

**דו"ח פרויקט סוף קורס מערכות לומדות ולמידה عمוקה
גילוי דלקת ריאות (פניאומוניה) מצילומי רנטגן**



מגישיים : רועי סביר - 318402484

עדו בן הרוש - 316439116

נור זכור - 312356868

מרצה : פרופ' אמיר אדלר

תאריך : 05/03/2025

תוכן עניינים

| | |
|----------|---|
| 3..... | מבוא : |
| 4..... | דרישות הפרויקט: |
| 5..... | משימות |
| 5..... | משימה 1 : תכנון רשתות נוירוניים לפענוח דלקת ריאות |
| 5..... | משימה 2 : אימון הרשות והערכת ביצועים |
| 6..... | משימה 3 : השוואת אלגוריתמי אימון ושיפור הביצועים |
| 6..... | משימה 4 : סיווג רב-קטgoriy (Multi-Class Classification) |
| 7..... | פתרונות |
| 7..... | רשת נוירונים קונבולוציונית (CNN) |
| 7..... | פתרון משימה 1 : |
| 10 | פתרון משימה 2 : |
| 11 | פתרון משימה 3 : |
| 32 | פתרון משימה 4 : |
| 42 | רשת Transfer Learning |
| 42 | פתרון משימה 1 : |
| 43 | פתרון משימה 2: |
| 46 | פתרון משימה 3: |

מבוא :

דלקת ריאות (Pneumonia) היא זיהום שפוגע בריאות, בעיקר בשקי קי' האויר הזרים האחראים על חילוף הגזים בגוף. כאשר מתפתח זיהום באזוריים אלה, יכולת הגוף להעביר חמצן לדם נפגעת, מה שעשול לגרום לקשי נשימה ולסיבוכים בריאותיים.

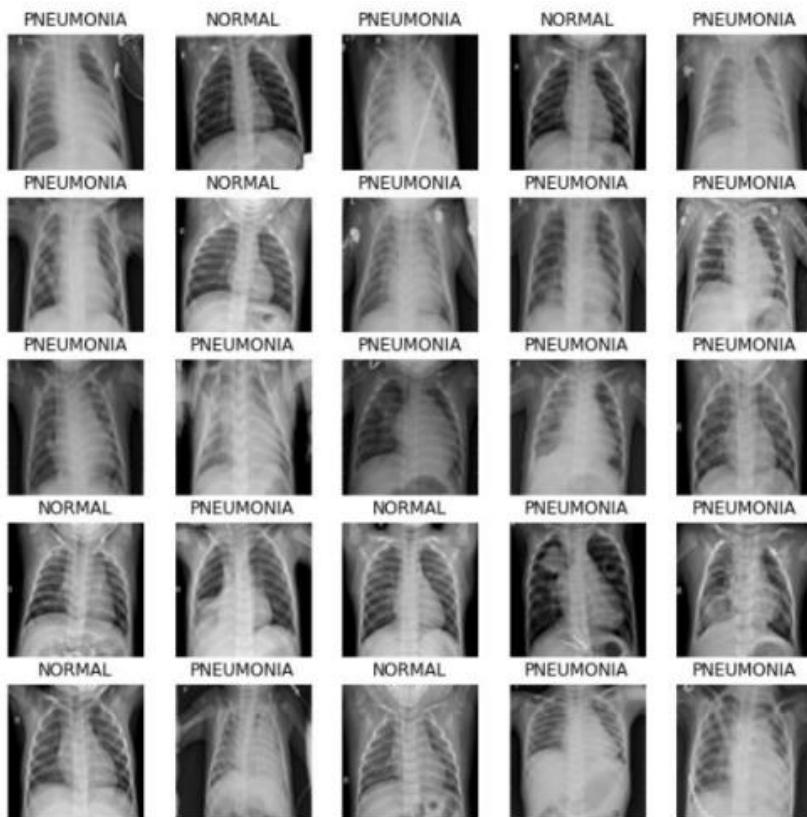
הגורמים העיקריים לדלקת ריאות הם חיידקים או וירוסים, אך לעיתים רחוקות, בעיקר בקרב אנשים עם מערכת חיסון מוחלשת, גם פטריות עלולות לגרום לה. מחלות נגיפיות, כמו שפעת, עשוות להוביל בהמשך להתרחשות דלקת ריאות ממוקור חיידק.

בין הסימנים השכיחים של המחלת ניתן למנות חום גבוה, צמרמות, שיעול, קושי בנשימה וכאבים באזור החזה. האבחון נעשה באמצעות צילום רנטגן של החזה ובדיקות דם.

בחורף ובמהלך התפרצויות של מחלות נਸימתיות כמו קורונה, ישנה עלייה משמעותית במספר החוליםים, מה שיזכר צריך בפתרון עיל ומחרה לפענוח צילומי רנטגן.

הפרויקט מציע מערכת מבוססת למידה عمוקה (Deep Learning) שמסוגלת לבצע פעולה אוטומטי של צילומי חזה בזמן אמת בדיקן של מעל 90%. פתרון זה נועד להקל על עבודות הרדיולוגים ולהפחית את העומס במערכות הבריאות.

המערכת מבוססת על אלפי צילומי רנטגן מותוך מאגר הכלל תמונות של מטופלים בריאים וחוליים. באמצעות רשת נירוניים מסווג (CNN) האלגוריתם לומד לזהות הבדלים בין תמונות תקיןות לכלה המצביעות על דלקת ריאות, ובכך מאפשר סיווג מהיר ומדויק של צילומים חדשים.



דרישות הפרויקט:

- בפרויקט זה נדרשנו לפתח ולשים פתרונות מבוססי רשות ניירונים מסוג CNN לizihi דלקת ריאות מתוך אלפי צילומי רנטגן אמתיים, אשר עברו אבחון על ידי רופאים מומחים בתחום. התמונות ילקחו מקור הנתונים בקשר הבא : [data](#).
- התמונות יסוגו לפי סוג דלקת הריאות השונות: דלקת ריאות חיידקית (Bacterial) ודלקת ריאות נגיפית (Viral).
- הערכת ביצועי הרשת תבוצע באמצעות סט בדיקה (Test Set) אשר יהיה נפרד לחילוץ מסט האימון (Train Set) כך שלא תהיה חפיפה בין הקבוצות.
- רשת אשר תציג ביצועים טובים ויחשב לכך עם שיעור שגיאה נמוך מ-7% (כלומר, דיוק העולה על 93%).
- סט הבדיקה יוכל לפחות 200 תמונות של מטופלים עם דלקת ריאות (100 מכל סוג) וכן 200 תמונות של מטופלים ללא דלקת ריאות.

משימות

משימה 1 : תכנון רשתות נוירוניים לפענוח דלקת ריאות

יש לתכנן שתי רשתות נוירוניים عمוקות (CNN) לצורך זיהוי מקרים חיוביים של דלקת ריאות מתוך תמונות רפואיות:

- רשת CNN המבוססת על Transfer Learning.
- רשת CNN הנבנית מאפס, ללא שימוש ב-Transfer Learning.

כל רשת תקבל תמונה כקלט ותنبي הסתברות (ערך רציף בתחום [0,1]) לכך שהתמונה מייצגת מקרה חיובי של דלקת ריאות, כאשר 1 מעיד על證ודאות מלאה של קיומ הדלקת.

משימה 2 : אימון הרשתות והערכת ביצועים

יש לבצע אימון לשתי הרשתות שתוכננו במשימה הראשונה, תוך פירוט הפרמטרים הבאים:

- אלגוריתם האימון
- מספר ה-Epochs
- ערך ה-Learning Rate
- גודל ה-Mini-Batch

עבור הרשת המבוססת על Transfer Learning, יש לבדוק שני אופני אימון:

- הקפאת פרמטרי רשת הבסיס – עדכון מתבצע רק בשכבות חדשות שהוספנו.
- עדכון חלק משכבות רשת הבסיס בנוסף לשכבות חדשות. – Fine-Tuning

לאחר האימון, יש להשוות את ביצועי שתי השיטות ולבחן איזו מהן השיגה את הדיקט הגבוה ביותר. בנוסף, עבור כל אחת מהרשתות, יש לשרטט גרף Precision-Recall, שבו כל נקודה תציג חישוב עבור סף קבלת החלטה שונה (בטווחו 0.1 עד 0.9, בקפיצות של 0.05). על אותו גרף, יש לסמן את נקודות ה-F-Score המוחשבות לכל צמד ערכים של Precision ו-Recall.

לבסוף, יש לקבוע את ערך הסף האופטימלי שבו מתקיים ערך ה-F-Score הגבוה ביותר.

משימה 3 : השוואת אלגוריתמי אימון ושיפור הביצועים

יש לבדוק את ביצועי הרשותות תוך שימוש באלגוריתמי האימון הבאים, תוך שינוי מספר ה-Epochs וה-Learning Rate לכל אלגוריתם:

- SGD (Stochastic Gradient Descent) •
- Momentum עם SGD •
- Adam •
- RMSprop •

לאחר מציאת האלגוריתם שסיפק את תוצאות הדיק הטובות ביותר, יש להפעיל מנגנון Early Stopping ולבוחן האם חל שיפור נוסף בביטויים. לכל אחד מאלגוריתמי האימון יש לשרטט גרף התוכנות שיציג את תהליך האימון לאורך הזמן, כולל:

- Train Loss & Validation Loss •
- Train Accuracy & Validation Accuracy •

סת הווידציה יכול לפחותות 50 תמונות חיוביות לדלקת ריאות (25 מכל סוג) ו-50 תמונות שליליות (ללא דלקת ריאות). סט זה ילקח מתוך סט האימון אך ישאר בלתי תלוי בסט הבדיקה הסופי.

משימה 4 : סיווג רב-קטgoriy (Multi-Class Classification)

יש להתאים את הרשות מהמשימה הראשונה (הרשות ללא Transfer Learning) כך שתוכל לבצע סיווג שלוש קטגוריות:

- אין דלקת ריאות •
- דלקת ריאות חידקית •
- דלקת ריאות נגיפית •

לאחר מכן, יש לבצע מחדש את משימה 3 עבור הרשות המותאמת ולבסוף להציג מטריצת בלבול (Confusion Matrix) עבור סט הבדיקה, תוך שימוש באלגוריתם שסיפק את התוצאות הטובות ביותר.

פתרונות

רשת ניורונית קונבולוציונית (CNN)

פתרון משימה 1:

את הרשת המוצגת קיבלנו לאחר מספר ניסיונות שביצענו כך שבכל פעם שינוינו או הוספנו שכבות נוספות. על מנת לבנות מבנה בסיסי שעה על דרישותינו השתמשנו במעטבות שביצענו במהלך הקורס, בנוסף לשימוש באינטרנט.

פירוט המודל :

| פרמטרים ¹ | | | | שם השכבה |
|----------------------|-------|----------|--------------|----------|
| פיזטרים | גרעין | אקטיבציה | פיזטרים | |
| 32 | (3,3) | Relu | Conv2D | |
| | (2,2) | | MaxPooling2D | |
| 64 | (3,3) | Relu | Conv2D | |
| | (2,2) | | MaxPooling2D | |
| 64 | (3,3) | Relu | Conv2D | |
| | 0.1 | | Dropout | |
| | (2,2) | | MaxPooling2D | |
| 64 | (3,3) | Relu | Conv2D | |
| | (2,2) | | MaxPooling2D | |
| 128 | (3,3) | Relu | Conv2D | |
| | 0.2 | | Dropout | |
| | (2,2) | | MaxPooling2D | |
| 256 | (3,3) | Relu | Conv2D | |
| | 0.3 | | Dropout | |
| | (2,2) | | MaxPooling2D | |
| | | | Flatten | |
| | 0.3 | | Dropout | |
| | | Sigmoid | Dense | |

- מידי הגרעין עבר כל שכבות הקונוונצייה הינן: (3,3).

¹ בכלל השכבות Conv2D ו MaxPooling2D נעשה שימוש $\text{padding} = \text{same}$ וב- 2×2 strides = 1.

דוח הסיכום לאחר הרצת הרשות :

| Layer (type) | Output Shape | Param # |
|--------------------------------|----------------------|---------|
| conv2d (Conv2D) | (None, 180, 180, 32) | 320 |
| max_pooling2d (MaxPooling2D) | (None, 90, 90, 32) | 0 |
| conv2d_1 (Conv2D) | (None, 90, 90, 64) | 18,496 |
| max_pooling2d_1 (MaxPooling2D) | (None, 45, 45, 64) | 0 |
| conv2d_2 (Conv2D) | (None, 45, 45, 64) | 36,928 |
| dropout (Dropout) | (None, 45, 45, 64) | 0 |
| max_pooling2d_2 (MaxPooling2D) | (None, 23, 23, 64) | 0 |
| conv2d_3 (Conv2D) | (None, 23, 23, 64) | 36,928 |
| max_pooling2d_3 (MaxPooling2D) | (None, 12, 12, 64) | 0 |
| conv2d_4 (Conv2D) | (None, 12, 12, 128) | 73,856 |
| dropout_1 (Dropout) | (None, 12, 12, 128) | 0 |
| max_pooling2d_4 (MaxPooling2D) | (None, 6, 6, 128) | 0 |
| conv2d_5 (Conv2D) | (None, 6, 6, 256) | 295,168 |
| dropout_2 (Dropout) | (None, 6, 6, 256) | 0 |
| max_pooling2d_5 (MaxPooling2D) | (None, 3, 3, 256) | 0 |
| flatten (Flatten) | (None, 2304) | 0 |
| dropout_3 (Dropout) | (None, 2304) | 0 |
| dense (Dense) | (None, 1) | 2,305 |

צלום הקוד לבניית הרשת

```
# Build the model
CNN = models.Sequential()

CNN.add(layers.Conv2D(32, (3, 3), padding = 'same' ,activation='relu', input_shape=(180, 180, 1)))
CNN.add(layers.MaxPooling2D((2,2) , strides = 2 , padding = 'same'))
CNN.add(layers.Conv2D(64 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))

CNN.add(layers.MaxPooling2D((2,2) , strides = 2 , padding = 'same'))
CNN.add(layers.Conv2D(64 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
CNN.add(Dropout(0.1))

CNN.add(layers.MaxPooling2D((2,2) , strides = 2 , padding = 'same'))
CNN.add(layers.Conv2D(128 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
CNN.add(Dropout(0.2))

CNN.add(layers.MaxPooling2D((2,2) , strides = 2 , padding = 'same'))
CNN.add(layers.Conv2D(256 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
CNN.add(Dropout(0.3))

CNN.add(layers.MaxPooling2D((2,2) , strides = 2 , padding = 'same'))

CNN.add(layers.Flatten())
CNN.add(Dropout(0.3))
CNN.add(layers.Dense(1, activation = 'sigmoid'))

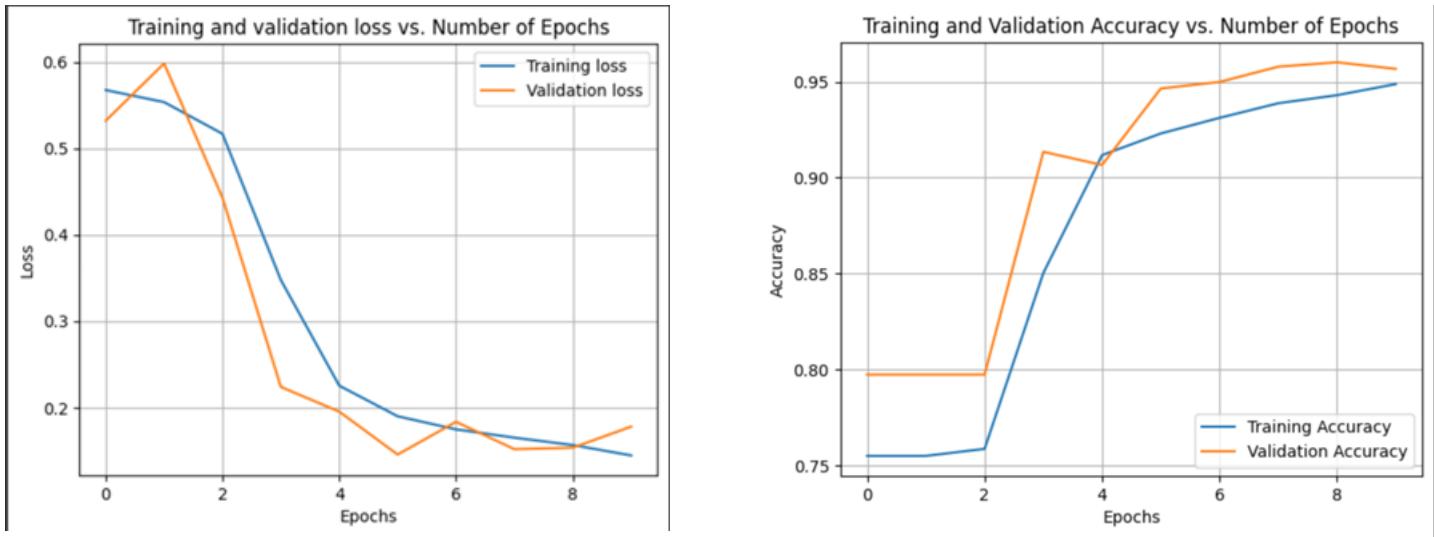
# Compile the model
CNN.compile(optimizer = Adagrad(learning_rate=0.01),loss='binary_crossentropy',metrics=['accuracy'])

CNN.summary()

history = CNN.fit(train_norm, train_label, epochs = 10, batch_size = 20,validation_data=(val_norm, val_label), verbose=1)
```

פתרון משימה 2 :

על מנת להגיע לתוצאות מספקות של רשת CNN השתמשנו ב - EPOCHS = 10.
להלן התוצאות שקיבלנו לאחר הרצת המודל :

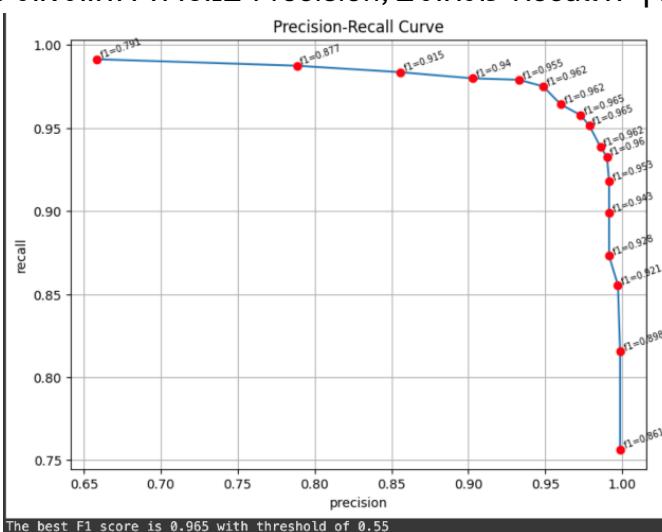


Test accuracy: 0.9402730464935303

Test loss: 0.18859414756298065

ניתן לראות כי קיימת התאמה יחסית גבוהה בין גרפּ האימון לגרפּ הווילדייזה, הן במדד הדיווק והן במדד הקנס. בעת הפעלת המודל על סט הבדיקה, התקבל דיווק של 94%, דבר המצביע על כך שהמודל מציג ביצועים טובים ועומד בדרישות הפרויקט, אשר קבעו רף דיווק של מעל 93% בסט הבדיקה.

