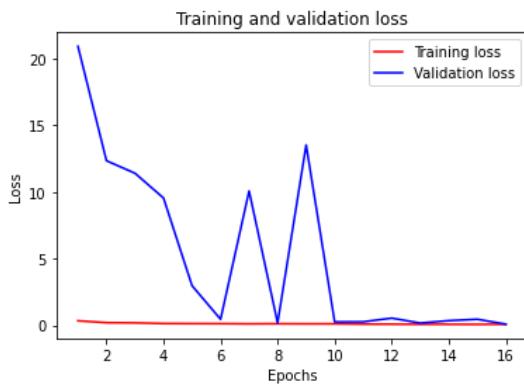
## חומר לתוכן עניינים

```

g2D)
conv2d_25 (Conv2D)      (None, 125, 125, 96)    41568
dropout_20 (Dropout)    (None, 125, 125, 96)    0
batch_normalization_25 (Batch Normalization) (None, 125, 125, 96)    384
max_pooling2d_25 (MaxPooling2D) (None, 63, 63, 96)    0
g2D)
conv2d_26 (Conv2D)      (None, 63, 63, 96)    83040
batch_normalization_26 (Batch Normalization) (None, 63, 63, 96)    384
chNormalization)
max_pooling2d_26 (MaxPooling2D) (None, 32, 32, 96)    0
g2D)
conv2d_27 (Conv2D)      (None, 32, 32, 192)   166080
dropout_21 (Dropout)    (None, 32, 32, 192)   0
batch_normalization_27 (Batch Normalization) (None, 32, 32, 192)    768
chNormalization)
max_pooling2d_27 (MaxPooling2D) (None, 16, 16, 192)    0
g2D)
conv2d_28 (Conv2D)      (None, 16, 16, 192)   331968
dropout_22 (Dropout)    (None, 16, 16, 192)   0
batch_normalization_28 (Batch Normalization) (None, 16, 16, 192)    768
chNormalization)
max_pooling2d_28 (MaxPooling2D) (None, 8, 8, 192)    0
g2D)
conv2d_29 (Conv2D)      (None, 8, 8, 384)    663936
dropout_23 (Dropout)    (None, 8, 8, 384)    0
batch_normalization_29 (Batch Normalization) (None, 8, 8, 384)    1536
chNormalization)
max_pooling2d_29 (MaxPooling2D) (None, 4, 4, 384)    0
g2D)
flatten_4 (Flatten)    (None, 6144)        0
dense_8 (Dense)        (None, 128)         786560
dropout_24 (Dropout)   (None, 128)         0
dense_9 (Dense)        (None, 1)          129
=====

Total params: 2,077,793
Trainable params: 2,075,777
Non-trainable params: 2,016
Loss of the model is - 0.23495358228683472
Accuracy of the model is - 91.02563858032227 %

```



## סעיף א - רשות ב' - הוספה של שכבה

הוספנו בסוף הרשות השנייה שכבה מכיוון שלאחר שחקרנו את הרשות Resnet50v2 הבנו שיש בה 50 שכבות קונבולוציה שכבר מצמצמים את גודל התמונה לממדים : 8x8x2048 ותוספה של עוד שכבות קונבולוציה כנראה שכבר לא יתרמו. מצד שני הוספות שכבות נסתרות לשכבות פרטפטורונים קיימות יכולה להעלות את רמת הדיק מול הטסט והתאורה טוענת שככל שיותר שכבות קיימות יש סיכוי גדול יותר להעלות את רמת הדיק אך לא בחררת. נבדוק תיאוריה זאת על המודל שלנו על ידי הוספה שכבה אחת. השכבה המוסיפה : 128 פרטפטורונים עם פונקציית הפעלה Relu עם Dropout של 0.1.

תוצאות הרצה עם ההוספה

רשות ב'							
סעיף א'							
fine tune				only MLP train			
Loss	Accuracy%	F Score%	Threshold	Loss	Accuracy%	F Score%	Threshold
0.198	93.26	95.29	0.2	0.24	91.02	94.4	0.75

הדיק עלה מעט מ 93.1% ל 93.26% לאחר אימון הקצה.

Optimizer: Adam , lr: 0.001 , EPOHCS: 20 , Percptorn factor: 7 , MoreLayer: yes

```
### CNN transfer learning network Model ###
### Add layer to the Network###
### Optimizer: Adam , lr: 0.001 , EPOHCS: 20 , Percptorn_factor: 7 , MoreLayer: yes ###
```

Model: "sequential"

```
Layer (type)      Output Shape       Param #  Trainable
=====
resnet50v2 (Functional)  (None, 8, 8, 2048)    23564800  Y
global_average_pooling2d (G (None, 2048)      0        Y
lobalAveragePooling2D)

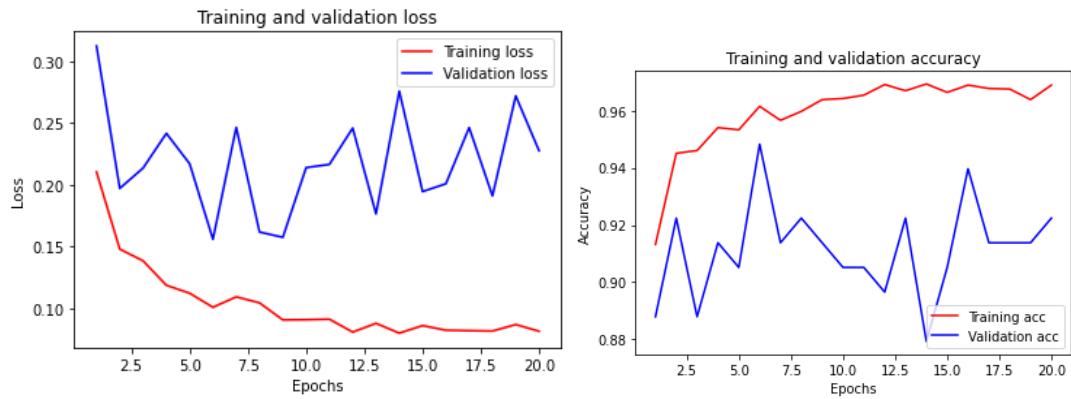
dense (Dense)     (None, 128)        262272   Y
dropout (Dropout) (None, 128)        0        Y
dense_1 (Dense)   (None, 128)        16512    Y
dropout_1 (Dropout) (None, 128)        0        Y
dense_2 (Dense)   (None, 1)          129      Y
=====

Total params: 23,843,713
Trainable params: 278,913
Non-trainable params: 23,564,800
```

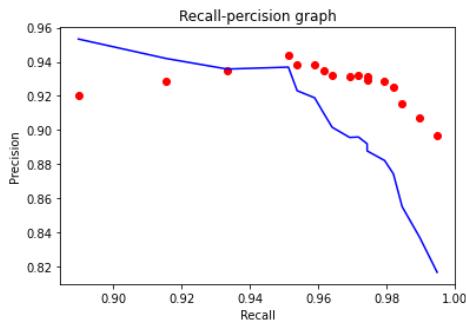
The history is now!

```
20/20 [=====] - 2s 77ms/step - loss: 0.2405 - accuracy: 0.9103
Loss of the model is - 0.2404867261648178
20/20 [=====] - 1s 65ms/step - loss: 0.2405 - accuracy: 0.9103
Accuracy of the model is - 91.02563858032227 %
```

## חומר לתוכן עניינים



Adam\_lr0.001\_EPOHCS20.h5



The f\_score max is: 0.9440203562340967

The corresponding threshold for the maximum f\_score is: 0.75  
Model: "sequential"

Layer (type)	Output Shape	Param #	Trainable
<hr/>			
resnet50v2 (Functional)	(None, 8, 8, 2048)	23564800	Y
<hr/>			
global_average_pooling2d (G (None, 2048))	(None, 2048)	0	Y
<hr/>			
dense (Dense)	(None, 128)	262272	Y
dropout (Dropout)	(None, 128)	0	Y
dense_1 (Dense)	(None, 128)	16512	Y
dropout_1 (Dropout)	(None, 128)	0	Y
dense_2 (Dense)	(None, 1)	129	Y
<hr/>			
Total params:	23,843,713		
Trainable params:	15,512,961		
Non-trainable params:	8,330,752		

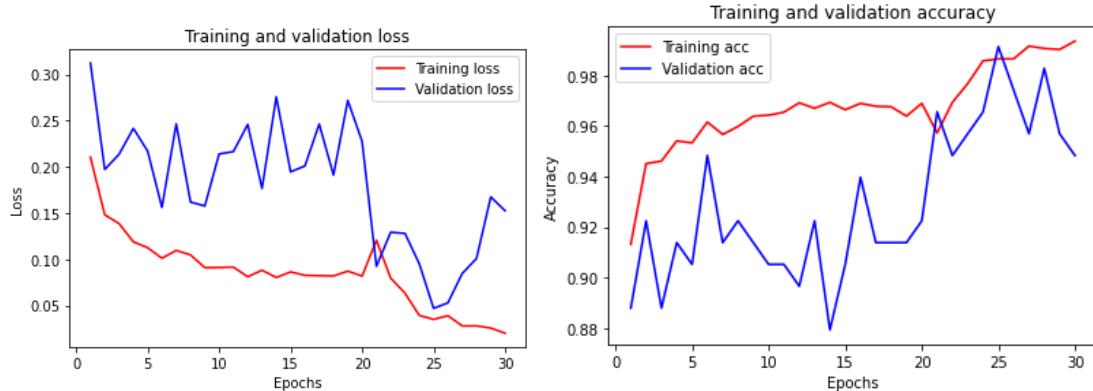
The history is return!

20/20 [=====] - 2s 65ms/step - loss: 0.1984 - accuracy: 0.9327

Loss of the model is - 0.198404923081398

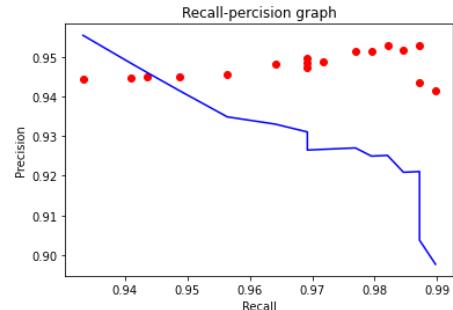
20/20 [=====] - 1s 65ms/step - loss: 0.1984 - accuracy: 0.9327

Accuracy of the model is - 93.2692289352417 %



## חזר לתוכן עניינים

Adam\_lr0.001\_EPOHCS20\_FineTune.h5



The f\_score max is: 0.952970297029703  
The corresponding threshold for the maximum f\_score is: 0.2

## סעיף ב' - רשות ב' - הוספת שכבות נסתרות

בהמשך לסעיף הקודם, ראיינו שהוספה של שכבה אחת של פרספטרונים שיפורה את רמת הדיווק. ויתכן שיותר שכבות יתרמו עוד, נבדוק את המספר האופטימלי של הוספת השכבות. נבצע 5 שינויים ונציג את התוצאות השונות

### סיכום התוצאות:

רשות ב'								Layer added	
fine tune				only MLP train					
Loss	Accuracy%	F Score%	Threshold	Loss	Accuracy%	F Score%	Threshold		
0.21	93.58	95.58	0.6	0.23	91.18	93.5	0.7	1	
0.27	92.68	95.13	0.65	0.232	91.185	93.9	0.7	2	
0.213	94.23	95.59	0.55	0.241	91.025	93.8	0.65	3	
0.29	92.948	95.42	0.9	0.233	92.147	94.33	0.65	4	
0.242	93.43			0.238	91.5	94.2	0.7	5	

על ידי הוספה של 3 שכבות, הדיווק עלה יפה מ 93.1% ל 94.23% לאחר אימון הקצה.

### פירוט התוצאות:

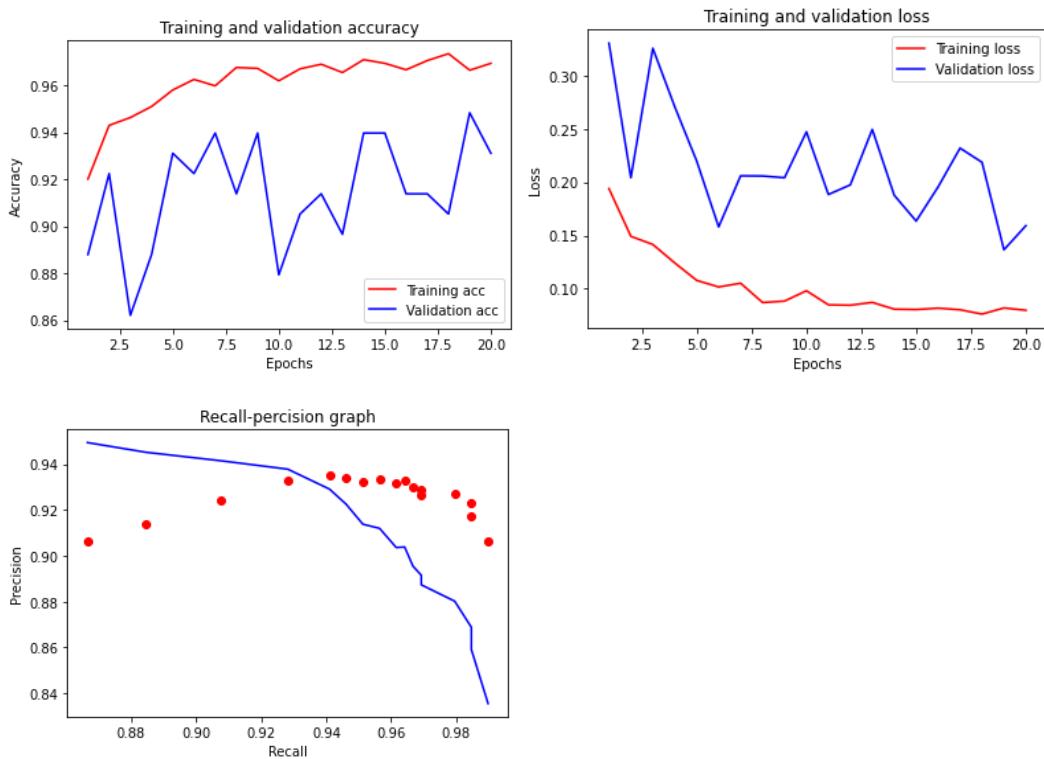
Optimizer: Adam , lr: 0.001 , EPOHCS: 20 , added layers: 1  
### Optimizer: Adam , lr: 0.001 , EPOHCS: 20 , added layers: 1 ###

Downloading data from [https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50v2\\_weights\\_tf\\_dim\\_ordering\\_tf\\_kernels\\_notop.h5](https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50v2_weights_tf_dim_ordering_tf_kernels_notop.h5)  
94674944/94668760 [=====] - 0s 0us/step  
94683136/94668760 [=====] - 0s 0us/step  
Model: "sequential"

Layer (type)	Output Shape	Param #	Trainable
<hr/>			
resnet50v2 (Functional)	(None, 8, 8, 2048)	23564800	Y
<hr/>			
global_average_pooling2d (G (None, 2048)	0	Y	
lobalAveragePooling2D)			
<hr/>			
dense (Dense)	(None, 128)	262272	Y
dropout (Dropout)	(None, 128)	0	Y
dense_1 (Dense)	(None, 128)	16512	Y
dropout_1 (Dropout)	(None, 128)	0	Y
dense_2 (Dense)	(None, 1)	129	Y
<hr/>			
Total params: 23,843,713			
Trainable params: 278,913			
Non-trainable params: 23,564,800			

## חזר לתוכן עניינים

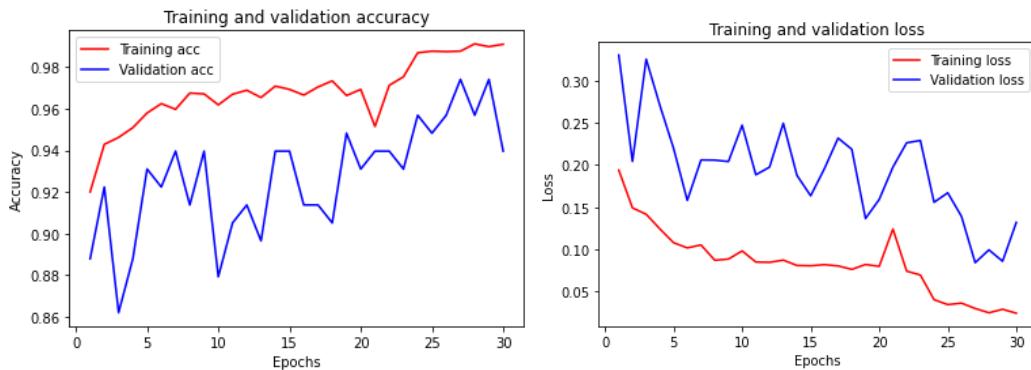
Loss of the model is - 0.23464716970920563  
 20/20 [=====] - 1s 64ms/step - loss: 0.2346 - accuracy: 0.9119  
 Accuracy of the model is - 91.18589758872986 %



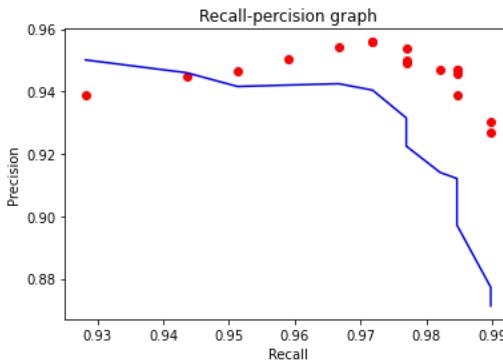
The f\_score max is: 0.935031847133758  
 The corresponding threshold for the maximum f\_score is: 0.7000000000000001  
 Model: "sequential"

Layer (type)	Output Shape	Param #	Trainable
<hr/>			
resnet50v2 (Functional)	(None, 8, 8, 2048)	23564800	Y
<hr/>			
global_average_pooling2d (G (None, 2048))	(None, 2048)	0	Y
lobalAveragePooling2D)			
dense (Dense)	(None, 128)	262272	Y
dropout (Dropout)	(None, 128)	0	Y
dense_1 (Dense)	(None, 128)	16512	Y
dropout_1 (Dropout)	(None, 128)	0	Y
dense_2 (Dense)	(None, 1)	129	Y
<hr/>			
Total params:	23,843,713		
Trainable params:	15,512,961		
Non-trainable params:	8,330,752		

Loss of the model is - 0.21850773692131042  
 Accuracy of the model is - 93.58974099159241 %



## חומר לתוכן עניינים

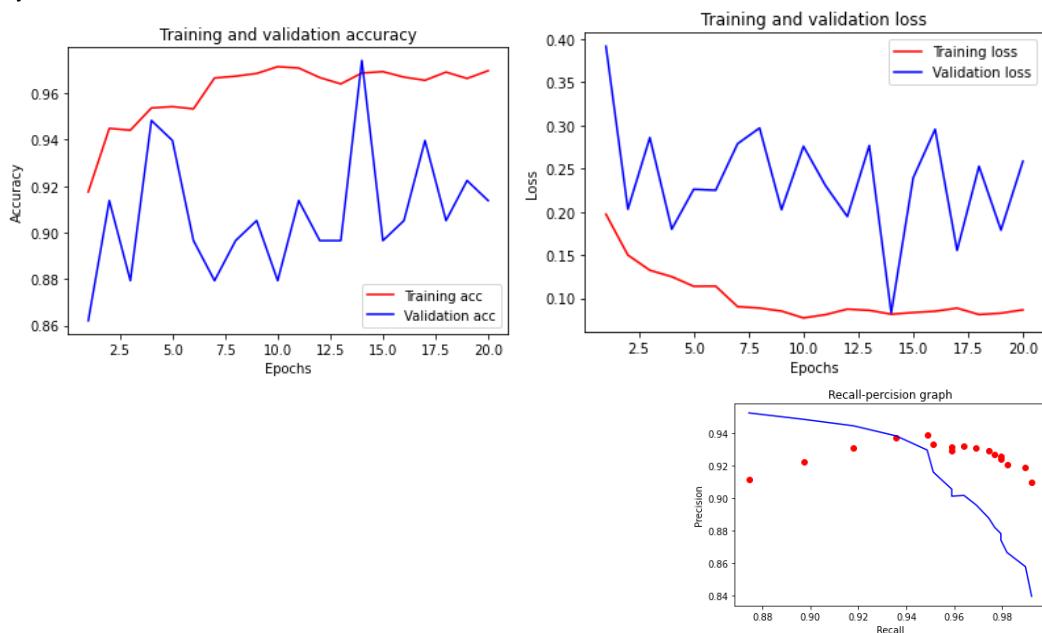


Optimizer: Adam , lr: 0.001 , EPOHCS: 20 , added layers: 2

### Optimizer: Adam , lr: 0.001 , EPOHCS: 20 , added layers: 2 ###

Model: "sequential\_1"

Layer (type)	Output Shape	Param #	Trainable
=====			
resnet50v2 (Functional)	(None, 8, 8, 2048)	23564800	Y
global_average_pooling2d_1	(None, 2048)	0	Y
(GlobalAveragePooling2D)			
dense_3 (Dense)	(None, 128)	262272	Y
dropout_2 (Dropout)	(None, 128)	0	Y
dense_4 (Dense)	(None, 128)	16512	Y
dropout_3 (Dropout)	(None, 128)	0	Y
dense_5 (Dense)	(None, 128)	16512	Y
dropout_4 (Dropout)	(None, 128)	0	Y
dense_6 (Dense)	(None, 1)	129	Y
=====			
Total params:	23,860,225		
Trainable params:	295,425		
Non-trainable params:	23,564,800		
Loss of the model is -	0.23214465379714966		
20/20 [=====] - 1s 65ms/step - loss: 0.2321 - accuracy: 0.9119			
Accuracy of the model is -	91.18589758872986 %		



The f\_score max is: 0.9390862944162437

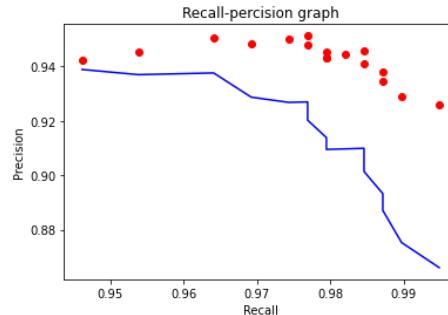
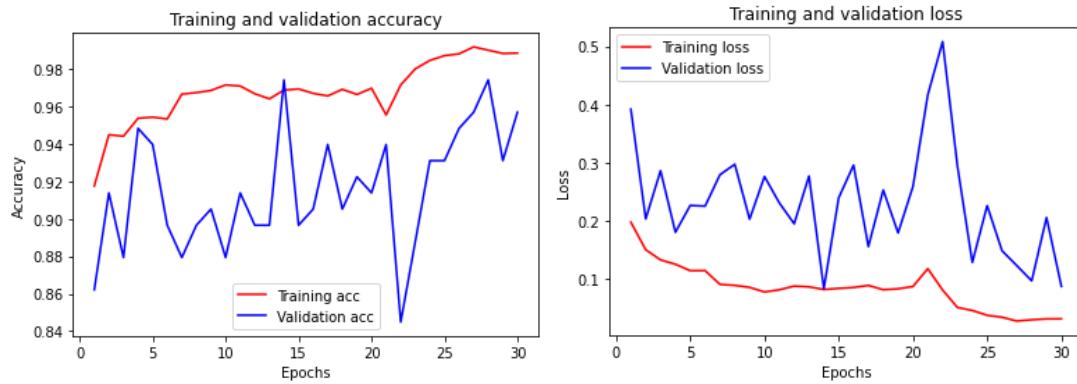
The corresponding threshold for the maximum f\_score is: 0.7000000000000001

Model: "sequential\_1"

## חזר לתוכן עניינים

Layer (type)	Output Shape	Param #	Trainable
<hr/>			
resnet50v2 (Functional)	(None, 8, 8, 2048)	23564800	Y
global_average_pooling2d_1 (None, 2048) (GlobalAveragePooling2D)		0	Y
dense_3 (Dense)	(None, 128)	262272	Y
dropout_2 (Dropout)	(None, 128)	0	Y
dense_4 (Dense)	(None, 128)	16512	Y
dropout_3 (Dropout)	(None, 128)	0	Y
dense_5 (Dense)	(None, 128)	16512	Y
dropout_4 (Dropout)	(None, 128)	0	Y
dense_6 (Dense)	(None, 1)	129	Y
<hr/>			
Total params:	23,860,225		
Trainable params:	15,529,473		
Non-trainable params:	8,330,752		

Loss of the model is - 0.27147433161735535  
 20/20 [=====] - 1s 65ms/step - loss: 0.2715 - accuracy: 0.9263  
 Accuracy of the model is - 92.62820482254028 %



The f\_score max is: 0.951310861423221  
 The corresponding threshold for the maximum f\_score is: 0.65

Optimizer: Adam , lr: 0.001 , EPOHCS: 20 , added layers: 3  
 ### Optimizer: Adam , lr: 0.001 , EPOHCS: 20 , added layers: 3 ###

Model: "sequential\_2"

Layer (type)	Output Shape	Param #	Trainable
<hr/>			
resnet50v2 (Functional)	(None, 8, 8, 2048)	23564800	Y
global_average_pooling2d_2 (None, 2048) (GlobalAveragePooling2D)		0	Y
dense_7 (Dense)	(None, 128)	262272	Y
dropout_5 (Dropout)	(None, 128)	0	Y
dense_8 (Dense)	(None, 128)	16512	Y

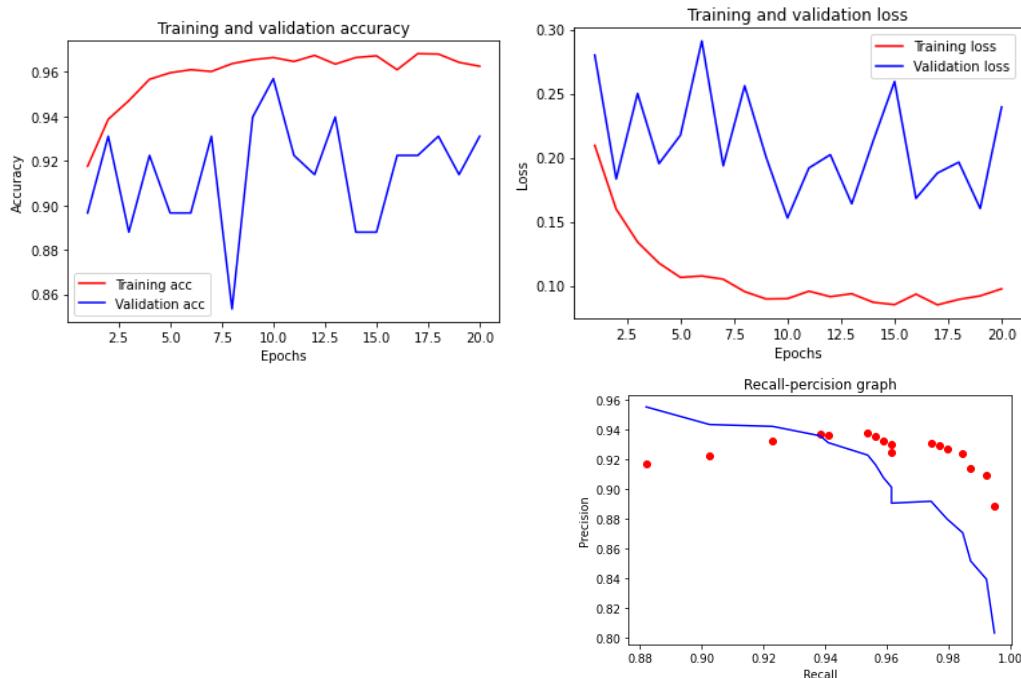
## חזר לתוכן עניינים

dropout_6 (Dropout)	(None, 128)	0	Y
dense_9 (Dense)	(None, 128)	16512	Y
dropout_7 (Dropout)	(None, 128)	0	Y
dense_10 (Dense)	(None, 128)	16512	Y
dropout_8 (Dropout)	(None, 128)	0	Y
dense_11 (Dense)	(None, 1)	129	Y

---

```
Total params: 23,876,737
Trainable params: 311,937
Non-trainable params: 23,564,800
```

The history is now!  
Loss of the model is - 0.24112319946289062  
20/20 [=====] - 1s 65ms/step - loss: 0.2411 - accuracy: 0.9103  
Accuracy of the model is - 91.02563858032227 %



The f\_score max is: 0.9382093316519547  
The corresponding threshold for the maximum f\_score is: 0.65  
Model: "sequential\_2"

---

Layer (type)	Output Shape	Param #	Trainable
resnet50v2 (Functional)	(None, 8, 8, 2048)	23564800	Y
global_average_pooling2d_2 (None, 2048)	0	Y	(GlobalAveragePooling2D)
dense_7 (Dense)	(None, 128)	262272	Y
dropout_5 (Dropout)	(None, 128)	0	Y
dense_8 (Dense)	(None, 128)	16512	Y
dropout_6 (Dropout)	(None, 128)	0	Y
dense_9 (Dense)	(None, 128)	16512	Y
dropout_7 (Dropout)	(None, 128)	0	Y
dense_10 (Dense)	(None, 128)	16512	Y
dropout_8 (Dropout)	(None, 128)	0	Y
dense_11 (Dense)	(None, 1)	129	Y

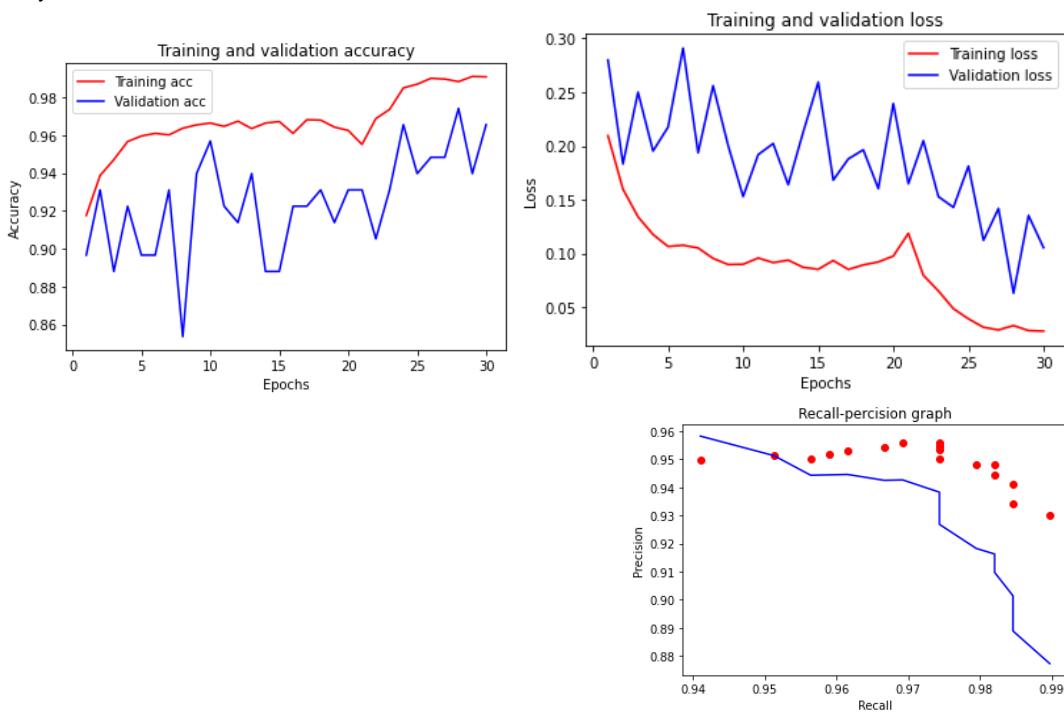
---

```
Total params: 23,876,737
Trainable params: 15,545,985
Non-trainable params: 8,330,752
```

---

## חזר לתוכן עניינים

Loss of the model is - 0.2136000394821167  
 20/20 [=====] - 1s 65ms/step - loss: 0.2136 - accuracy: 0.9423  
 Accuracy of the model is - 94.2307710647583 %



The f\_score max is: 0.9559748427672956  
 The corresponding threshold for the maximum f\_score is: 0.55

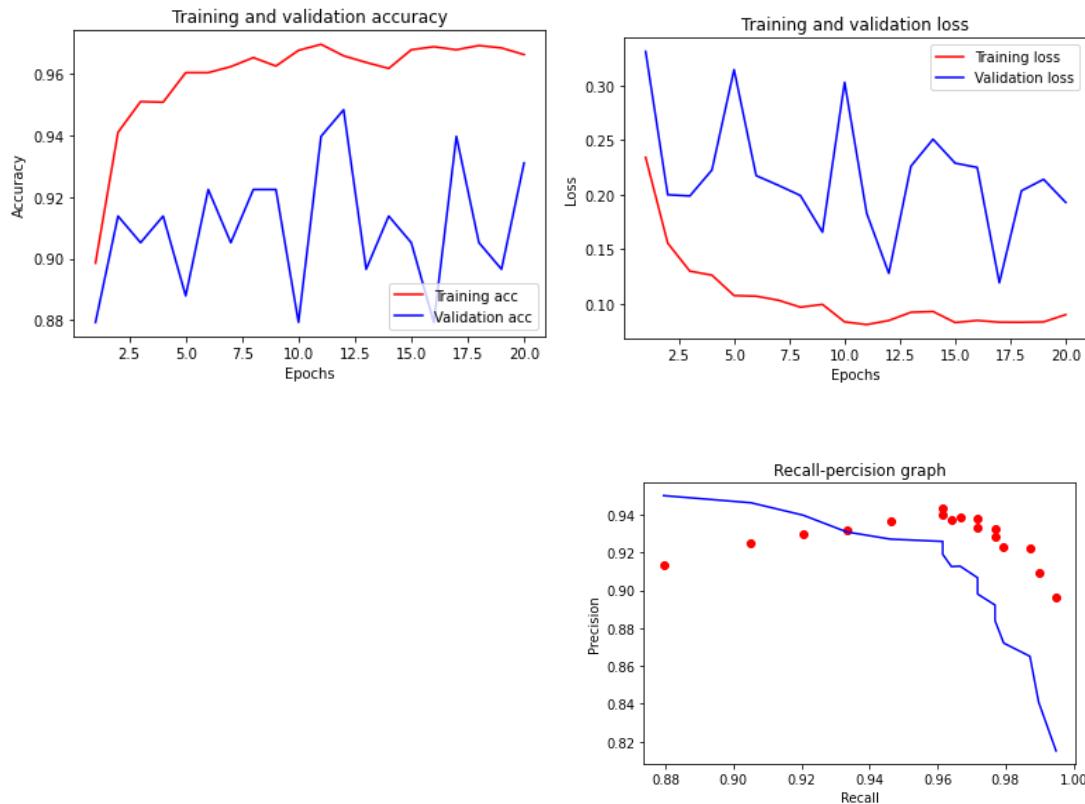
Optimizer: Adam , lr: 0.001 , EPOHCS: 20 , added layers:4  
 ### Optimizer: Adam , lr: 0.001 , EPOHCS: 20 , added layers: 4 ###