כעת, נציג את גרפּ ה Recall-Precision -כטלוֹת ב, במטרה לזהות את ערך ה F1 -



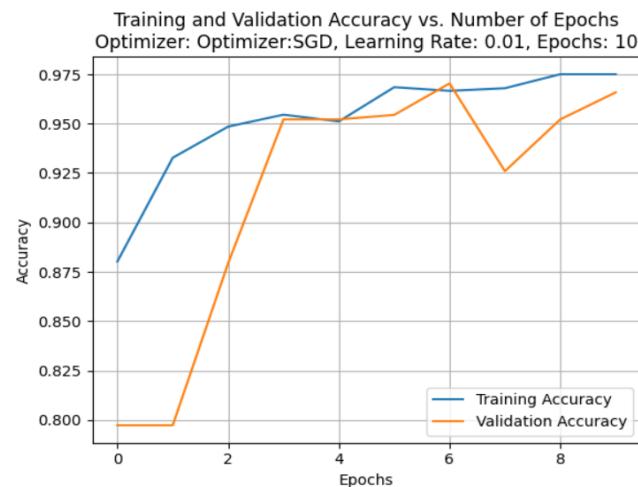
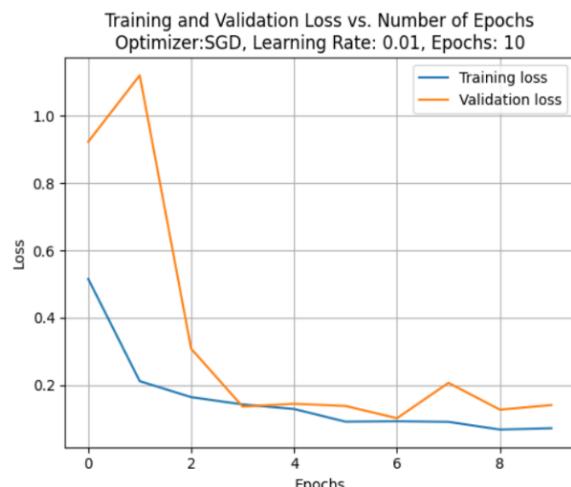
ניתן לראות כי עבור סך הסתברות של 0.55 (כלומר, אם הפלט של הרשת גבוה מערך זה, התמונה תסwoג כחולה), מתקיים יחס של 0.965.

פתרון משימה 3

כלל האלגוריתמים הורצו על Batch-size של 20 חלקות.

: SGD

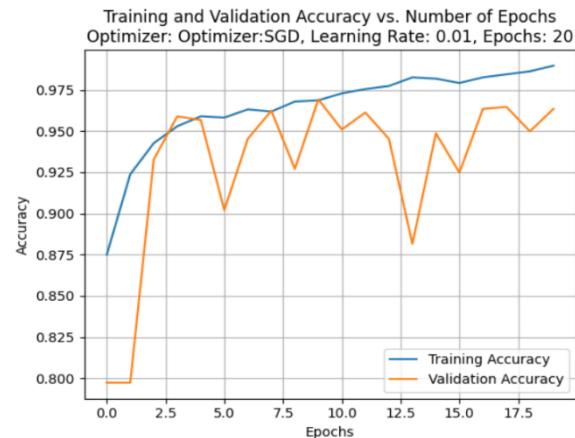
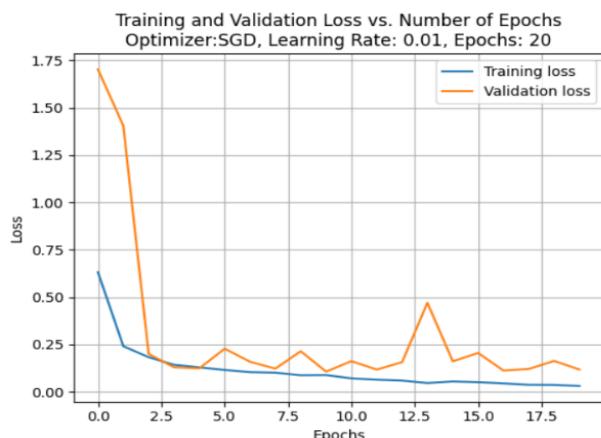
: LR=0.01, EPOCHS=10



Test accuracy: 0.9445392489433289

Test loss: 0.17095471918582916

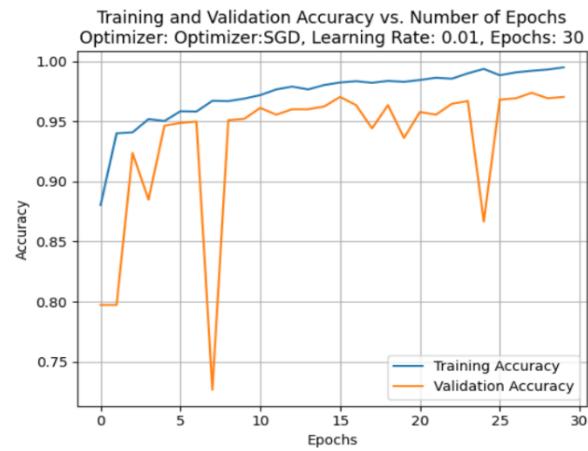
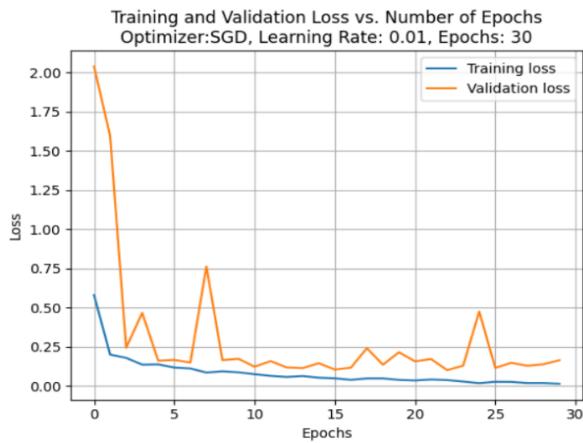
: LR=0.01, EPOCHS=20



Test accuracy: 0.947098970413208

Test loss: 0.16901102662086487

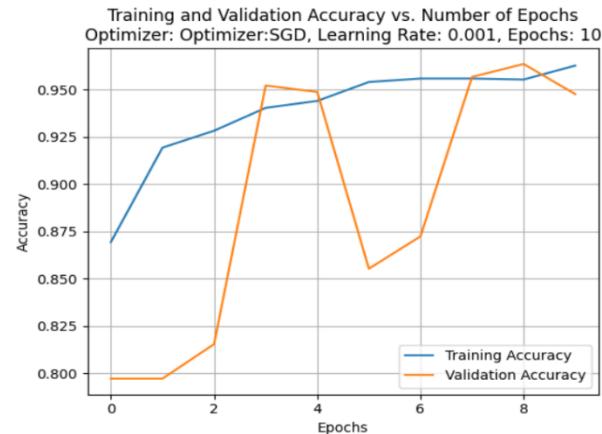
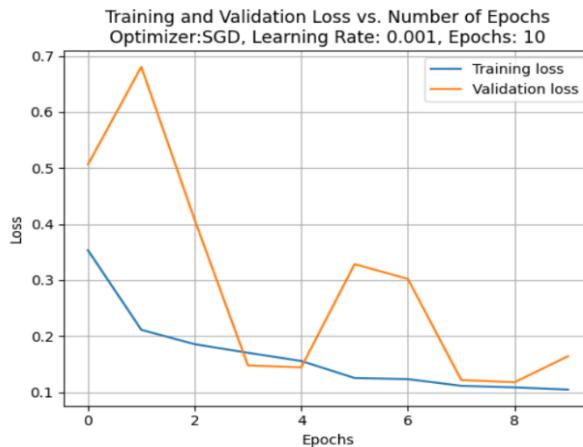
: LR=0.01, EPOCHS=30



Test accuracy: 0.9505119323730469

Test loss: 0.2589838206768036

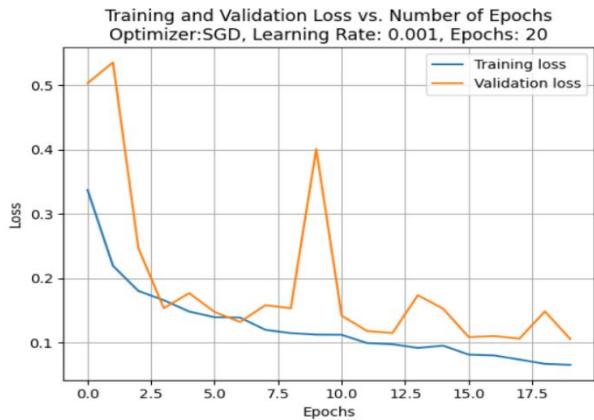
: LR=0.001, EPOCHS=10



Test accuracy: 0.94112628698349

Test loss: 0.16301468014717102

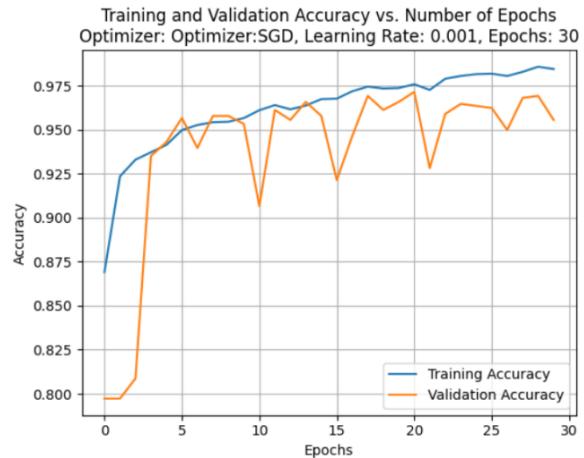
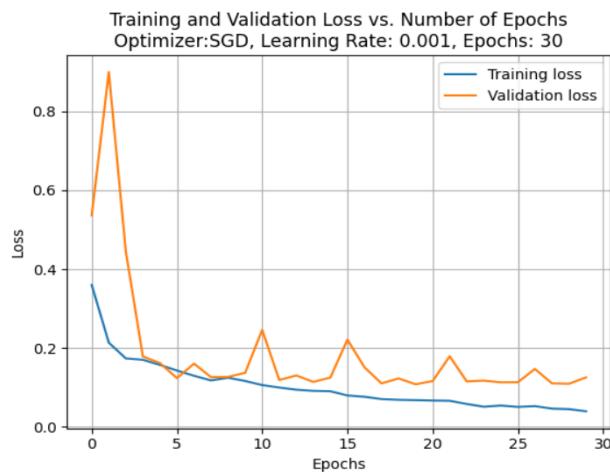
:LR=0.001, EPOCHS=20



Test accuracy: 0.9556313753128052

Test loss: 0.13209833204746246

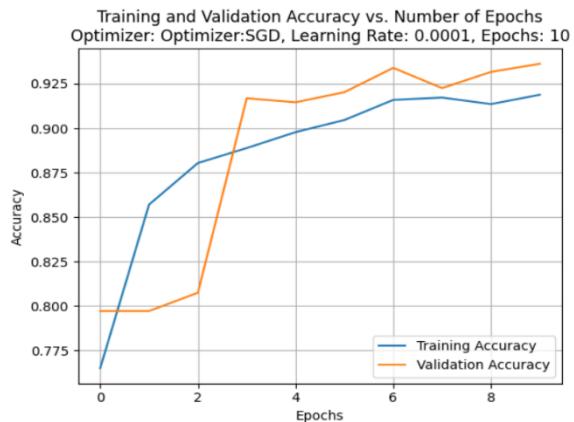
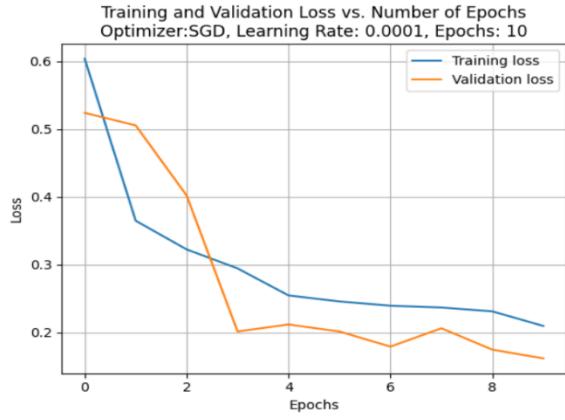
: LR=0.001, EPOCHS=30



Test accuracy: 0.9445392489433289

Test loss: 0.1781369000673294

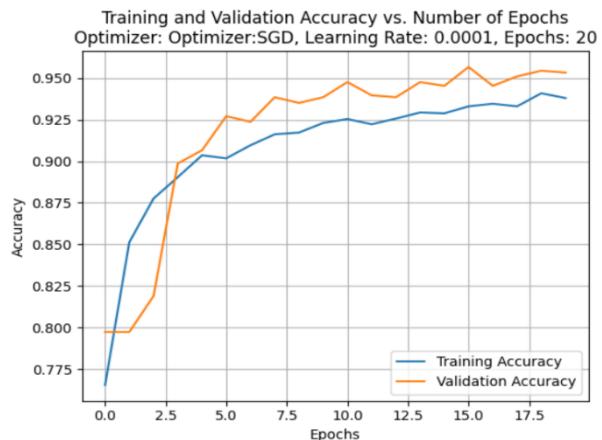
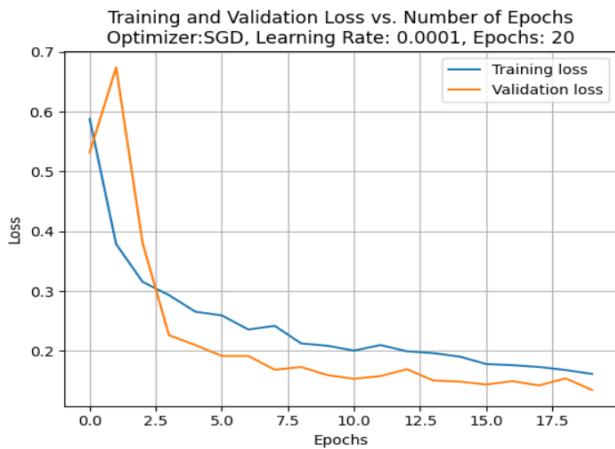
:LR=0.0001, EPOCHS=10



Test accuracy: 0.9274743795394897

Test loss: 0.19265210628509521

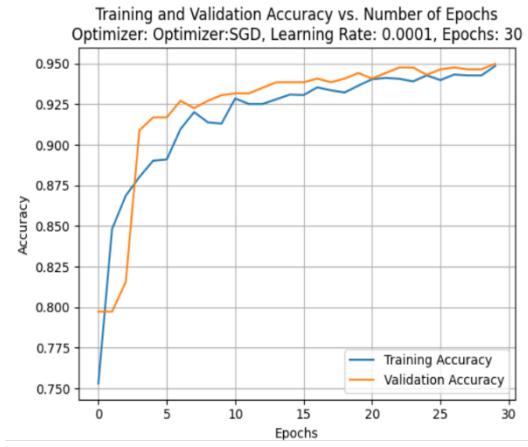
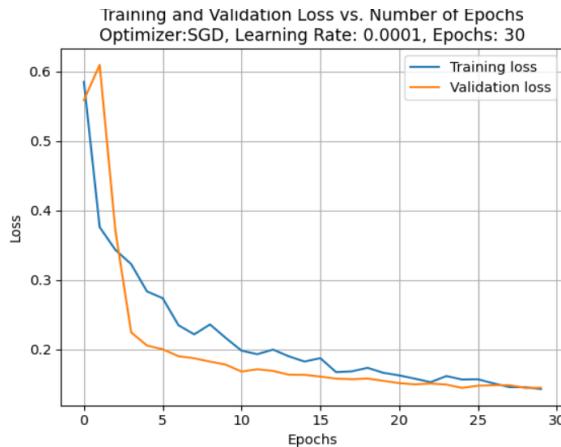
:LR=0.0001, EPOCHS=20