Model: "sequential\_3"

Layer (type)	Output Shape	Param #	Trainable
<hr/>			
resnet50v2 (Functional)	(None, 8, 8, 2048)	23564800	Y
<hr/>			
global_average_pooling2d_3 (None, 2048) (GlobalAveragePooling2D)		0	Y
dense_12 (Dense)	(None, 128)	262272	Y
dropout_9 (Dropout)	(None, 128)	0	Y
dense_13 (Dense)	(None, 128)	16512	Y
dropout_10 (Dropout)	(None, 128)	0	Y
dense_14 (Dense)	(None, 128)	16512	Y
dropout_11 (Dropout)	(None, 128)	0	Y
dense_15 (Dense)	(None, 128)	16512	Y
dropout_12 (Dropout)	(None, 128)	0	Y
dense_16 (Dense)	(None, 128)	16512	Y
dropout_13 (Dropout)	(None, 128)	0	Y
dense_17 (Dense)	(None, 1)	129	Y
<hr/>			
Total params:	23,893,249		
Trainable params:	328,449		
Non-trainable params:	23,564,800		

The history is now!  
 Loss of the model is - 0.23351943492889404  
 20/20 [=====] - 1s 66ms/step - loss: 0.2335 - accuracy: 0.9215  
 Accuracy of the model is - 92.14743375778198 %

## חומר לתוכן עניינים



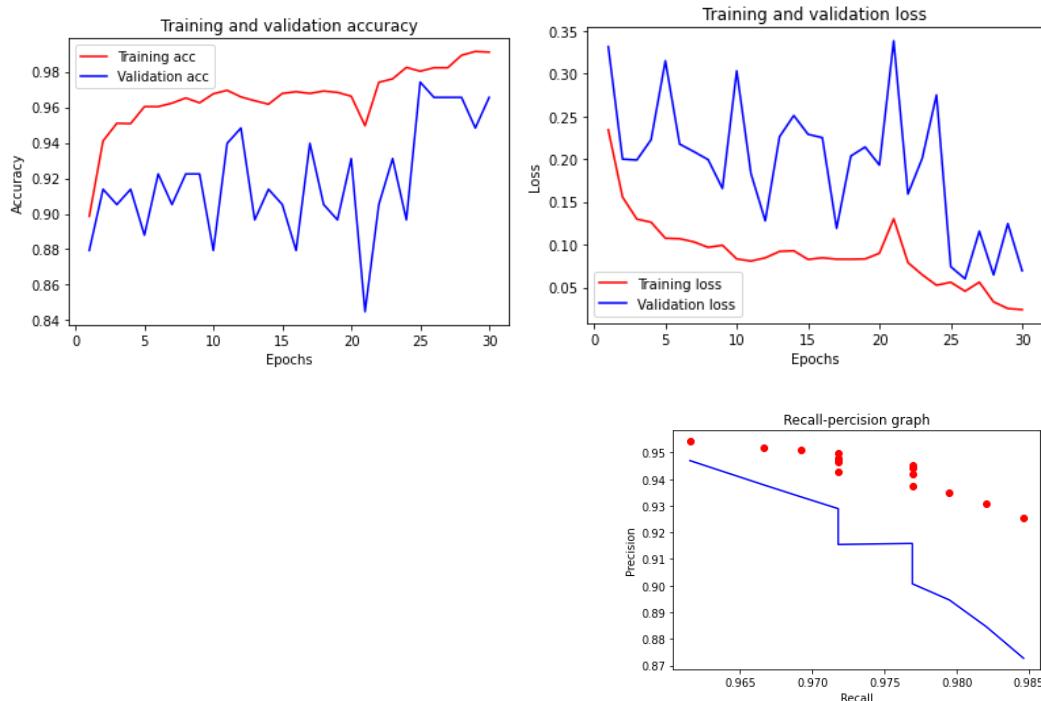
The f\_score max is: 0.9433962264150944

The corresponding threshold for the maximum f\_score is: 0.65

Model: "sequential\_3"

Layer (type)	Output Shape	Param #	Trainable
<hr/>			
resnet50v2 (Functional)	(None, 8, 8, 2048)	23564800	Y
<hr/>			
global_average_pooling2d_3 (None, 2048)	(GlobalAveragePooling2D)	0	Y
dense_12 (Dense)	(None, 128)	262272	Y
dropout_9 (Dropout)	(None, 128)	0	Y
dense_13 (Dense)	(None, 128)	16512	Y
dropout_10 (Dropout)	(None, 128)	0	Y
dense_14 (Dense)	(None, 128)	16512	Y
dropout_11 (Dropout)	(None, 128)	0	Y
dense_15 (Dense)	(None, 128)	16512	Y
dropout_12 (Dropout)	(None, 128)	0	Y
dense_16 (Dense)	(None, 128)	16512	Y
dropout_13 (Dropout)	(None, 128)	0	Y
dense_17 (Dense)	(None, 1)	129	Y
<hr/>			
Total params:	23,893,249		
Trainable params:	15,562,497		
Non-trainable params:	8,330,752		
<hr/>			
Loss of the model is -	0.2908949851989746		
20/20 [=====] - 1s 65ms/step - loss: 0.2909 - accuracy: 0.9295			
Accuracy of the model is -	92.94871687889099 %		

## חזר לתוכן עניינים



The f\_score max is: 0.9541984732824428

The corresponding threshold for the maximum f\_score is: 0.9

Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , added layers: 5

Mounted at /content/drive

### CNN transfer learning network Model ###

### mission\_31B: add 5 hidden layers ###

### Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , added layers: 5 ###

Downloading data from [https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50v2\\_weights\\_tf\\_dim\\_ordering\\_tf\\_kernels\\_notop.h5](https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50v2_weights_tf_dim_ordering_tf_kernels_notop.h5)

94674944/94668760 [=====] - 1s 0us/step

94683136/94668760 [=====] - 1s 0us/step

Model: "sequential"

Layer (type)	Output Shape	Param #	Trainable
<hr/>			
resnet50v2 (Functional)	(None, 8, 8, 2048)	23564800	Y
global_average_pooling2d (G (None, 2048)	0	Y	
lobalAveragePooling2D)			
dense (Dense)	(None, 128)	262272	Y
dropout (Dropout)	(None, 128)	0	Y
dense_1 (Dense)	(None, 128)	16512	Y
dropout_1 (Dropout)	(None, 128)	0	Y
dense_2 (Dense)	(None, 128)	16512	Y
dropout_2 (Dropout)	(None, 128)	0	Y
dense_3 (Dense)	(None, 128)	16512	Y
dropout_3 (Dropout)	(None, 128)	0	Y
dense_4 (Dense)	(None, 128)	16512	Y
dropout_4 (Dropout)	(None, 128)	0	Y
dense_5 (Dense)	(None, 128)	16512	Y
dropout_5 (Dropout)	(None, 128)	0	Y

## חזר לתוכן עניינים

dense\_6 (Dense) (None, 1) 129 Y

```
=====
Total params: 23,909,761
Trainable params: 344,961
Non-trainable params: 23,564,800
```

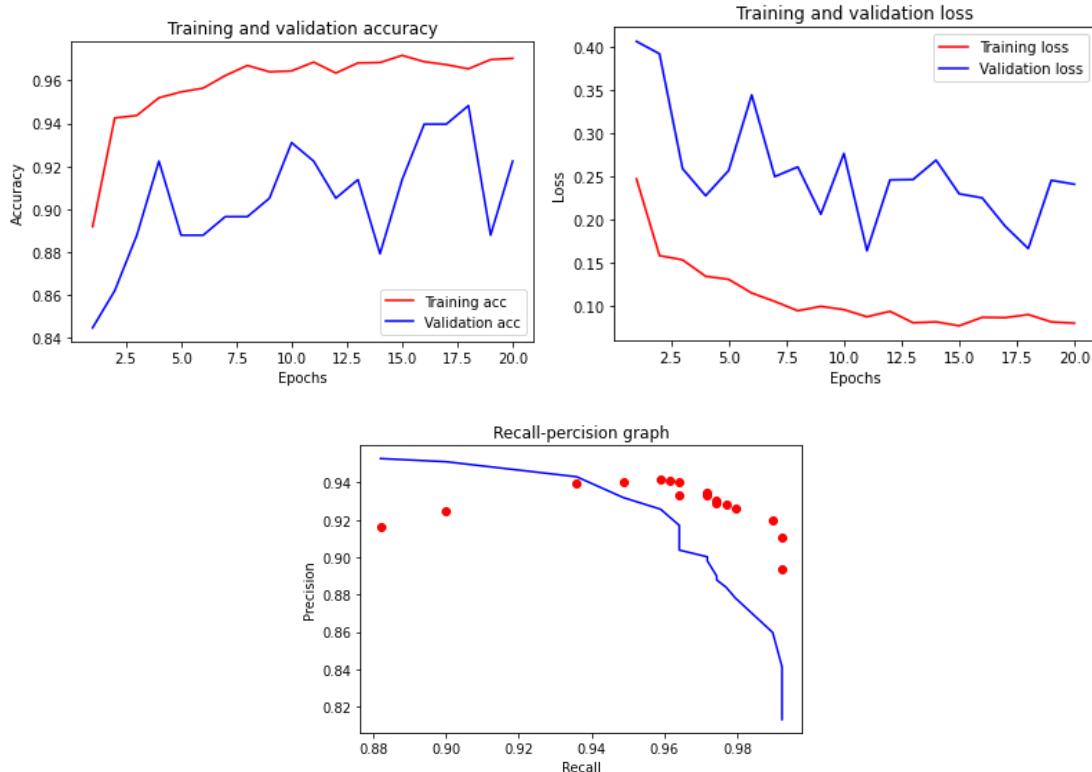
The history is now!

20/20 [=====] - 2s 77ms/step - loss: 0.2382 - accuracy: 0.9151

Loss of the model is - 0.23815812170505524

20/20 [=====] - 1s 65ms/step - loss: 0.2382 - accuracy: 0.9151

Accuracy of the model is - 91.50640964508057 %



The f\_score max is: 0.9420654911838792

The corresponding threshold for the maximum f\_score is: 0.7000000000000001

Model: "sequential"

Layer (type)	Output Shape	Param #	Trainable
--------------	--------------	---------	-----------

```
=====
resnet50v2 (Functional) (None, 8, 8, 2048) 23564800 Y
```

global\_average\_pooling2d (G (None, 2048) 0 Y  
lobalAveragePooling2D)

dense (Dense) (None, 128) 262272 Y

dropout (Dropout) (None, 128) 0 Y

dense\_1 (Dense) (None, 128) 16512 Y

dropout\_1 (Dropout) (None, 128) 0 Y

dense\_2 (Dense) (None, 128) 16512 Y

dropout\_2 (Dropout) (None, 128) 0 Y

dense\_3 (Dense) (None, 128) 16512 Y

dropout\_3 (Dropout) (None, 128) 0 Y

dense\_4 (Dense) (None, 128) 16512 Y

dropout\_4 (Dropout) (None, 128) 0 Y

dense\_5 (Dense) (None, 128) 16512 Y

dropout\_5 (Dropout) (None, 128) 0 Y

dense\_6 (Dense) (None, 1) 129 Y

```
=====
Total params: 23,909,761
```

## חזר לתוכן עניינים

Trainable params: 15,579,009  
Non-trainable params: 8,330,752

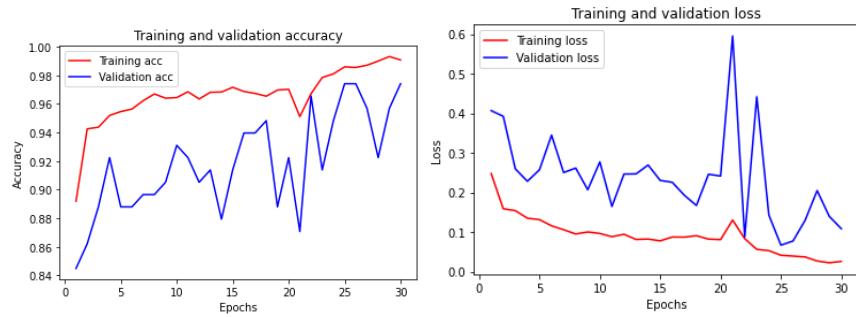
The history is return!

20/20 [=====] - 2s 65ms/step - loss: 0.2421 - accuracy: 0.9343

Loss of the model is - 0.2421025186777115

20/20 [=====] - 1s 65ms/step - loss: 0.2421 - accuracy: 0.9343

Accuracy of the model is - 93.42948794364929 %



## 3.2 מנגנון

2. בדקו את ביצועי הרשותות עם אלגוריתמי האימון הבאים (בדקו השפעת מספר ה EPOCHS וה- LEARNING RATE לכל אלגוריתם):

- אלגוריתם SGD
- אלגוריתם MOMENTUM
- אלגוריתם ADAM
- אלגוריתם RMSPROP

### רשות א'

#### סיכום התוצאות

ריכזו את תוצאות של כלל הרכזות שבוצעו בסעיף זה:

רשות א'						
Loss	Accuracy%	F Score%	Threshold	lr	epoch	
0.70	62.5	76.90	0.10	0.1	8	Adam
0.74	77.88	89.90	0.90	0.01	8	
0.53	75.00	88.60	0.15	0.001	8	
0.60	77.24	84.79	0.55	0.1	16	
0.29	91.02	92.98	0.50	0.01	16	
<b>0.25</b>	<b>92.14</b>	<b>94.41</b>	<b>0.70</b>	<b>0.001</b>	<b>16</b>	
0.66	64.90	81.50	0.20	0.1	24	
	89.90	92.50	0.60	0.01	24	
	91.00	93.50	0.80	0.001	24	
0.77	62.50	76.87	0.10	0.1	8	
<b>1.37</b>	67.63	75.58	0.10	0.01	8	RMSPROP
0.55	75.32	87.00	0.15	0.001	8	
0.70	62.50	76.90	0.10	0.1	16	
0.72	88.30	93.45	0.90	0.01	16	
<b>0.27</b>	<b>92.47</b>	<b>94.30</b>	<b>0.60</b>	<b>0.001</b>	<b>16</b>	
0.70	62.50	76.90	0.10	0.1	24	
	89.10	92.80	0.80	0.01	24	
0.30	91.50	93.44	0.50	0.001	24	
0.59	79.00	92.53	0.90	0.1	8	SGD
0.58	82.53	91.67	0.80	0.01	8	
0.54	80.44	91.40	0.85	0.001	8	
0.29	89.58	92.92	0.65	0.1	16	
0.36	90.70	93.44	0.60	0.01	16	
0.41	83.97	91.59	0.80	0.001	16	
0.54	86.53	91.86		0.1	24	
0.43	87.66	92.82	0.80	0.01	24	
0.40	86.22	91.56	0.75	0.001	24	
0.69	62.50	76.90	0.10	0.1	8	SGD with momentum
	57.21	72.75	0.10	0.01	8	
	75.80	91.06	0.90	0.001	8	
	62.50	76.90	0.10	0.1	16	
	86.38	92.09	0.90	0.01	16	

## חזר לתוכן עניינים

	88.62	92.36	0.70	0.001	16	
1.23	6.5			0.1	24	
	90.38	93.54	0.65	0.01	24	
	88.94	92.94	0.80	0.001	24	
1.37	92.47	94.41	0.90	<b>max</b>		
0.25	57.21	72.75	0.10	<b>min</b>		

- |            |  |
|------------|--|
| best case  |  |
| worse case |  |
1. התאים הירוקים מייצגים את המקorra הטוב ביותר (הכי רצוי).  
 2. התאים האדומים מייצגים את המקorra חci גרווע (הכי לא רצוי).  
 3. ניתן לראות שהרשנות עם הפרמטרים המסומנים ב**הציגה כחוליה** -  
 קיבלו את דיקט גבואה, F score��, וכן גם loss נמוך מאד.  
 משום ששינה התלבטות מי הרשות הנבחרת ביצועים, הגיענו למסקנה  
 שניקח את הרשות עם F score הגבוה ו loss הנמוך יותר, מאחר והיא "ניצחה" בקטגוריות  
 הפרמטרים הרצויים.
- Optimizer: Adam , lr: 0.001 , EPOHCS: 16  
 זו גם רשות הרפרנס שלנו.

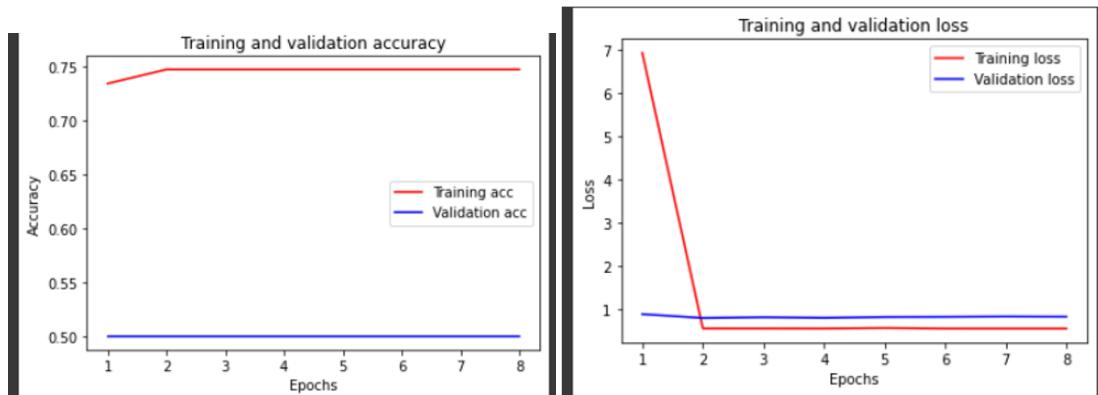
## פירוט תוצאות

Optimizer: Adam , lr: 0.1 , EPOHCS: 8

Accuracy of the model is - **62.5%**

Loss of the model is - **0.695**

F Score max of the model is - **76.9%** for **0.1** threshold



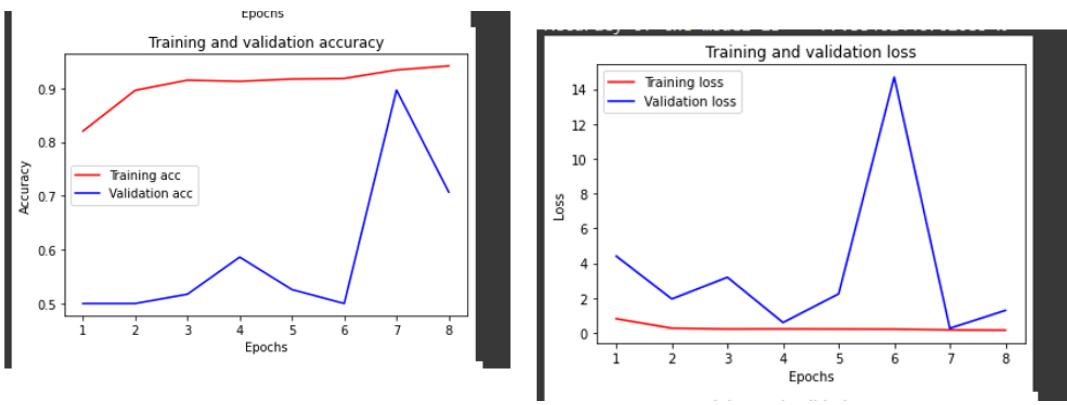
Optimizer: Adam , lr: 0.01 , EPOHCS: 8

Accuracy of the model is - **77.884%**

Loss of the model is - **0.742**

F Score max of the model is - **89.9%** for **0.9** threshold

## חזר לתוכן עניינים

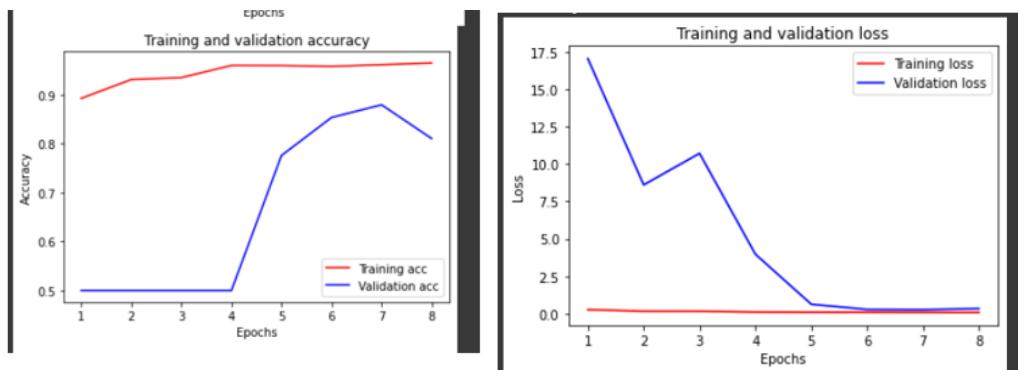


Optimizer: Adam , lr: 0.001 , EPOCHS: 8

Accuracy of the model is - **75%**

Loss of the model is - **0.53**

F Score max of the model is - **88.6%** for **0.15** threshold

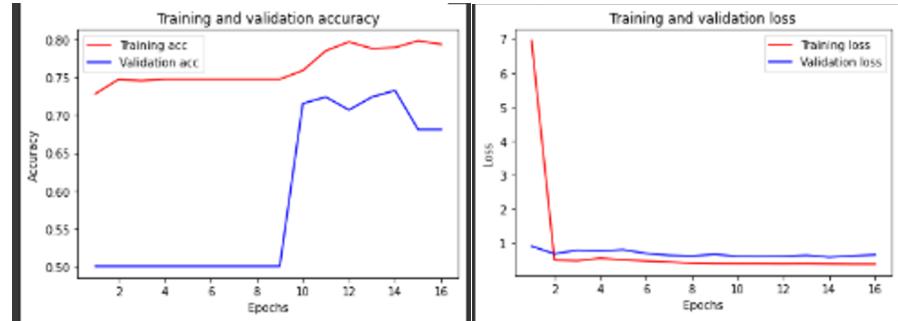


Optimizer: Adam , lr: 0.1 , EPOCHS: 16

Accuracy of the model is - **77.24%**

Loss of the model is - **0.596**

F Score max of the model is - **84.79%** for **0.55** threshold



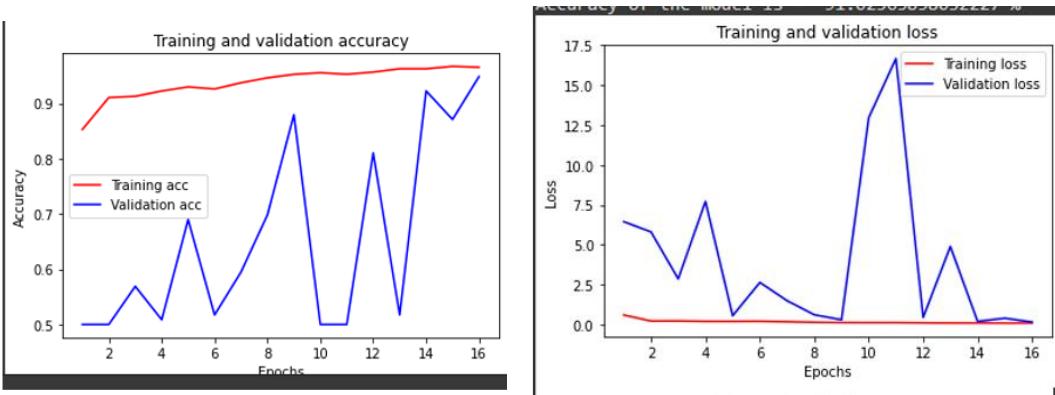
Optimizer: Adam , lr: 0.01 , EPOCHS: 16

Accuracy of the model is - **91.02%**

Loss of the model is - **0.29**

F Score max of the model is - **92.98%** for **0.5** threshold

## חומר לתוכן עניינים

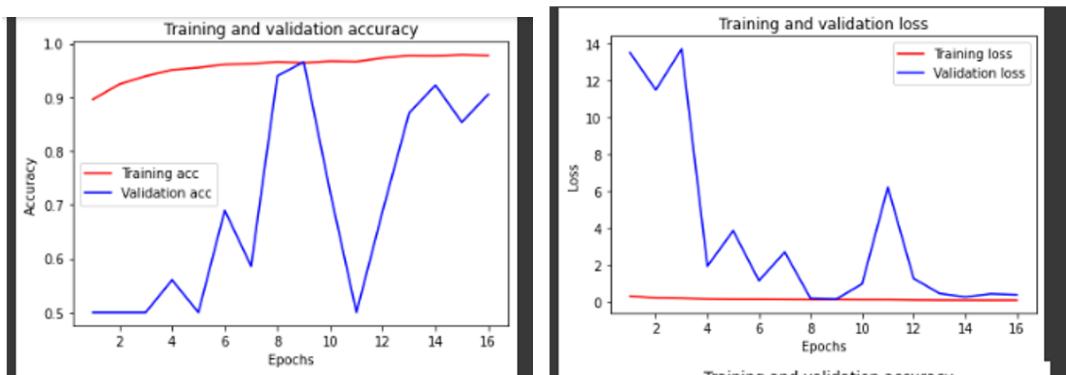


Optimizer: Adam , lr: 0.001 , EPOCHS: 16

Accuracy of the model is - **92.14%**

Loss of the model is – **0.249**

F Score max of the model is - **94.41%** for **0.7** threshold



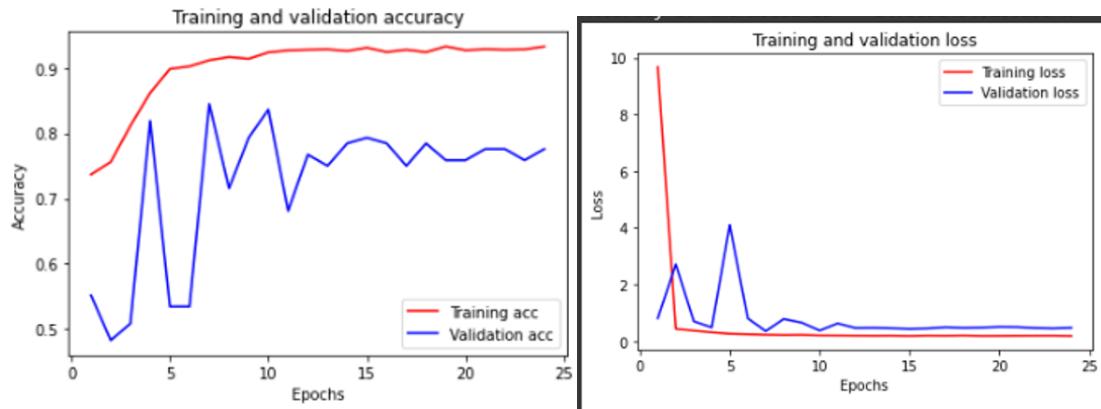
Optimizer: Adam , lr: 0.1 , EPOCHS: 24

Accuracy of the model is - **64.9%**

Loss of the model is – **0.658**

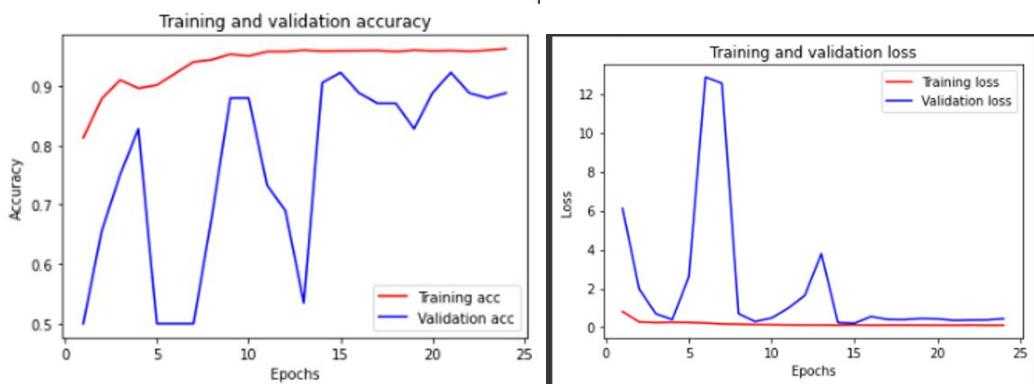
F Score max of the model is - **81.5%** for **0.2** threshold

## חומר לתוכן עניינים



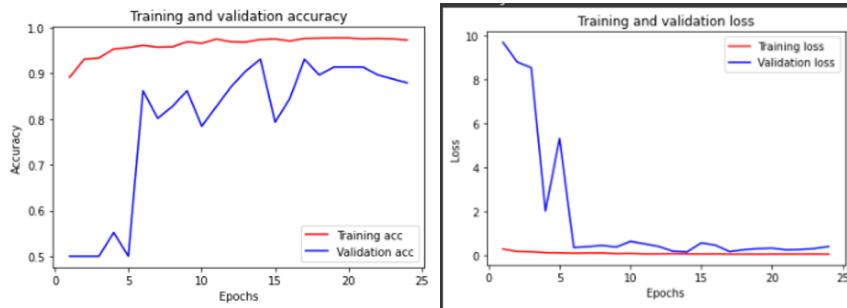
Optimizer: Adam , lr: 0.01 , EPOCHS: 24

Accuracy of the model is - **89.9%**  
F Score max of the model is - **92.5%** for **0.6** threshold



Optimizer: Adam , lr: 0.001 , EPOCHS: 24

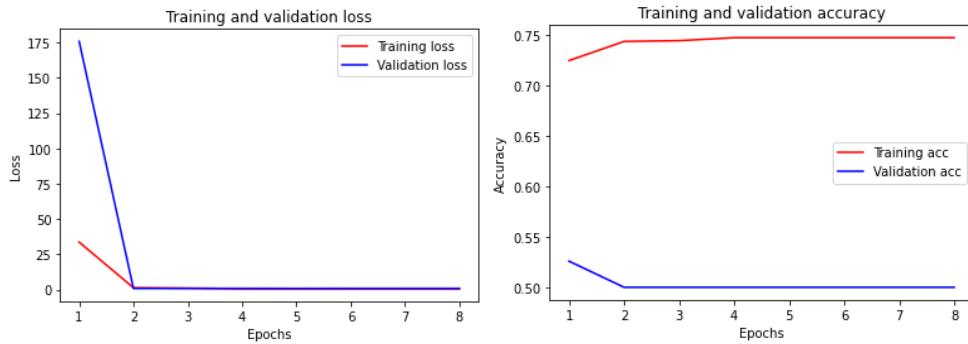
Accuracy of the model is - **91%**  
F Score max of the model is - **93.5%** for **0.8** threshold



Optimizer: RMSprop , lr: 0.1 , EPOCHS: 8

Accuracy of the model is - **62.5%**  
Loss of the model is - **0.773**  
F Score max of the model is - **76.8%** for **0.1** threshold

## חזר לתוכן עניינים

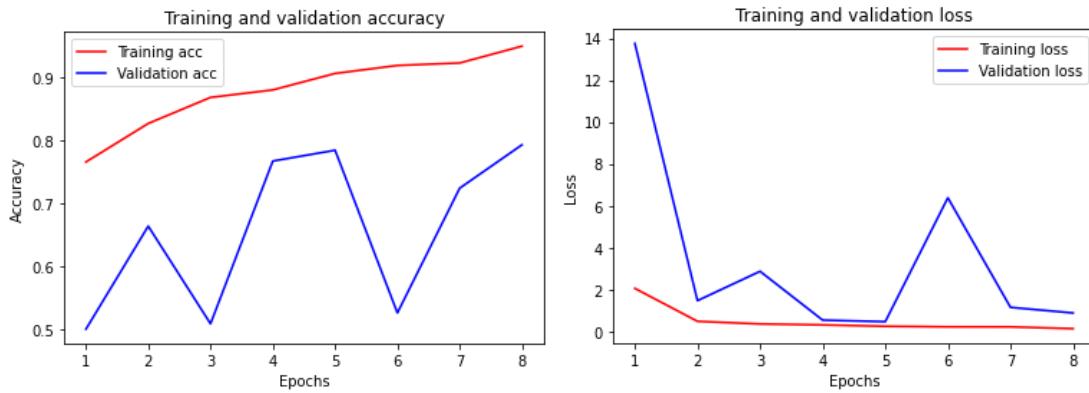


Optimizer: RMSprop , lr: 0.01 , EPOCHS: 8

Accuracy of the model is - **67.7%**

Loss of the model is – **1.372**

F Score max of the model is - **75.5%** for **0.1** threshold

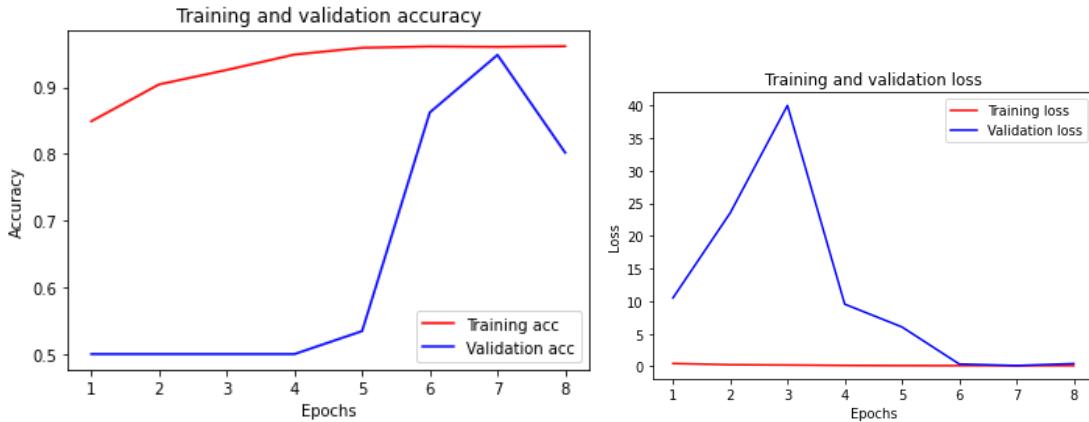


Optimizer: RMSprop , lr: 0.001 , EPOCHS: 8

Accuracy of the model is - **75.3%**

Loss of the model is – **0.550**

F Score max of the model is - **87%** for **0.15** threshold



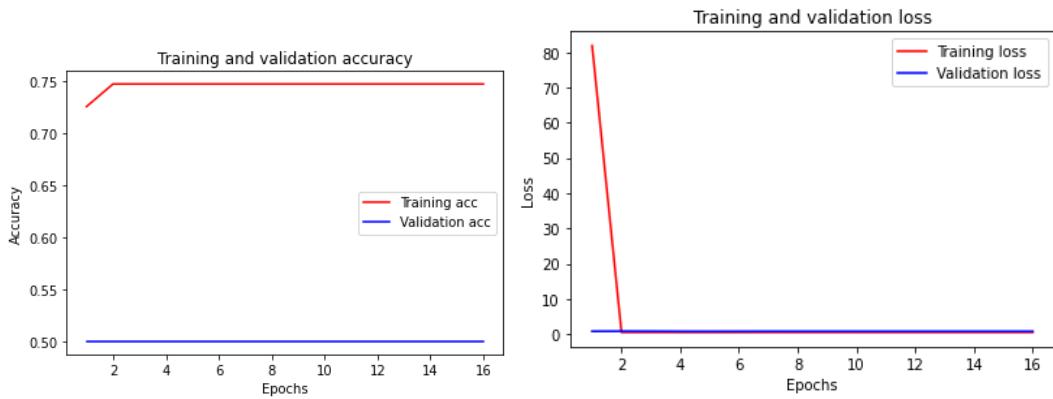
Optimizer: RMSprop , lr: 0.1 , EPOCHS: 16

Accuracy of the model is - **62.5%**

Loss of the model is – **0.698**

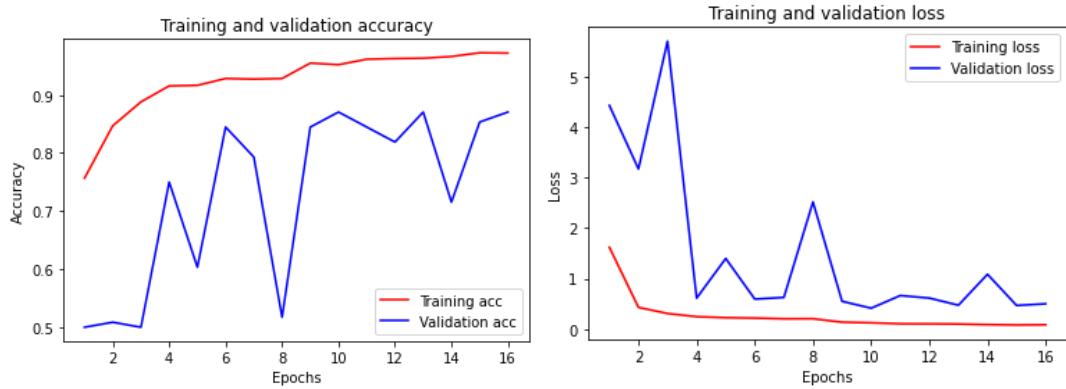
F Score max of the model is - **76.9%** for **0.1** threshold

## חזר לתוכן עניינים



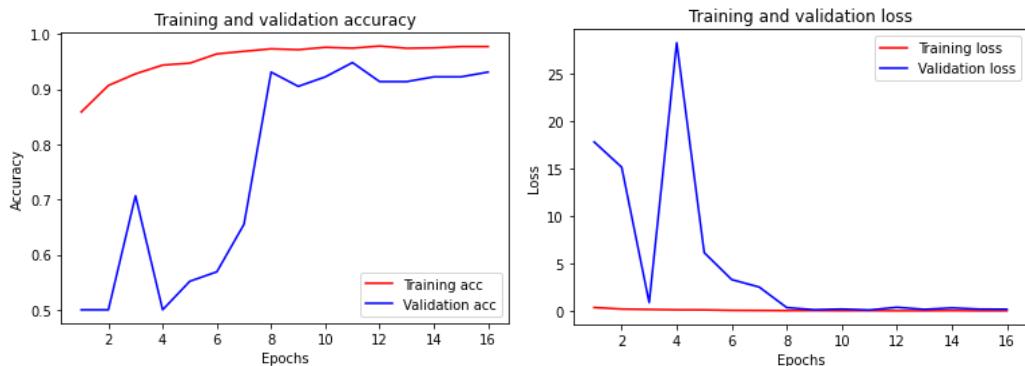
Optimizer: RMSprop , lr: 0.01 , EPOHCS: 16

Accuracy of the model is - **88.3%**  
 Loss of the model is - **0.717**  
 threshold **0.9** for F Score max of the model is - **93.4%**



Optimizer: RMSprop , lr: 0.001 , EPOHCS: 16

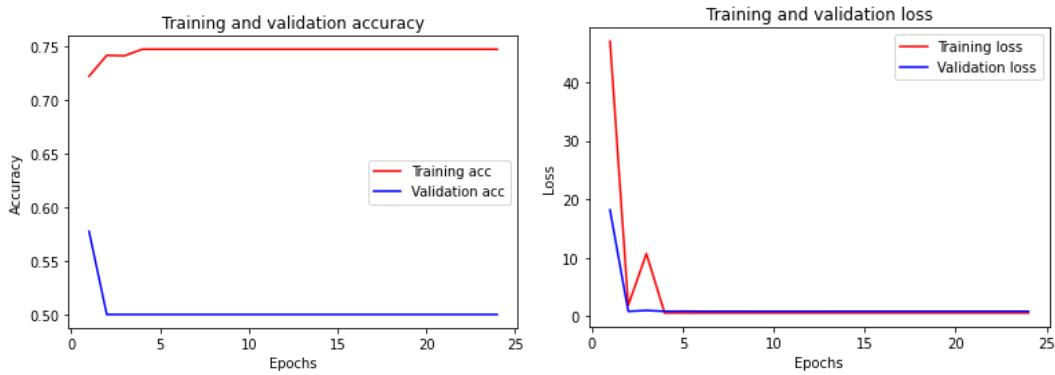
Accuracy of the model is - **92.46%**  
 Loss of the model is - **0.268**  
 threshold **0.6** for F Score max of the model is - **94.3%**



Optimizer: RMSprop , lr: 0.1 , EPOHCS: 24

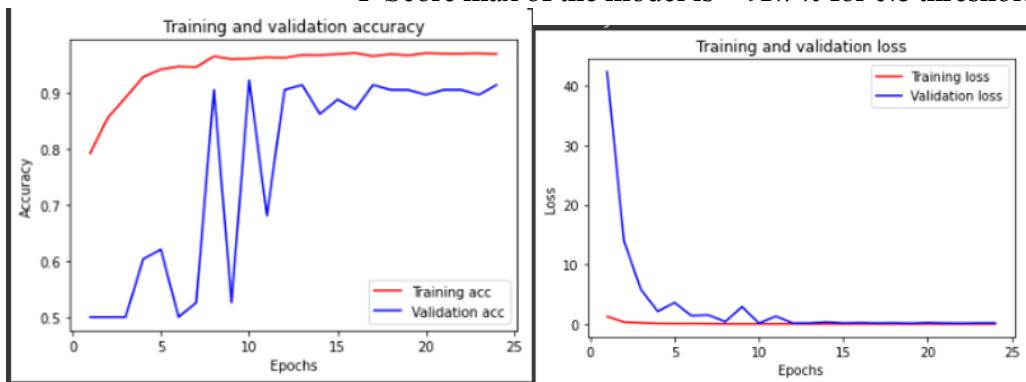
Accuracy of the model is - **62.5%**  
 Loss of the model is - **0.698**  
 F Score max of the model is - **76.9%** for **0.1** threshold

## חזר לתוכן עניינים



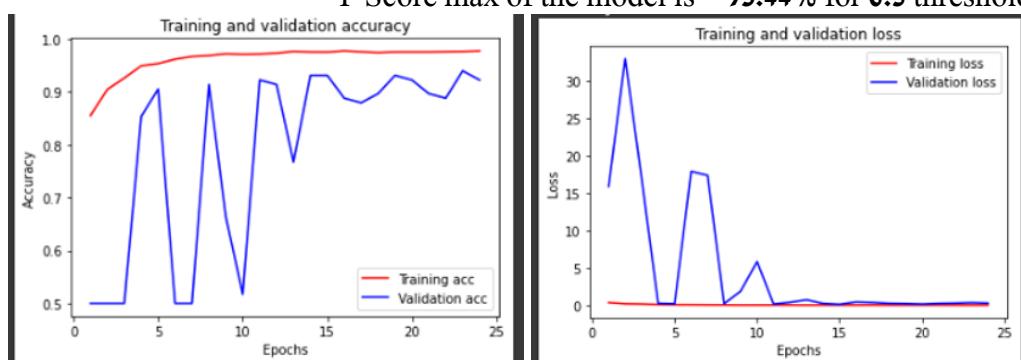
Optimizer: RMSprop , lr: 0.01 , EPOCHS: 24

Accuracy of the model is - **89.1%**  
F Score max of the model is - **92.7%** for **0.8** threshold



Optimizer: RMSprop , lr: 0.001 , EPOCHS: 24

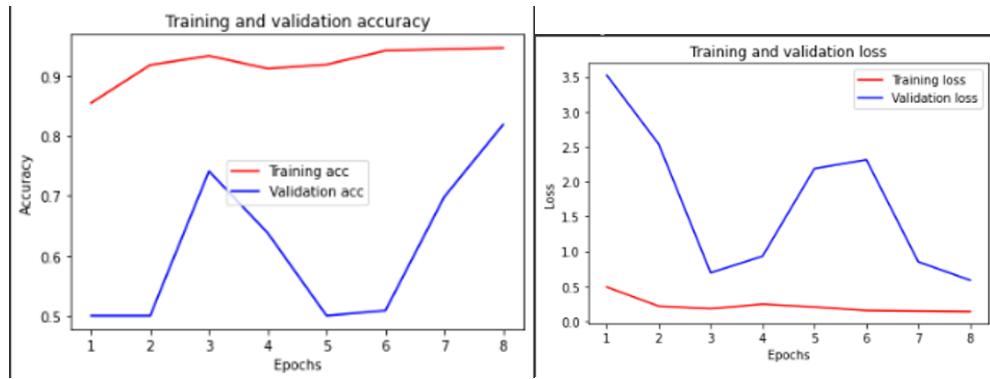
Accuracy of the model is - **91.5%**  
Loss of the model is - **0.297**  
F Score max of the model is - **93.44%** for **0.5** threshold



Optimizer: SGD , lr: 0.1 , EPOCHS: 8

Accuracy of the model is - **79%**  
Loss of the model is - **0.585**  
F Score max of the model is - **92.53%** for **0.9** threshold

## חזר לתוכן עניינים

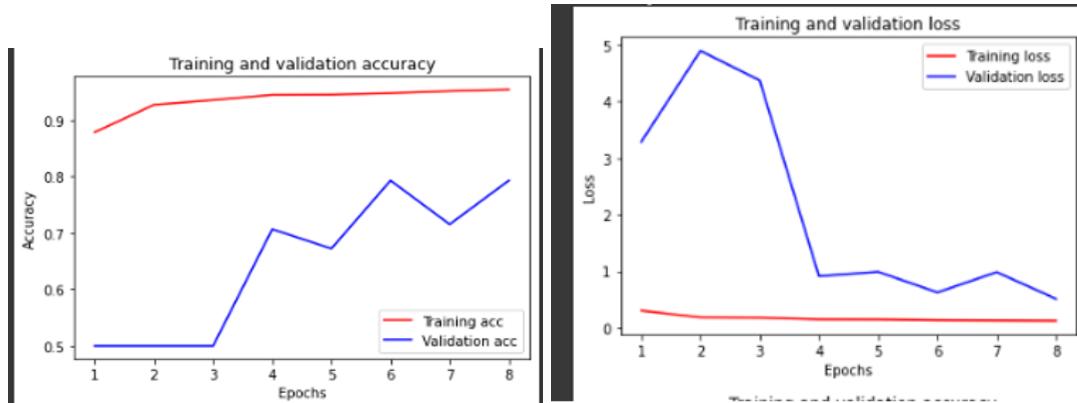


Optimizer: SGD , lr: 0.01 , EPOCHS: 8

Accuracy of the model is - **82.53%**

Loss of the model is - **0.576**

F Score max of the model is - **91.67%** for **0.8** threshold

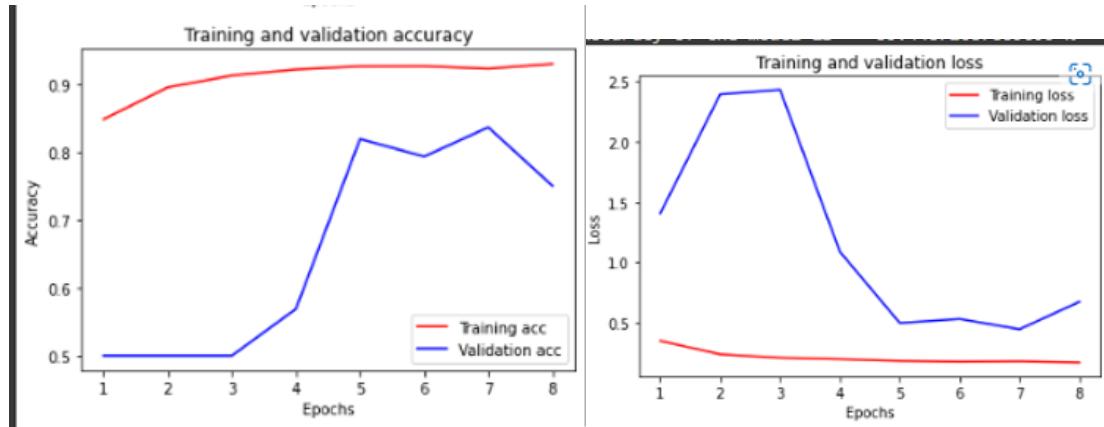


Optimizer: SGD , lr: 0.001 , EPOCHS: 8

Accuracy of the model is - **80.44%**

Loss of the model is - **0.543**

F Score max of the model is - **91.4%** for **0.85** threshold



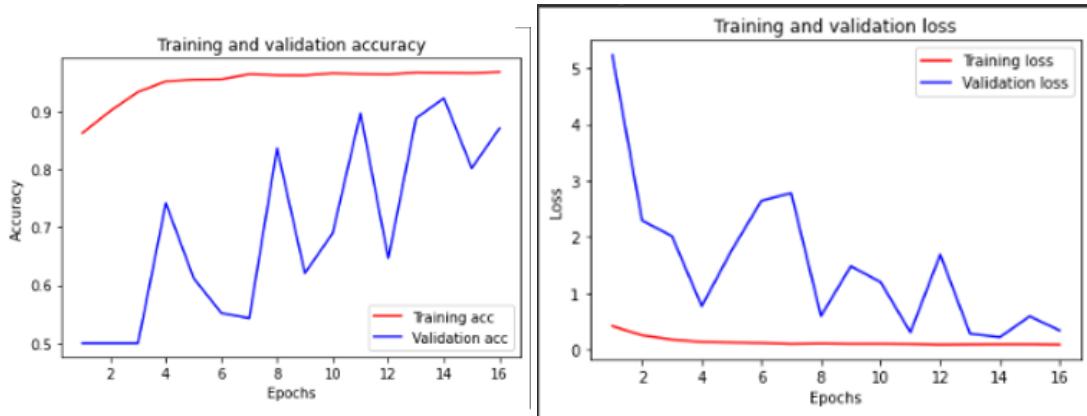
Optimizer: SGD , lr: 0.1 , EPOCHS: 16

Accuracy of the model is - **89.58%**

Loss of the model is - **0.292**

F Score max of the model is - **92.92%** for **0.65** threshold

## חזר לתוכן עניינים

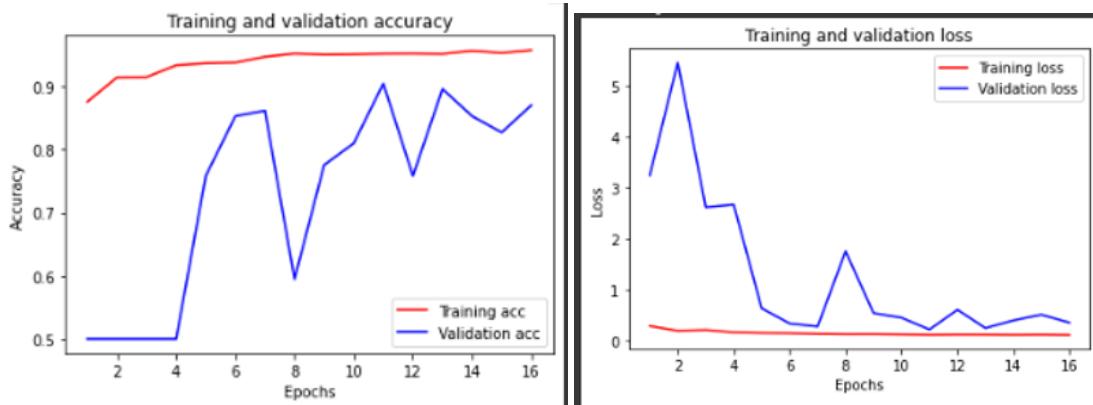


Optimizer: SGD , lr: 0.01 , EPOCHS: 16

Accuracy of the model is - **90.7%**

Loss of the model is - **0.355**

F Score max of the model is - **93.44%** for **0.6** threshold

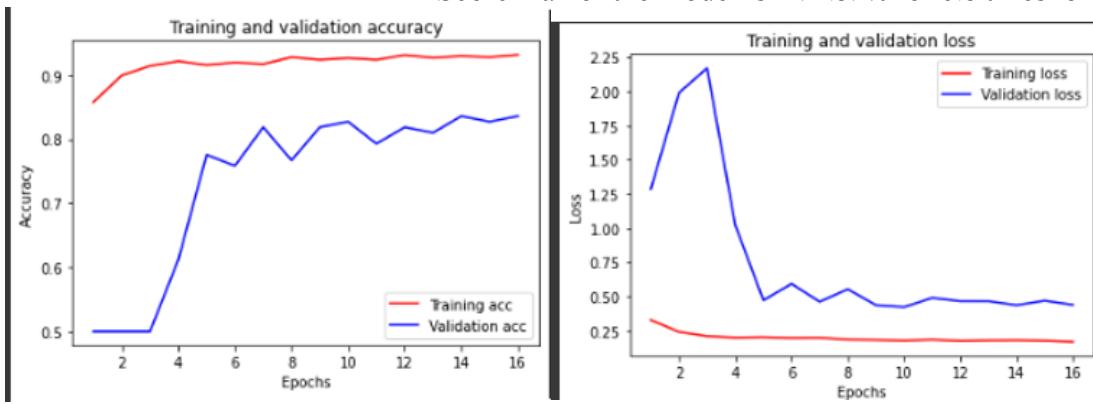


Optimizer: SGD , lr: 0.001 , EPOCHS: 16

Accuracy of the model is - **83.97%**

Loss of the model is - **0.41**

F Score max of the model is - **91.59%** for **0.8** threshold



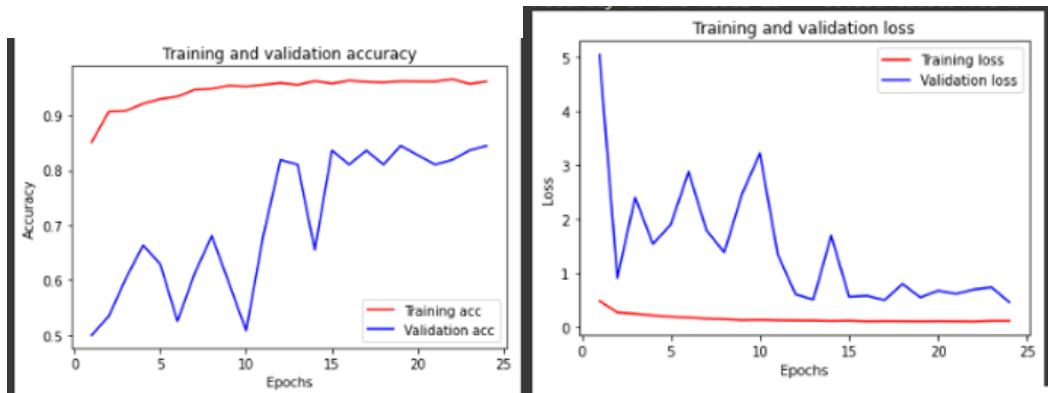
Optimizer: SGD , lr: 0.1 , EPOCHS: 24

Accuracy of the model is - **86.53%**

Loss of the model is - **0.541**

F Score max of the model is - **91.86%**

## חומר לתוכן עניינים

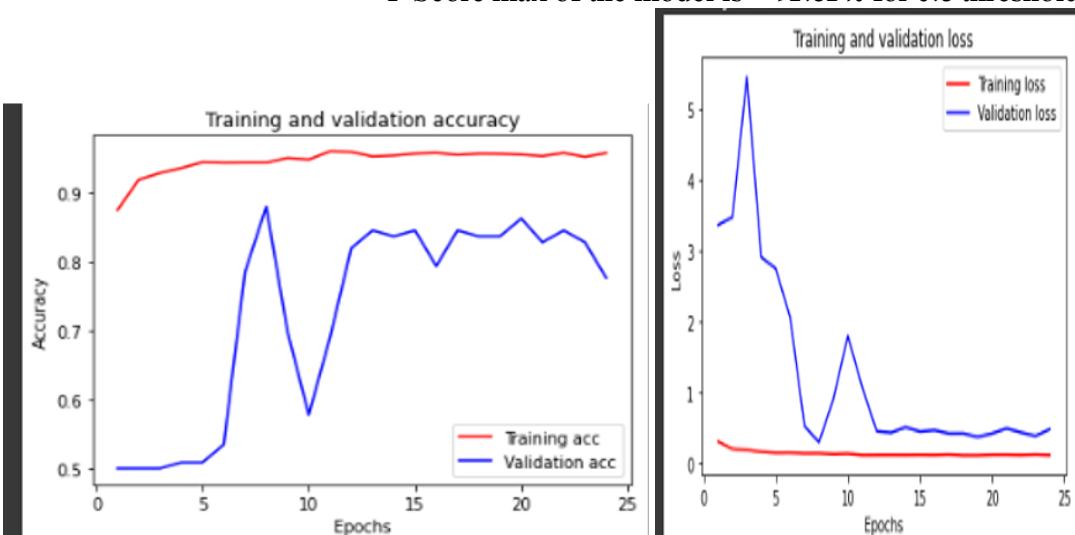


Optimizer: SGD , lr: 0.01 , EPOCHS: 24

Accuracy of the model is - **87.66%**

Loss of the model is - **0.427**

F Score max of the model is - **92.82%** for **0.8** threshold

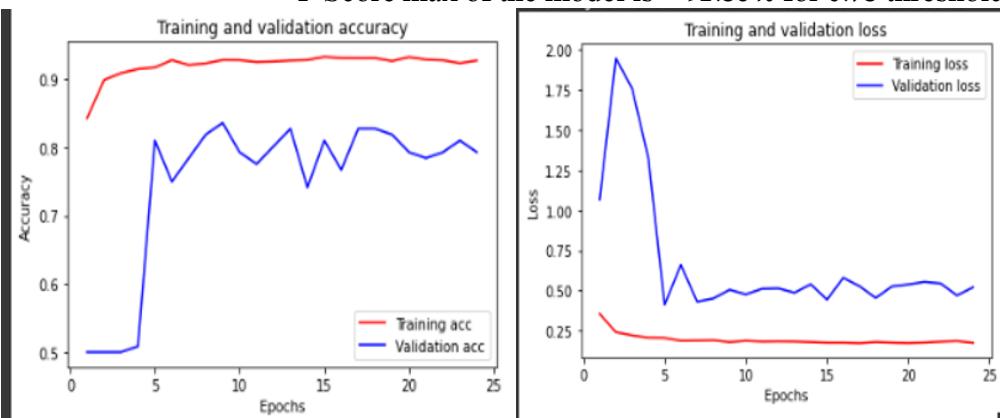


Optimizer: SGD , lr: 0.001 , EPOCHS: 24

Accuracy of the model is - **86.22%**

Loss of the model is - **0.401**

F Score max of the model is - **91.56%** for **0.75** threshold



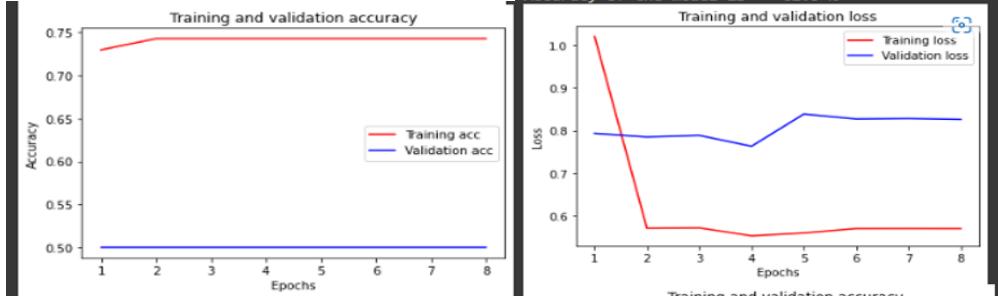
## חזר לתוכן עניינים

Optimizer: MSGD , lr: 0.1 , EPOHCS: 8

Accuracy of the model is - **62.5%**

Loss of the model is - **0.694**

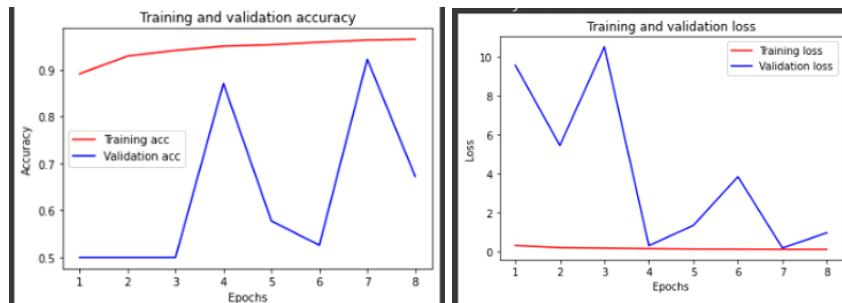
F Score max of the model is - **76.9%** for **0.1** threshold



Optimizer: MSGD , lr: 0.01 , EPOHCS: 8

Accuracy of the model is - **57.21%**

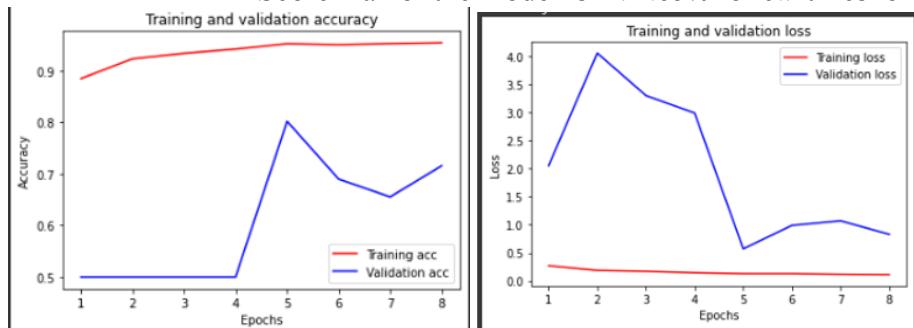
F Score max of the model is - **72.75%** for **0.1** threshold



Optimizer: MSGD , lr: 0.001 , EPOHCS: 8

Accuracy of the model is - **75.80%**

F Score max of the model is - **91.06%** for **0.9** threshold

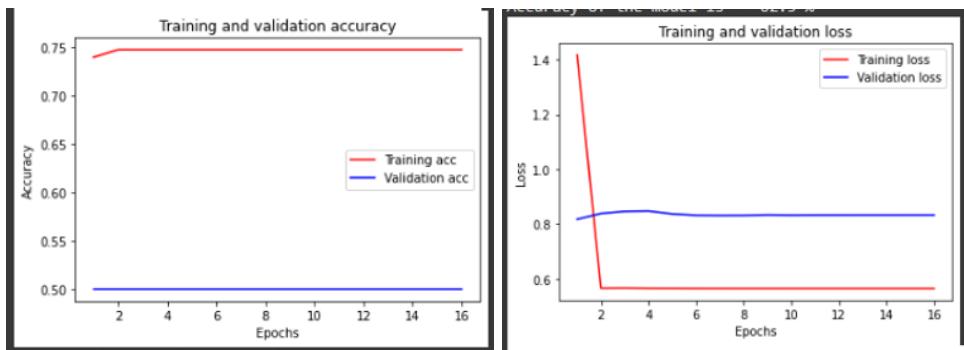


Optimizer: MSGD , lr: 0.1 , EPOHCS: 16

Accuracy of the model is - **62.5%**

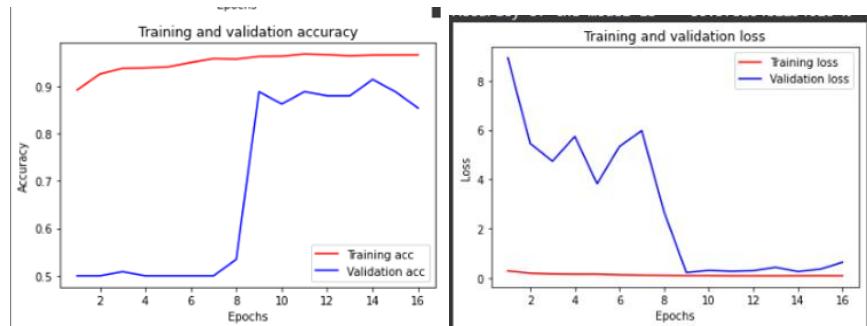
F Score max of the model is - **76.9%** for **0.1** threshold

## חומר לתוכן עניינים



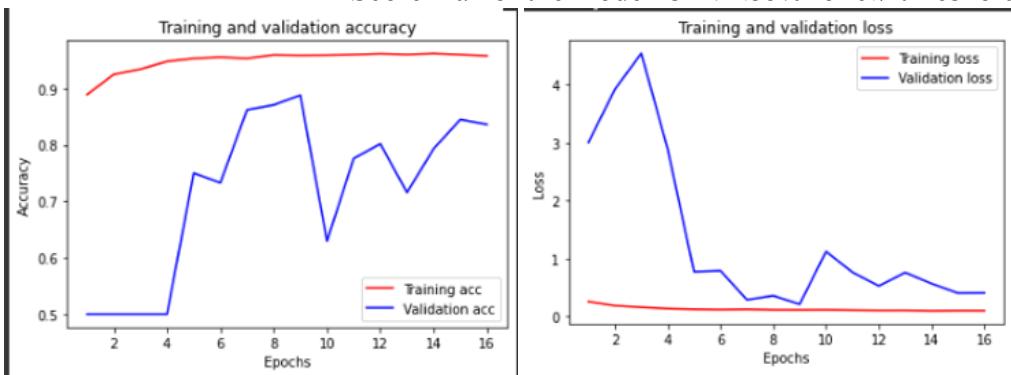
Optimizer: MSGD , lr: 0.01 , EPOHCS: 16

Accuracy of the model is - **86.378%**  
F Score max of the model is - **92.09%** for **0.9** threshold



Optimizer: MSGD , lr: 0.001 , EPOHCS: 16

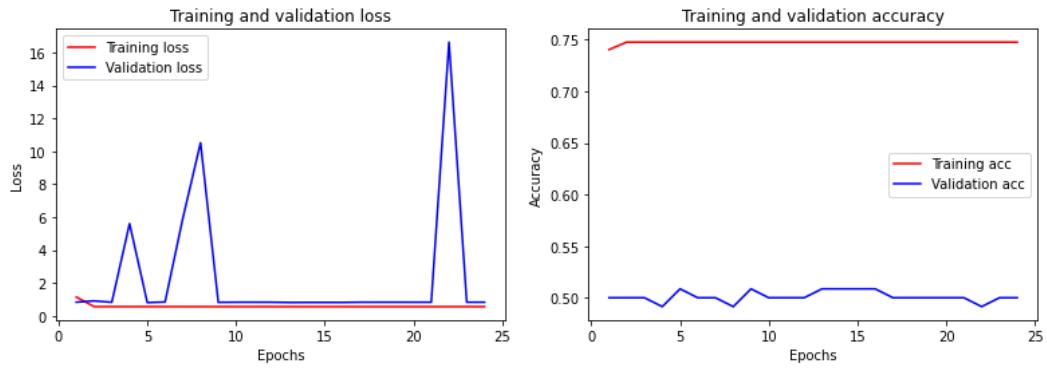
Accuracy of the model is - **88.621%**  
F Score max of the model is - **92.36%** for **0.7** threshold



Optimizer: MSGD , lr: 0.1 , EPOHCS: 24

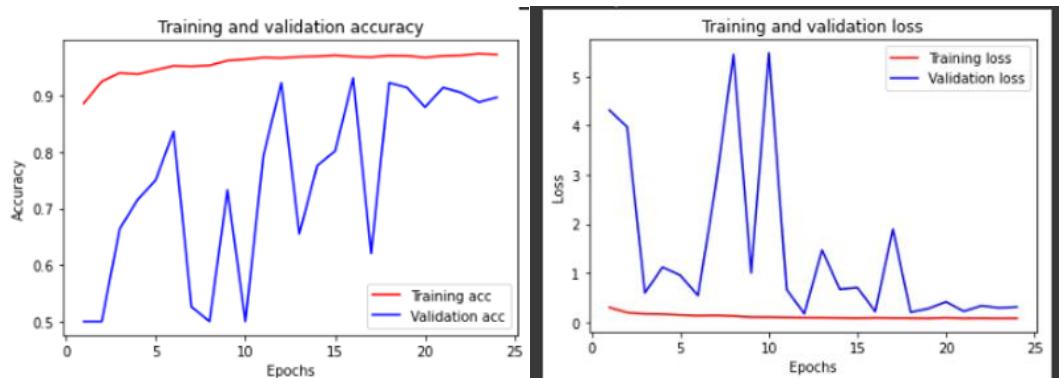
Loss of the model is - 1.2346349954605103  
Accuracy of the model is - 62.5 %

## חזר לתוכן עניינים



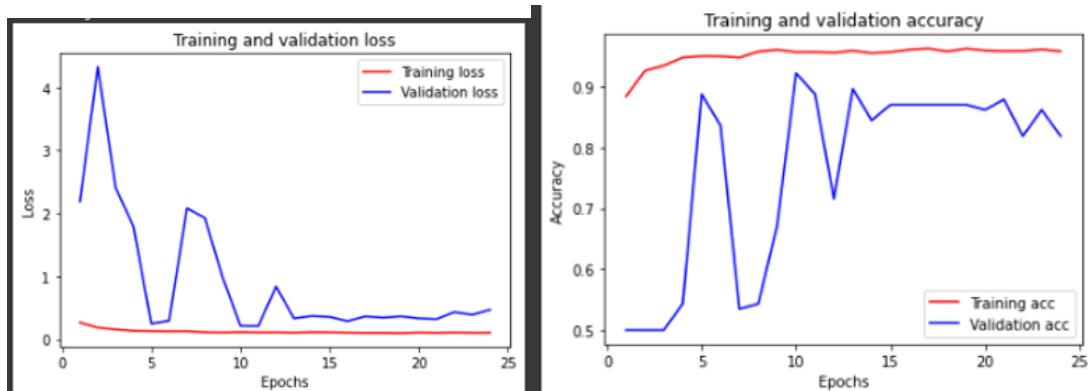
Optimizer: MSGD , lr: 0.01 , EPOCHS: 24

Accuracy of the model is - **90.384%**  
F Score max of the model is - **93.54%** for **0.65** threshold



Optimizer: MSGD , lr: 0.001 , EPOCHS: 24

Accuracy of the model is - **88.942%**  
F Score max of the model is - **92.94%** for **0.8** threshold



**רשת ב'**

**סיכום התוצאות ומסקנות:**

ריכזו את תוצאות של כל הרכזות שבוצעו בסעיף זה :

רשת ב'					
fine tune		only MLP train		lr	epoch
acc %	loss	acc %	loss		
87.50	0.41	90.38	0.30	0.1	8
90.71	0.28	91.35	0.25	0.1	8
91.83	0.30	91.51	0.22	0.001	8
91.51	0.24	90.54	0.34	0.1	16
92.15	0.25	91.51	0.23	0.01	16
89.58	0.44	91.19	0.23	0.001	16
92.31	0.28	90.71	0.24	0.1	24
92.79	0.24	91.19	0.24	0.01	24
93.59	0.23	91.19	0.24	0.001	24
88.94	5.47	89.90	0.59	0.1	8
91.83	1.42	91.03	0.29	0.01	8
92.79	0.64	91.35	0.25	0.001	8
79.81	3.29	89.90	0.30	0.1	16
92.79	0.26	92.15	0.30	0.01	16
93.27	0.37	91.51	0.28	0.001	16
91.99	0.51	91.35	0.29	0.1	24
93.59	0.21	91.35	0.26	0.01	24
<b>94.39</b>	<b>0.20</b>	<b>91.51</b>	<b>0.26</b>	<b>0.001</b>	<b>24</b>
93.11	0.22	90.71	0.24	0.1	8
89.10	0.28	90.06	0.25	0.01	8
77.08	0.51	85.42	0.34	0.001	8
92.15	0.21	92.31	0.24	0.1	16
90.87	0.26	91.35	0.25	0.01	16
80.29	0.48	86.54	0.33	0.001	16
93.59	0.23	91.83	0.25	0.1	24
92.15	0.25	91.51	0.25	0.01	24
79.81	0.48	87.34	0.33	0.001	24
93.27	0.26	90.38	0.26	0.1	8
93.11	0.21	91.99	0.25	0.01	8
89.42	0.27	91.35	0.24	0.001	8
92.47	0.22	62.50	0.70	0.1	16
93.43	0.21	91.83	0.24	0.01	16
90.54	0.26	91.35	0.25	0.001	16
93.75	0.19	91.19	0.26	0.1	24
93.27	0.25	92.31	0.25	0.01	24
91.99	0.23	89.90	0.26	0.001	24
<b>94.39</b>	<b>5.47</b>	<b>92.31</b>	<b>0.70</b>	<b>max</b>	
77.08	0.19	62.50	0.22	<b>min</b>	

## חזרה לתוכן עניינים

best case
worse case

- .1. התאים הירוקים מייצגים את המקורה הטוב ביותר (הכי רצוי).
- .2. התאים האדומים מייצגים את המקורה הכי גרווע (הכי לא רצוי).
- .3. ניתן לראות שחרשת עם הפרמטרים המסומנים **בחדשה בחוליה** -  
קיבלה את הדיקוק הגבוה ביותר, וכן גם loss נמוך מאוד. לכן, זו נחברה כרשות עם הביצועים הטובים ביותר.