Test accuracy: 0.9377133250236511

Test loss: 0.18168789148330688

:LR=0.0001, EPOCHS=30

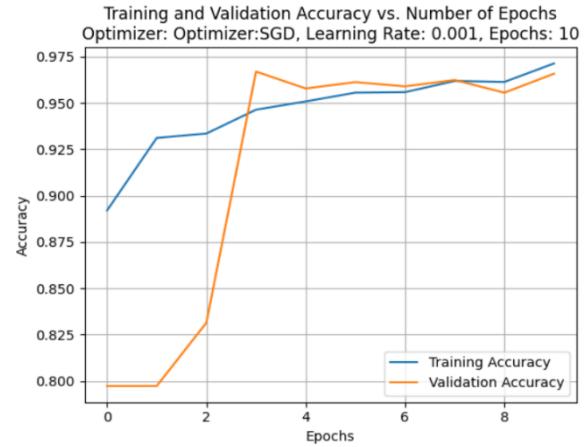
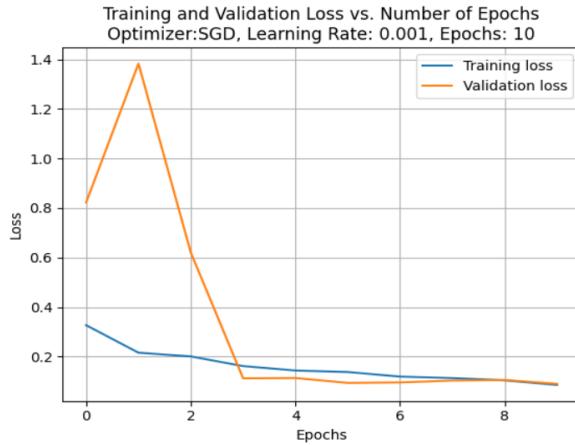


Test accuracy: 0.9402730464935303

Test loss: 0.16169407963752747

: SGD, With Momentum = 0.9

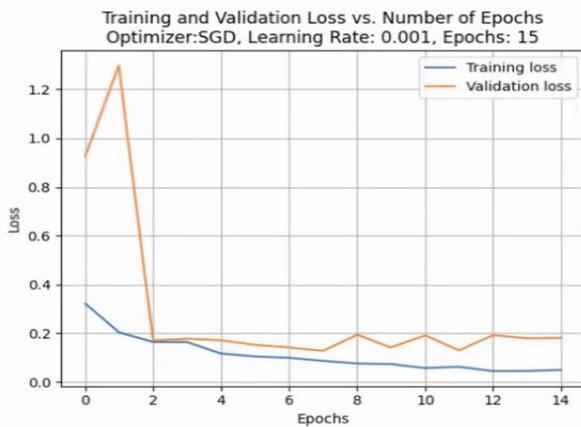
: LR=0.001,EPOCHS=10



Test accuracy: 0.953071653842926

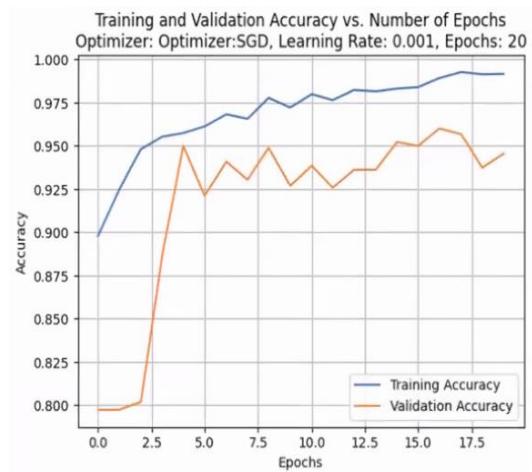
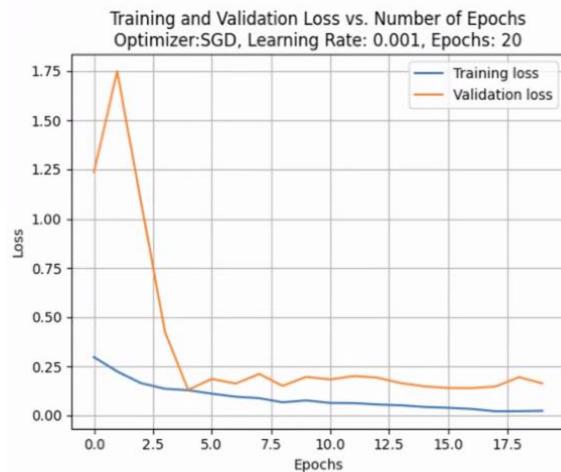
Test loss: 0.12411794811487198

: LR=0.001,EPOCHS=15



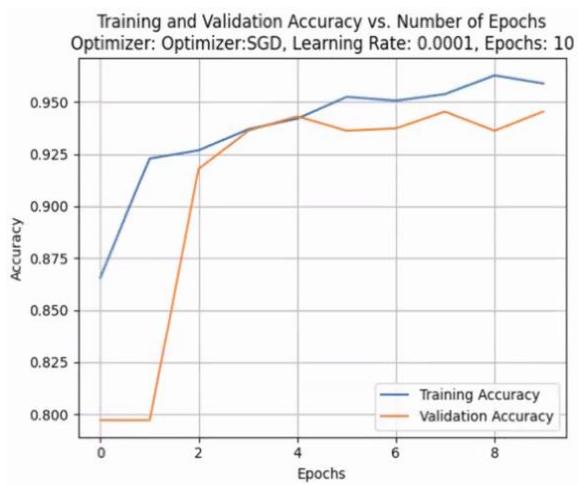
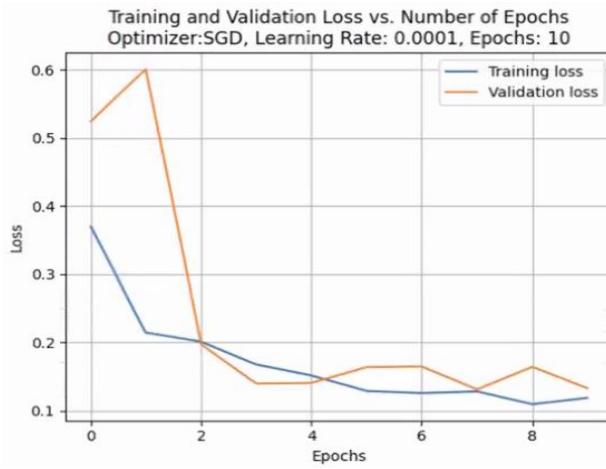
Test accuracy: 0.9505119323730469
Test loss: 0.14171287417411804

: LR=0.001,EPOCHS=20



Test accuracy: 0.9539248943328857
Test loss: 0.12988801300525665

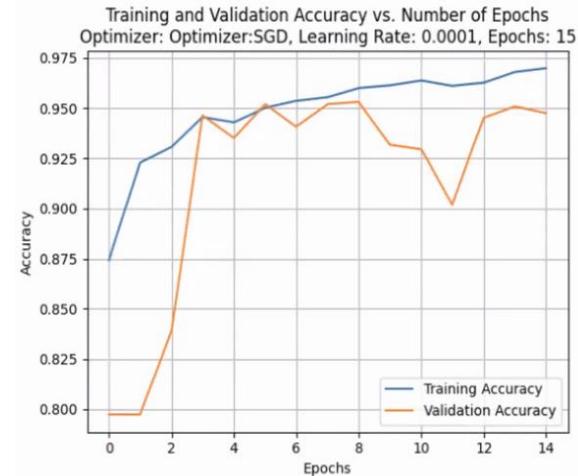
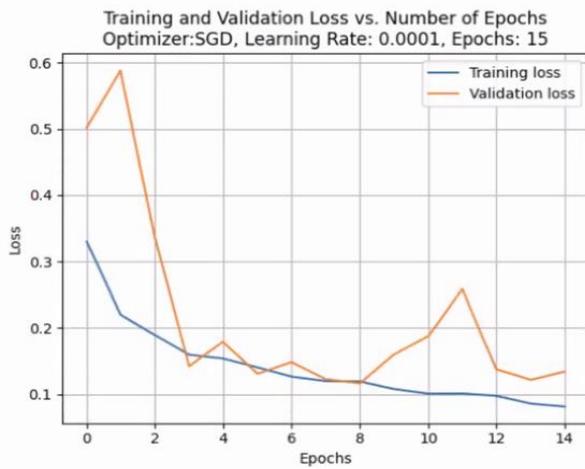
: LR=0.0001,EPOCHS=10



Test accuracy: 0.9215016961097717

Test loss: 0.18962462246418

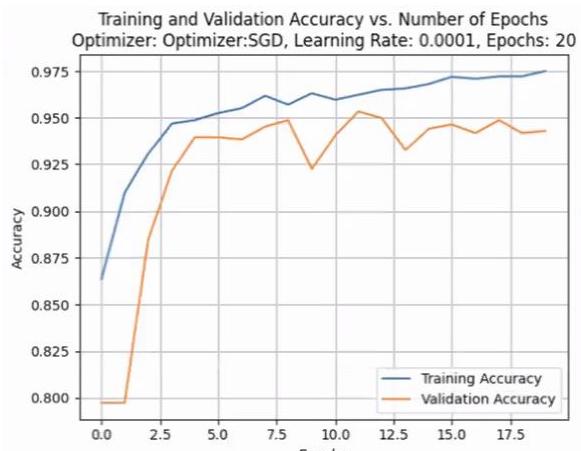
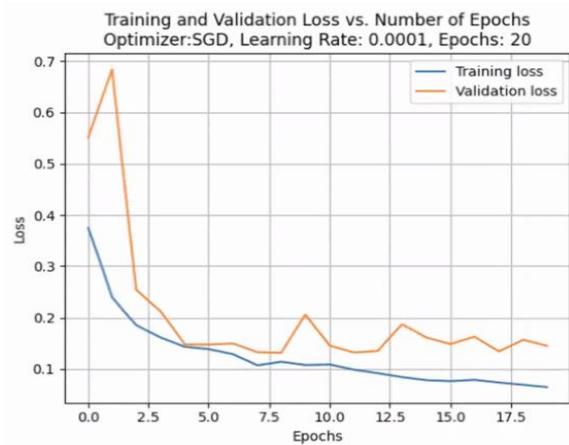
: LR=0.0001,EPOCHS=15



Test accuracy: 0.9488054513931274

Test loss: 0.13189218938350677

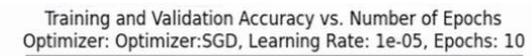
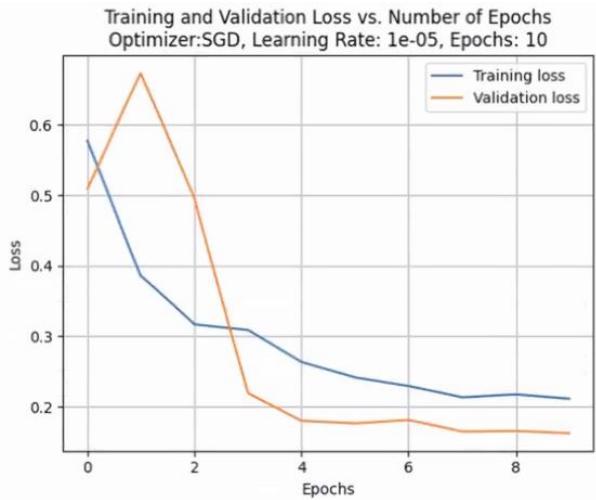
: LR=0.0001,EPOCHS=20



Test accuracy: 0.9453924894332886

Test loss: 0.14088751375675201

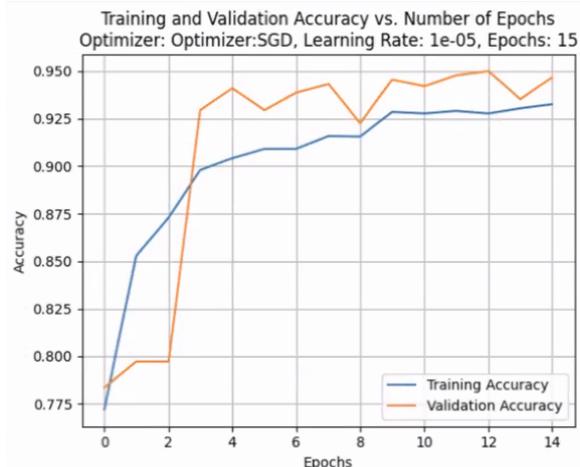
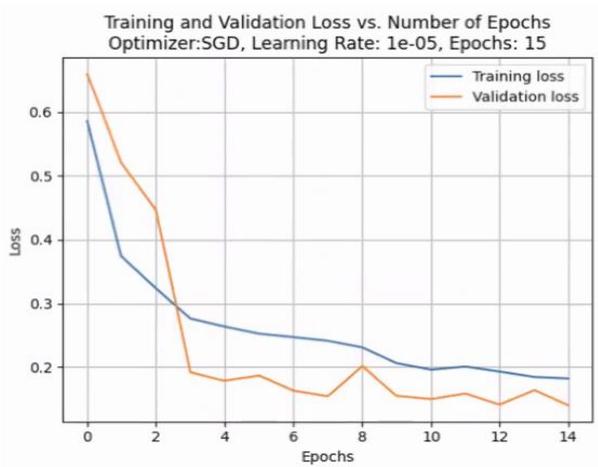
: LR=0.00001,EPOCHS=10



Test accuracy: 0.921501696109771

Test loss: 0.22580046951770782

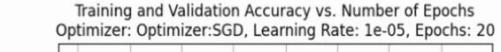
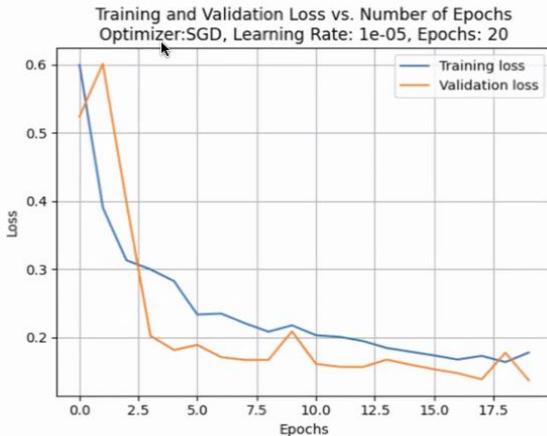
: LR=0.00001,EPOCHS=15



Test accuracy: 0.9300341010093689

Test loss: 0.19203270971775055

: LR=0.00001,EPOCHS=20

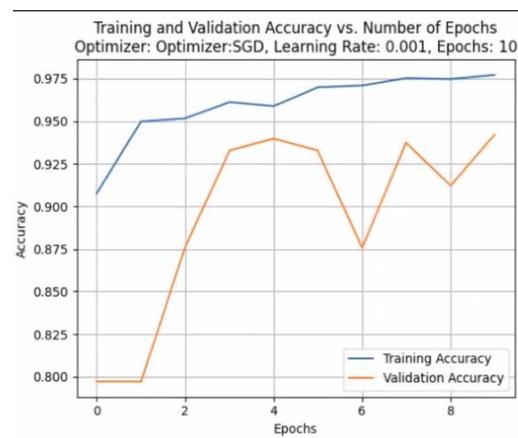
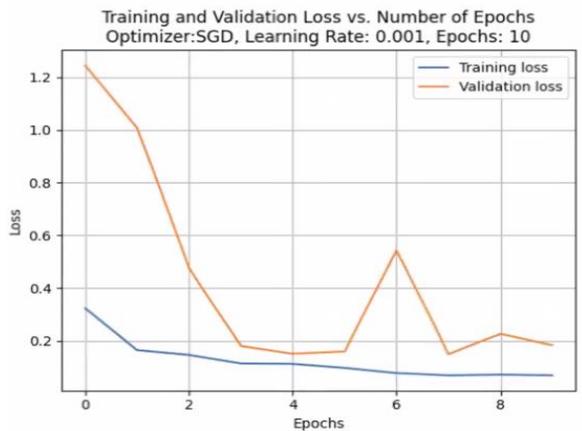


Test accuracy: 0.9308874011039734

Test loss: 0.18593551218509674

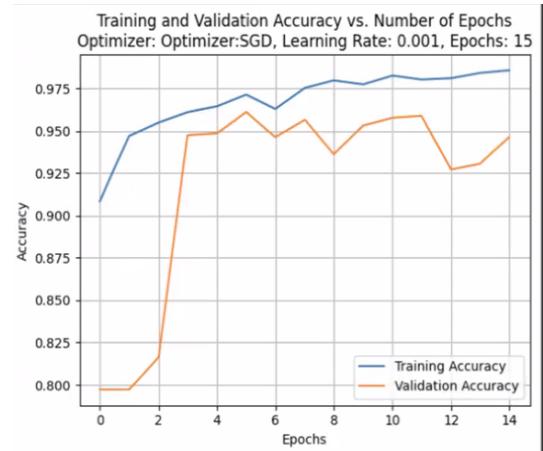
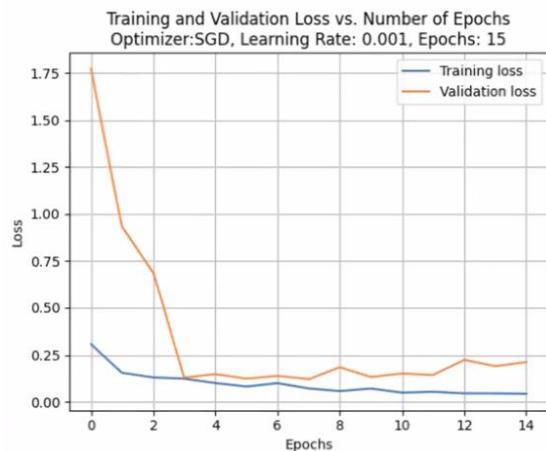
:ADAM