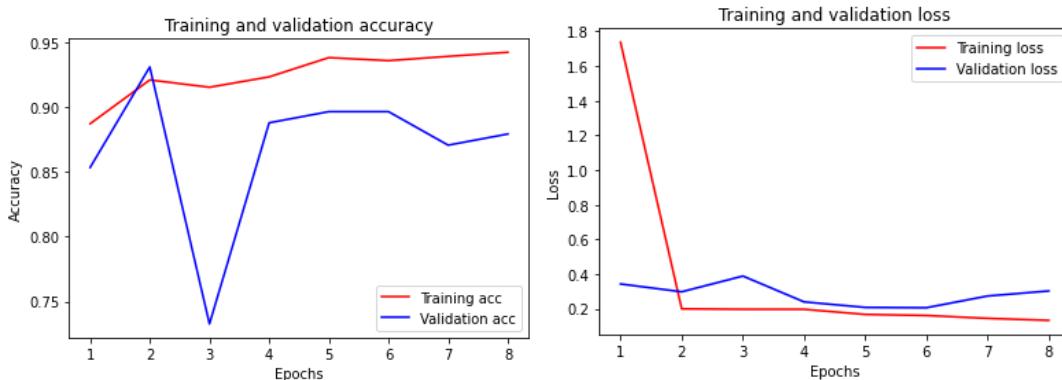
Optimizer: RMSPROP, lr: 0.001 , EPOHCS: 24

- .4. ניתן לראות באופן עקבי שביצוע fine tuning משפר חן את הדיקוק וחן את loss.

### פירוט התוצאות:

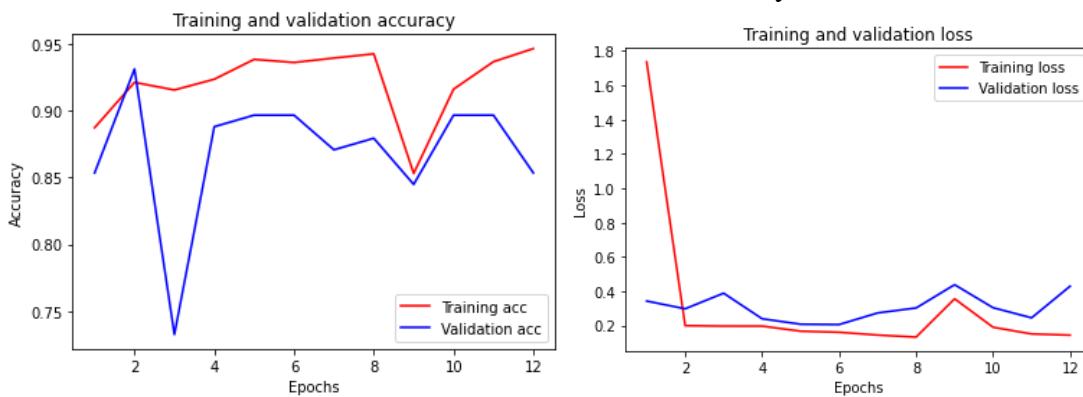
Optimizer: Adam , lr: 0.1 , EPOHCS: 8

Loss of the model is - 0.29675355553627014  
Accuracy of the model is - 90.38461446762085 %



### Fine tune

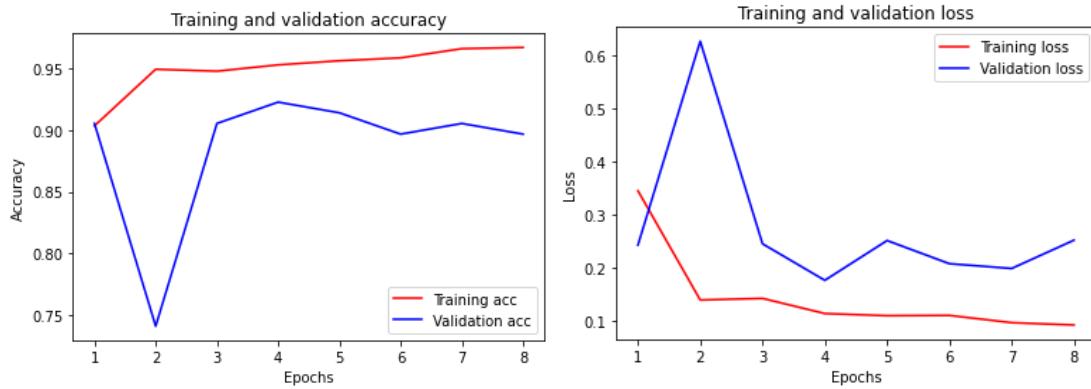
Loss of the model is - 0.4081336259841919  
Accuracy of the model is - 87.5 %



Optimizer: Adam , lr: 0.01 , EPOHCS: 8

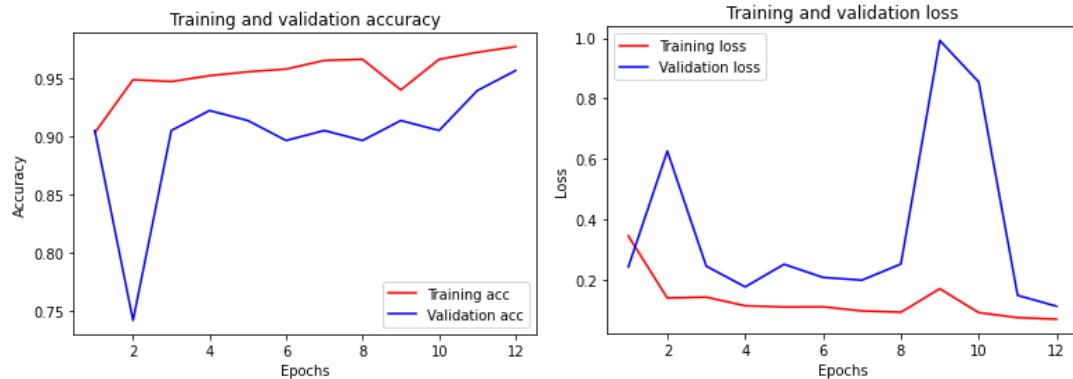
Loss of the model is - 0.25329700112342834  
Accuracy of the model is - 91.34615659713745 %

## חזר לתוכן עניינים



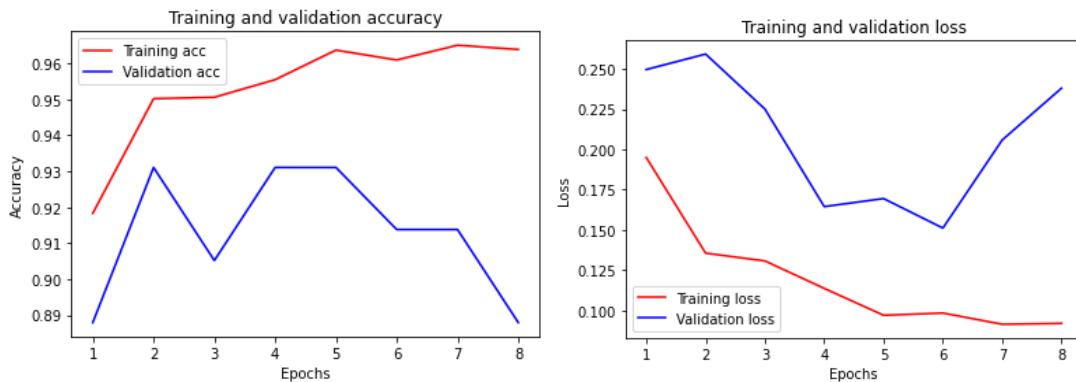
### Fine tune

Loss of the model is - 0.28157126903533936  
Accuracy of the model is - 90.70512652397156 %



Optimizer: Adam , lr: 0.001 , EPOCHS: 8

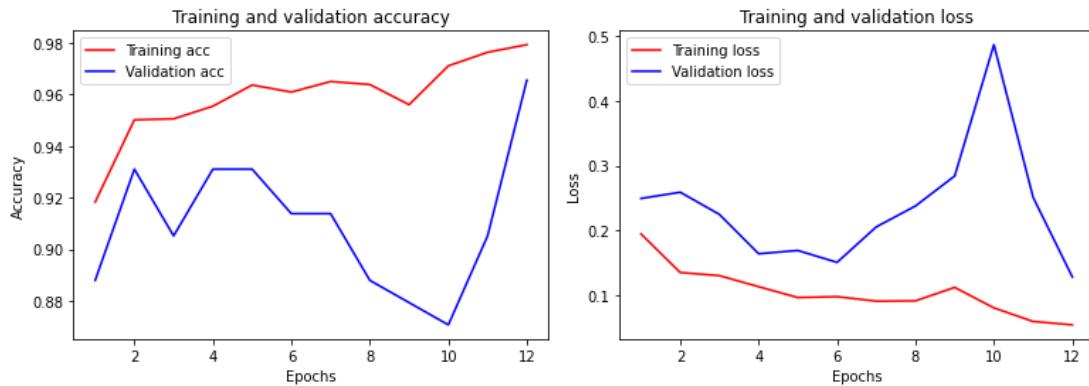
Loss of the model is - 0.22043582797050476  
Accuracy of the model is - 91.50640964508057 %



### Fine tune

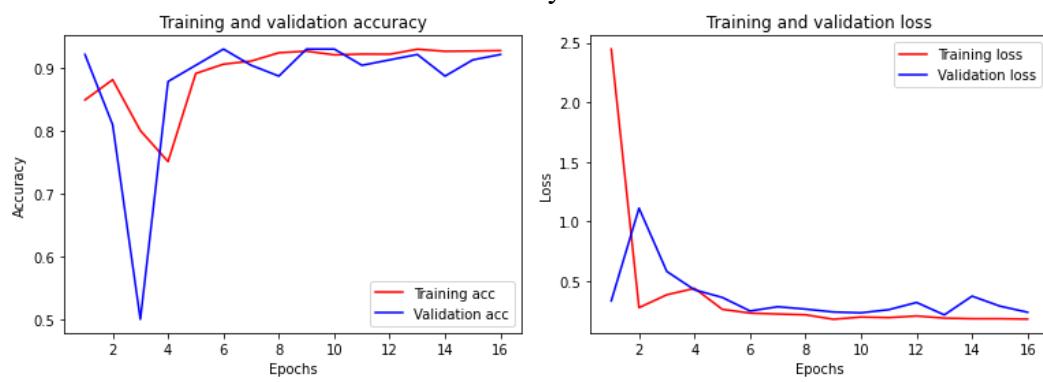
Loss of the model is - 0.29779648780822754  
Accuracy of the model is - 91.82692170143127 %

## חזר לתוכן עניינים



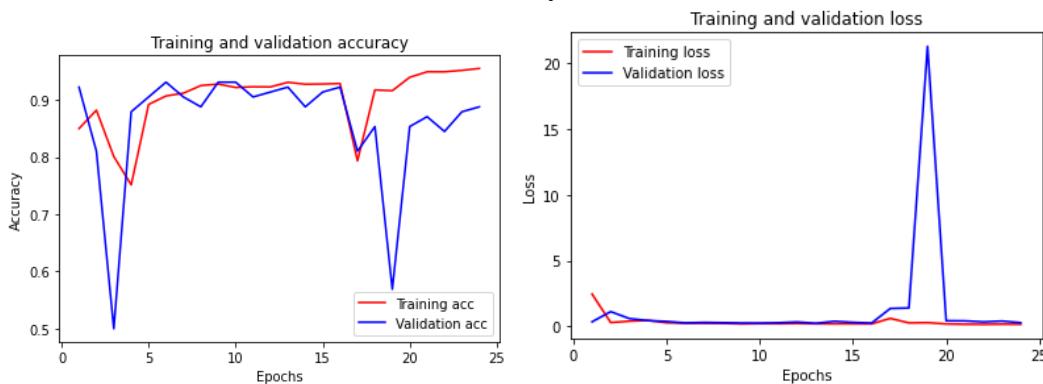
Optimizer: Adam , lr: 0.1 , EPOCHS: 16

Loss of the model is - 0.3387520909309387  
Accuracy of the model is - 90.54487347602844 %



## Fine tune

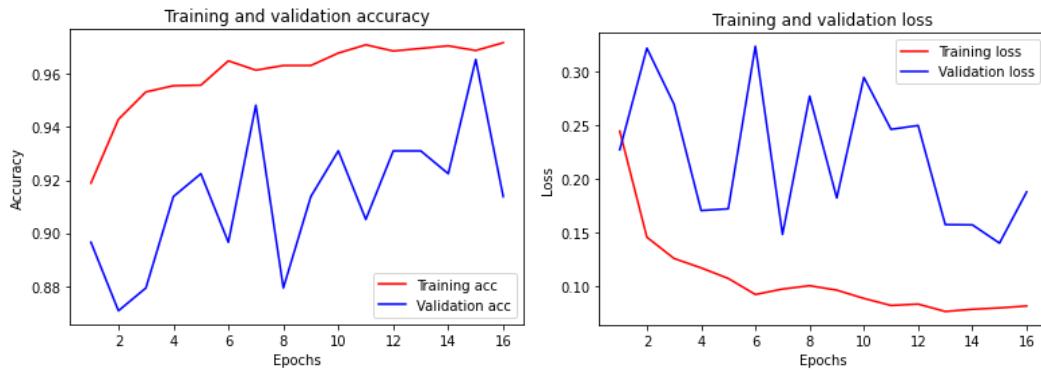
Loss of the model is - 0.24400374293327332  
Accuracy of the model is - 91.50640964508057 %



Optimizer: Adam , lr: 0.01 , EPOCHS: 16

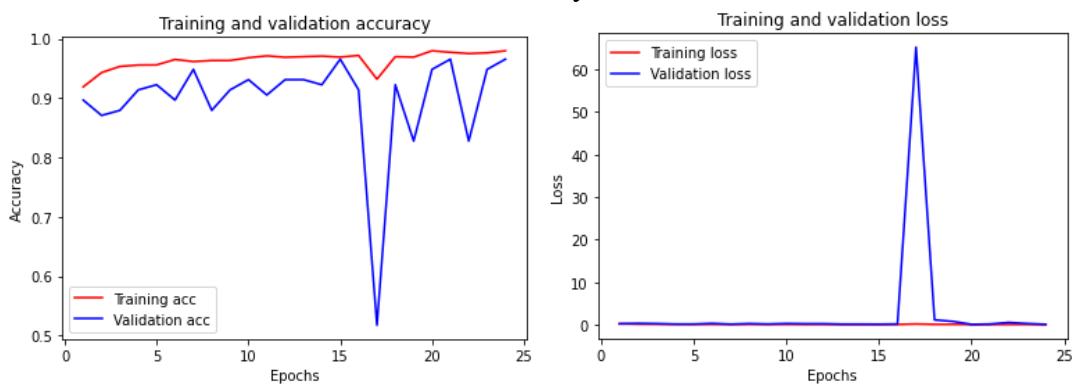
Loss of the model is - 0.23392905294895172  
Accuracy of the model is - 91.50640964508057 %

## חזר לתוכן עניינים



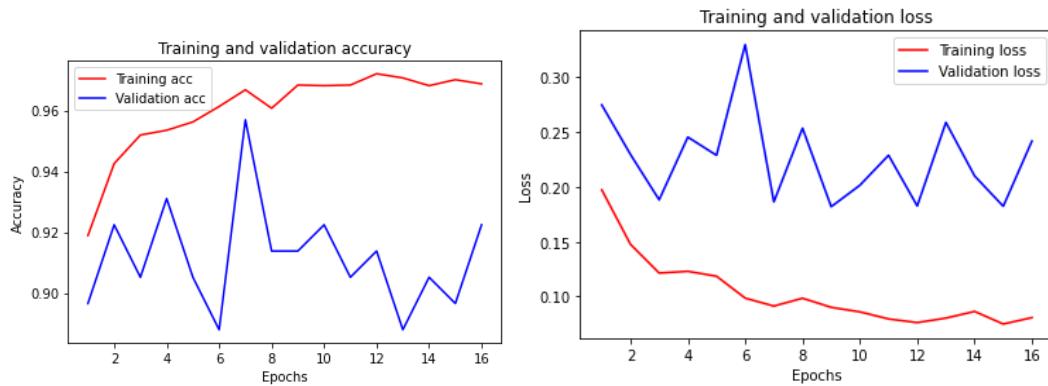
### Fine tune

Loss of the model is - 0.2466564178466797  
 Accuracy of the model is - 92.14743375778198 %



Optimizer: Adam , lr: 0.001 , EPOCHS: 16

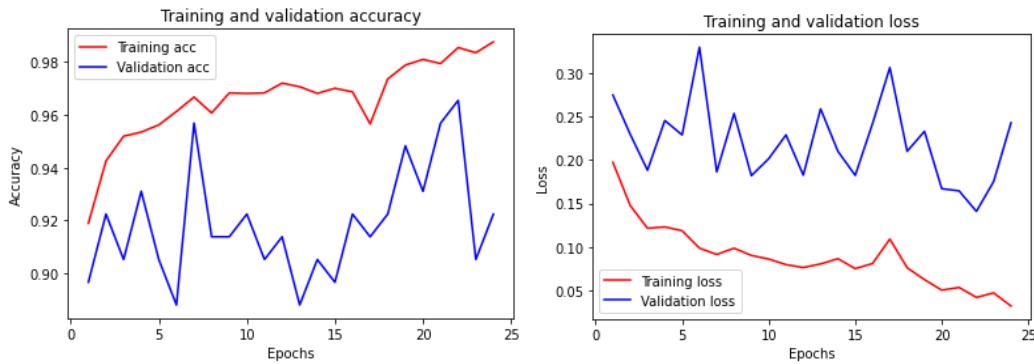
Loss of the model is - 0.2346874177455902  
 Accuracy of the model is - 91.18589758872986 %



### Fine tune

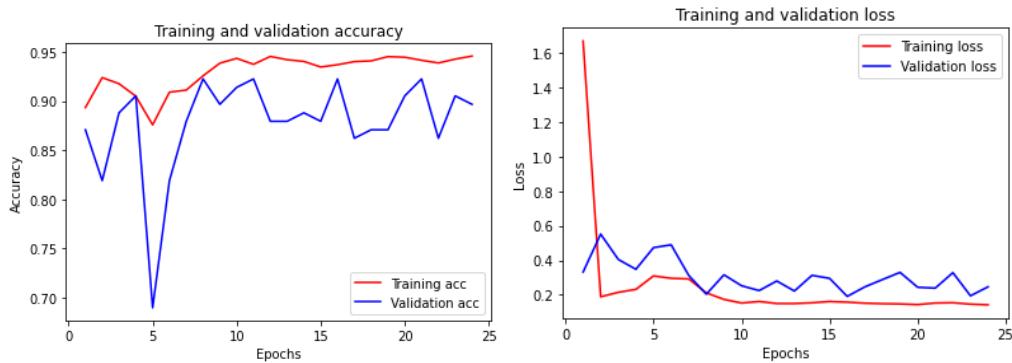
Loss of the model is - 0.4402116537094116  
 Accuracy of the model is - 89.58333134651184 %

## חזר לתוכן עניינים



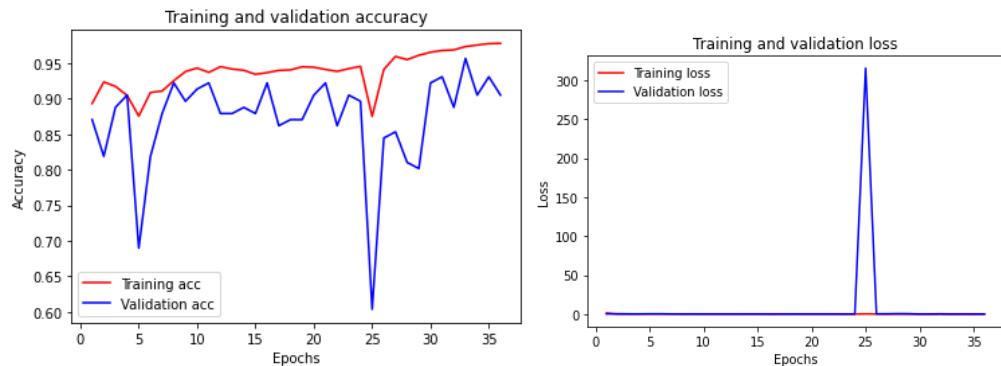
Optimizer: Adam , Ir: 0.1 , EPOCHS: 24

Loss of the model is - 0.2400968223810196  
Accuracy of the model is - 90.70512652397156 %



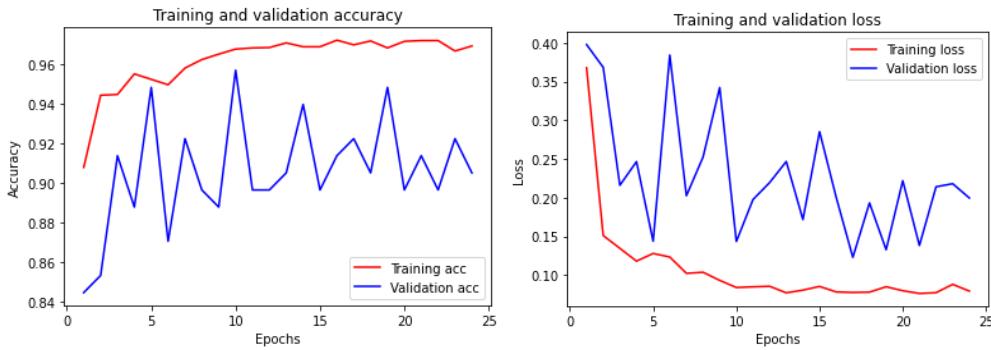
**Fine tune**

Loss of the model is - 0.28242212533950806  
Accuracy of the model is - 92.30769276618958 %



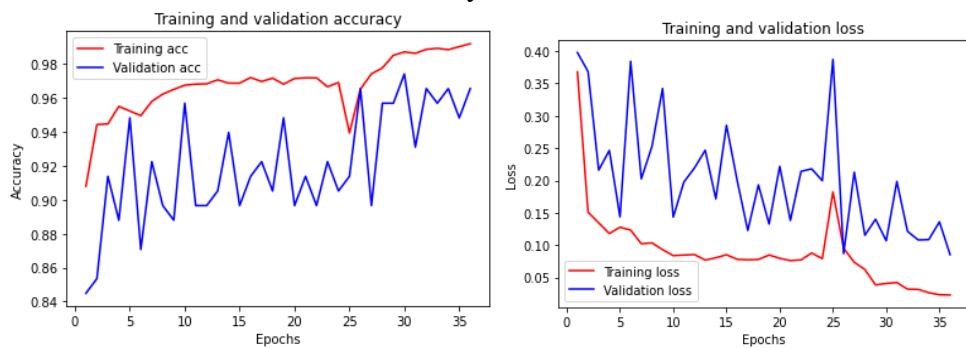
Optimizer: Adam , Ir: 0.01 , EPOCHS: 24

Loss of the model is - 0.24312454462051392  
Accuracy of the model is - 91.18589758872986 %



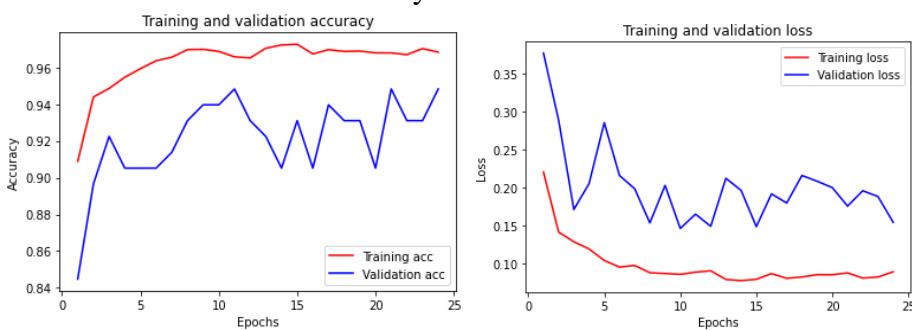
**Fine tune**

Loss of the model is - 0.23749332129955292  
Accuracy of the model is - 92.78846383094788 %



Optimizer: Adam , lr: 0.001 , EPOCHS: 24

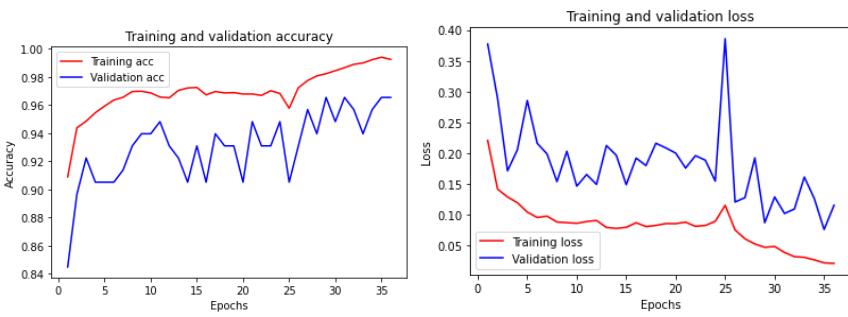
Loss of the model is - 0.23628661036491394  
Accuracy of the model is - 91.18589758872986 %



**Fine tune**

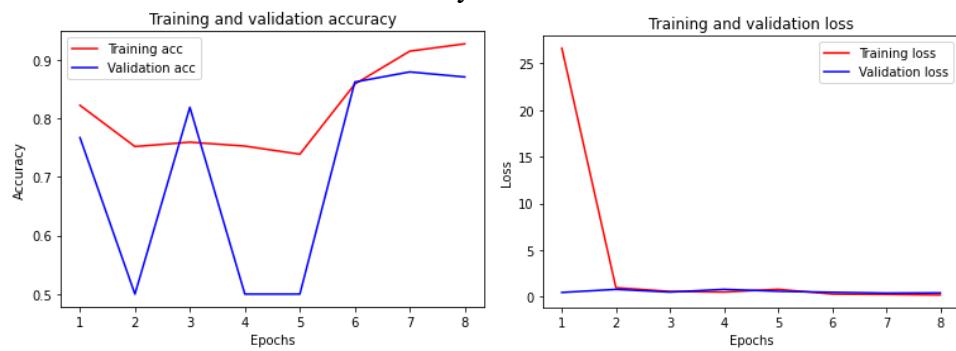
Loss of the model is - 0.22947081923484802  
Accuracy of the model is - 93.58974099159241 %

## חזר לתוכן עניינים



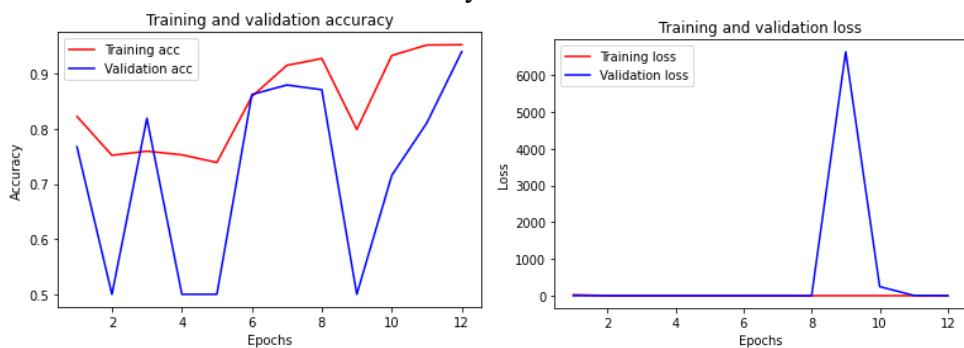
Optimizer: RMSprop , lr: 0.1 , EPOCHS: 8

Loss of the model is - 0.5898919105529785  
Accuracy of the model is - 89.90384340286255 %



## Fine tune

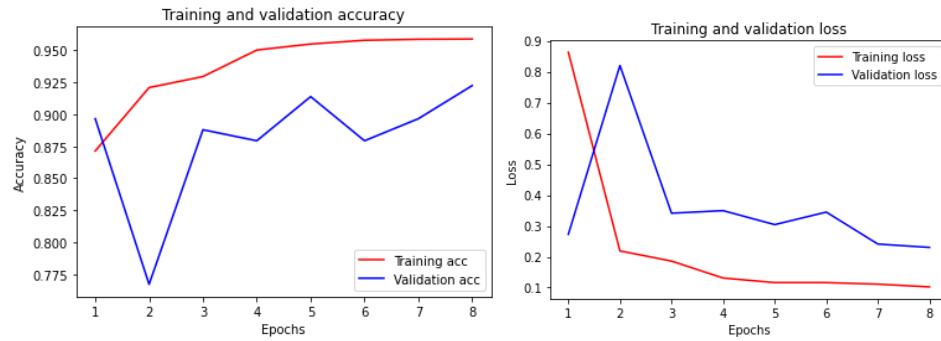
Loss of the model is - 5.473158836364746  
Accuracy of the model is - 88.94230723381042 %



Optimizer: RMSprop , lr: 0.01 , EPOCHS: 8

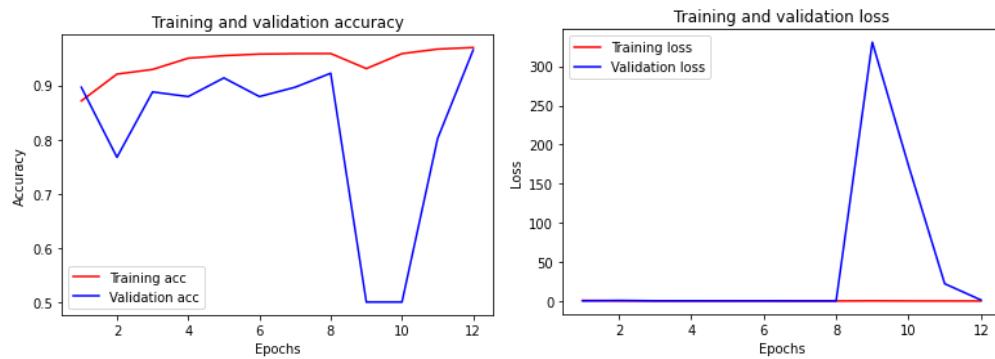
Loss of the model is - 0.28899654746055603  
Accuracy of the model is - 91.02563858032227 %

## חזר לתוכן עניינים



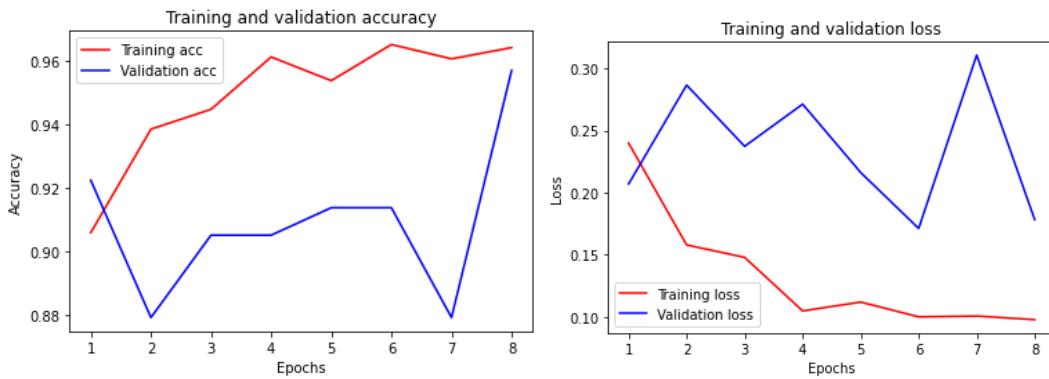
## Fine tune

Loss of the model is - 1.420469880104065  
Accuracy of the model is - 91.82692170143127 %



Optimizer: RMSprop , lr: 0.001 , EPOCHS: 8

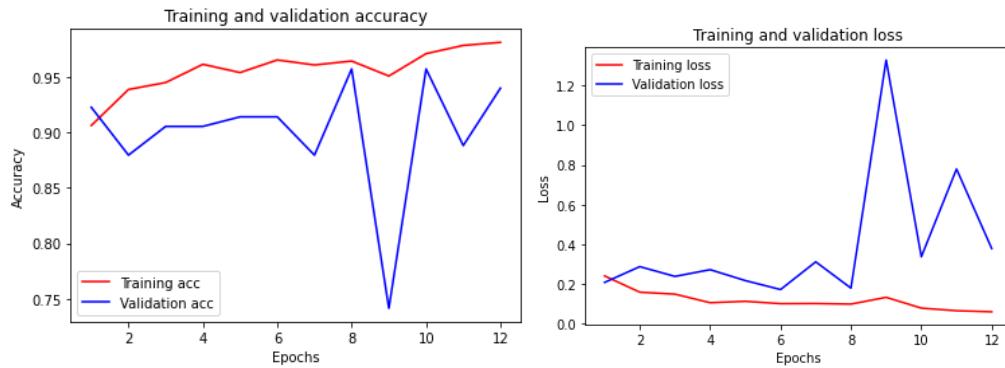
Loss of the model is - 0.25087055563926697  
Accuracy of the model is - 91.34615659713745 %



## Fine tune

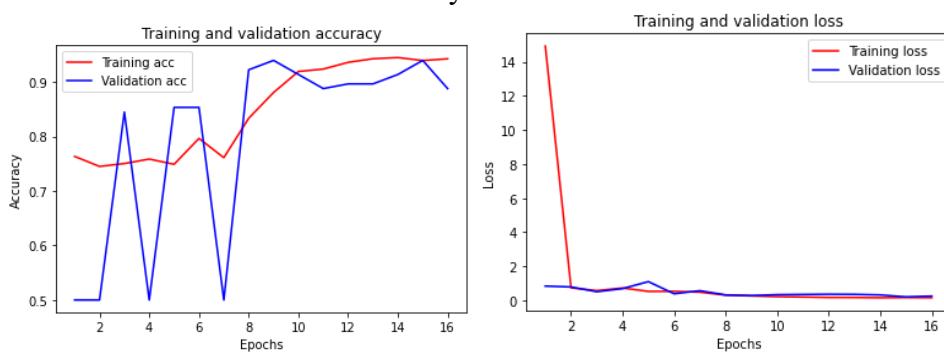
Loss of the model is - 0.6368541121482849  
Accuracy of the model is - 92.78846383094788 %

## חזר לתוכן עניינים



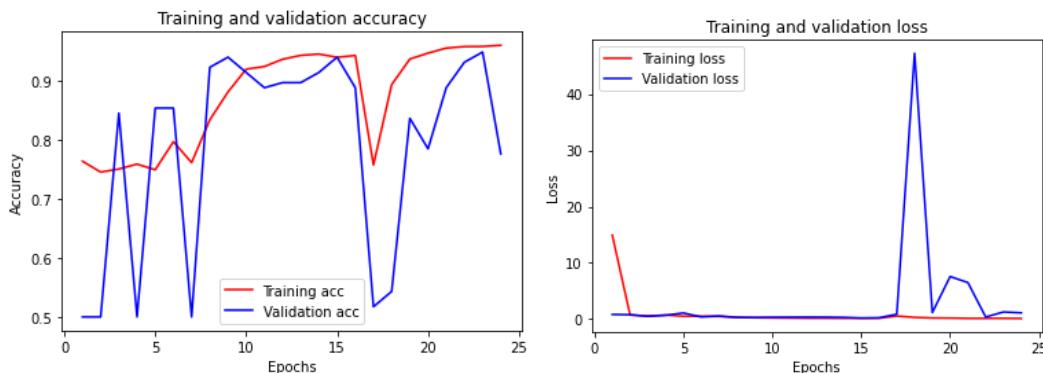
Optimizer: RMSprop , lr: 0.1 , EPOCHS: 16

Loss of the model is - 0.2997707724571228  
Accuracy of the model is - 89.90384340286255 %



**Fine tune**

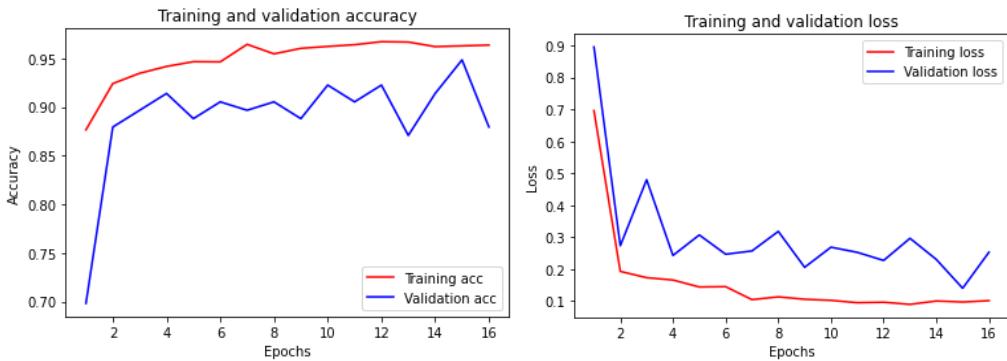
Loss of the model is - 3.291388034820556  
Accuracy of the model is - 79.80769276618958 %



Optimizer: RMSprop , lr: 0.01 , EPOCHS: 16

Loss of the model is - 0.30174288153648376  
Accuracy of the model is - 92.14743375778198 %

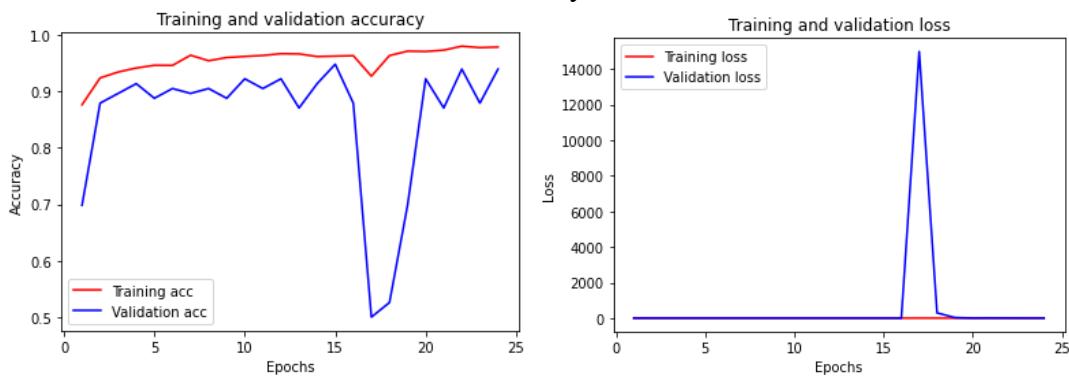
## חזר לתוכן עניינים



**Fine tune**

Loss of the model is - 0.2563861906528473

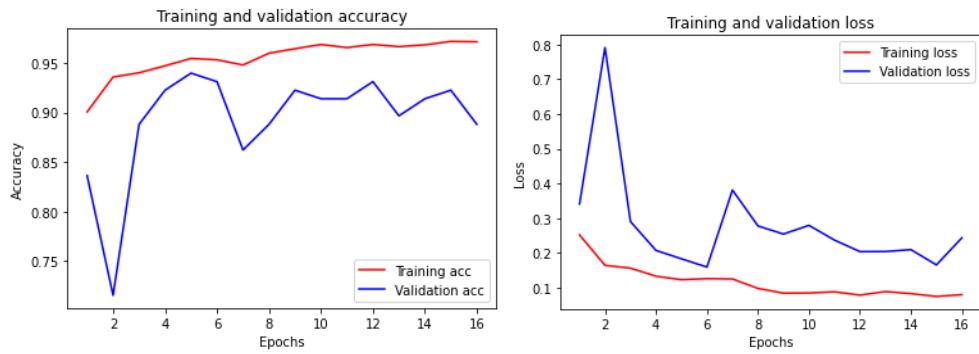
Accuracy of the model is - 92.78846383094788 %



Optimizer: RMSprop , lr: 0.001 , EPOCHS: 16

Loss of the model is - 0.2768539488315582

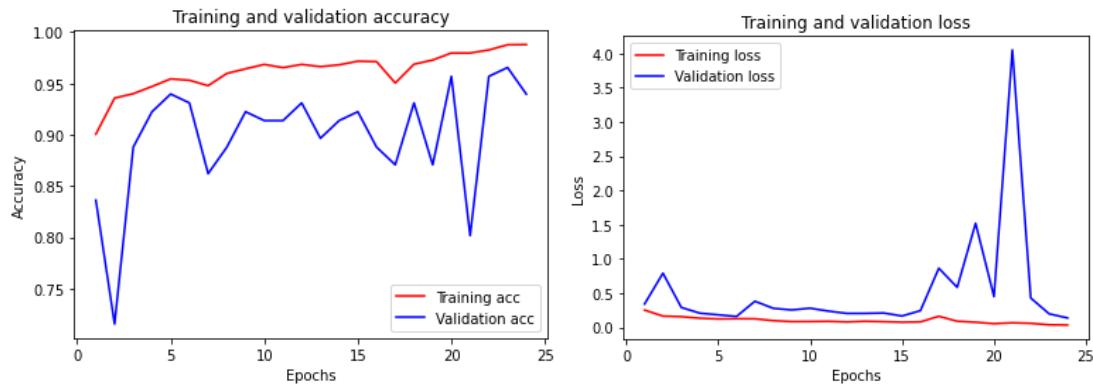
Accuracy of the model is - 91.50640964508057 %



**Fine tune**

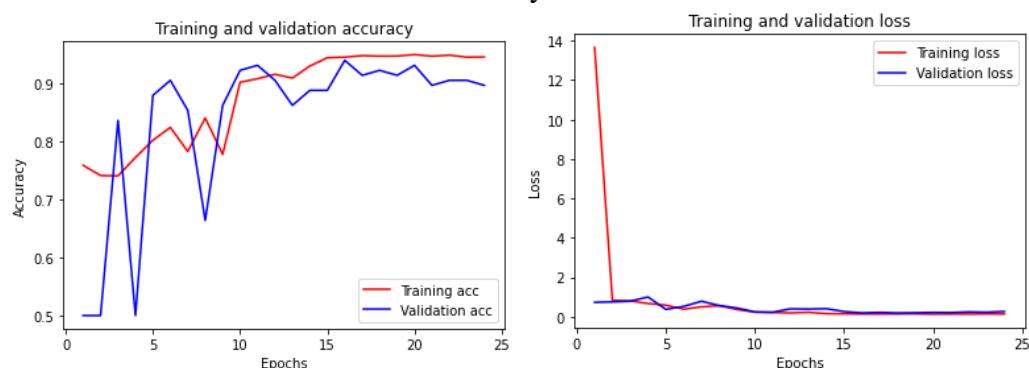
Loss of the model is - 0.36693623661994934  
Accuracy of the model is - 93.2692289352417 %

## חזר לתוכן עניינים



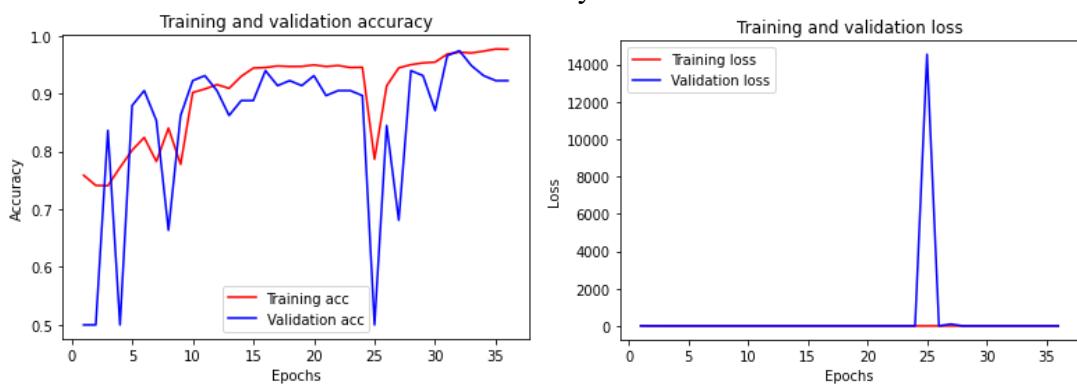
Optimizer: RMSprop , lr: 0.1 , EPOCHS: 24

Loss of the model is - 0.28736501932144165  
Accuracy of the model is - 91.34615659713745 %



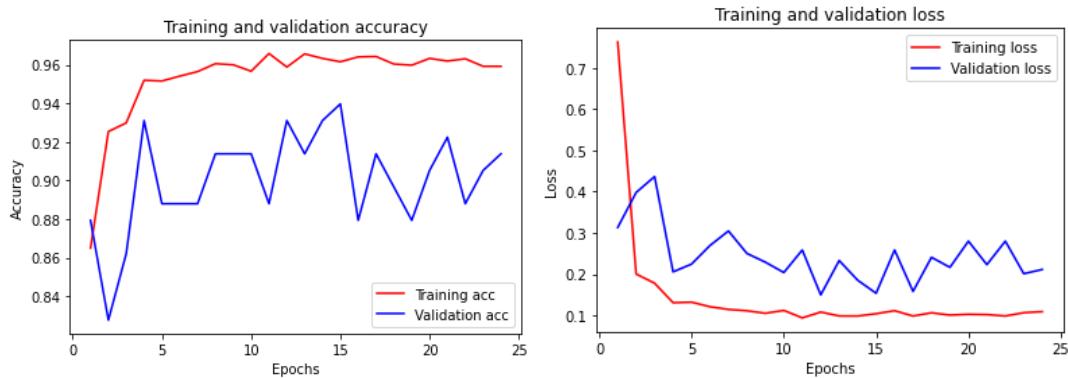
## Fine tune

Loss of the model is - 0.507497251033783  
Accuracy of the model is - 91.98718070983887 %



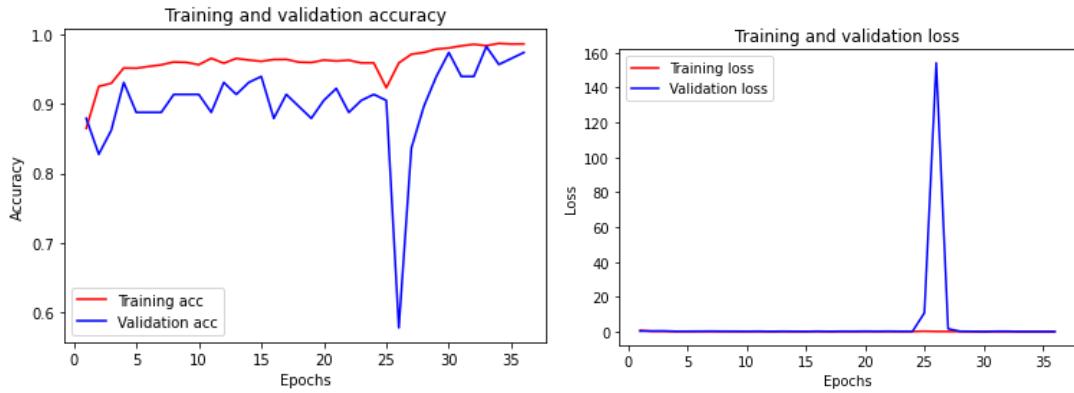
Optimizer: RMSprop , lr: 0.01 , EPOCHS: 24

Loss of the model is - 0.2580012381076813  
 Accuracy of the model is - 91.34615659713745 %



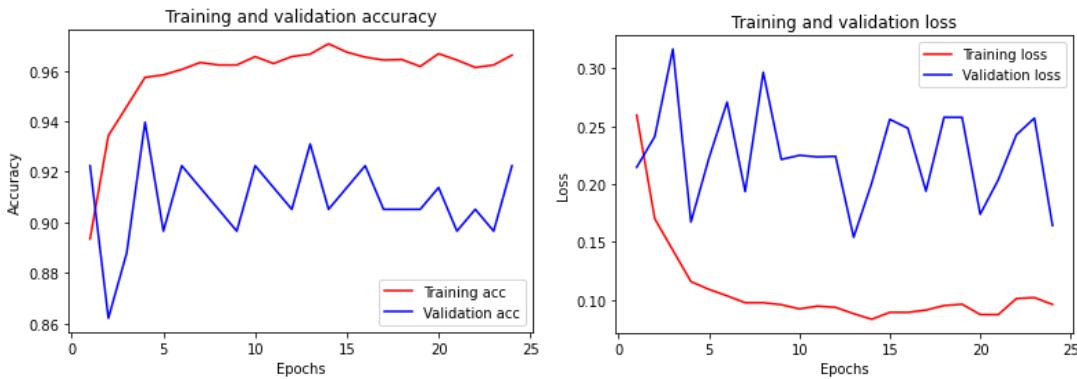
**Fine tune**

Loss of the model is - 0.2090732902288437  
 Accuracy of the model is - 93.58974099159241 %



Optimizer: RMSprop , lr: 0.001 , EPOCHS: 24

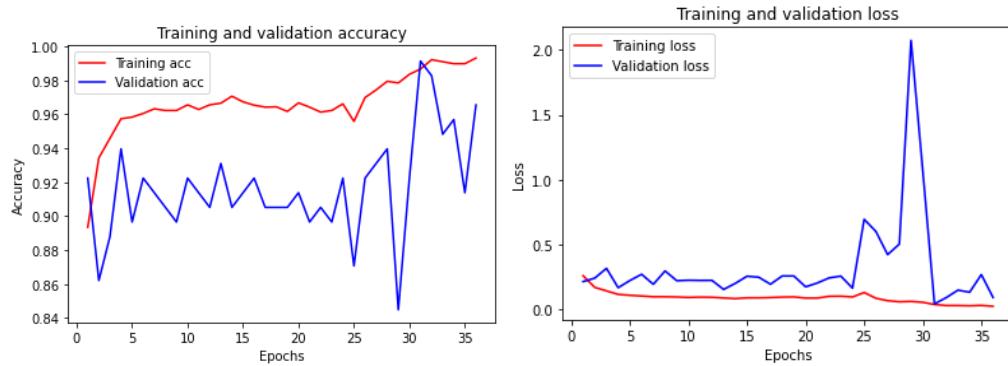
Loss of the model is - 0.2587624788284302  
 Accuracy of the model is - 91.50640964508057 %



**Fine tune**

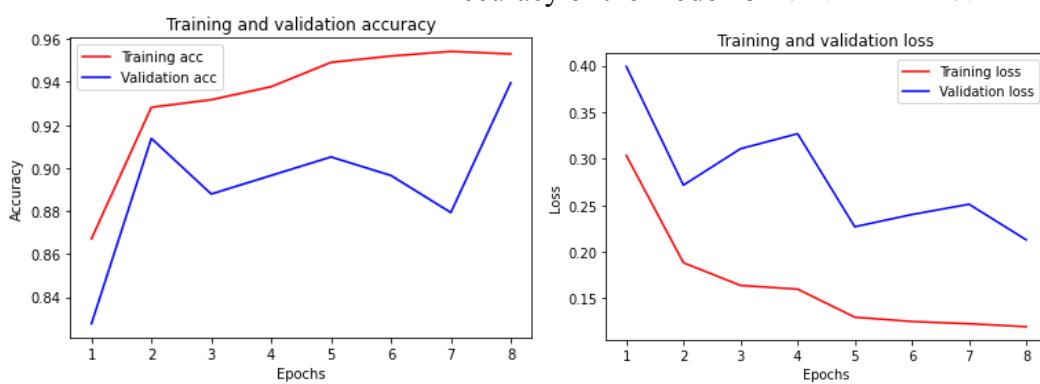
Loss of the model is - 0.20316524803638458  
 Accuracy of the model is - 94.39102411270142 %

## חומר לתוכן עניינים



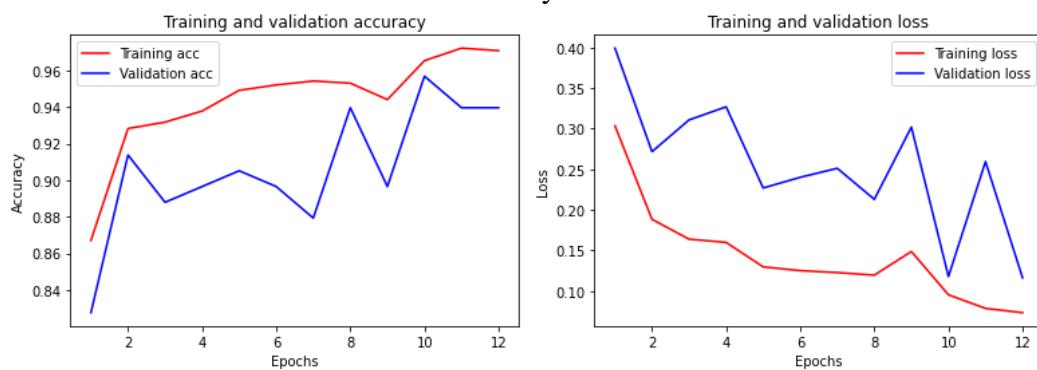
Optimizer: SGD , lr: 0.1 , EPOCHS: 8

Loss of the model is - 0.2427241951227188  
Accuracy of the model is - 90.70512652397156 %



**Fine tune**

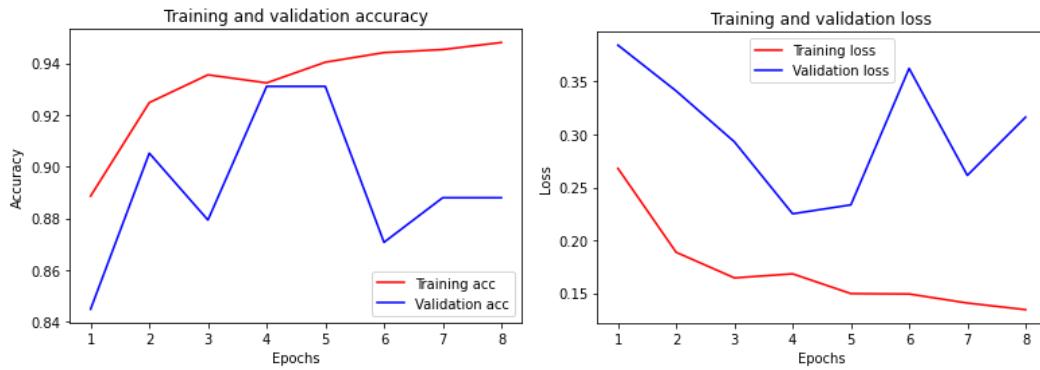
Loss of the model is - 0.22448047995567322  
Accuracy of the model is - 93.10897588729858 %



Optimizer: SGD , lr: 0.01 , EPOCHS: 8

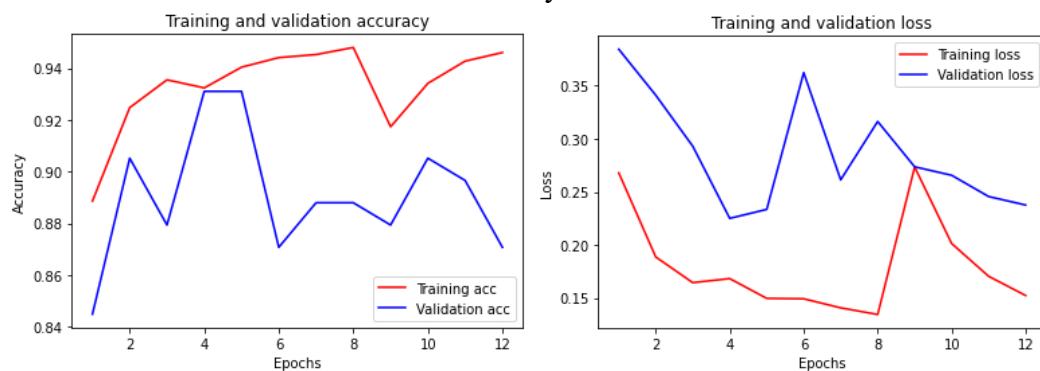
Loss of the model is - 0.2507004141807556  
Accuracy of the model is - 90.06410241127014 %

## חומר לתוכן עניינים



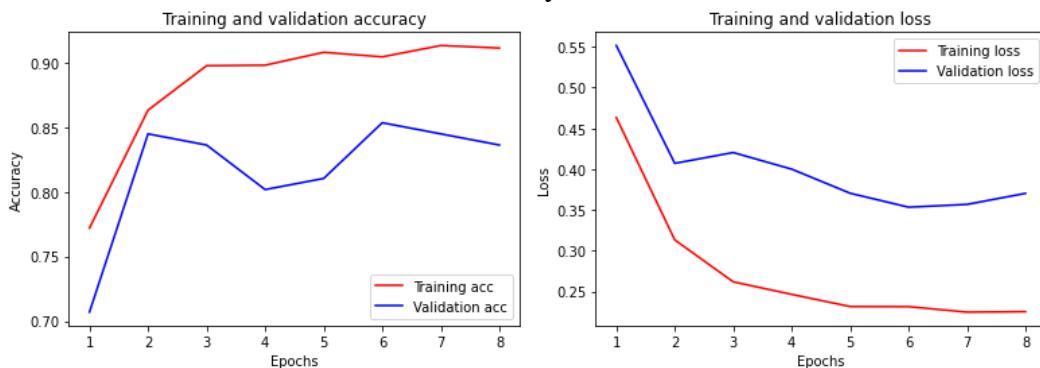
### Fine tune

Loss of the model is - 0.27658265829086304  
 Accuracy of the model is - 89.10256624221802 %



Optimizer: SGD , lr: 0.001 , EPOCHS: 8

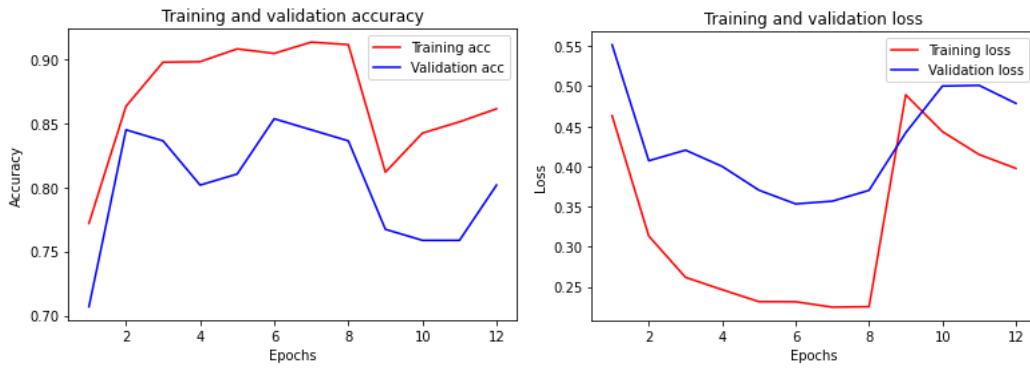
Loss of the model is - 0.3401344120502472  
 Accuracy of the model is - 85.41666865348816 %



### Fine tune

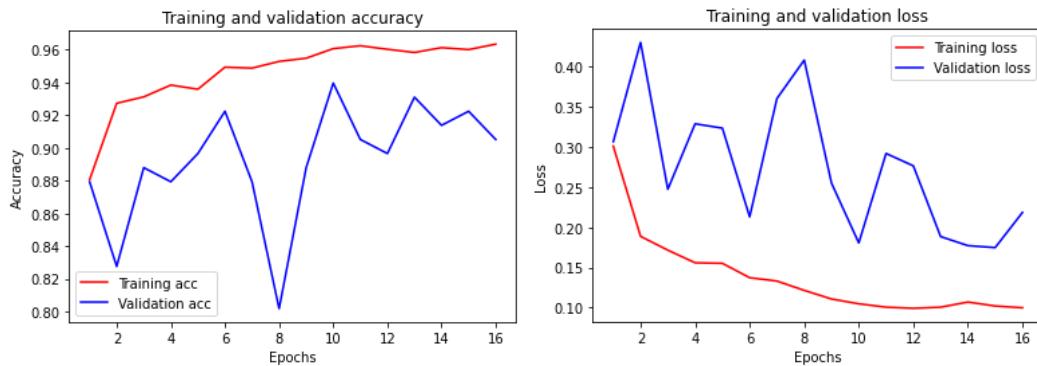
Loss of the model is - 0.514221727848053  
 Accuracy of the model is - 77.08333134651184 %

## חזר לתוכן עניינים



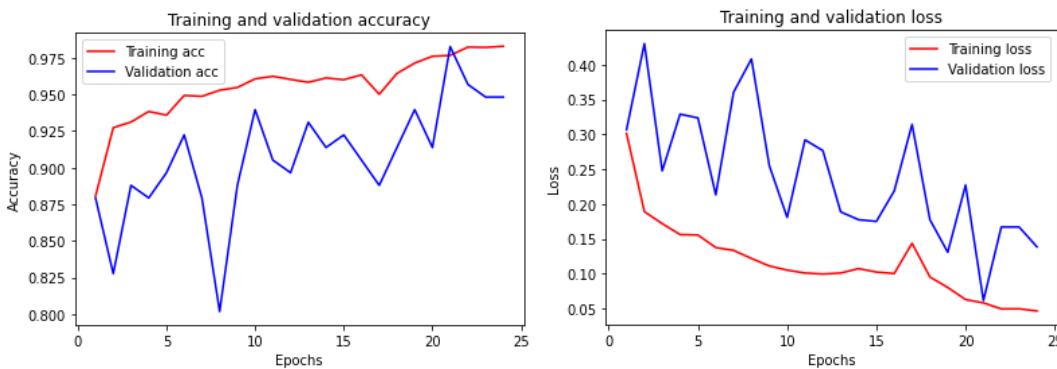
Optimizer: SGD , lr: 0.1 , EPOCHS: 16

Loss of the model is - 0.2415880262851715  
Accuracy of the model is - 92.30769276618958 %



**Fine tune**

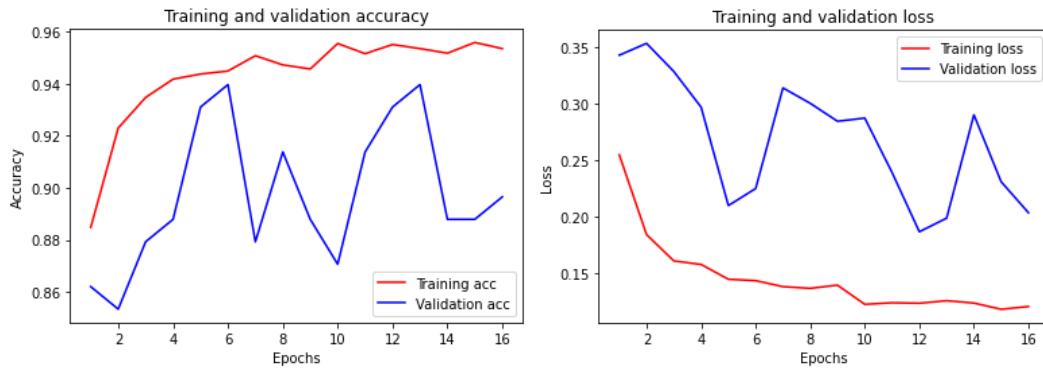
Loss of the model is - 0.21075816452503204  
Accuracy of the model is - 92.14743375778198 %



Optimizer: SGD , lr: 0.01 , EPOCHS: 16

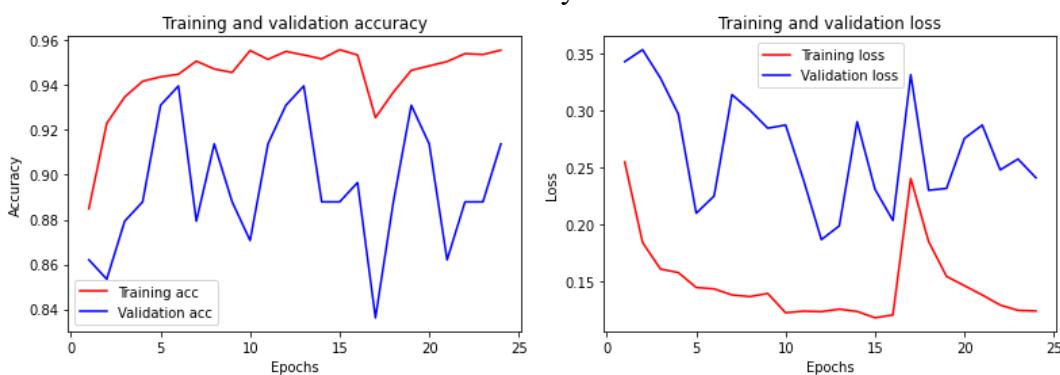
Loss of the model is - 0.24881480634212494  
Accuracy of the model is - 91.34615659713745 %

## חזר לתוכן עניינים



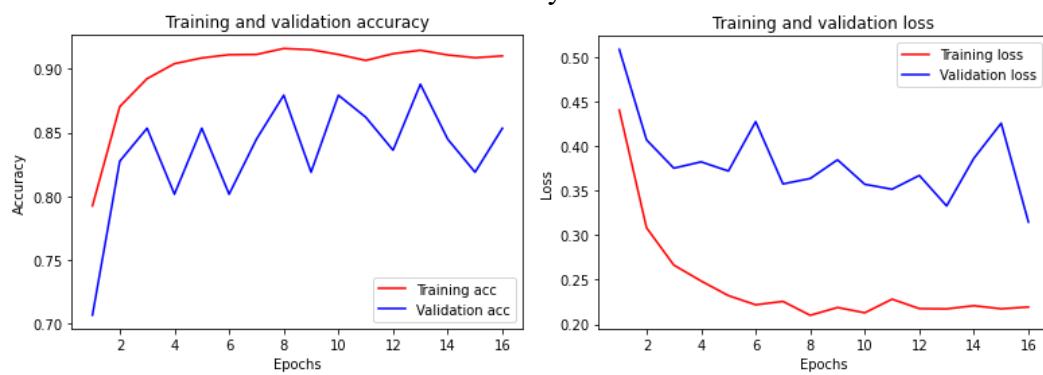
### Fine tune

Loss of the model is - 0.2616933286190033  
Accuracy of the model is - 90.86538553237915 %