: LR=0.001,EPOCHS=10



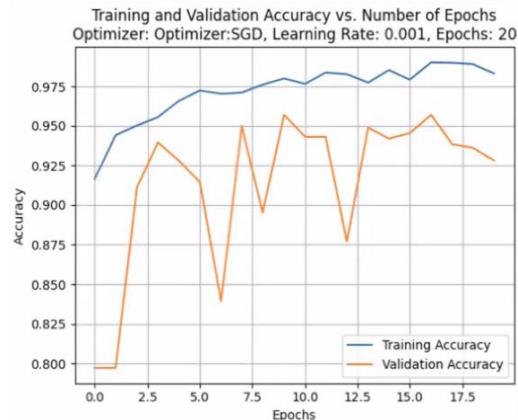
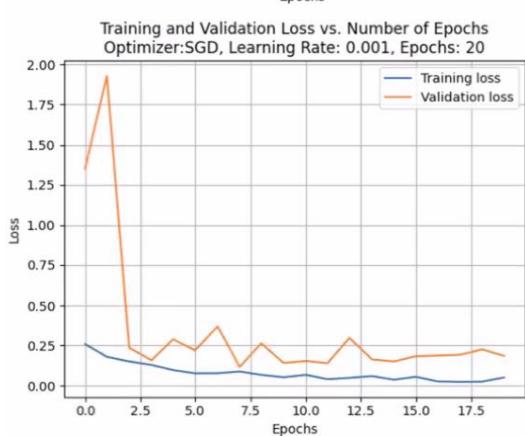
Test accuracy: 0.9172354936599731
 Test loss: 0.2662166953086853

: LR=0.001,EPOCHS=15



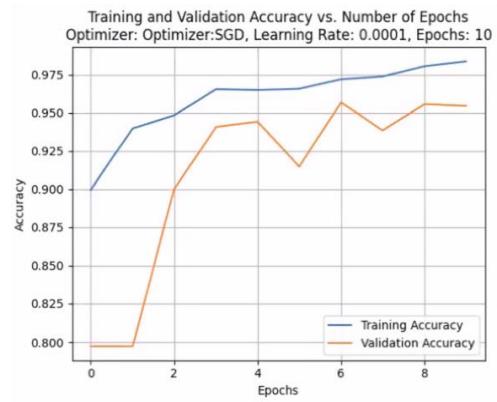
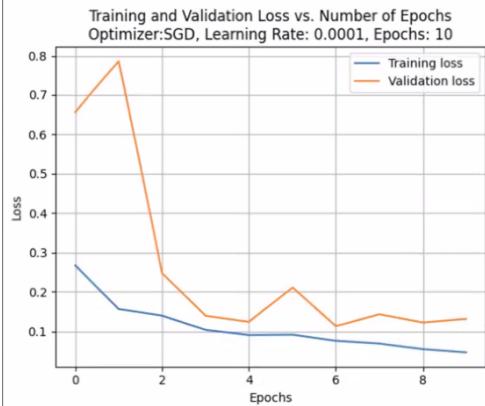
Test accuracy: 0.9129692912101746
 Test loss: 0.4013814628124237

: LR=0.001, EPOCHS=20



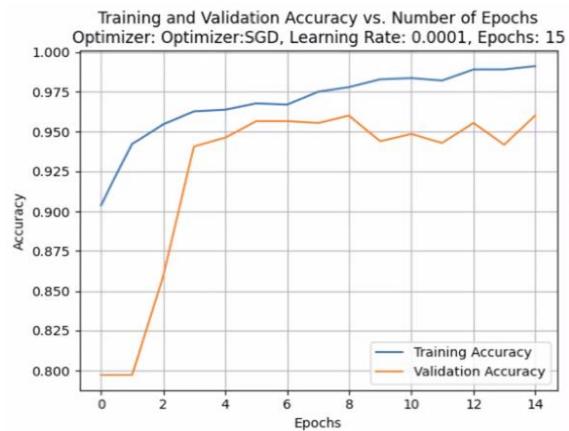
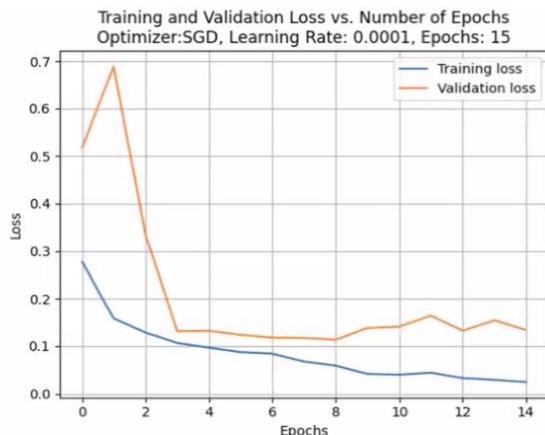
Test accuracy: 0.9249146580696106
Test loss: 0.1804673969745636

: LR=0.0001,EPOCHS=10



Test accuracy: 0.9249146580696106
Test loss: 0.2020854651927948

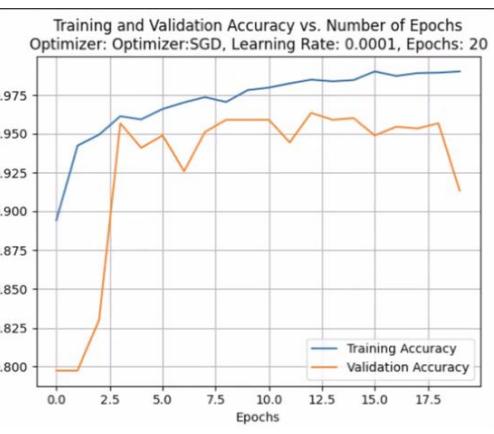
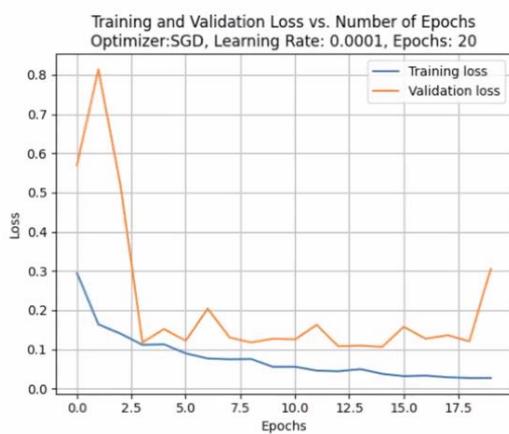
: LR=0.0001, EPOCHS=15



Test accuracy: 0.92662113904953

Test loss: 0.23440364003181458

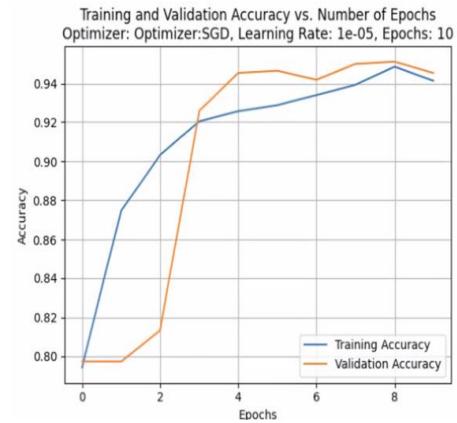
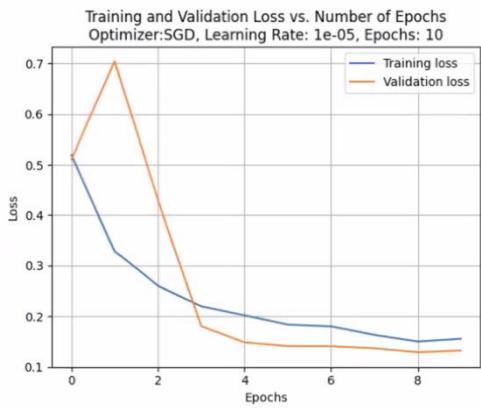
: LR=0.0001, EPOCHS=20



Test accuracy: 0.9334471225738525

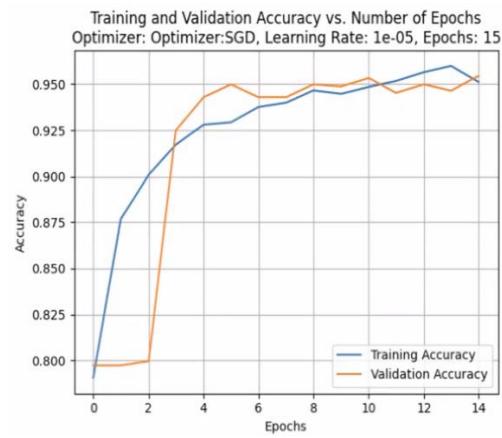
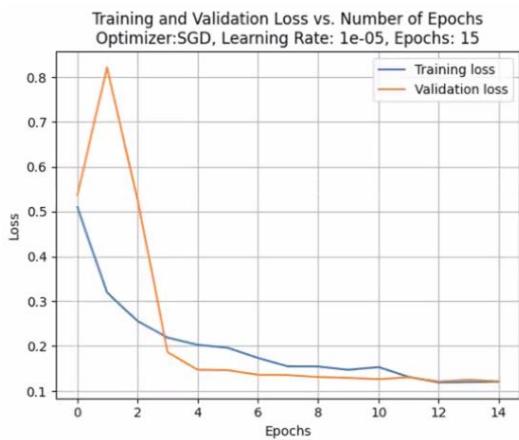
Test loss: 0.2188994586467743

: LR=0.00001, EPOCHS=10



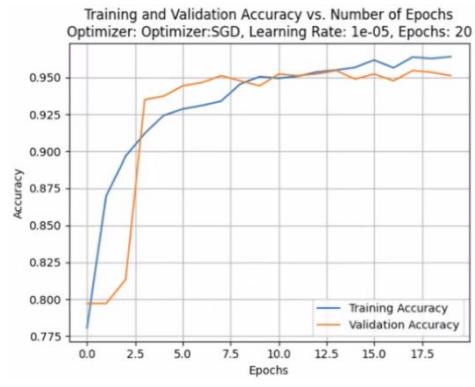
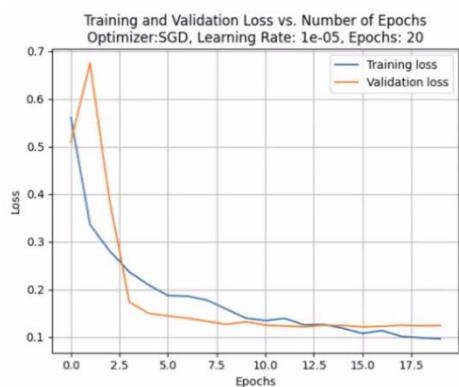
Test accuracy: 0.9325938820838928
Test loss: 0.16223299503326416

: LR=0.00001, EPOCHS=15



Test accuracy: 0.9419795274734497
Test loss: 0.15982593595981598

: LR=0.00001,EPOCHS=20

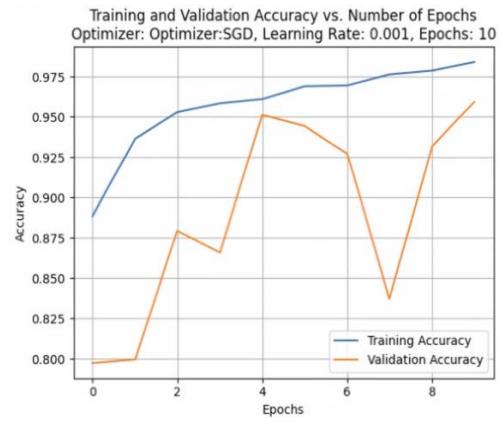
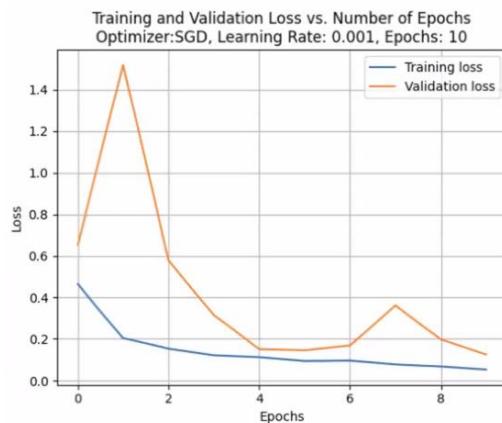


Test accuracy: 0.9428327679634094

Test loss: 0.14873535931110382

:RMSPROP

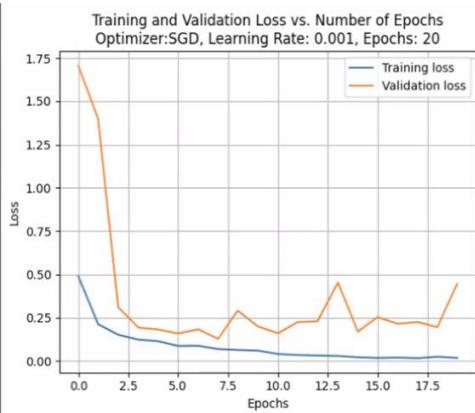
: LR=0.001,EPOCHS=10



Test accuracy: 0.9385665655136108

Test loss: 0.18515659868717194

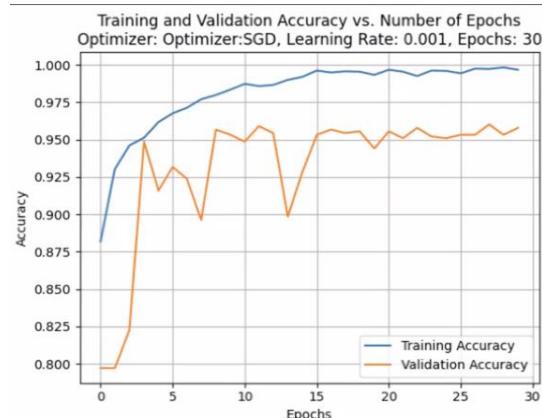
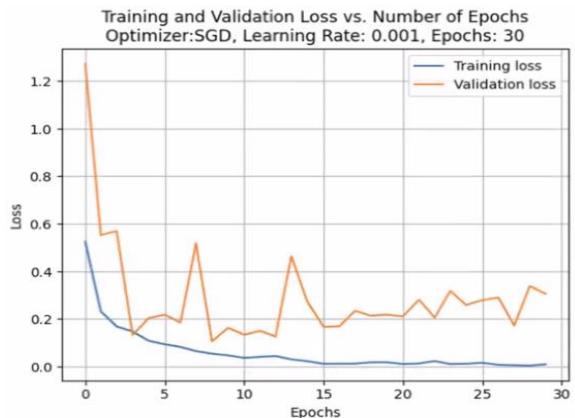
: LR=0.001,EPOCHS=20



Test accuracy: 0.8131399154663086

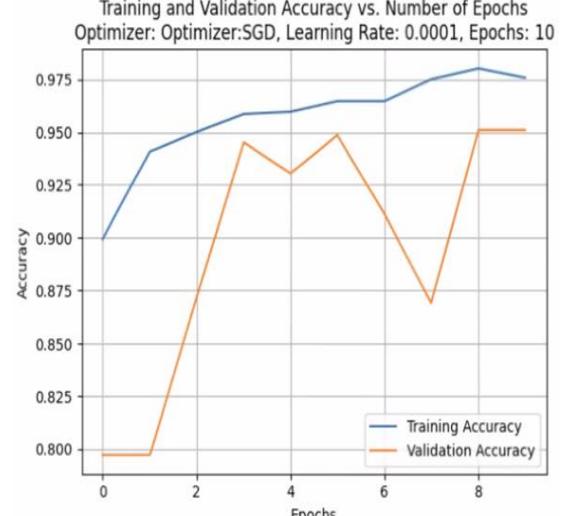
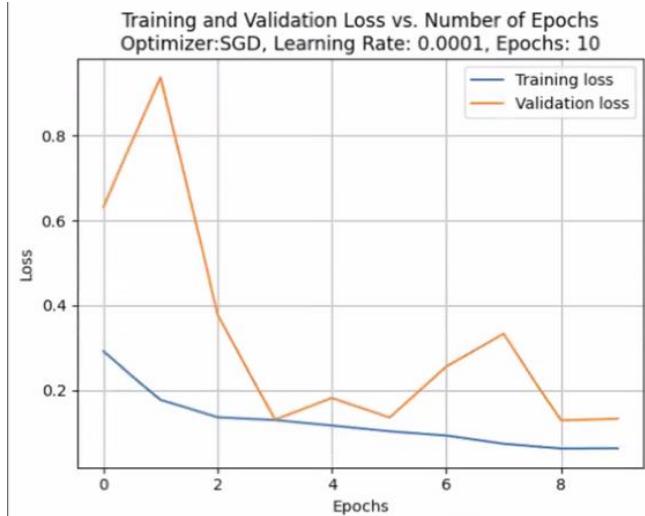
Test loss: 0.9475091695785522

: LR=0.001,EPOCHS=30



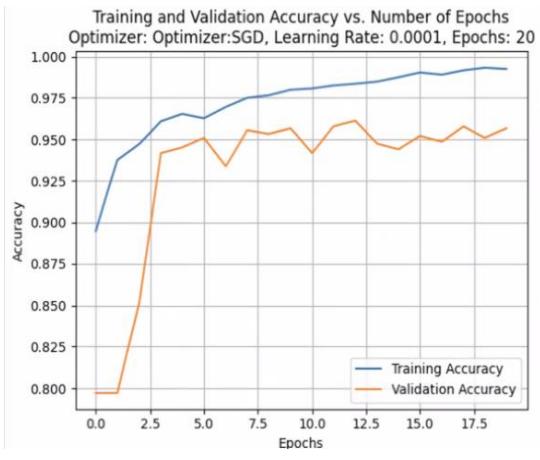
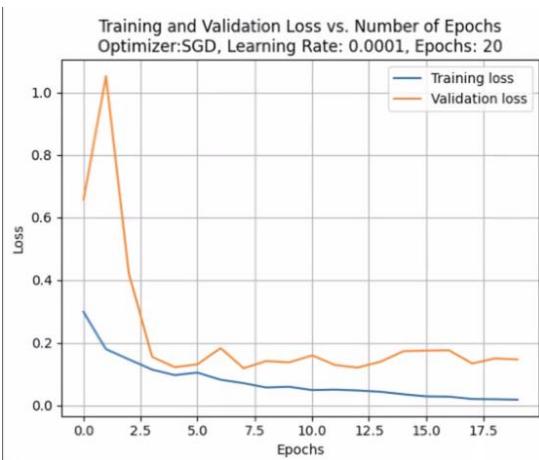
Test accuracy: 0.9496586918830872
Test loss: 0.3337664306163788