Optimizer: SGD , lr: 0.001 , EPOCHS: 16

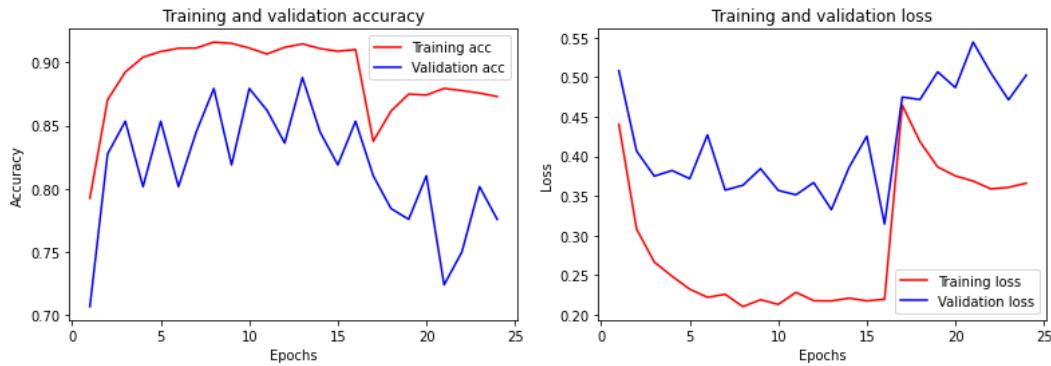
Loss of the model is - 0.325775146484375  
Accuracy of the model is - 86.53846383094788 %



### Fine tune

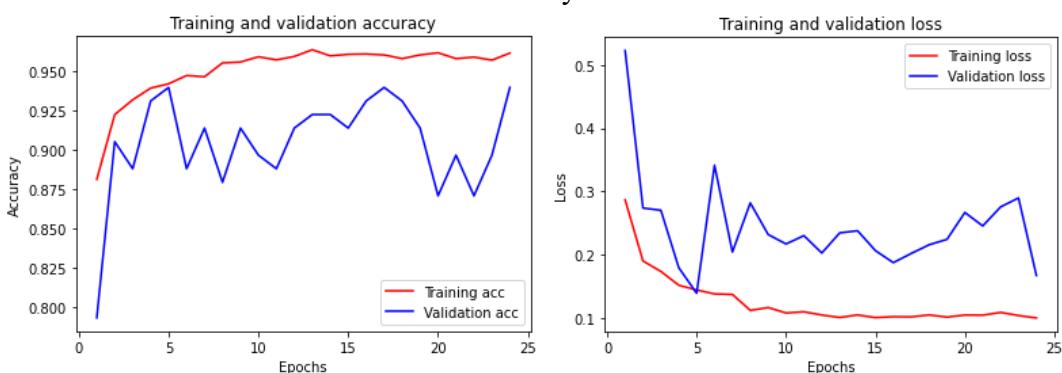
Loss of the model is - 0.4769095480442047  
Accuracy of the model is - 80.28846383094788 %

## חזר לתוכן עניינים



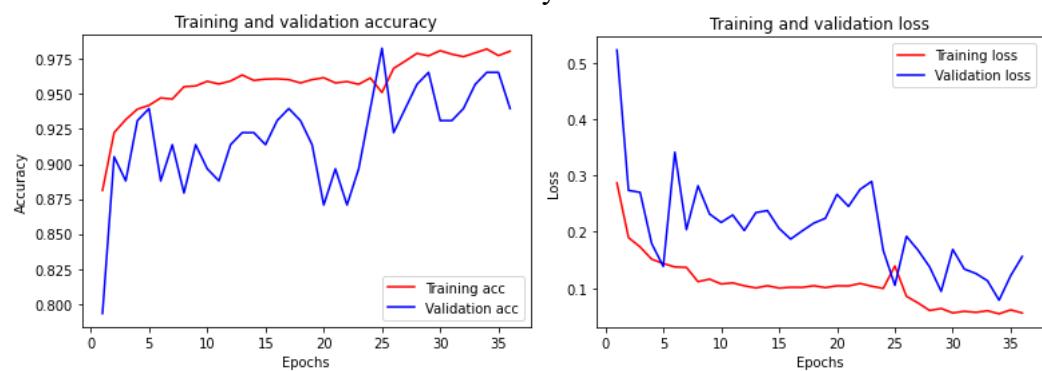
Optimizer: SGD , lr: 0.1 , EPOCHS: 24

Loss of the model is - 0.2455897480249405  
Accuracy of the model is - 91.82692170143127 %



## Fine tune

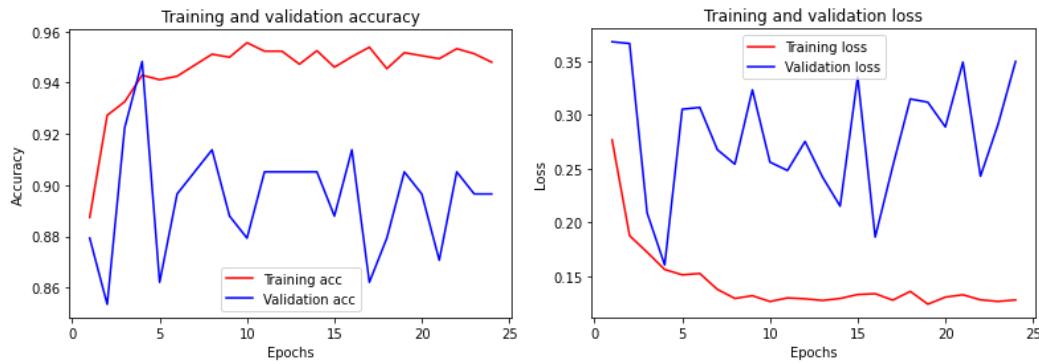
Loss of the model is - 0.2268601506948471  
Accuracy of the model is - 93.58974099159241 %



Optimizer: SGD , lr: 0.01 , EPOCHS: 24

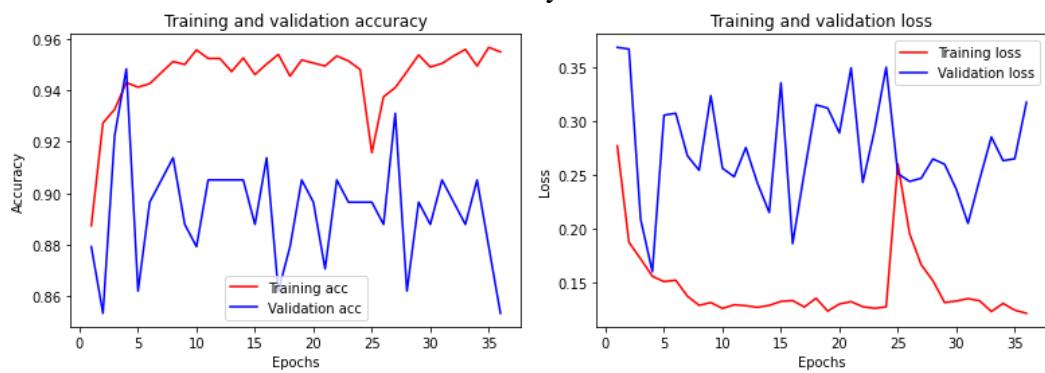
Loss of the model is - 0.25187206268310547  
Accuracy of the model is - 91.50640964508057 %

## חזר לתוכן עניינים



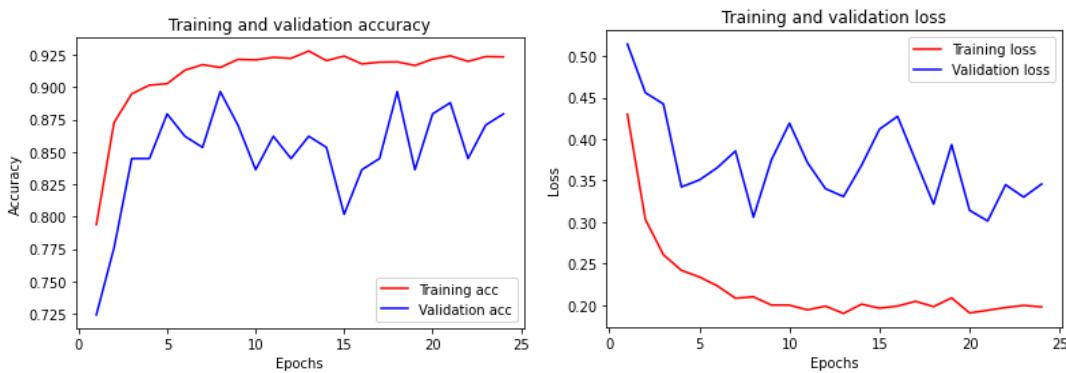
### Fine tune

Loss of the model is - 0.2450978010892868  
Accuracy of the model is - 92.14743375778198 %



Optimizer: SGD , lr: 0.001 , EPOCHS: 24

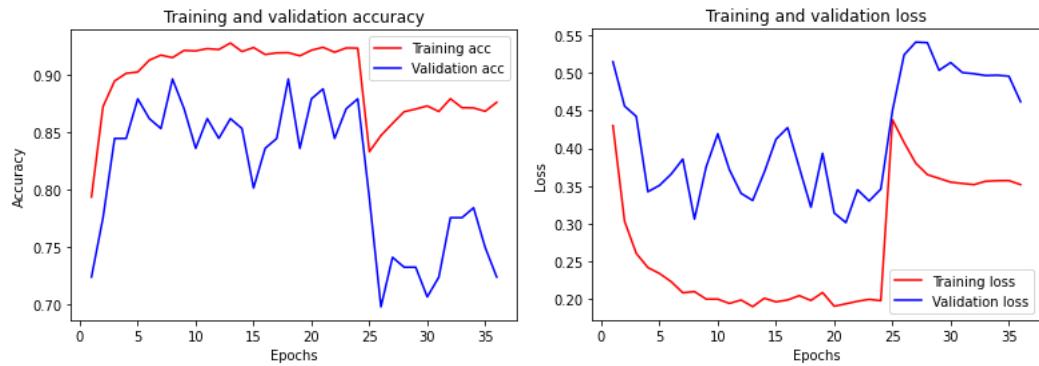
Loss of the model is - 0.3277052640914917  
Accuracy of the model is - 87.33974099159241 %



### Fine tune

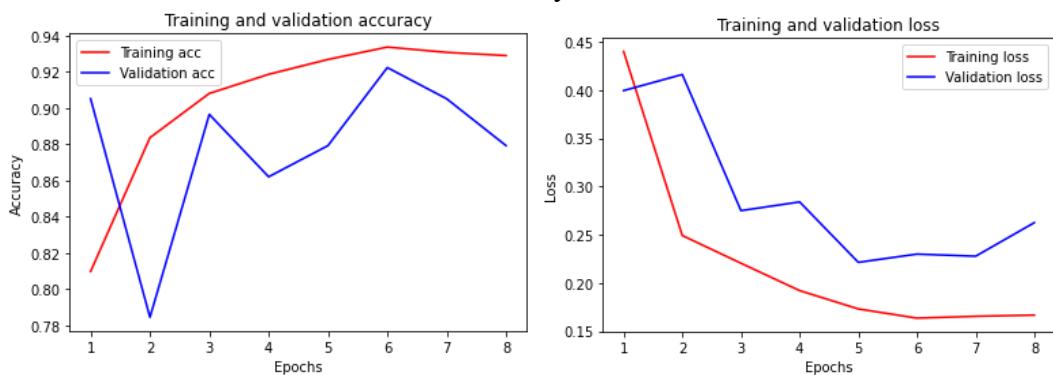
Loss of the model is - 0.48007574677467346  
Accuracy of the model is - 79.80769276618958 %

## חזר לתוכן עניינים



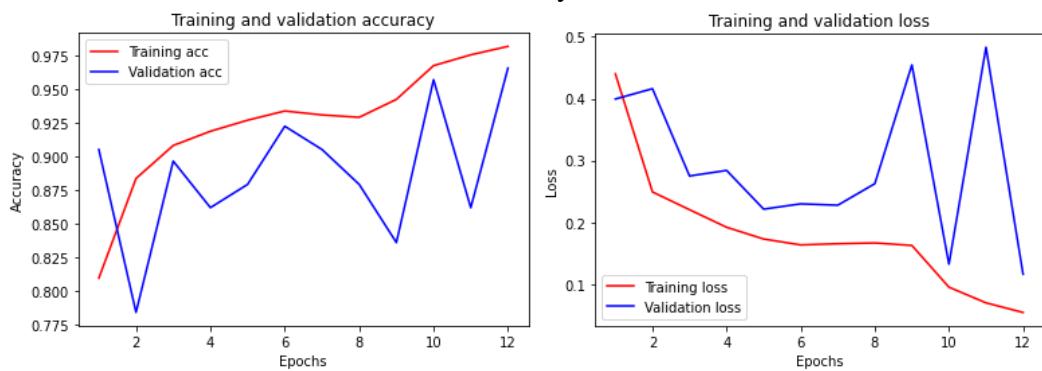
Optimizer: MSGD , lr: 0.1 , EPOCHS: 8

Loss of the model is - 0.2649281919002533  
Accuracy of the model is - 90.38461446762085 %



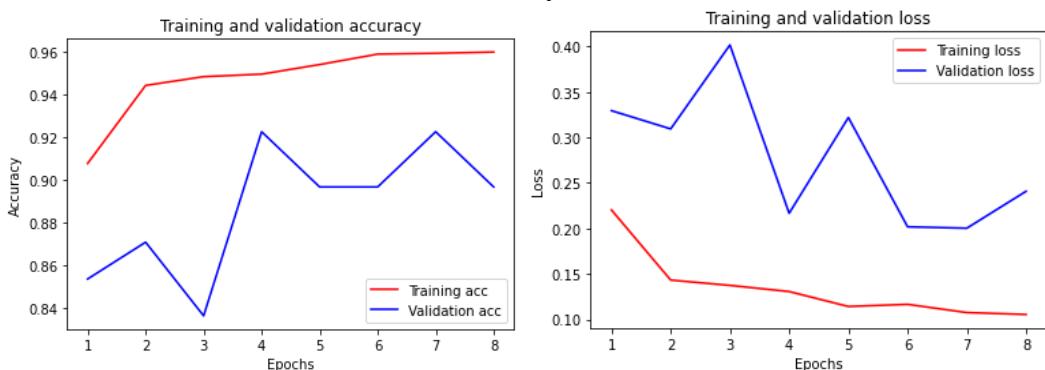
**Fine tune**

Loss of the model is - 0.26246997714042664  
Accuracy of the model is - 93.2692289352417 %



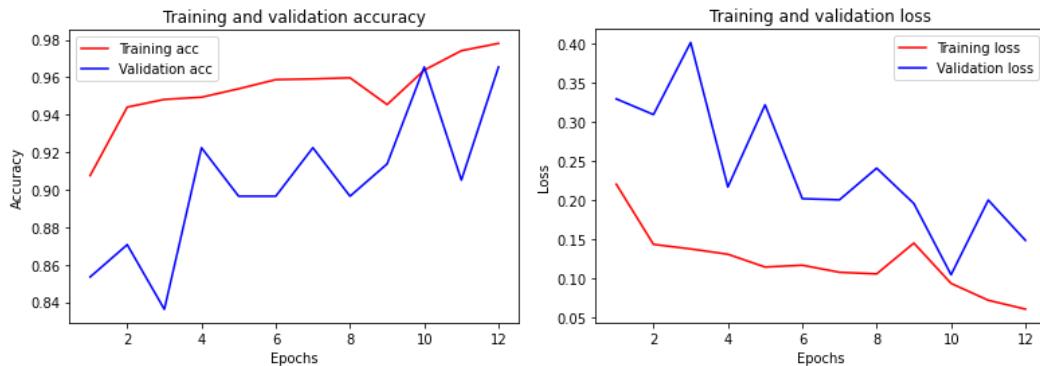
Optimizer: MSGD , lr: 0.01 , EPOCHS: 8

Loss of the model is - 0.25036028027534485  
Accuracy of the model is - 91.98718070983887 %



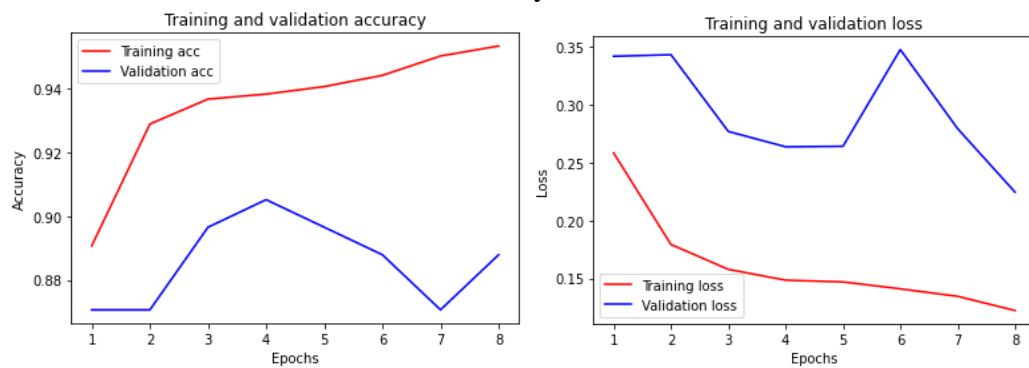
### Fine tune

Loss of the model is - 0.21113014221191406  
 Accuracy of the model is - 93.10897588729858 %



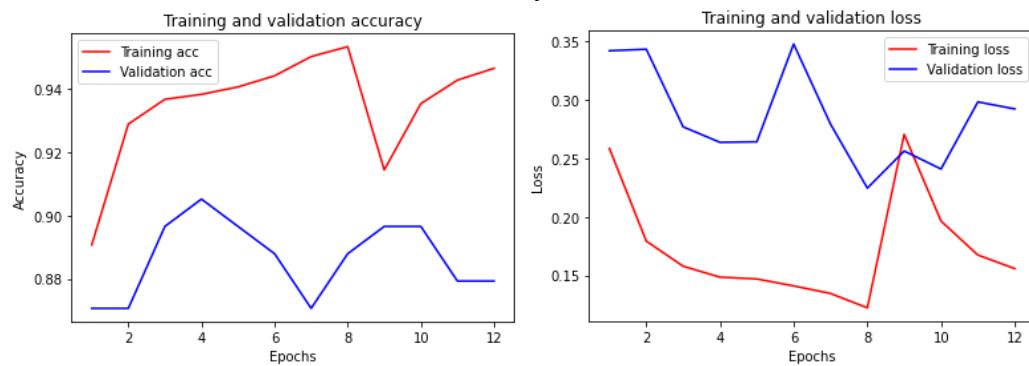
Optimizer: MSGD , lr: 0.001 , EPOCHS: 8

Loss of the model is - 0.23635070025920868  
 Accuracy of the model is - 91.34615659713745 %



### Fine tune

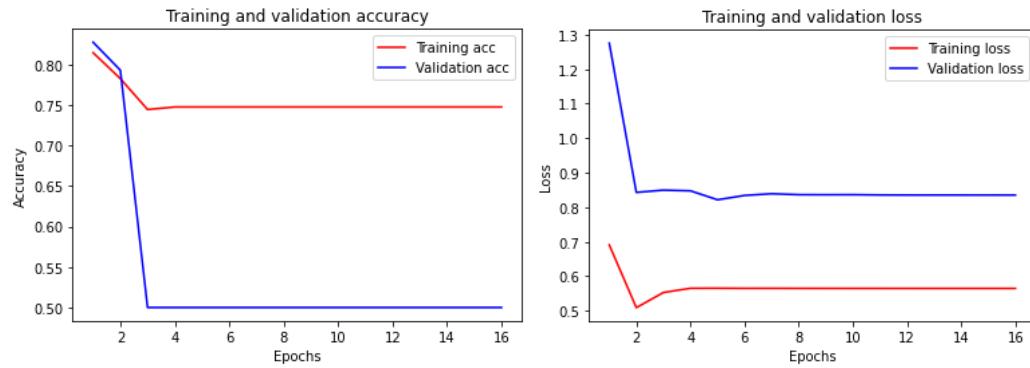
Loss of the model is - 0.26812058687210083  
 Accuracy of the model is - 89.42307829856873 %



Optimizer: MSGD , lr: 0.1 , EPOCHS: 16

Loss of the model is - 0.6987857818603516  
 Accuracy of the model is - 62.5 %

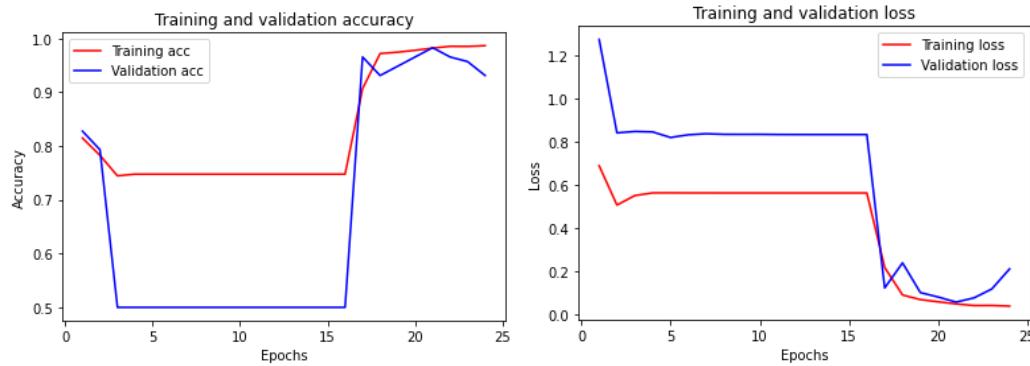
## חזר לתוכן עניינים



### Fine tune

Loss of the model is - 0.2226763814687729

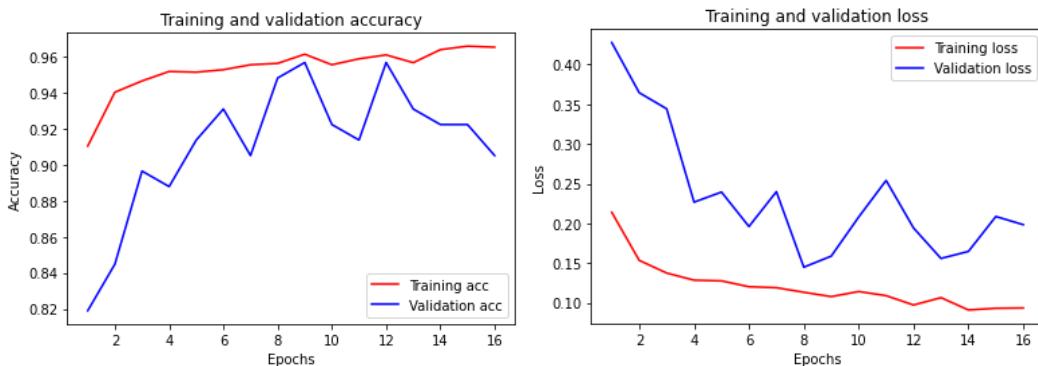
Accuracy of the model is - 92.46794581413269 %



Optimizer: MSGD , lr: 0.01 , EPOCHS: 16

Loss of the model is - 0.23749272525310516

Accuracy of the model is - 91.82692170143127 %

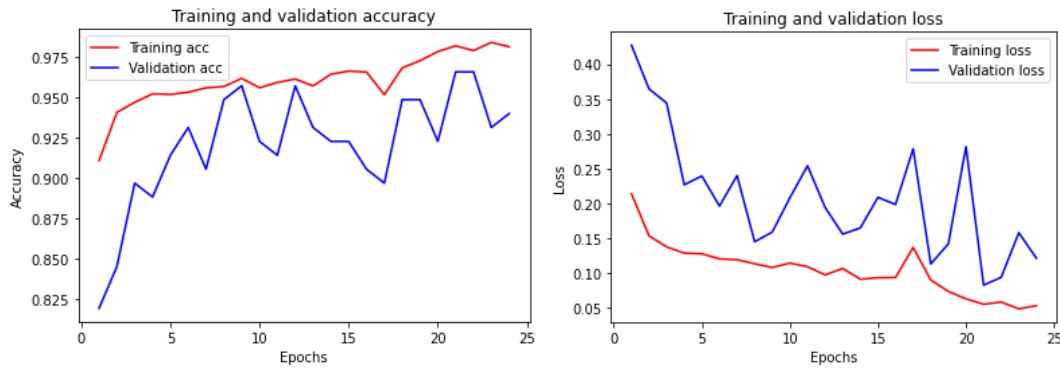


### Fine tune

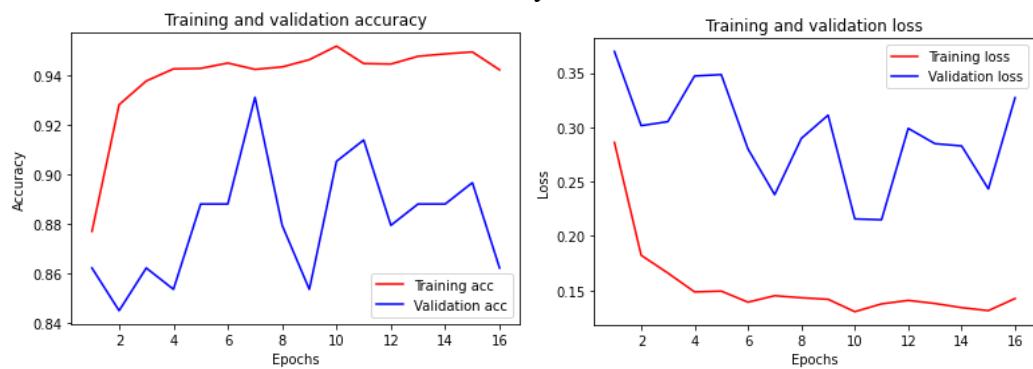
Loss of the model is - 0.21157097816467285

Accuracy of the model is - 93.42948794364929 %

## חזר לתוכן עניינים

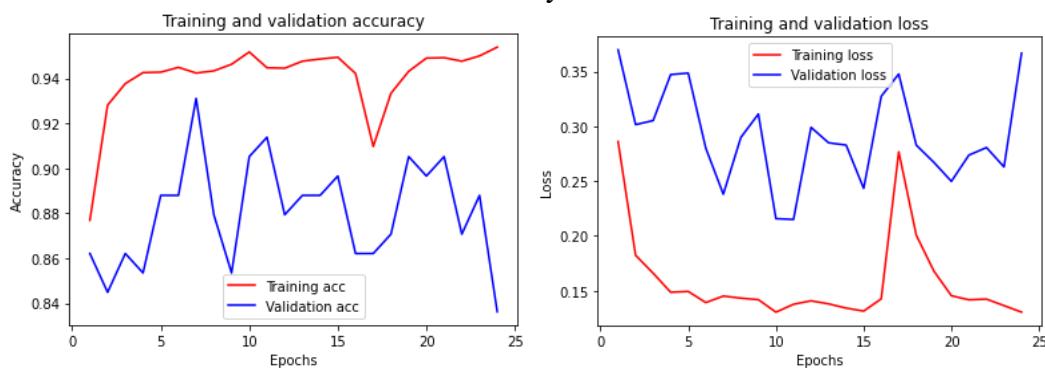


Optimizer: MSGD , lr: 0.001 , EPOCHS: 16  
 Loss of the model is - 0.24533207714557648  
 Accuracy of the model is - 91.34615659713745 %



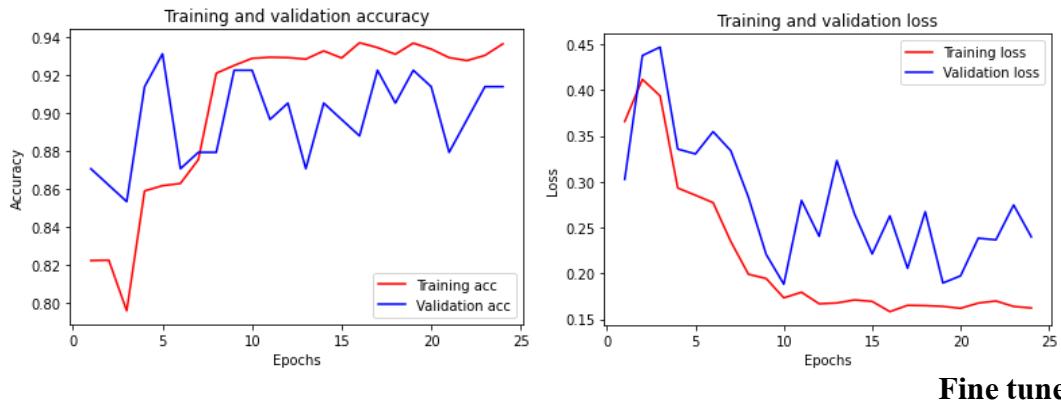
## Fine tune

Loss of the model is - 0.2625751197338104  
 Accuracy of the model is - 90.54487347602844 %



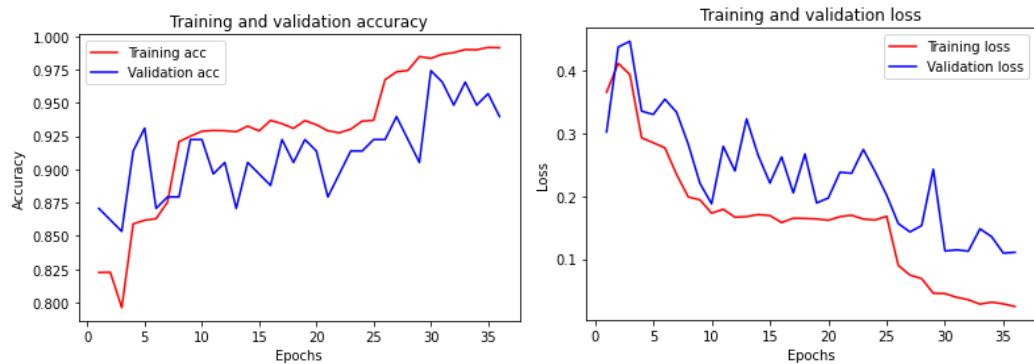
Optimizer: MSGD , lr: 0.1 , EPOCHS: 24  
 Loss of the model is - 0.25801050662994385  
 Accuracy of the model is - 91.18589758872986 %

## חזר לתוכן עניינים

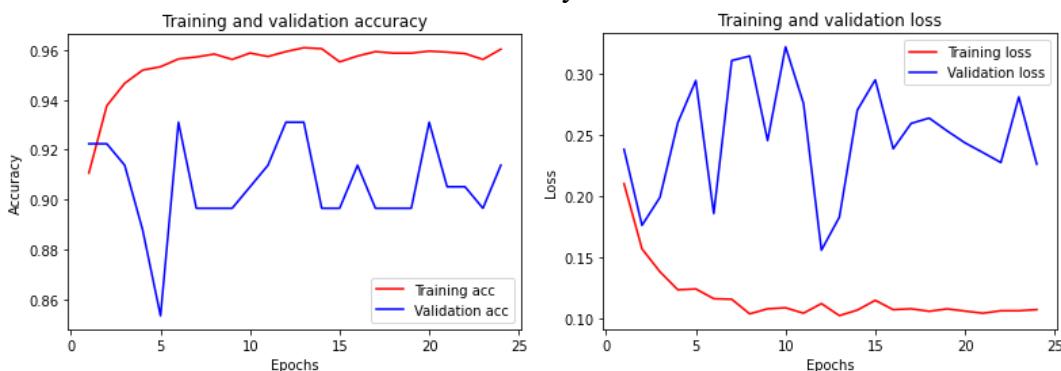


**Fine tune**

Loss of the model is - 0.18662378191947937  
Accuracy of the model is - 93.75 %



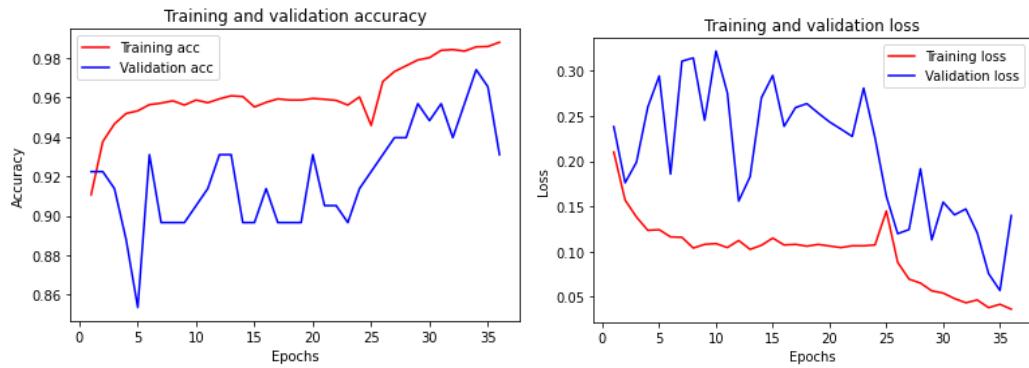
Optimizer: MSGD , lr: 0.01 , EPOCHS: 24  
Loss of the model is - 0.25147613883018494  
Accuracy of the model is - 92.30769276618958 %



**Fine tune**

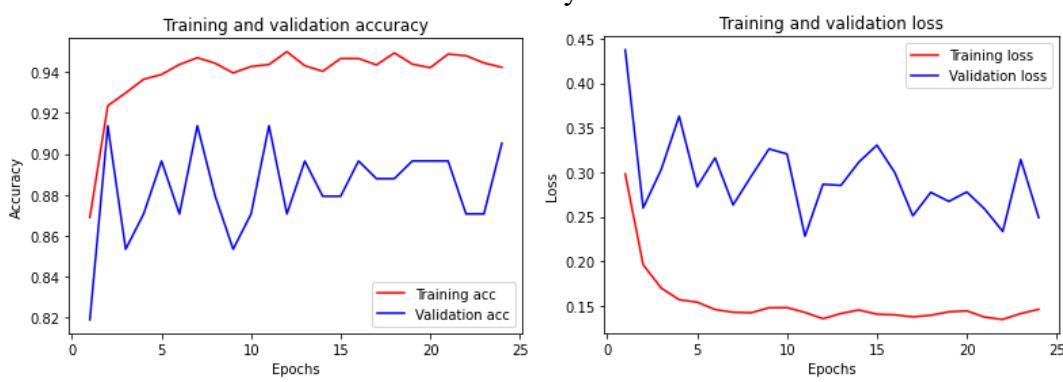
Loss of the model is - 0.24910785257816315  
Accuracy of the model is - 93.2692289352417 %

## חזר לתוכן עניינים



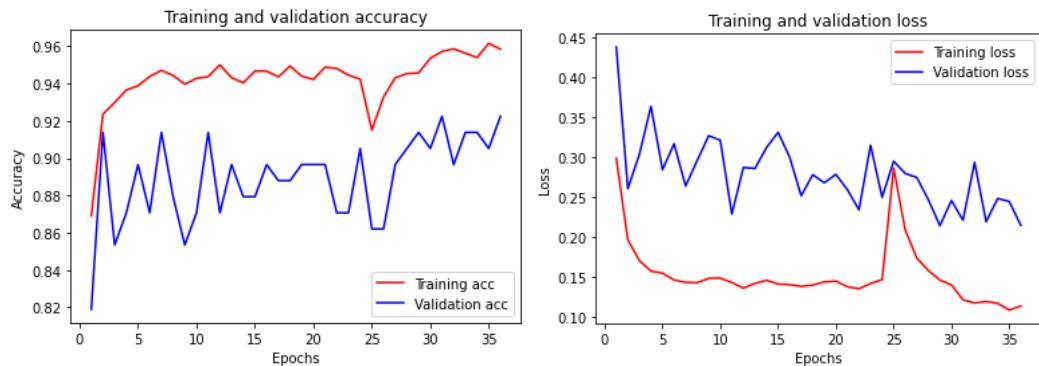
Optimizer: MSGD , lr: 0.001 , EPOCHS: 24

Loss of the model is - 0.2619633078575134  
Accuracy of the model is - 89.90384340286255 %



**Fine tune**

Loss of the model is - 0.23015598952770233  
Accuracy of the model is - 91.98718070983887 %



### 3.3 משימה

3. בדקו השפעת השינויים הבאים:

ה. DROPOUT - בדקו השפעת שינוי ההסתברות (זהו הפרמטר של DROPOUT) במידה וקיים שכבת DROPOUT.