: LR=0.0001,EPOCHS=10



Test accuracy: 0.9436860084533691
Test loss: 0.16502892971038818

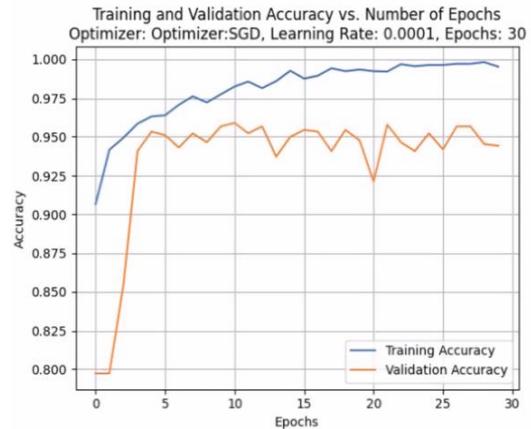
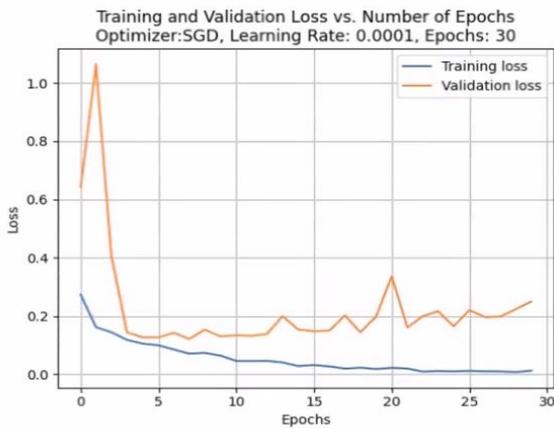
: LR=0.0001,EPOCHS=20



Test accuracy: 0.9522184133529663

Test loss: 0.19964164449546814

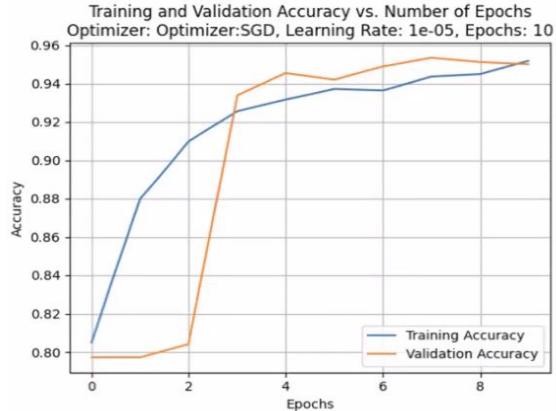
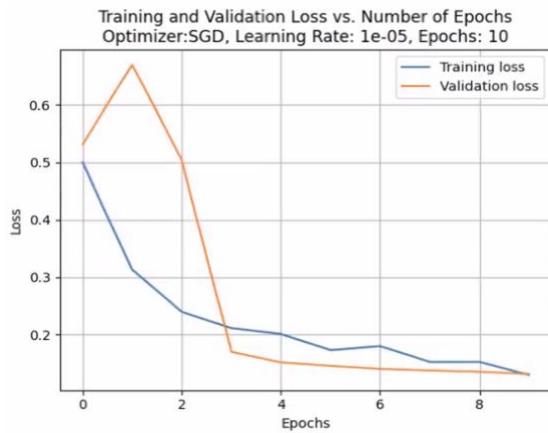
: LR=0.0001,EPOCHS=30



Test accuracy: 0.9462457299232483

Test loss: 0.17955610156059265

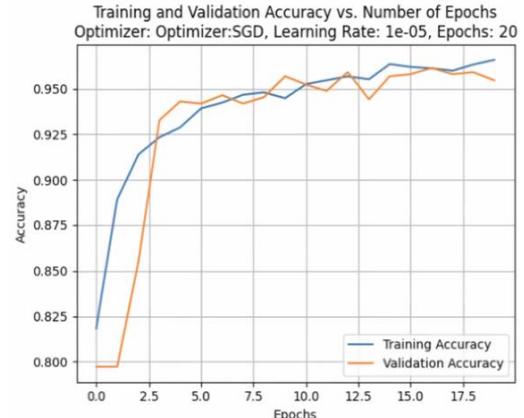
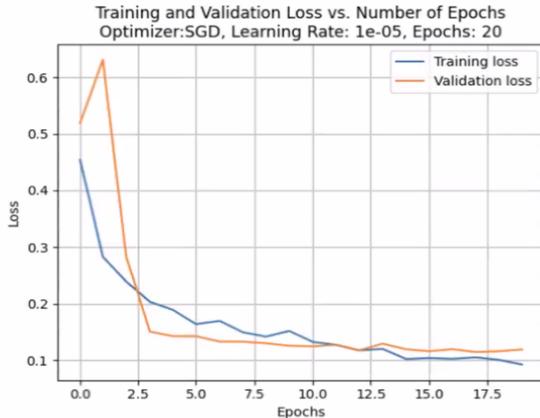
: LR=0.00001,EPOCHS=10



Test accuracy: 0.9402730464935303

Test loss: 0.16770856082439423

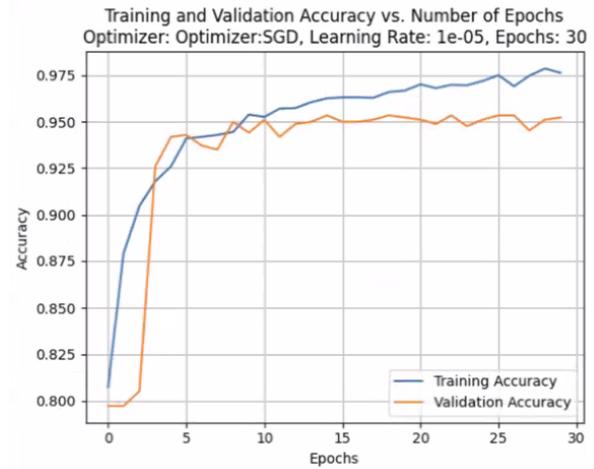
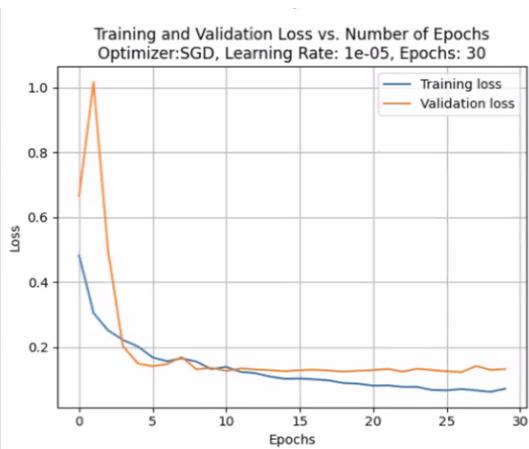
: LR=0.00001,EPOCHS=20



Test accuracy: 0.935153603553772

Test loss: 0.17534612119197845

: LR=0.00001,EPOCHS=30



Test accuracy: 0.9436860084533691

Test loss: 0.14121074974536896

סיכום התוצאות:

| SGD | | | |
|---------------|--------|--------|--------|
| Learning Rate | 0.01 | 0.01 | 0.0001 |
| Epochs | 10 | 20 | 30 |
| | 09445 | 0.9411 | 0.9274 |
| | 0.947 | 0.9526 | 0.9377 |
| | 0.9505 | 0.9445 | 0.9402 |

| SGD (With Momentum = 0.9) | | | |
|---------------------------|--------|---------|---------|
| Learning Rate | 0.001 | 0.0001 | 0.00001 |
| Epochs | 10 | 15 | 20 |
| | 0.9530 | 0.9215 | 0.9215 |
| | 0.9505 | 0.9488 | 0.93 |
| | 0.9539 | 0.94539 | 0.9308 |

| RMSPROP | | | |
|---------|--------|--------|---------|
| LR | 0.001 | 0.0001 | 0.00001 |
| EPOCHS | 10 | 20 | 30 |
| | 0.9385 | 0.9436 | 0.9402 |
| | 0.8131 | 0.9522 | 0.9351 |
| | 0.9496 | 0.9462 | 0.9436 |

| Adam | | | |
|---------------|--------|--------|---------|
| Learning Rate | 0.001 | 0.0001 | 0.00001 |
| Epochs | 10 | 15 | 20 |
| | 0.9172 | 0.9249 | 0.9325 |
| | 9129.0 | 0.9266 | 0.9419 |
| | 0.9249 | 0.9334 | 0.9428 |

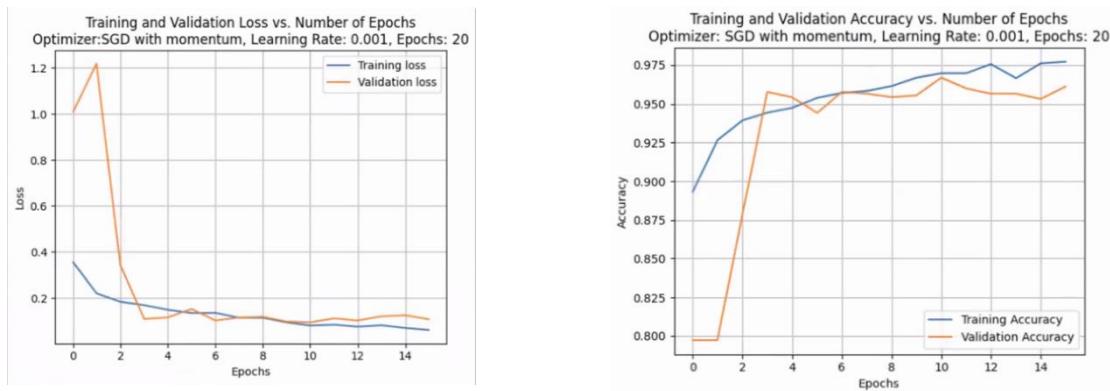
- התוצאה הגדולה ביותר על סט האימון התקבלה באלגוריתם SGD עם מומנטום של 0.9 עבור 20 EPOCHS = 0.9522
- נבחן את טיב התוצאות שקיבלנו עבור אלגוריתם זה עם מנגנון עצירה מוקדם – :Early Stopping

הารגומנטים שהכנסנו:

```
monitor='val_loss', patience=5, mode='min', restore_best_weights=True, verbose=1
```

```
Non-trainable params: 1,088 (4.25 KB)
Epoch 1/20
191/191      20s 62ms/step - accuracy: 0.8550 - loss: 0.4634 - val_accuracy: 0.7973 - val_loss: 1.0117
Epoch 2/20
191/191      7s 23ms/step - accuracy: 0.9313 - loss: 0.1947 - val_accuracy: 0.7973 - val_loss: 1.2192
Epoch 3/20
191/191      5s 24ms/step - accuracy: 0.9314 - loss: 0.2082 - val_accuracy: 0.8793 - val_loss: 0.3418
Epoch 4/20
191/191      4s 23ms/step - accuracy: 0.9431 - loss: 0.1709 - val_accuracy: 0.9579 - val_loss: 0.1082
Epoch 5/20
191/191      4s 23ms/step - accuracy: 0.9444 - loss: 0.1555 - val_accuracy: 0.9544 - val_loss: 0.1153
Epoch 6/20
191/191      5s 23ms/step - accuracy: 0.9584 - loss: 0.1227 - val_accuracy: 0.9442 - val_loss: 0.1526
Epoch 7/20
191/191      5s 23ms/step - accuracy: 0.9617 - loss: 0.1279 - val_accuracy: 0.9579 - val_loss: 0.1016
Epoch 8/20
191/191      5s 23ms/step - accuracy: 0.9507 - loss: 0.1277 - val_accuracy: 0.9567 - val_loss: 0.1149
Epoch 9/20
191/191      5s 23ms/step - accuracy: 0.9696 - loss: 0.0913 - val_accuracy: 0.9544 - val_loss: 0.1177
Epoch 10/20
191/191      5s 23ms/step - accuracy: 0.9640 - loss: 0.1025 - val_accuracy: 0.9556 - val_loss: 0.0969
Epoch 11/20
191/191      5s 23ms/step - accuracy: 0.9715 - loss: 0.0791 - val_accuracy: 0.9670 - val_loss: 0.0937
Epoch 12/20
191/191      5s 22ms/step - accuracy: 0.9735 - loss: 0.0725 - val_accuracy: 0.9601 - val_loss: 0.1110
Epoch 13/20
191/191      5s 23ms/step - accuracy: 0.9778 - loss: 0.0693 - val_accuracy: 0.9567 - val_loss: 0.1015
Epoch 14/20
191/191      4s 23ms/step - accuracy: 0.9653 - loss: 0.0848 - val_accuracy: 0.9567 - val_loss: 0.1202
Epoch 15/20
191/191      5s 23ms/step - accuracy: 0.9746 - loss: 0.0698 - val_accuracy: 0.9533 - val_loss: 0.1248
Epoch 16/20
191/191      5s 24ms/step - accuracy: 0.9766 - loss: 0.0658 - val_accuracy: 0.9613 - val_loss: 0.1069
Epoch 16: early stopping
Restoring model weights from the end of the best epoch: 11.
37/37      1s 9ms/step - accuracy: 0.9695 - loss: 0.0840
```

- ניתן לראות שמעבר לאפסוק 16 התקיים מנגנון העצירה המוקדמת.



Test accuracy: 0.9505119323730469

Test loss: 0.14201396703720093

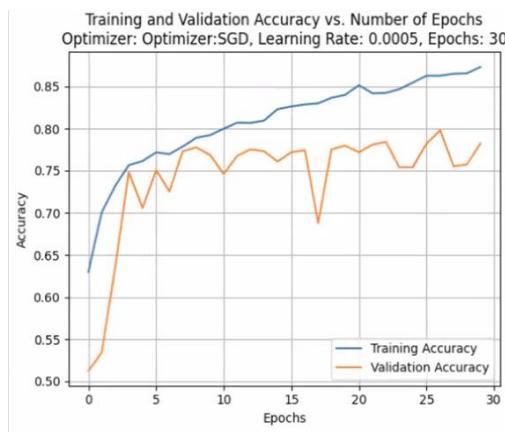
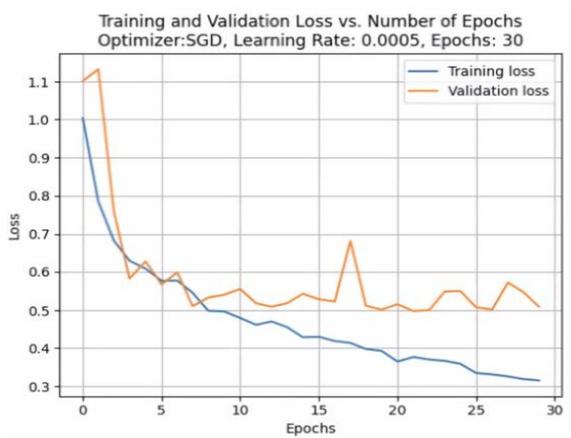
- ניתן לראות כי לא הושג שיפור עבורי מנגנון העצירה המוקדמת באלגוריתם SGD עם מומנטום 0.9.