ו. הפעילו מנגנון EARLY STOPPING בהתאם לאלגוריתם שנלמד בהרצאה (קובץ שקפים בנושא גולריזציה). האם הושג שיפור ביצועים באמצעות מנגנון זה?

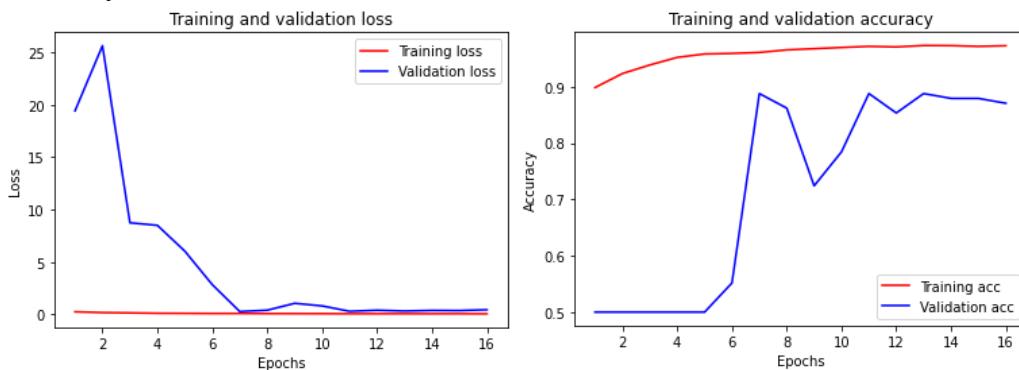
#### רשות א' סעיף א' - שינוי dropout probability.

רשות א'				
Loss	Accuracy%	F Score%	Threshold	dropout
0.27	90.22	93.47	0.80	0.1
0.31	89.58	93.07	0.90	0.2
0.43	87.98	92.81	0.90	0.3
5.08	62.66	77.15	0.90	0.4
1.46	73.55	84.23	0.85	0.5

Optimizer: Adam , lr: 0.001 , EPOCHS: 16 , dropout: 0.1

Loss of the model is - 0.27138131856918335

Accuracy of the model is - 90.22436141967773 %



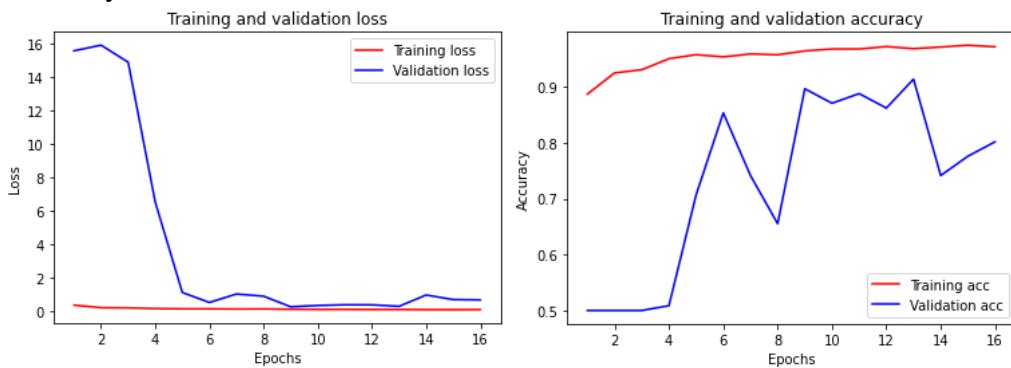
The f\_score max is: 0.9346991037131882

The corresponding threshold for the maximum f\_score is: 0.8

Optimizer: Adam , lr: 0.001 , EPOCHS: 16 , dropout: 0.2

Loss of the model is - 0.31630563735961914

Accuracy of the model is - 89.58333134651184 %



## חזר לתוכן עניינים

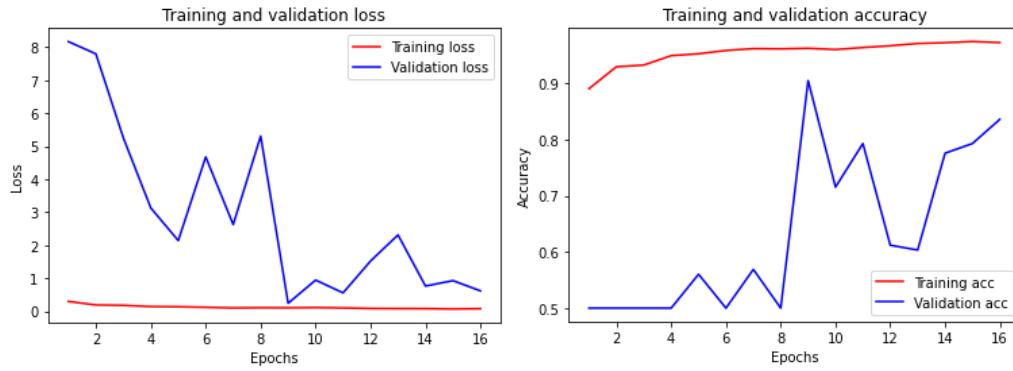
The f\_score max is: 0.9307692307692308

The corresponding threshold for the maximum f\_score is: 0.9

Optimizer: Adam , lr: 0.001 , EPOCHS: 16 , dropout: 0.3

Loss of the model is - 0.43890222907066345

Accuracy of the model is - 87.9807710647583 %



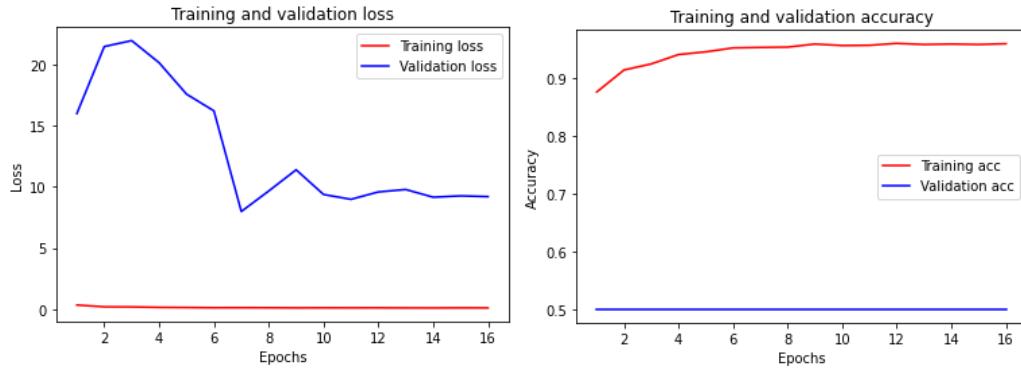
The f\_score max is: 0.9281210592686002

The corresponding threshold for the maximum f\_score is: 0.9

Optimizer: Adam , lr: 0.001 , EPOCHS: 16 , dropout: 0.4

Loss of the model is - 5.088131904602051

Accuracy of the model is - 62.66025900840759 %



The f\_score max is: 0.771513353115727

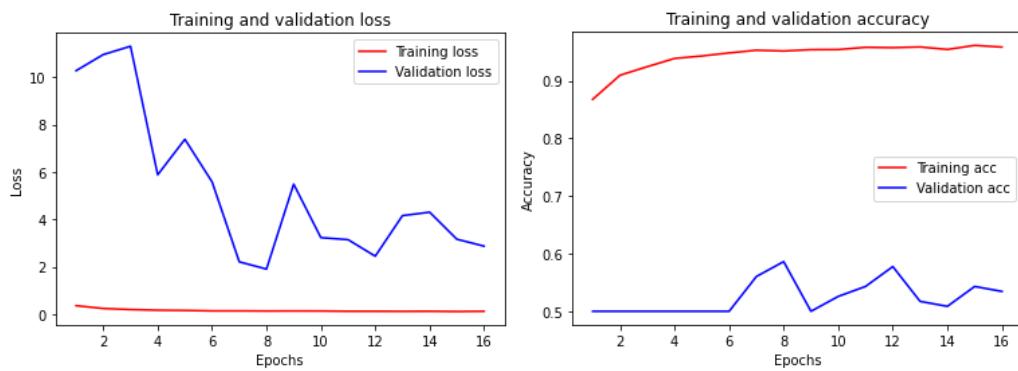
The corresponding threshold for the maximum f\_score is: 0.9

Optimizer: Adam , lr: 0.001 , EPOCHS: 16 , dropout: 0.5

Loss of the model is - 1.4698270559310913

Accuracy of the model is - 73.55769276618958 %

## חזר לתוכן עניינים



The f\_score max is: 0.8423475258918297

The corresponding threshold for the maximum f\_score is: 0.85

dropout	eccuracy	loss	ref model 1		
			drop out	accuracy	loss
0.1	90.224	0.2714	0.1	92.95%	0.23
0.2	89.58	0.3163			
0.3	87.98	0.4389			
0.4	62.66	5.0881			
0.5	73.557	1.4698			

מסקנה העלתה ה drop out לא שיפרה את accuracy ולא שיפרה את loss

## רשות א' סעיף ב' - add early stop and change patience -

### סיכום התוצאות

Adam	epoch=16	lr=0.001	batch=32
Loss	Accuracy%	Early stop patience	רשות א'
0.338	88.46	1	
0.74	72.27	2	
0.26	91.18	3	
<b>0.256</b>	<b>91.34</b>	<b>4</b>	
0.37	89.26	5	

התוצאה הטובה ביותר התקבלה עבור patience = 4

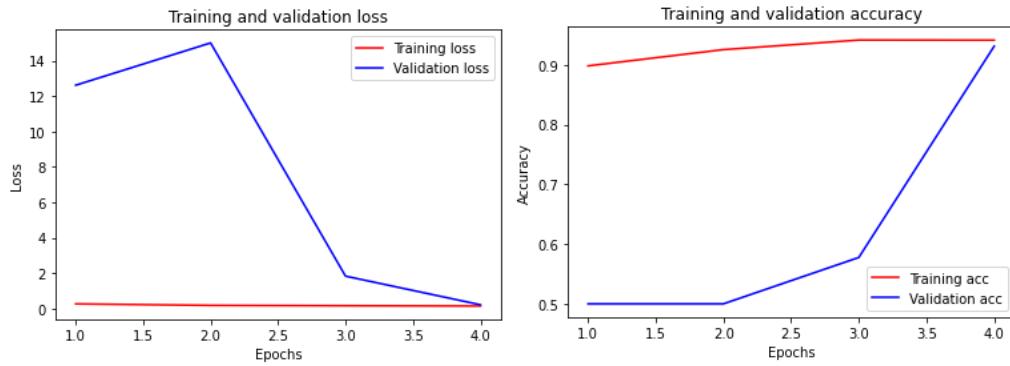
### פירוט התוצאות

Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , Early stop and patience: 1

Loss of the model is - 0.33810266852378845

Accuracy of the model is - 88.46153616905212 %

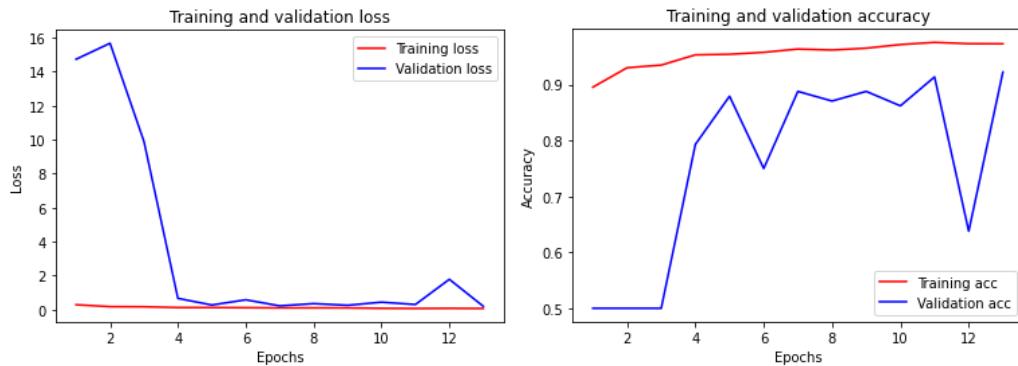
## חומר לתוכן עניינים



Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , Early stop and patience: 2

Loss of the model is - 0.7467219829559326

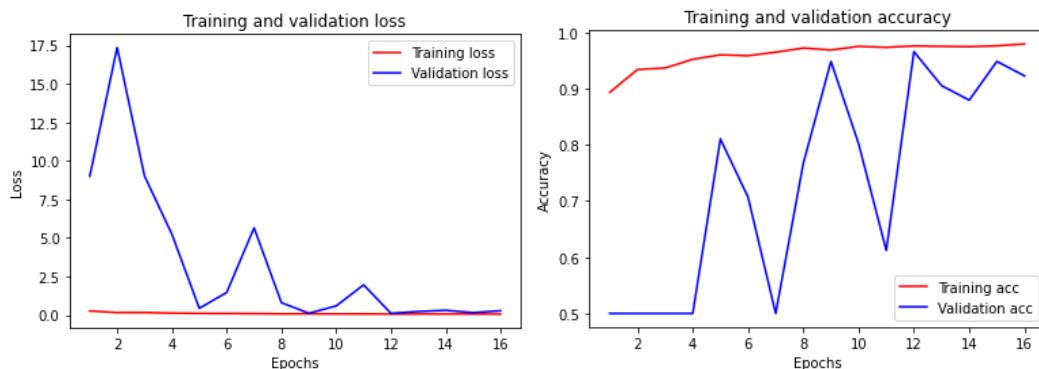
Accuracy of the model is - 72.27563858032227 %



Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , Early stop and patience: 3

Loss of the model is - 0.2600783109664917

Accuracy of the model is - 91.18589758872986 %

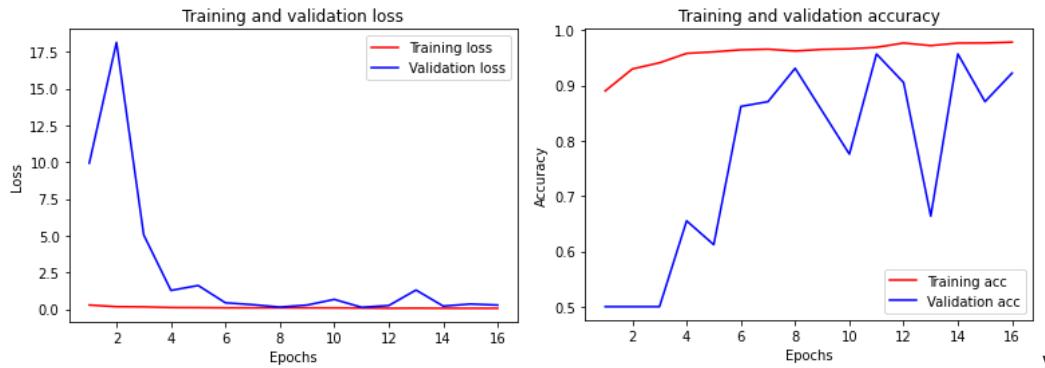


Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , Early stop and patience: 4

Loss of the model is - 0.2562257945537567

Accuracy of the model is - 91.34615659713745 %

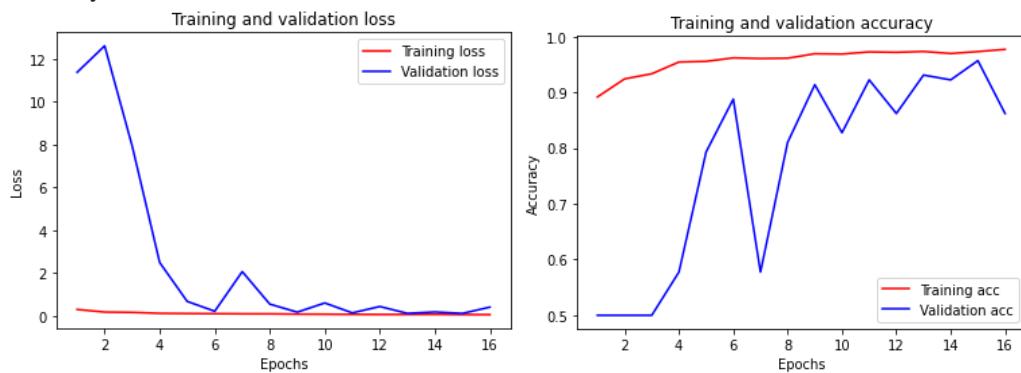
## חזר לתוכן עניינים



Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , Early stop and patience: 5

Loss of the model is - 0.3704276978969574

Accuracy of the model is - 89.26281929016113 %



## רשת ב' סעיף א' - שינוי drop

### סיכום התוצאות

Adam epoch=20 lr=0.001 batch=32

רשת ב'				
fine tune		only MLP train		dropout
Loss	Accuracy%	Loss	Accuracy%	
0.22	92.95	0.24	91.19	0.1
0.23	93.42	0.24	90.70	0.2
<b>0.20</b>	<b>93.27</b>	<b>0.24</b>	<b>91.50</b>	<b>0.3</b>
0.26	92.63	0.25	91.66	0.4
0.24	93.10	0.23	91.82	0.5

התוצאה הטובה ביותר ביותר התקבלה עבור dropout = 0.3

זאת מכיוון שהLOSS קטן משמעותית מהשאך.

לעומת accuracy שקרוב יחסית לתוצאה הטובה ביותר (0.2).

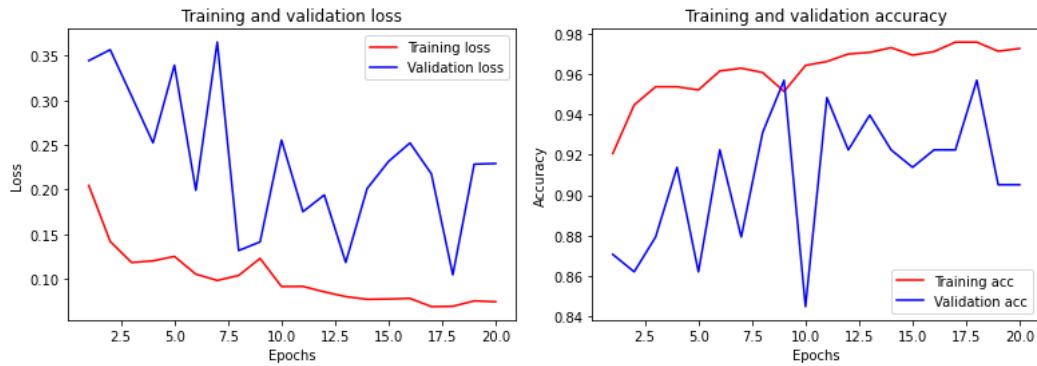
### פירוט התוצאות

Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , dropout: 0.1

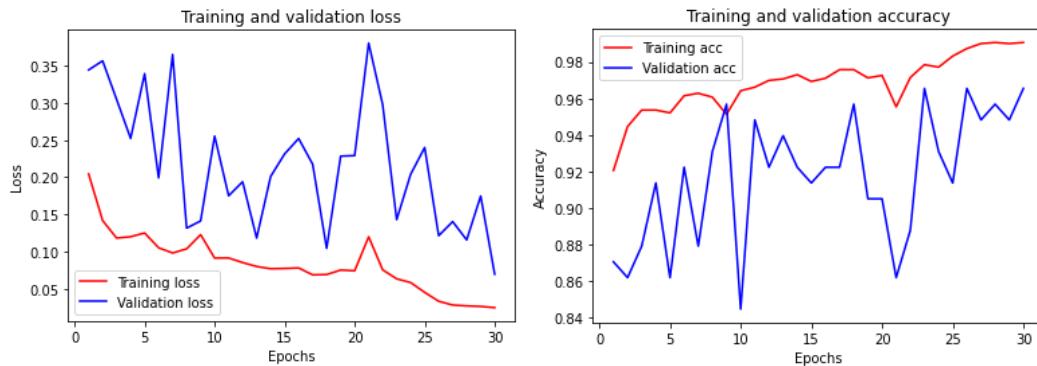
Loss of the model is - 0.24136535823345184

Accuracy of the model is - 91.18589758872986 %

## חזר לתוכן עניינים

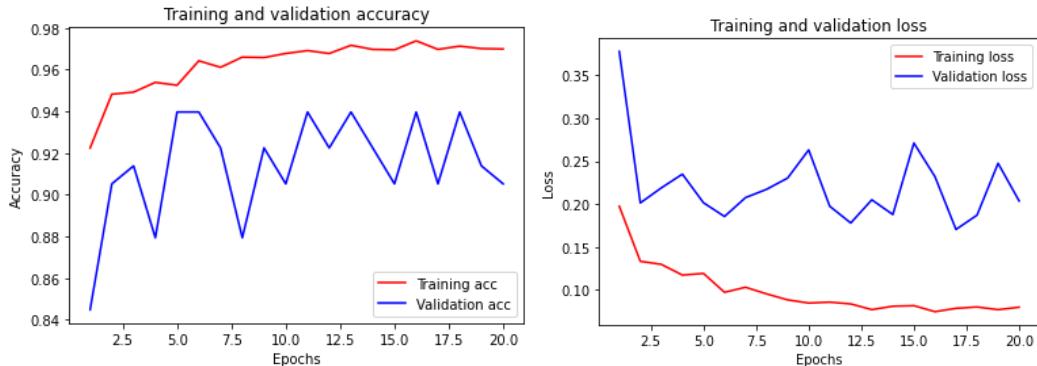


Loss of the model is - 0.21876946091651917  
 Accuracy of the model is - 92.94871687889099 %

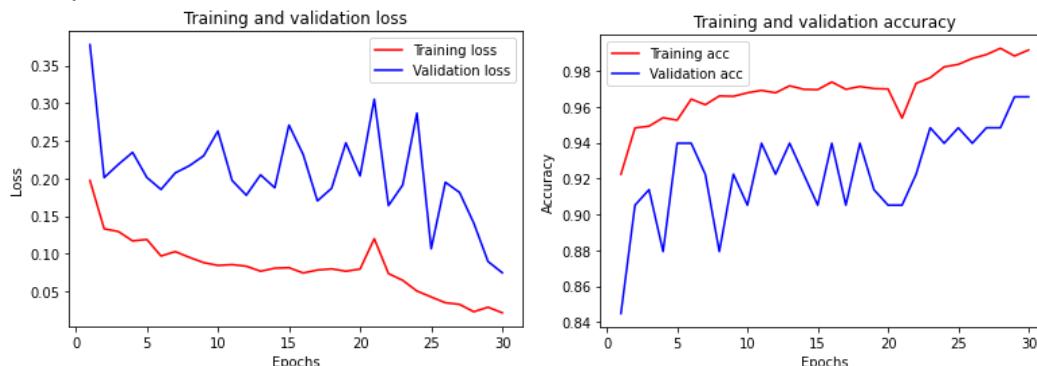


Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , dropout: 0.2

Loss of the model is - 0.24156707525253296  
 Accuracy of the model is - 90.70512652397156 %



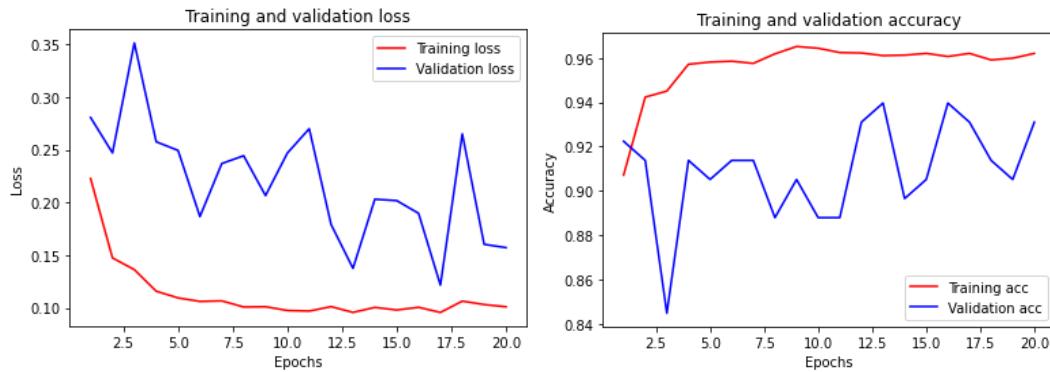
Loss of the model is - 0.23191498219966888  
 Accuracy of the model is - 93.42948794364929 %



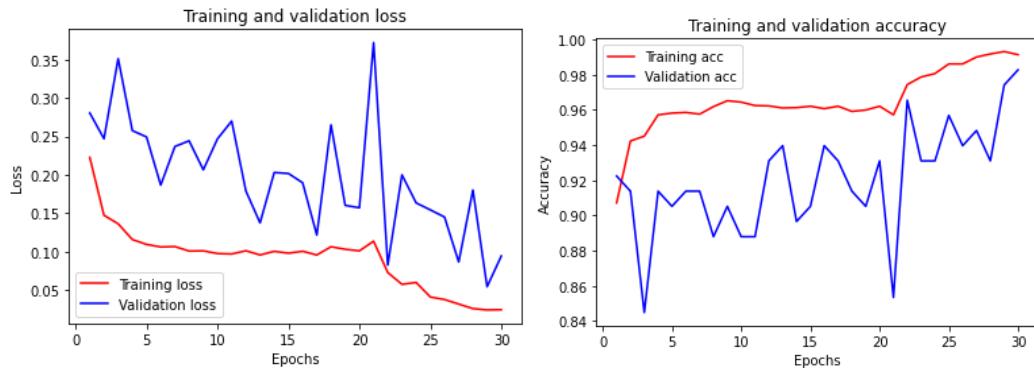
Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , dropout: 0.3

Loss of the model is - 0.2444939762353897  
 Accuracy of the model is - 91.50640964508057 %

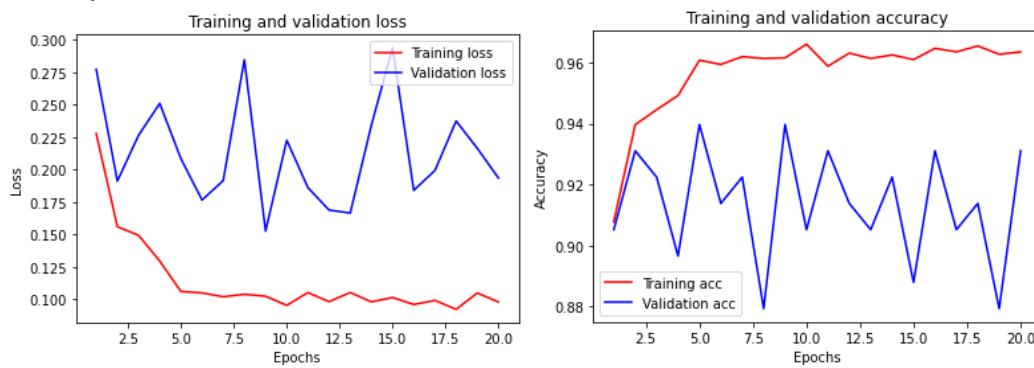
## חומר לתוכן עניינים



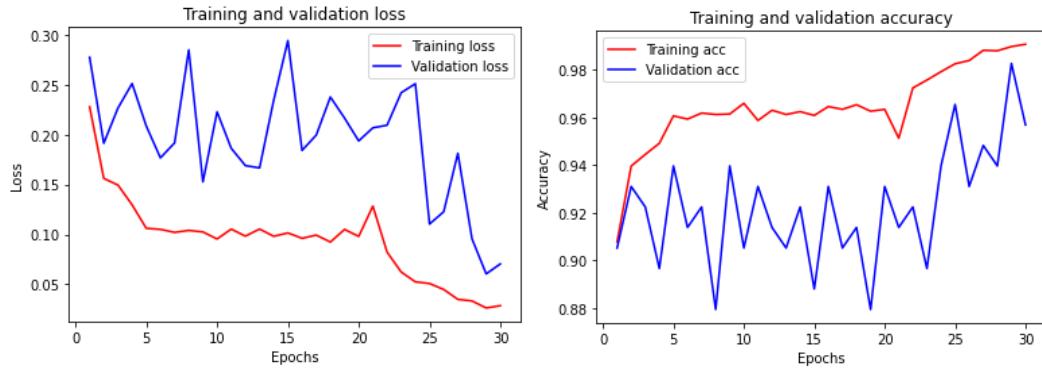
Loss of the model is - 0.20399989187717438  
 Accuracy of the model is - 93.2692289352417 %



Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , dropout: 0.4  
 Loss of the model is - 0.249877467751503  
 Accuracy of the model is - 91.66666865348816 %



Loss of the model is - 0.2597370743751526  
 Accuracy of the model is - 92.62820482254028 %

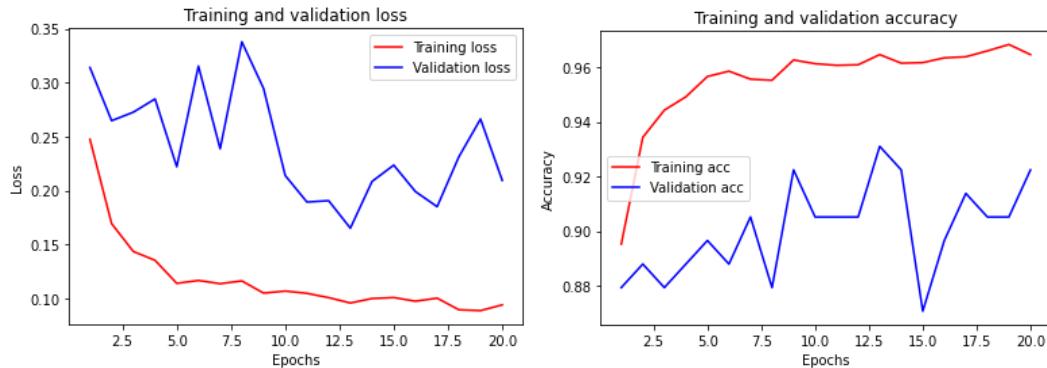


## חזר לתוכן עניינים

Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , dropout: 0.5

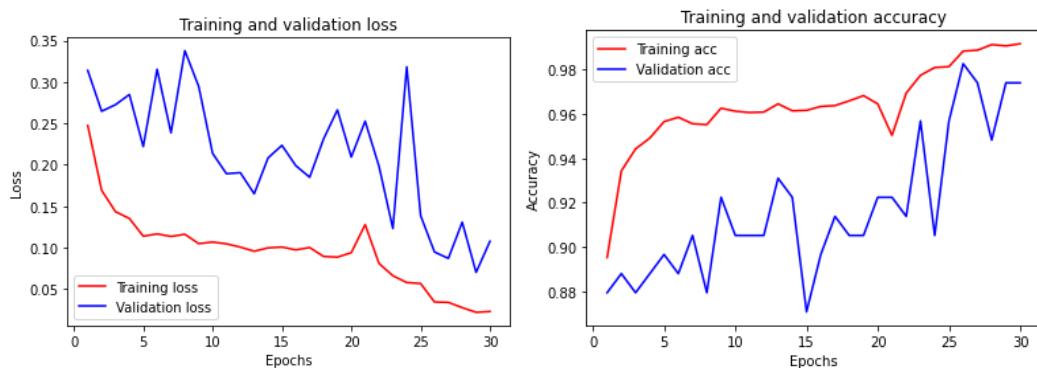
Loss of the model is - 0.23211489617824554

Accuracy of the model is - 91.82692170143127 %



Loss of the model is - 0.2431386262178421

Accuracy of the model is - 93.10897588729858 %



## רשות ב' סעיף ב' -

Adam epoch=20 lr=0.001 batch=32

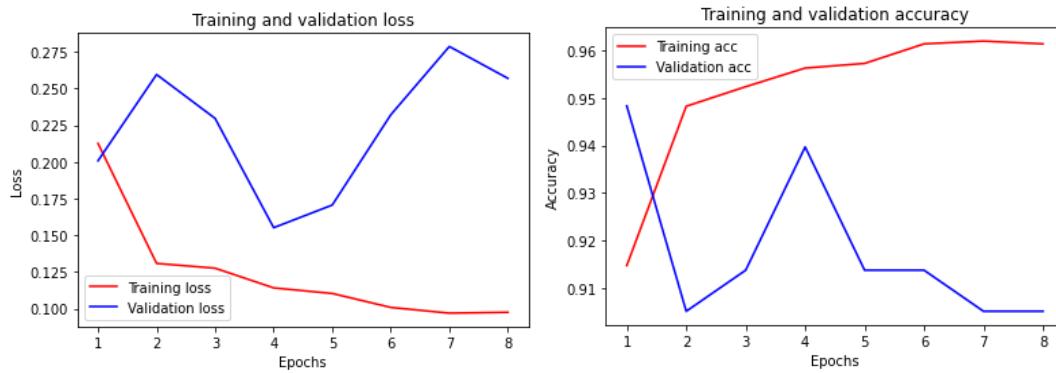
רשות ב'				
fine tune		only MLP train		Early stop patience
Loss	Accuracy%	Loss	Accuracy%	
0.24	93.43	0.24	91.66	1
0.26	93.59	0.24	91.34	2
<b>0.23</b>	<b>94.23</b>	<b>0.25</b>	<b>91.02</b>	<b>3</b>
0.32	92.31	0.25	91.50	4
0.58	92.62	0.24	91.02	5

Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , Early stop and patience: 1

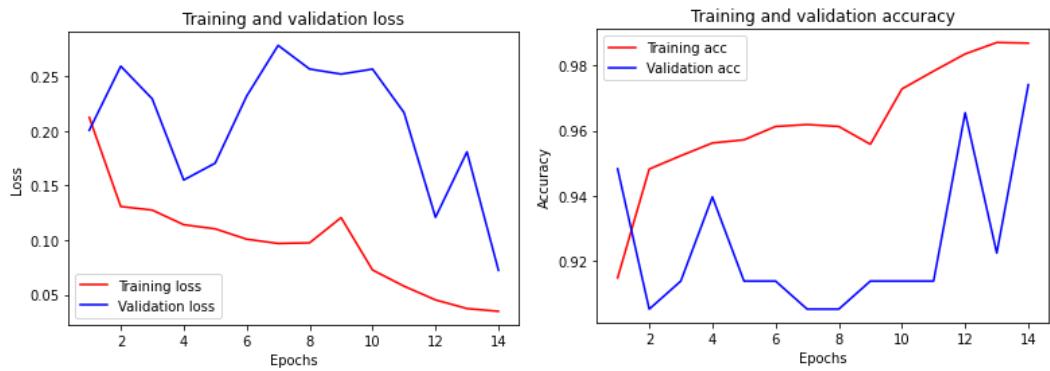
Loss of the model is - 0.23740611970424652

Accuracy of the model is - 91.66666865348816 %

## חומר לתוכן עניינים

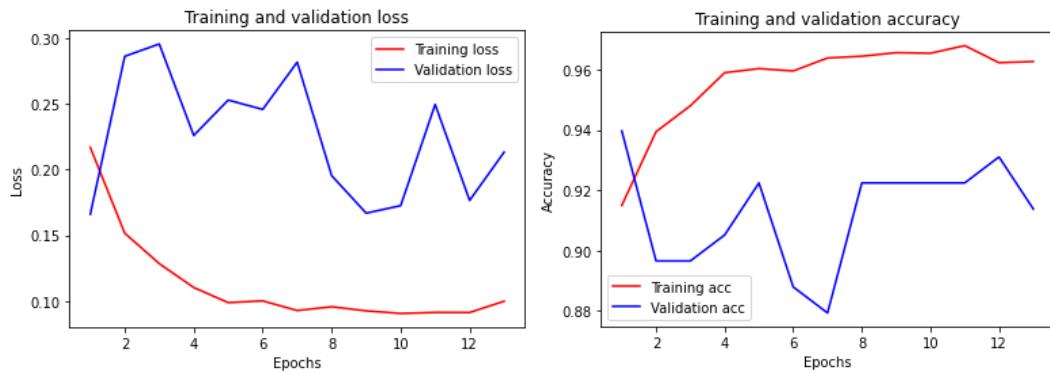


Loss of the model is - 0.24033886194229126  
 Accuracy of the model is - 93.42948794364929 %

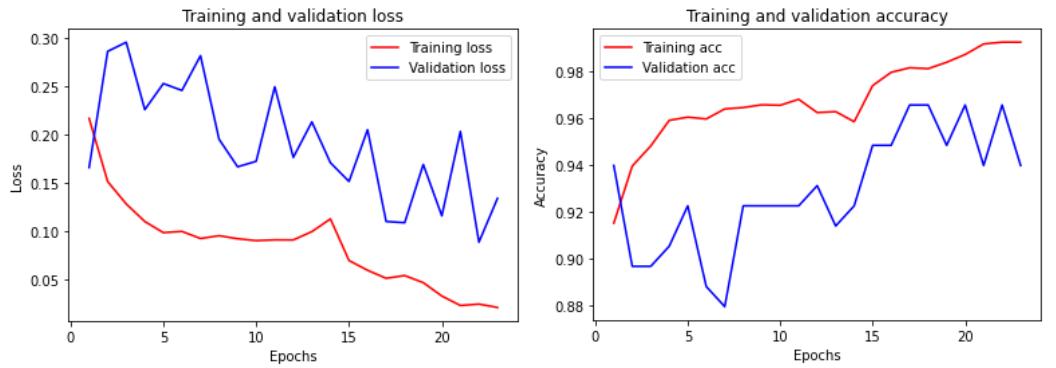


Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , Early stop and patience: 2

Loss of the model is - 0.24119721353054047  
 Accuracy of the model is - 91.34615659713745 %



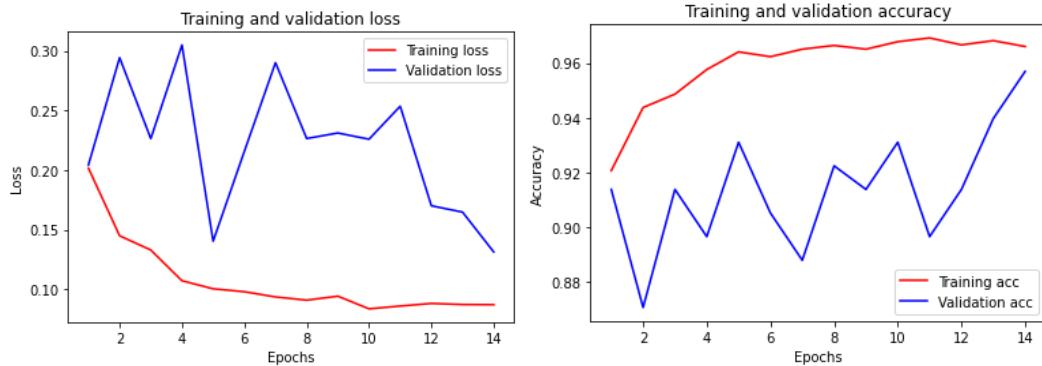
Loss of the model is - 0.25712987780570984  
 Accuracy of the model is - 93.58974099159241 %



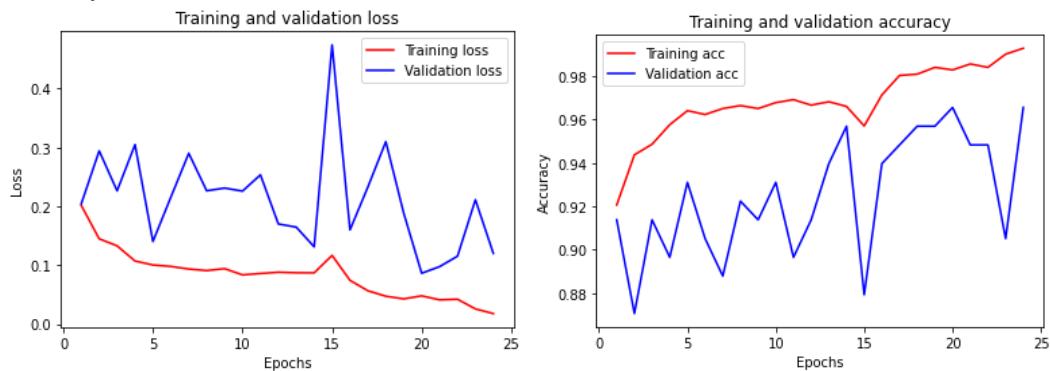
## חזר לתוכן עניינים

Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , Early stop and patience: 3

Loss of the model is - 0.24508140981197357  
Accuracy of the model is - 91.02563858032227 %

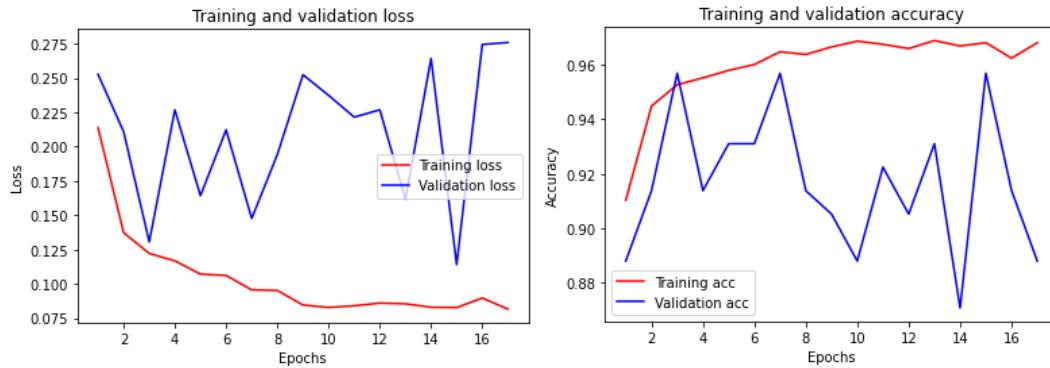


Loss of the model is - 0.23597079515457153  
Accuracy of the model is - 94.2307710647583 %



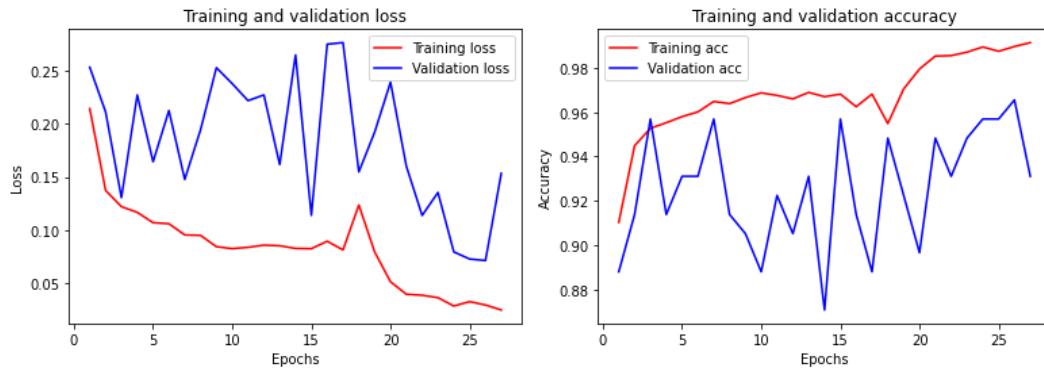
Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , Early stop and patience: 4

Loss of the model is - 0.25225937366485596  
Accuracy of the model is - 91.50640964508057 %



Loss of the model is - 0.3241201639175415  
Accuracy of the model is - 92.30769276618958 %

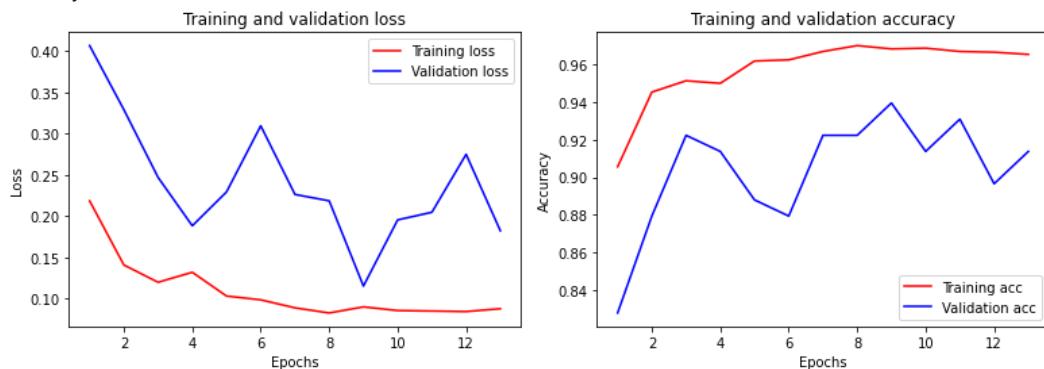
## חזרה לתוכן עניינים



Optimizer: Adam , lr: 0.001 , EPOCHS: 20 , Early stop and patience: 5

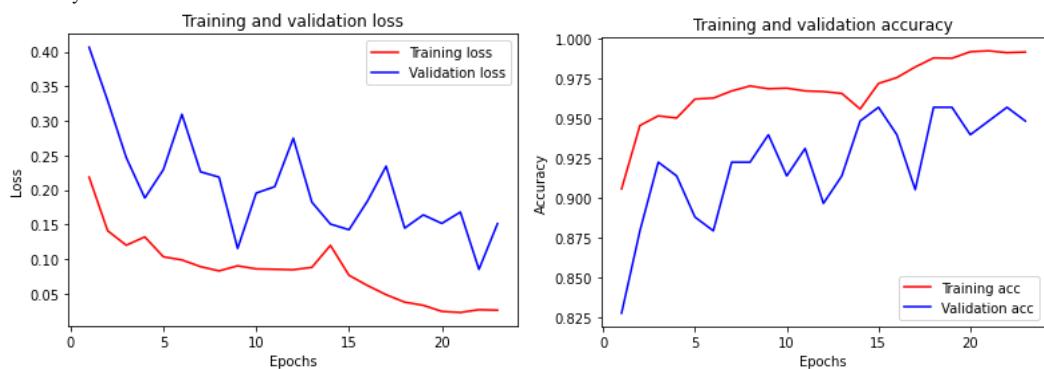
Loss of the model is - 0.24034340679645538

Accuracy of the model is - 91.02563858032227 %



Loss of the model is - 0.2585001587867737

Accuracy of the model is - 92.62820482254028 %



## בדיקות היתכנות

לקחנו את הפרמטרים הנבחרים משלבים 3 וניסינו להקליל לכדי רשות אחת בעבר כל רשות, ובדקנו את הביצועים שהתקבלו.

## רשות א'

הרצה עם הפרמטרים שנבחרו :

Optimizer: RMSprop , lr: 0.001 , EPOCHS: 16 , drop: 0.1 , Early stop and patience: 4

```
### CNN network Model ###
```

```
### Optimizer: RMSprop , lr: 0.001 , EPOCHS: 16 , drop: 0.1 , Early stop and
patience: 4 ###
```

## חזר לתוכן עניינים

Model: "sequential"

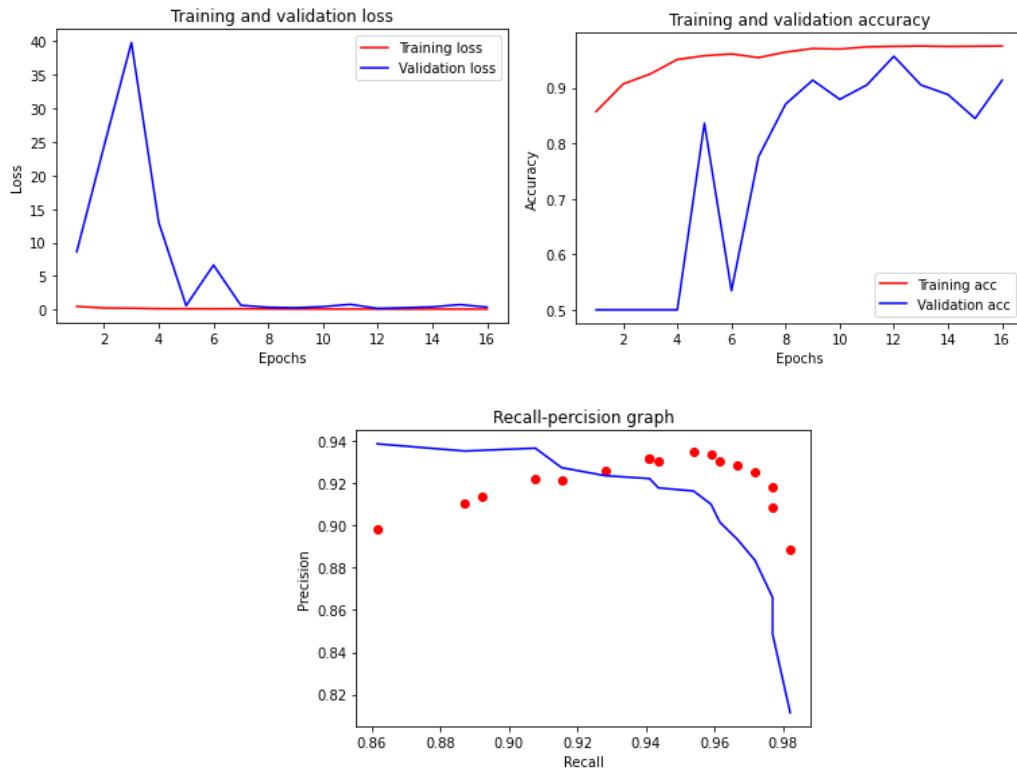
Layer (type)	Output Shape	Param #
<hr/>		
conv2d (Conv2D)	(None, 250, 250, 32)	320
batch_normalization (BatchNormalization)	(None, 250, 250, 32)	128
max_pooling2d (MaxPooling2D)	(None, 125, 125, 32)	0
conv2d_1 (Conv2D)	(None, 125, 125, 64)	18496
dropout (Dropout)	(None, 125, 125, 64)	0
batch_normalization_1 (BatchNormalization)	(None, 125, 125, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 63, 63, 64)	0
conv2d_2 (Conv2D)	(None, 63, 63, 64)	36928
batch_normalization_2 (BatchNormalization)	(None, 63, 63, 64)	256
max_pooling2d_2 (MaxPooling2D)	(None, 32, 32, 64)	0
conv2d_3 (Conv2D)	(None, 32, 32, 128)	73856
dropout_1 (Dropout)	(None, 32, 32, 128)	0
batch_normalization_3 (BatchNormalization)	(None, 32, 32, 128)	512
max_pooling2d_3 (MaxPooling2D)	(None, 16, 16, 128)	0
conv2d_4 (Conv2D)	(None, 16, 16, 128)	147584
dropout_2 (Dropout)	(None, 16, 16, 128)	0
batch_normalization_4 (BatchNormalization)	(None, 16, 16, 128)	512
max_pooling2d_4 (MaxPooling2D)	(None, 8, 8, 128)	0
conv2d_5 (Conv2D)	(None, 8, 8, 256)	295168
dropout_3 (Dropout)	(None, 8, 8, 256)	0
batch_normalization_5 (BatchNormalization)	(None, 8, 8, 256)	1024
max_pooling2d_5 (MaxPooling2D)	(None, 4, 4, 256)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 128)	524416
dropout_4 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129
<hr/>		

Total params: 1,099,585  
 Trainable params: 1,098,241  
 Non-trainable params: 1,344

---

Loss of the model is - 0.2856387495994568  
 Accuracy of the model is - 91.18589758872986 %

## חומר לתוכן עניינים



## רשות ב'

הרצה עם הפרמטרים שנבחרו :

Optimizer: RMSprop , lr: 0.001 , EPOHCS: 24 , drop: 0.3 , Early stop and patience: 3 ,  
MoreLayer: 3

```
### CNN transfer learning network Model ###
### test the Final Result with optimal parameters found ###
### Optimizer: RMSprop , lr: 0.001 , EPOHCS: 24 , drop: 0.3 , Early stop and
patience: 3 , MoreLayer: 3 ###
```

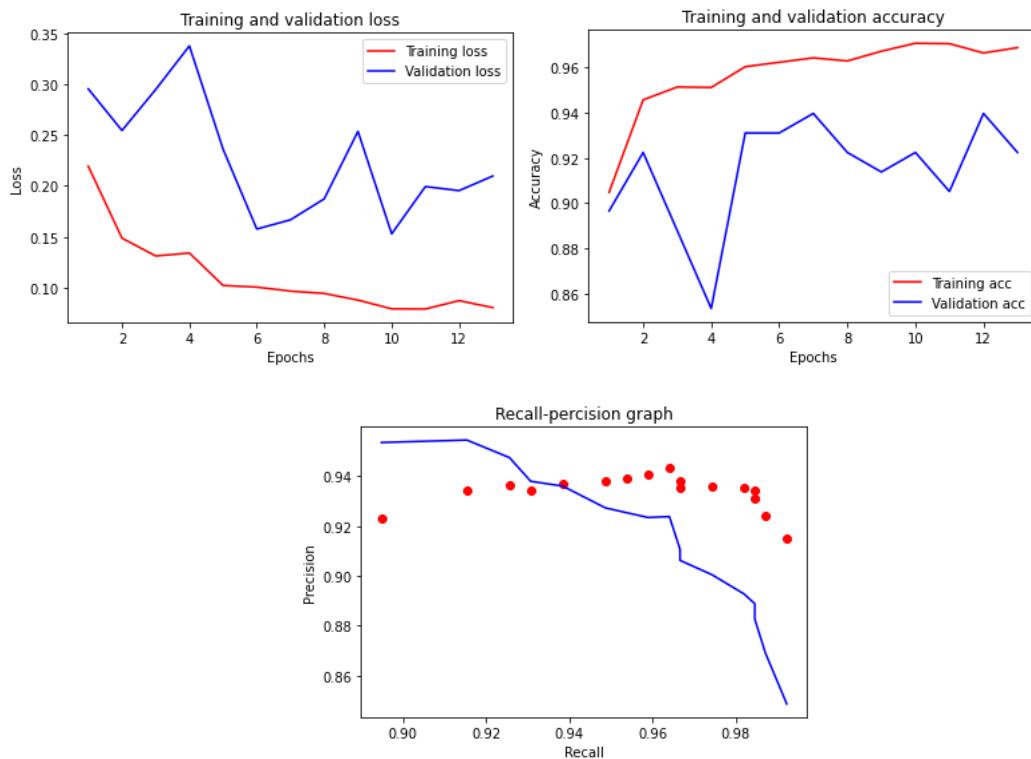
Model: "sequential"

Layer (type)	Output Shape	Param #	Trainable
<hr/>			
resnet50v2 (Functional)	(None, 8, 8, 2048)	23564800	Y
<hr/>			
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0	Y
dense (Dense)	(None, 128)	262272	Y
dropout (Dropout)	(None, 128)	0	Y
dense_1 (Dense)	(None, 128)	16512	Y
dropout_1 (Dropout)	(None, 128)	0	Y
dense_2 (Dense)	(None, 128)	16512	Y
dropout_2 (Dropout)	(None, 128)	0	Y
dense_3 (Dense)	(None, 128)	16512	Y
dropout_3 (Dropout)	(None, 128)	0	Y

## חומר לתוכן עניינים

```
dense_4 (Dense)           (None, 1)          129      Y
=====
Total params: 23,876,737
Trainable params: 311,937
Non-trainable params: 23,564,800
```

Loss of the model is - 0.22509434819221497  
 Accuracy of the model is - 92.78846383094788 %



The f\_score max is: 0.9435382685069009  
 The corresponding threshold for the maximum f\_score is: 0.5

### fine tune

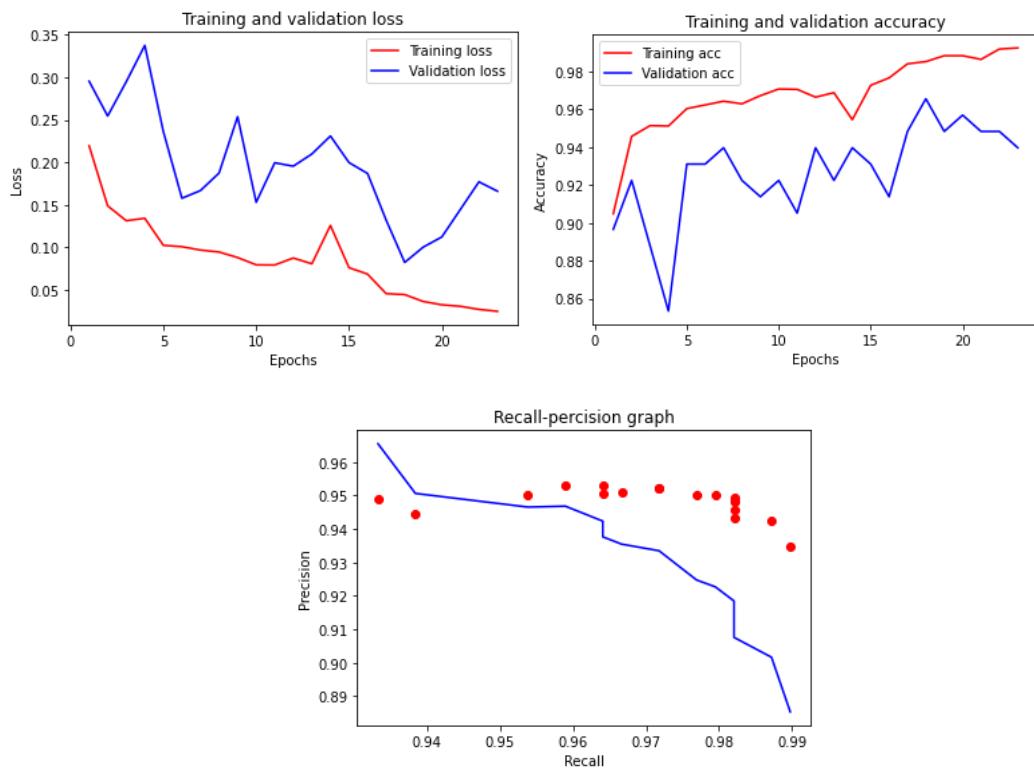
```
Layer (type)          Output Shape         Param #  Trainable
=====
resnet50v2 (Functional) (None, 8, 8, 2048) 23564800  Y
global_average_pooling2d (G (None, 2048) 0  Y
lobalAveragePooling2D)

dense (Dense)          (None, 128)        262272   Y
dropout (Dropout)       (None, 128)        0  Y
dense_1 (Dense)         (None, 128)        16512    Y
dropout_1 (Dropout)     (None, 128)        0  Y
dense_2 (Dense)         (None, 128)        16512    Y
dropout_2 (Dropout)     (None, 128)        0  Y
dense_3 (Dense)         (None, 128)        16512    Y
dropout_3 (Dropout)     (None, 128)        0  Y
dense_4 (Dense)         (None, 1)          129      Y
=====

Total params: 23,876,737
Trainable params: 15,545,985
Non-trainable params: 8,330,752
```

## חומר לתוכן עניינים

Loss of the model is - 0.19996796548366547  
 Accuracy of the model is - 93.91025900840759 %



The f\_score max is: 0.953105196451204

The corresponding threshold for the maximum f\_score is: 0.7000000000000001

### סיכום התוצאות:

Loss	Accuracy%	F Score%	Threshold	רפרנס	רשות א'
0.23	92.948	94.45	0.5	רפרנס	
0.285	91.185	93.467	0.45	סופי	
0.24	91.025	-	-	only MLP train	רפרנס
0.23	93.1	95.71	0.8	fine tune	
0.225	92.788	94.35	0.5	only MLP train	רשות ב'
0.2	93.91	95.3	0.7	fine tune	סופי

ניתן לראות ששילוב כלל הפרמטרים שיצאו בסעיפים הקודמים לא בהכרח תורם לשיפור ביצועי הרשות.

למשל, ברשות א' דוגמא ראיינו ירידה בביטויים, בעוד שברשות ב' ראיינו עלייה בביטויים.

## מסקנות כלליות

ראינו במהלך הפרויקט כי דוקא שינוי בפרמטר יחיד יכול להוביל לביצועים גבוהים יותר מאשר שילוב של מספר שינויים בפרמטרים.  
למשל:

עבור רשת ב' - כאשר שינו רק את patience=3 early stop קיבלנו דוקא דיוק גבוהה יותר של 94.23% מאשר סיכום כל השינויים היחידים כפי שעשינו בסוף עם מספר פרמטרים. כמו כן, כאשר ביצענו הוספה שכבות בלבד, ראינו 3 שכבות נסתרות לבlok MLP של רשת ב' הגענו לשיפור דומה ב accuracy של הרשת: 94.23%

הופעה זו ברורה לנו מכיוון שרשותנו נוירונים שלנו עושים שימוש בפונקציות אקטיבציה לא לינאריות (כמו sigmoid) ולכן בעצם אינן מערכות לינאריות.

הרשות הנבחרת שהגיעה לתוצאות הטובות ביותר הינה רשת ב' בעלת הפרמטרים הבאים:

Optimizer: RMSprop

lr: 0.001

EPOHCS: 24

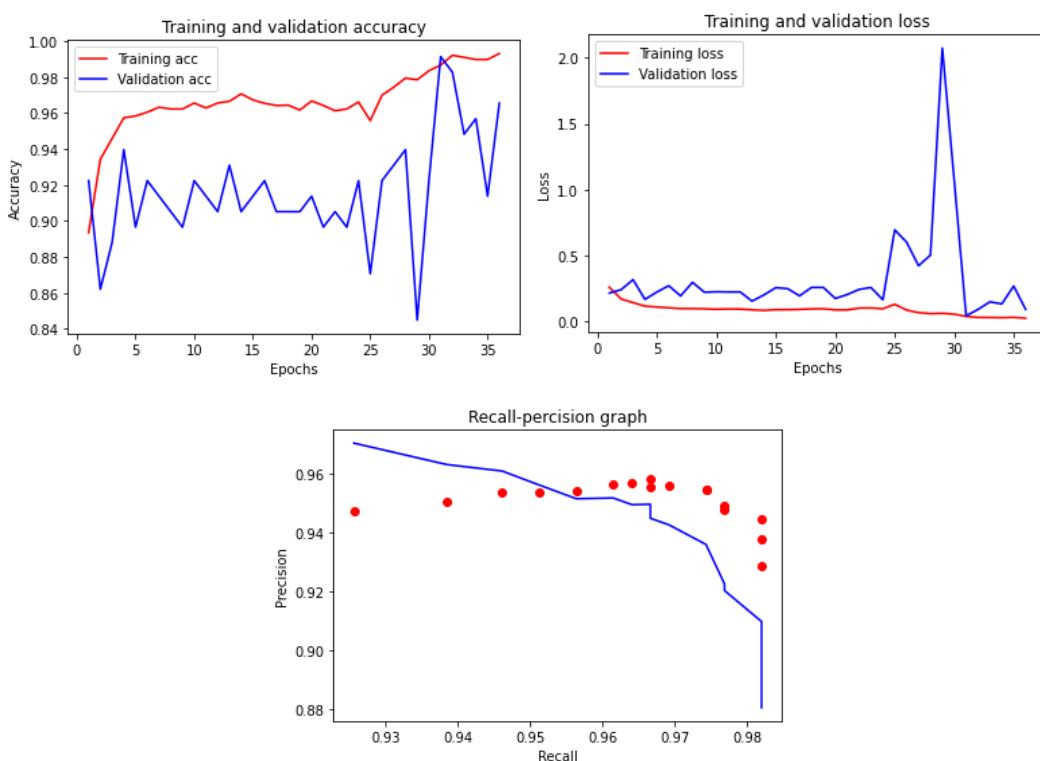
לא שינוי שכבות, ולא dropout, לא early stopping, תוצאות הרשת הן:

Loss of the model is - 0.20316524803638458

Accuracy of the model is - 94.39102411270142 %

The f\_score max is: 0.9580686149936468

The corresponding threshold for the maximum f\_score is: 0.55



## סיכום

[https://en.wikipedia.org/wiki/Multilayer\\_perceptron](https://en.wikipedia.org/wiki/Multilayer_perceptron)

[https://en.wikipedia.org/wiki/Rectifier\\_\(neural\\_networks\)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks))

[https://en.wikipedia.org/wiki/Stochastic\\_gradient\\_descent](https://en.wikipedia.org/wiki/Stochastic_gradient_descent)

[https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network)

[https://en.wikipedia.org/wiki/Artificial\\_neural\\_network](https://en.wikipedia.org/wiki/Artificial_neural_network)

<https://www.pico.net/kb/the-role-of-bias-in-neural-networks>

<https://www.kaggle.com/code/madz2000/pneumonia-detection-using-cnn-92-6-accuracy/comments>

[https://keras.io/api/callbacks/reduce\\_lr\\_on\\_plateau](https://keras.io/api/callbacks/reduce_lr_on_plateau)

<https://www.kaggle.com/code/sanwal092/intro-to-cnn-using-keras-to-predict-pneumonia>

[https://en.wikipedia.org/wiki/Training,\\_validation,\\_and\\_test\\_data\\_sets](https://en.wikipedia.org/wiki/Training,_validation,_and_test_data_sets)

[https://www.tensorflow.org/api\\_docs/python/tf/keras/preprocessing/image/ImageDataGenerator](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator)

<https://vijayabhaskar96.medium.com/tutorial-image-classification-with-keras-flow-from-directory-and-generators-95f75ebe5720>

[https://www.tensorflow.org/api\\_docs/python/tf/keras/layers/Conv2D](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv2D)

<https://machinelearningmastery.com/difference-test-validation-datasets>

<https://www.kaggle.com/code/swetash/transfer-learning-detect-pneumonia-cnn-94-acc/notebook>

<https://keras.io/api/applications/resnet>

[https://keras.io/guides/transfer\\_learning](https://keras.io/guides/transfer_learning)

<https://www.mdpi.com/2076-3417/12/6/2851/htm>

<https://www.kaggle.com/code/saumitandhari/resnet50v2/notebook>

[https://keras.io/api/layers/regularization\\_layers/dropout/](https://keras.io/api/layers/regularization_layers/dropout/)

cnn-network

<https://www.kaggle.com/code/tavetisyan/x-ray-pneumonia-detection-97-test-accuracy>

מקור לבסיס רשות ב':

<https://www.kaggle.com/code/swetash/transfer-learning-detect-pneumonia-cnn-94-acc/notebook>

<https://www.geeksforgeeks.org/python-opencv-cv2-imread-method>

<https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning>

[https://www.researchgate.net/figure/Architectures-of-the-best-fine-tuning-models-namely-VGG16-ResNet50V2-and-DenseNet169\\_fig3\\_344341113](https://www.researchgate.net/figure/Architectures-of-the-best-fine-tuning-models-namely-VGG16-ResNet50V2-and-DenseNet169_fig3_344341113)

[https://www.researchgate.net/publication/341677998\\_A\\_modified\\_deep\\_convolutional\\_neural\\_network\\_for\\_detecting\\_COVID-19\\_and\\_pneumonia\\_from\\_chest\\_X-ray\\_images\\_based\\_on\\_the\\_concatenation\\_of\\_Xception\\_and\\_ResNet50V2](https://www.researchgate.net/publication/341677998_A_modified_deep_convolutional_neural_network_for_detecting_COVID-19_and_pneumonia_from_chest_X-ray_images_based_on_the_concatenation_of_Xception_and_ResNet50V2)

<https://stackoverflow.com/questions/48491728/what-is-the-behavior-of-same-padding-when-stride-is-greater-than-1>

מידע על דלקת ריאות:

<https://www.wikirefua.org.il/w/index.php/%D7%93%D7%9C%D7%A7%D7%AA%D7%A8%D7%99%D7%90%D7%95%D7%AA - Pneumonia>