פתרון משימה 4

בשעיף זה השתמשנו ב- 60, EPOCHS=30 על כל שיעור למידה של 0.005, ו- 0.0001 עם Batch-size=20

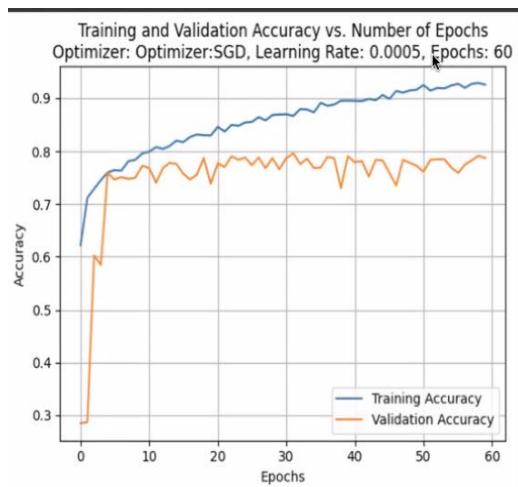
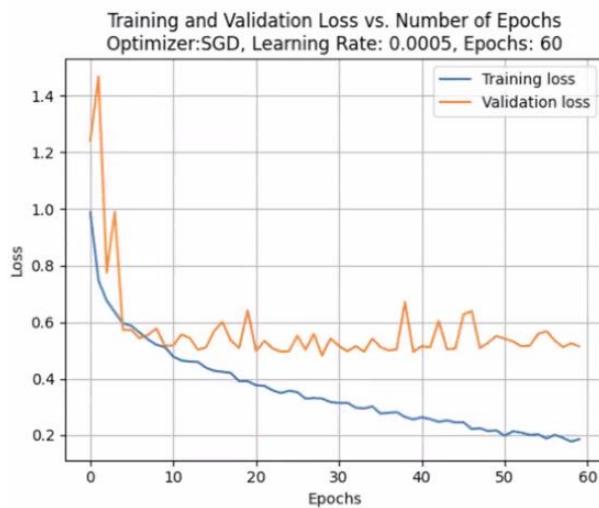
: SGD

: LR=0.0005,EPOCHS=30



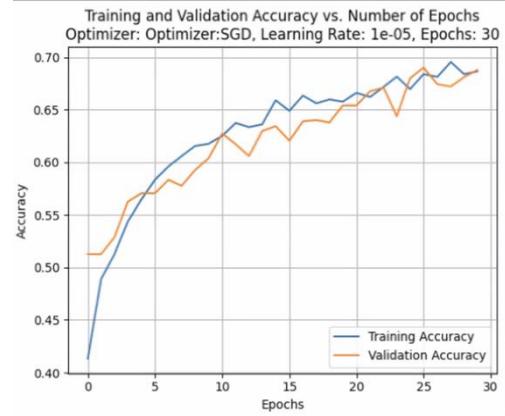
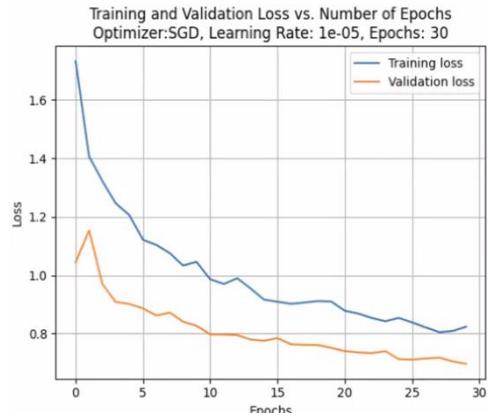
Test accuracy: 0.841296911239624
Test loss: 0.4035588502883911

: LR=0.0005,EPOCHS=60



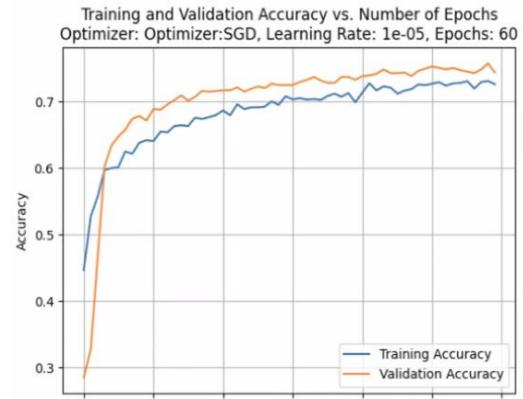
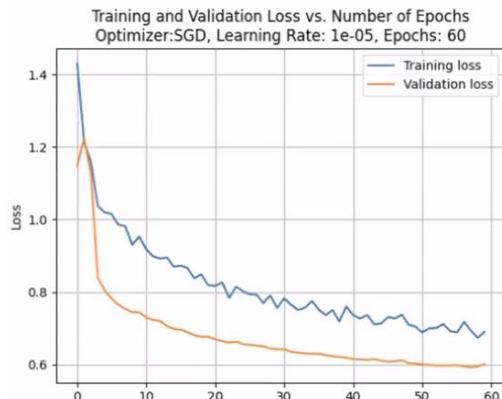
Test accuracy: 0.829351544380188
Test loss: 0.4250079095363617

: LR=0.00001, EPOCHS=30



Test accuracy: 0.7329351305961609
 Test loss: 0.6400948166847229

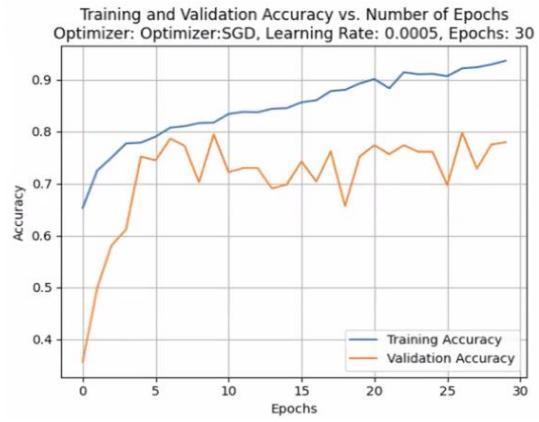
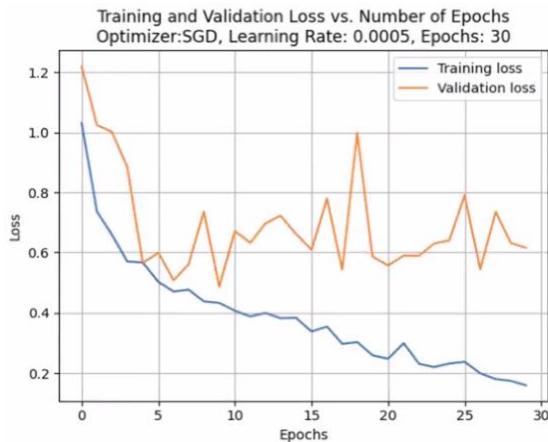
: LR=0.00001, EPOCHS=60



Test accuracy: 0.7986348271369934
 Test loss: 0.5140733122825623

: SGD, With Momentum = 0.9

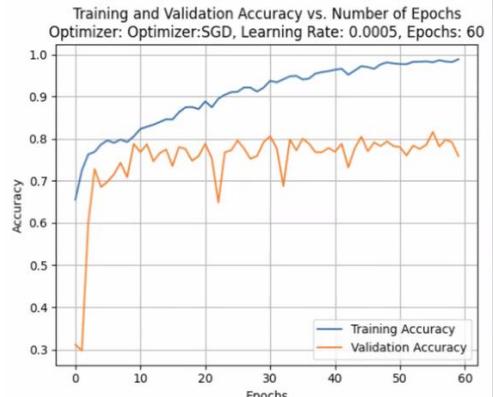
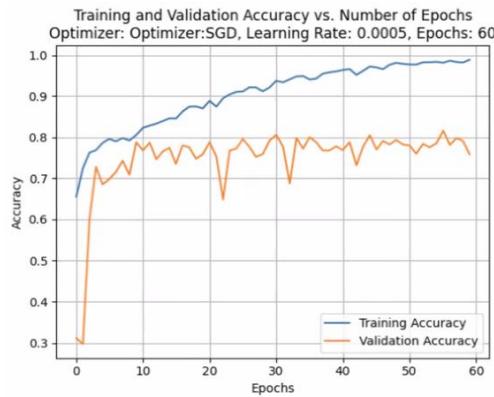
: LR=0.0005,EPOCHS=30



Test accuracy: 0.835324227809906

Test loss: 0.46957939863204956

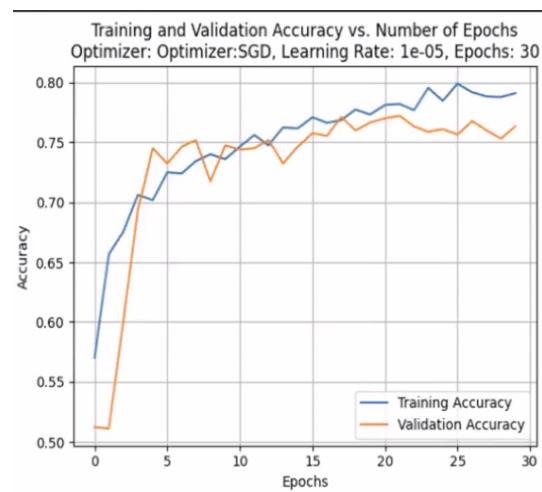
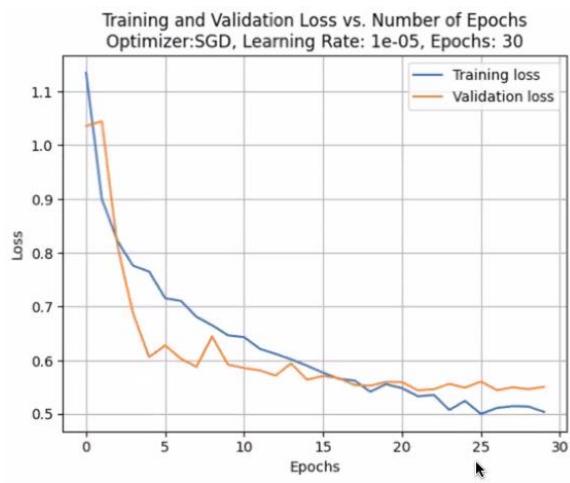
: LR=0.0005,EPOCHS=60



Test accuracy: 0.7679181098937988

Test loss: 0.8959608674049377

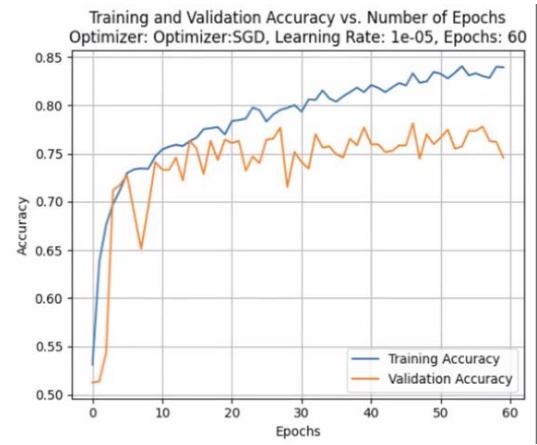
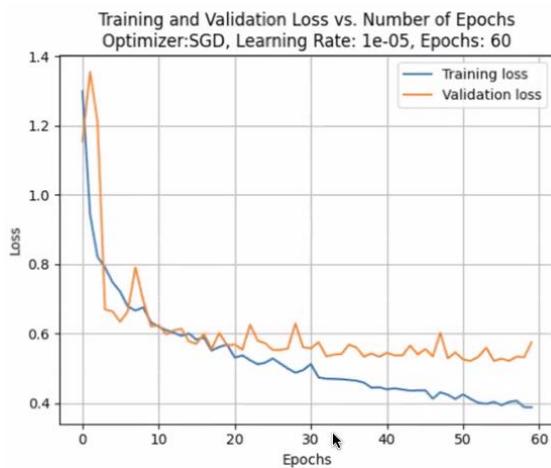
: LR=0.00001,EPOCHS=30



Test accuracy: 0.8259385824203491

Test loss: 0.44234755635261536

: LR=0.00001,EPOCHS=60

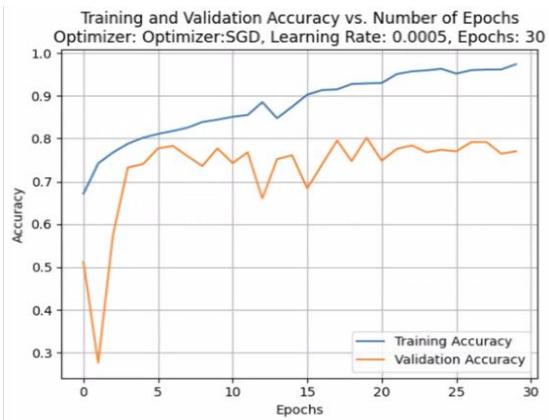
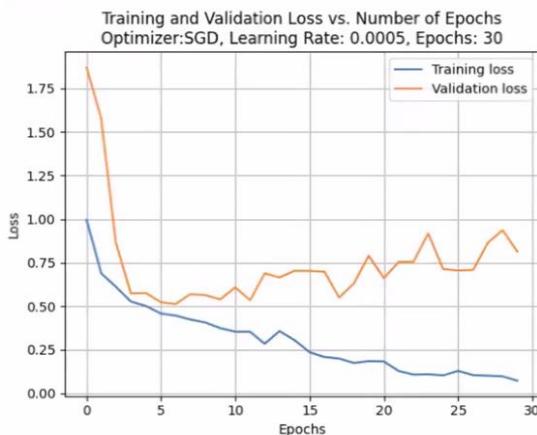


Test accuracy: 0.8063139915466309

Test loss: 0.45169296860694885

: ADAM

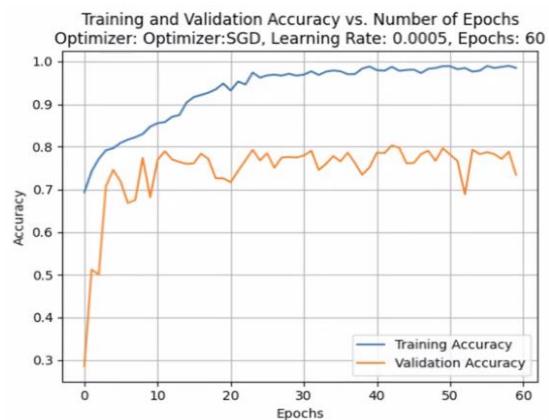
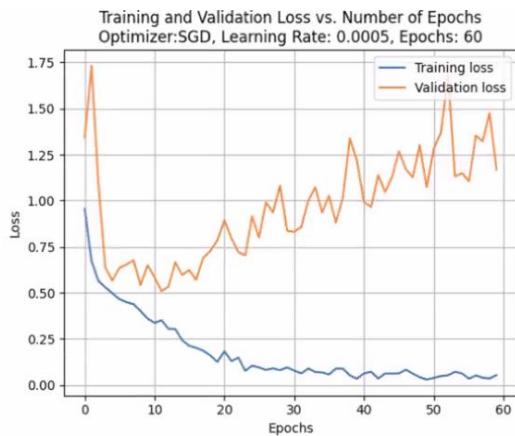
: LR=0.0005,EPOCHS=30



Test accuracy: 0.8011945486068726

Test loss: 0.6836934089660645

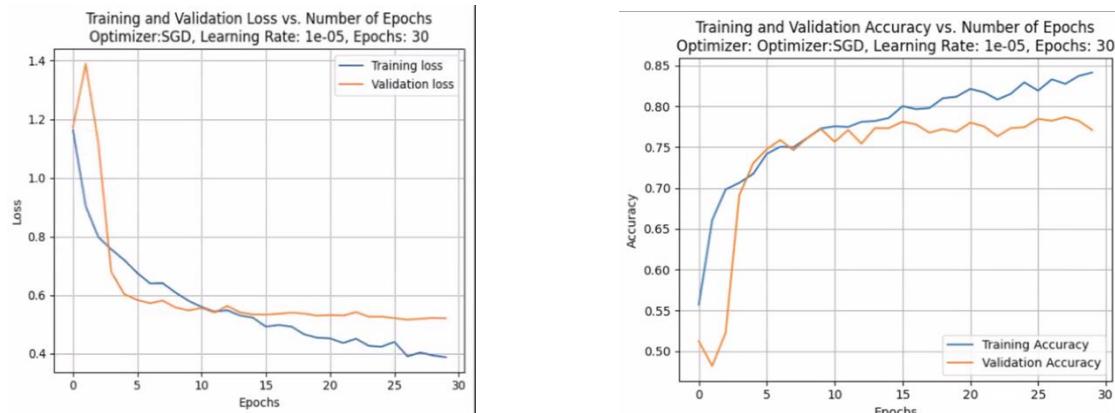
: LR=0.0005,EPOCHS=60



Test accuracy: 0.770477831363678

Test loss: 1.1424576044082642

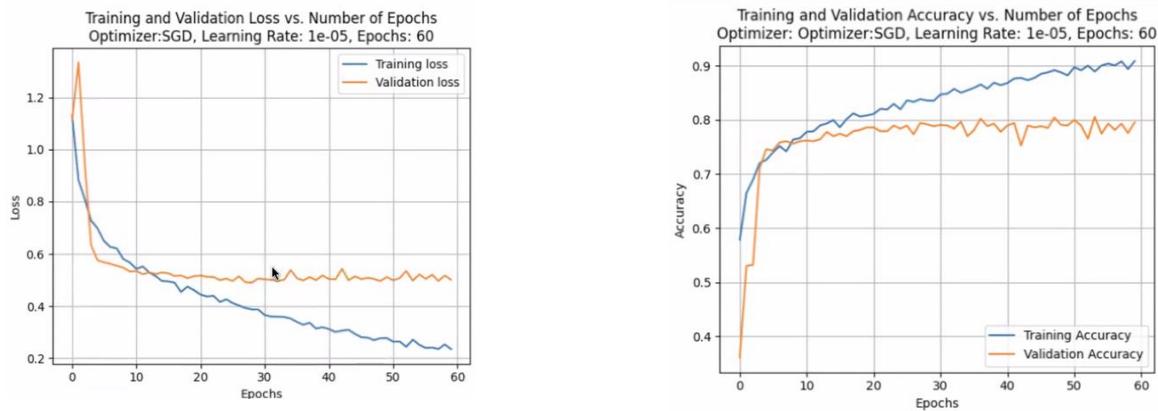
: LR=0.00001,EPOCHS=30



Test accuracy: 0.8199658989906311

Test loss: 0.44494539499282837

: LR=0.00001,EPOCHS=60

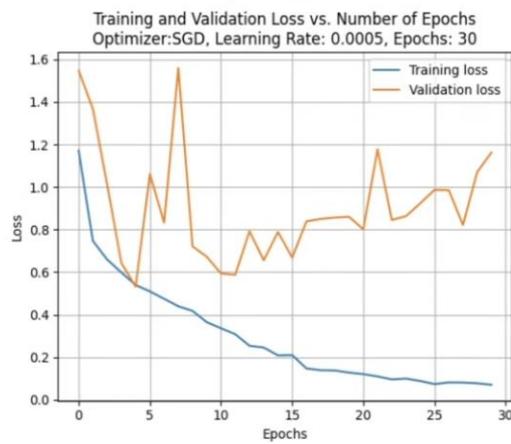


Test accuracy: 0.835324227809906

Test loss: 0.40670353174209595

: RMSPROP

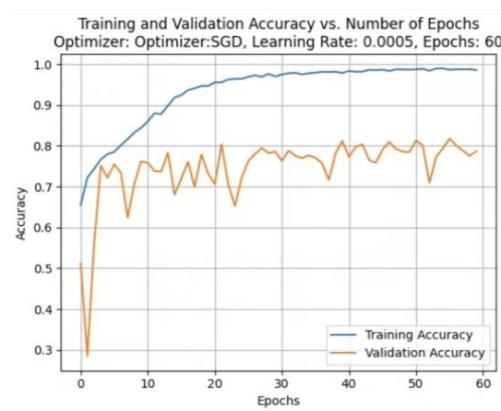
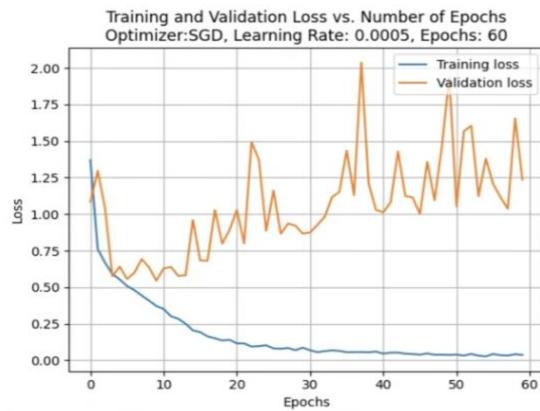
: LR=0.0005,EPOCHS=30



Test accuracy: 0.849829375743866

Test loss: 0.69105464220047

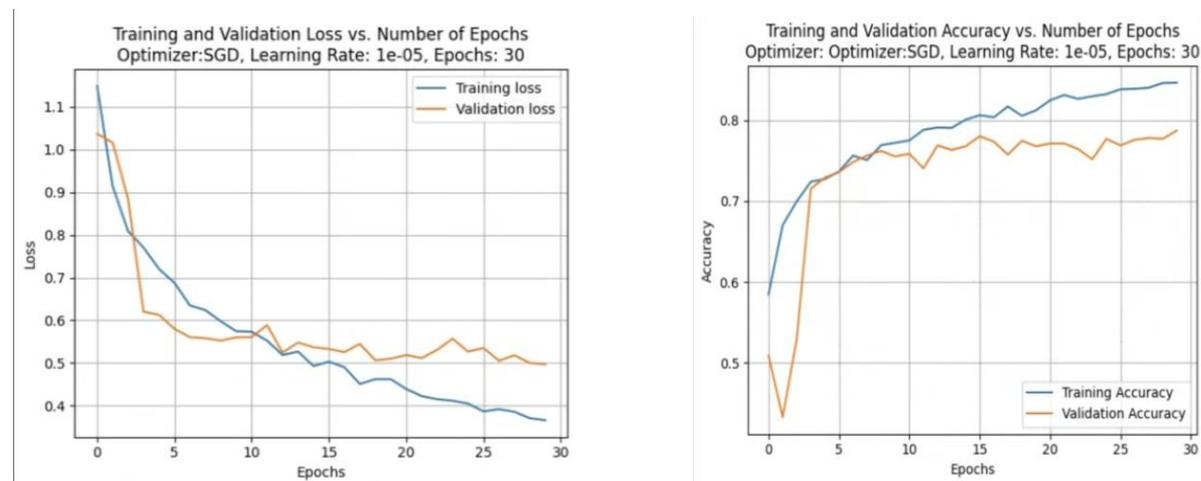
: LR=0.0005,EPOCHS=60



Test accuracy: 0.835324227809906

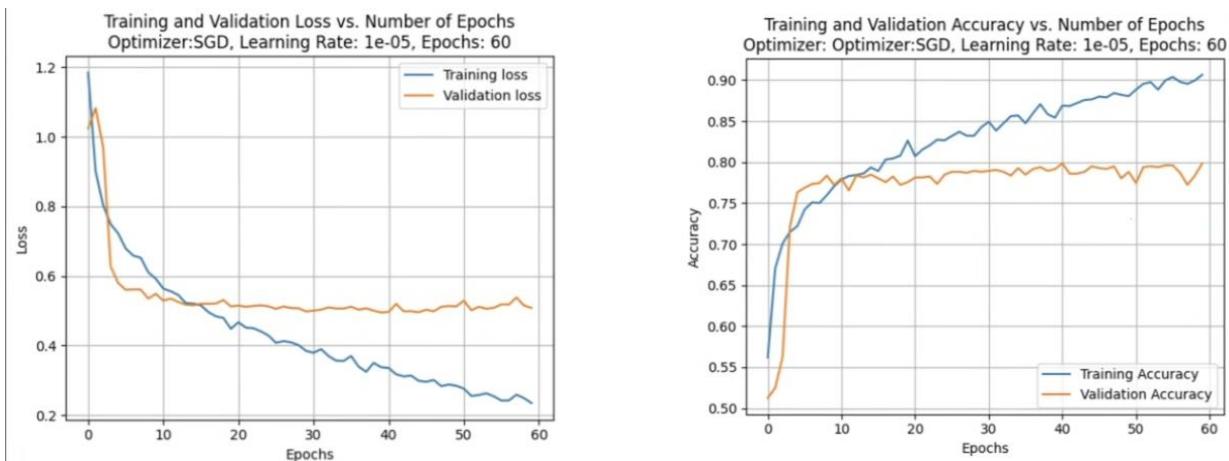
Test loss: 0.9223475456237793

: LR=0.00001,EPOCHS=30



Test accuracy: 0.82337886095047
Test loss: 0.417412668466568

: LR=0.00001,EPOCHS=60



Test accuracy: 0.8344709873199463
Test loss: 0.4058295488357544

סיכום התוצאות:

| SGD | | |
|---------------|--------|--------|
| Learning Rate | Epochs | |
| 0.0005 | 30 | 0.841 |
| 0.00001 | 60 | 0.7329 |

| SGD (With Momentum = 0.9) | | |
|---------------------------|--------|--------|
| Learning Rate | Epochs | |
| 0.0005 | 30 | 0.8353 |
| 0.00001 | 60 | 0.8259 |

| RMSprop | | |
|---------------|--------|--------|
| Learning Rate | Epochs | |
| 0.0005 | 30 | 0.8498 |
| 0.00001 | 60 | 0.8233 |

| Adam | | |
|---------------|--------|--------|
| Learning Rate | Epochs | |
| 0.0005 | 30 | 0.8011 |
| 0.00001 | 60 | 0.8199 |

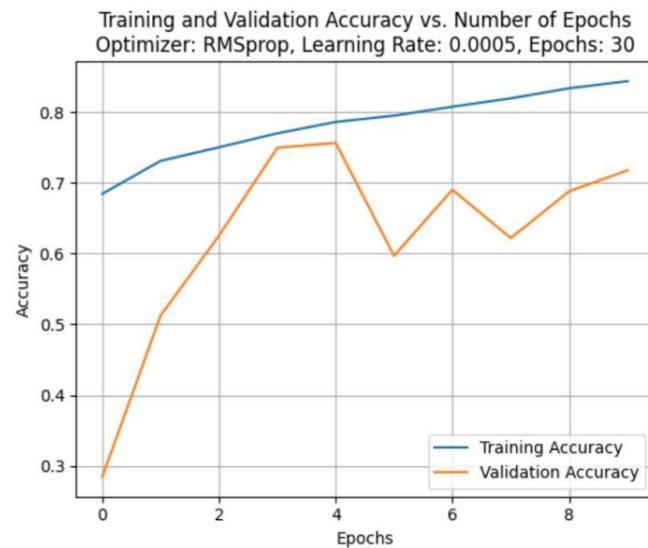
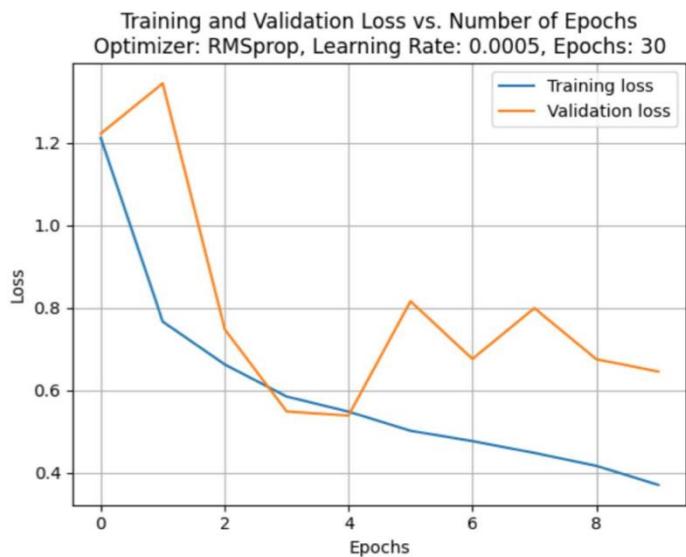
- ניתן לראות כי עבור RMSPROP עם שיעור למידה 0.0005 ואפוקס 30 קיבלנו את התוצאה הגבוהה ביותר על סט הבדיקה.

```

Epoch 28/60
232/232 [=====] - 10s 42ms/step - loss: 0.2927 - accuracy: 0.8804 - val_loss: 0.3893 - val_accuracy: 0.8700
Epoch 29/60
232/232 [=====] - 10s 43ms/step - loss: 0.2868 - accuracy: 0.8819 - val_loss: 0.4680 - val_accuracy: 0.8400
Epoch 30/60
232/232 [=====] - 10s 43ms/step - loss: 0.2727 - accuracy: 0.8868 - val_loss: 0.4004 - val_accuracy: 0.8500
Epoch 31/60
232/232 [=====] - ETA: 0s - loss: 0.2686 - accuracy: 0.8933Restoring model weights from the end of the best epoch: 16.
232/232 [=====] - 10s 43ms/step - loss: 0.2686 - accuracy: 0.8933 - val_loss: 0.3811 - val_accuracy: 0.8500
Epoch 31: early stopping
36/36 [=====] - 1s 13ms/step - loss: 0.5133 - accuracy: 0.7861

```

- נבע צעט את המנגנון העצירה המוקדמת עבור האלגוריתם RMSPROP עם שיעור למידה 0.0005 ואפוקס 30.

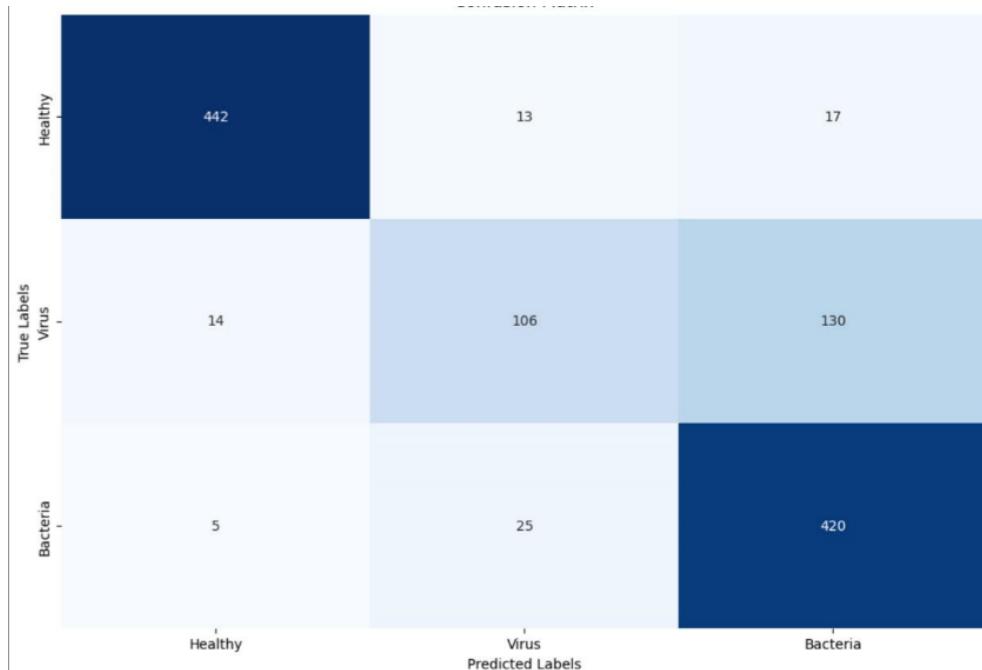


Test accuracy: 0.8259385824203491

Test loss: 0.4141954779624939

- ניתן לראות כי עברו מנגנון עזירה המוקדמת לא הביא לשיפור בתוצאות ביחס לאלגוריתם RMSPROP ללא העזירה המקודמת.

להלן מטריצת הבלבול עבור ההרצתה:



Class 0: Precision = 0.9588, Recall = 0.9364
Class 1: Precision = 0.7361, Recall = 0.4240
Class 2: Precision = 0.7407, Recall = 0.9333

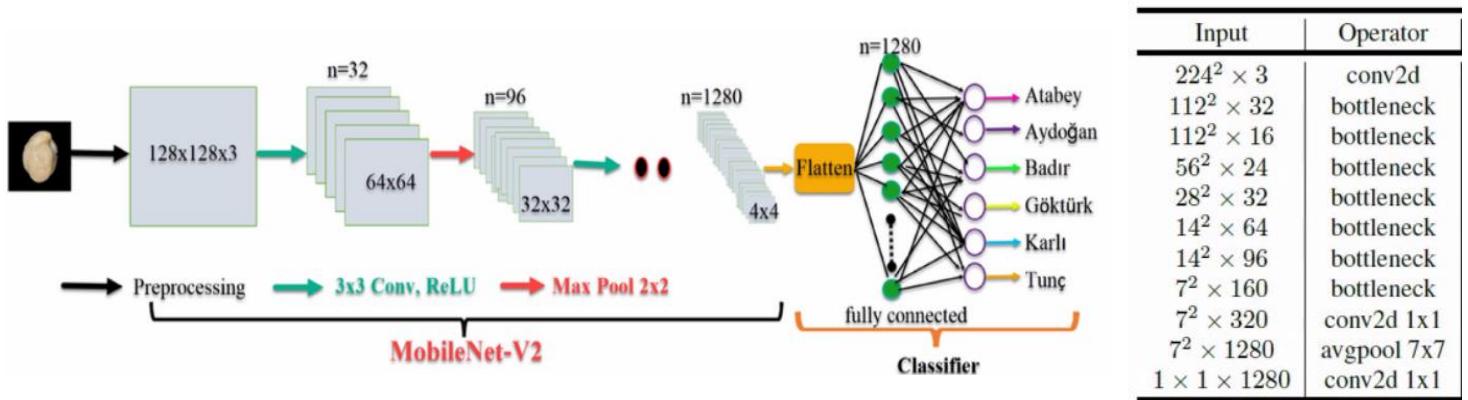
רשת Transfer Learning

פתרון משימה 1 :

המודל:

לבנייה רשת מסווג זה השתמשנו ברשת MobileNetV2, שבה נזורנו במהלך המעבדה.

רשת זו כוללת את השכבות הבאות (154 שכבות) :



- נציין כי קלט הרשת יכול להיות כל תמונה בכל גודל ובכל צבע.
- את כל השכבות הcranio בעבר, למעט שכבה Bottleneck

לאחר מספר ניסיונות לשיפור ביצועי הרשת, הוספנו את השכבות הבאות לרשת הבסיס :

```
# Additional layers
x = tf.keras.layers.Conv2D(32, (3, 3), activation='relu', padding='same')(x)
x = tf.keras.layers.MaxPooling2D((2, 2))(x)
x = tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same')(x)
x = tf.keras.layers.MaxPooling2D((2, 2))(x)
x = tf.keras.layers.Conv2D(128, (3, 3), activation='relu', padding='same')(x)

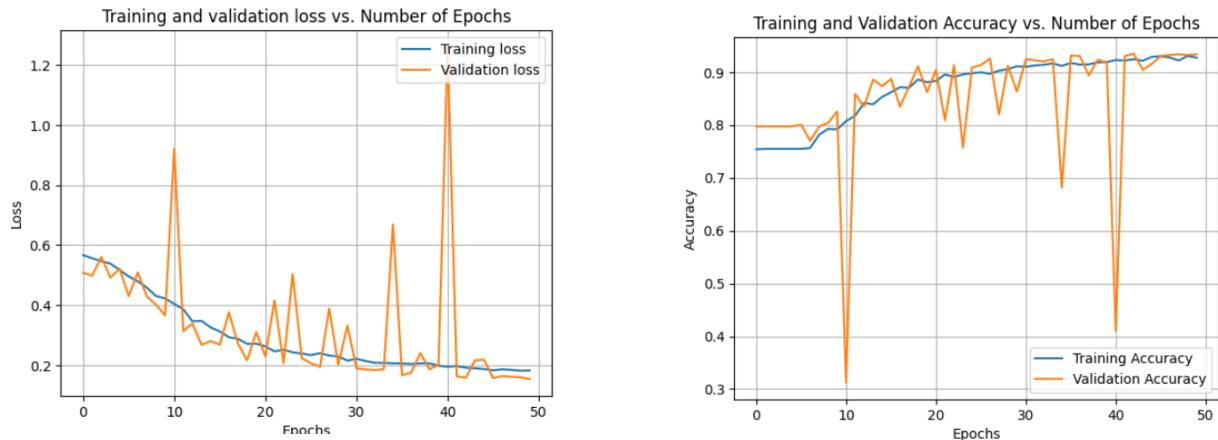
x = global_average_layer(x)
x = tf.keras.layers.Dropout(0.1)(x)
```

| Layer (type) | Output Shape | Param # |
|--|---------------------|-----------|
| input_layer_37 (InputLayer) | (None, 180, 180, 3) | 0 |
| true_divide_18 (TrueDivide) | (None, 180, 180, 3) | 0 |
| subtract_18 (Subtract) | (None, 180, 180, 3) | 0 |
| mobilenetv2_1.00_224 (Functional) | (None, 6, 6, 1280) | 2,257,984 |
| conv2d_54 (Conv2D) | (None, 6, 6, 32) | 368,672 |
| max_pooling2d_39 (MaxPooling2D) | (None, 3, 3, 32) | 0 |
| conv2d_55 (Conv2D) | (None, 3, 3, 64) | 18,496 |
| max_pooling2d_40 (MaxPooling2D) | (None, 1, 1, 64) | 0 |
| conv2d_56 (Conv2D) | (None, 1, 1, 128) | 73,856 |
| global_average_pooling2d_18 (GlobalAveragePooling2D) | (None, 128) | 0 |
| dropout_18 (Dropout) | (None, 128) | 0 |
| dense_35 (Dense) | (None, 1) | 129 |

פתרון משימה 2:

שכבות מוקפאות:

נ裏ץ את הרשת שמיישנו כאשר שכבות רשות הבסיס מוקפאות, תוך שימוש באלגוריתם Adagrad עם שיעור למידה של 0.01 וערך דעיכה (decay) של $1e-6$. מספר ה-epochs יהיה 50, וגודל ה- Mini-Batch יעמוד על 20.

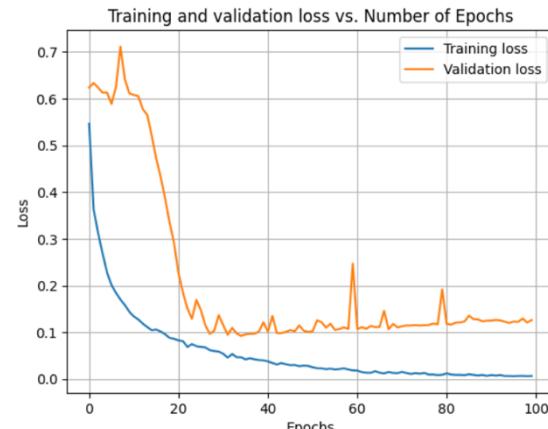
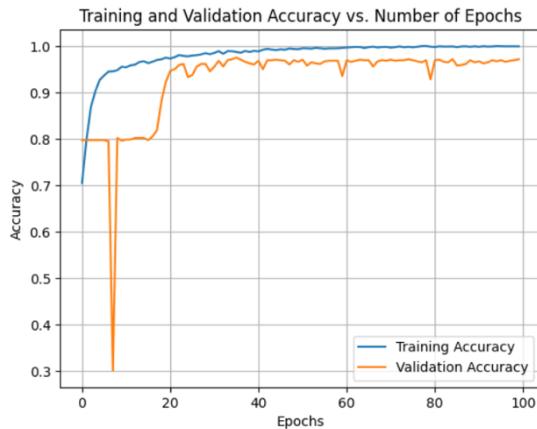


Test accuracy: 0.9488054513931274

Test loss: 0.1849689781665802

:Fine-Tuning

נريץ את הרשת שמימשנו כאשר 60 השכבות הראשונות של רשת הבסיס מוקפאות והאלגוריתם הינו Adagrad עם שיעור למידה של $0.001 \cdot 1e-6$. כמו כן, מספר האפוקס הם 100 וגודל Mini-Batch-הו 20.

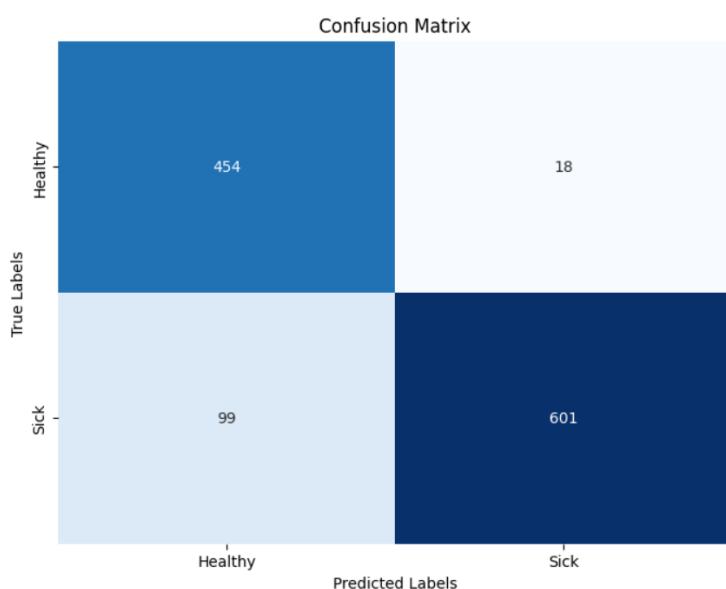


Test accuracy: 0.9155290126800537

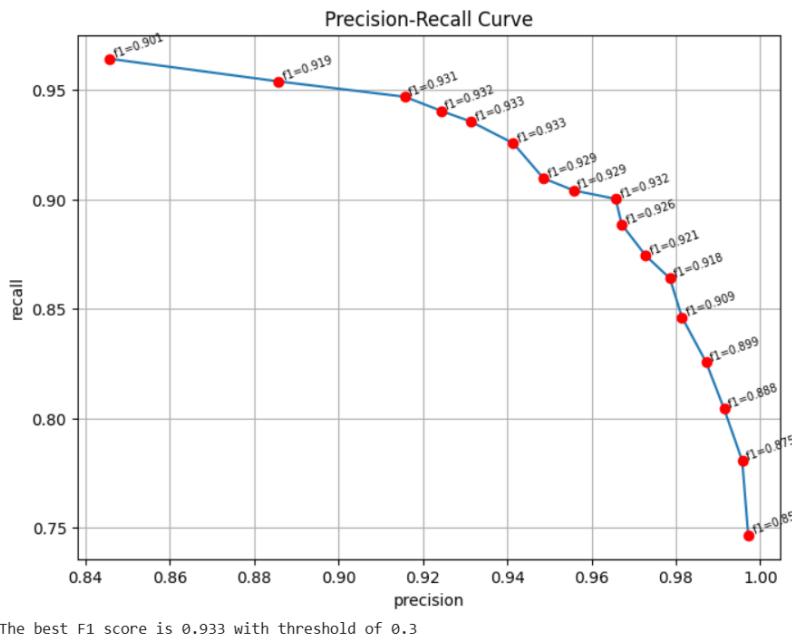
Test loss: 0.22931496798992157

- נשים לב כי עבורי הרשת עם השכבות המוקפאות קיבלנו תוצאות טובות יותר.

- ניתור כעת את מטריצת הבלבול של רשת TL עם השכבות המוקפאות :



ניצור כעת את גраф Precision-Recall



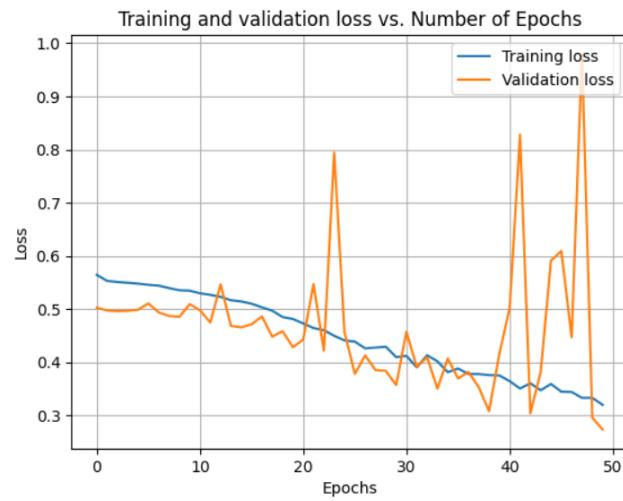
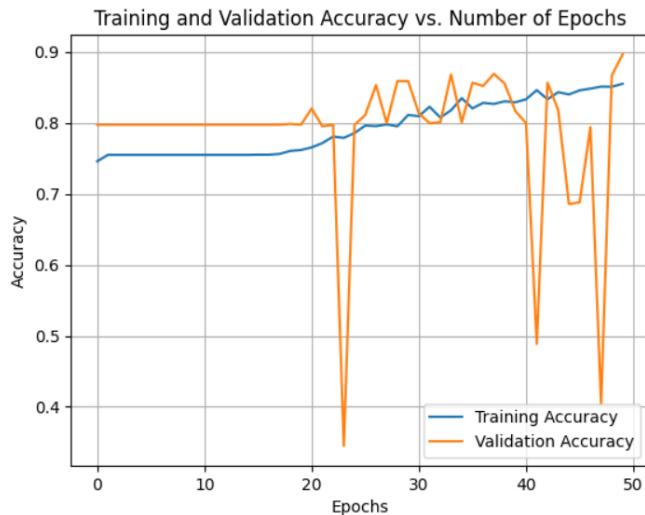
- ניתן לראות כי עבור הסטברות סף של 0.3 (כלומר אם נמצא אם בМОצה הרשות התקבלה הסטברות גבואה יותר או זיהו הצללים יסועג כחולה) עם יחס של 0.933.
- נבחר את הרשות שהניבה את התוצאות הטובות ביותר – רשות TL עם השכבות המוקפות.

פתרון משימה 3:

את משימה זו, הרצינו על שיעור למידה של 0.001 עם אפוקס 50-1 80-1 20-1

:SGD

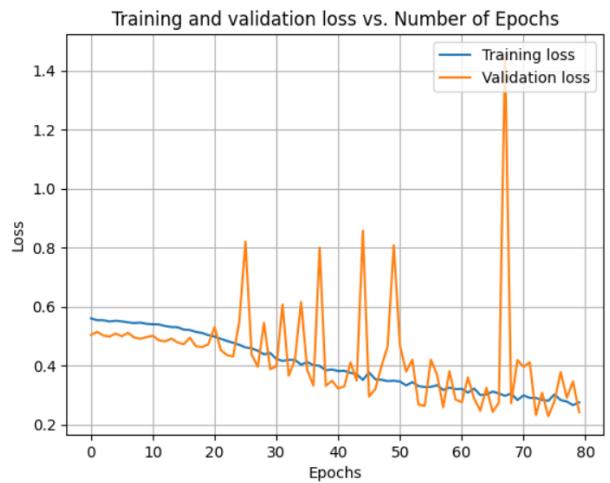
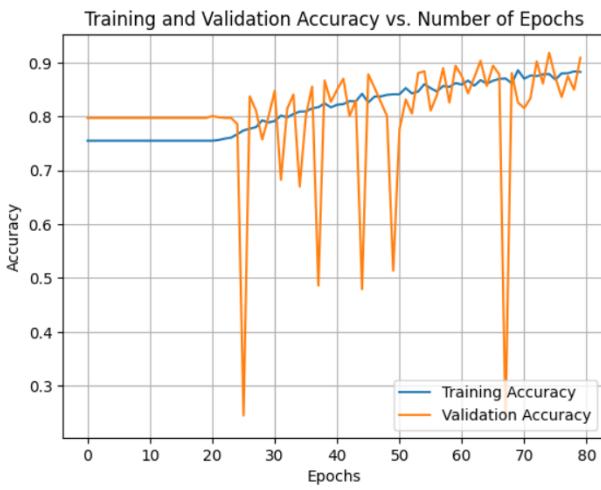
R=0.001, EPOCHS=50



Test accuracy: 0.8344709873199463

Test loss: 0.35770201683044434

: LR=0.001,EPOCHS=80

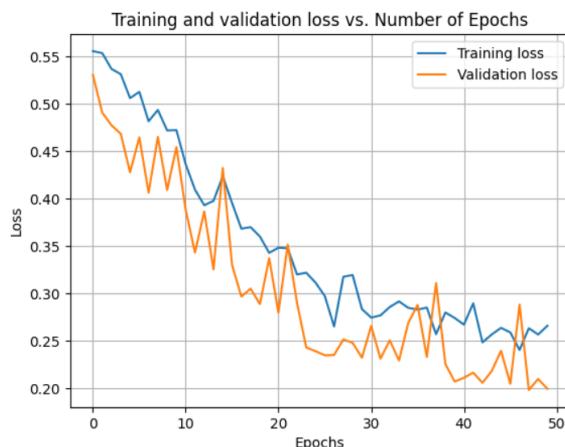


Test accuracy: 0.8831058144569397

Test loss: 0.2890828251838684

:SGD, With Momentum=0.9

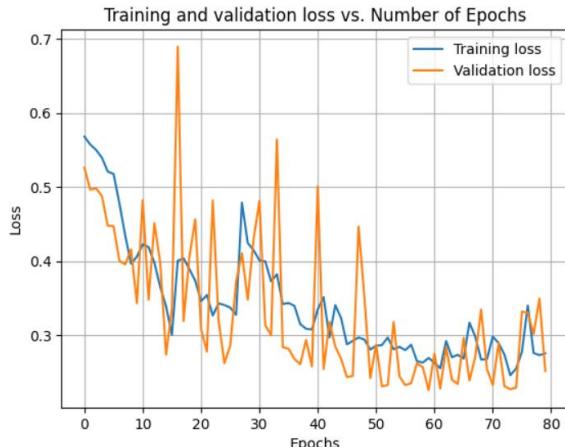
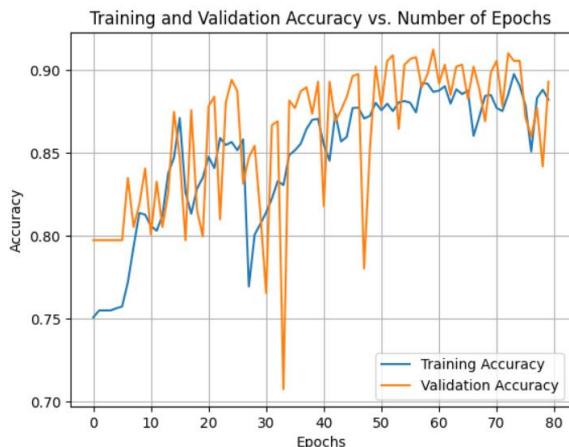
R=0.001, EPOCHS=50



Test accuracy: 0.8976109027862549

Test loss: 0.26317423582077026

R=0.001, EPOCHS=80

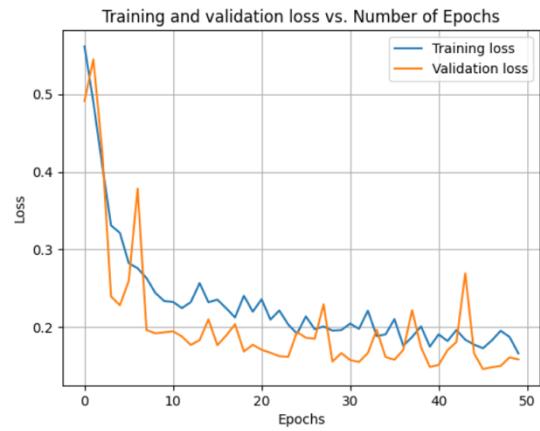
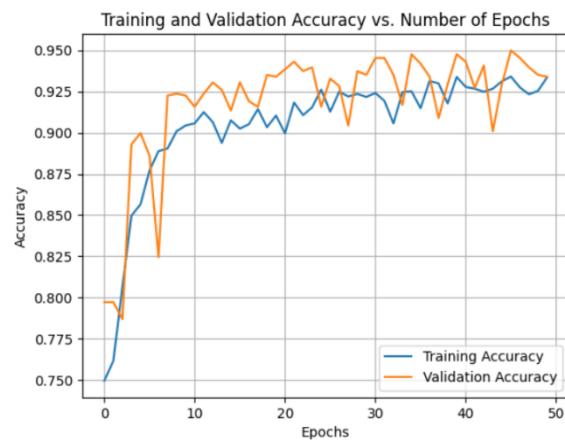


Test accuracy: 0.8208191394805908

Test loss: 0.3876152038574219

:Adam

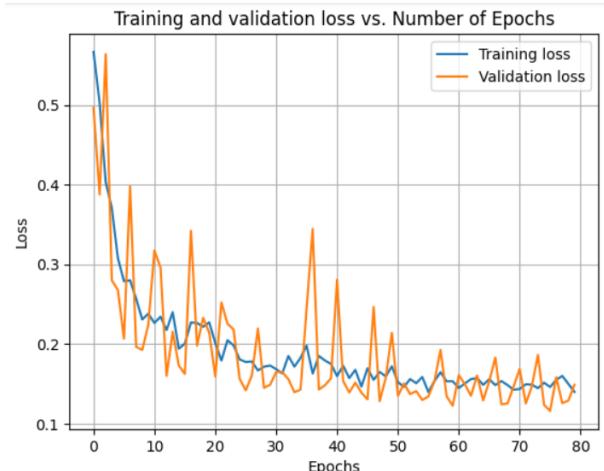
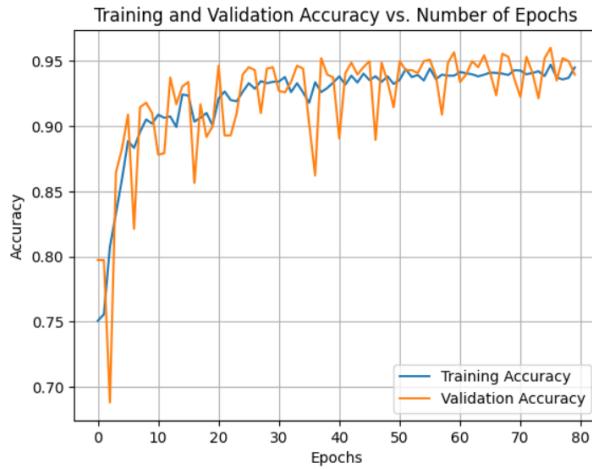
R=0.001, EPOCHS=50



Test accuracy: 0.888225257396698

Test loss: 0.2910398244857788

R=0.001, EPOCHS=80

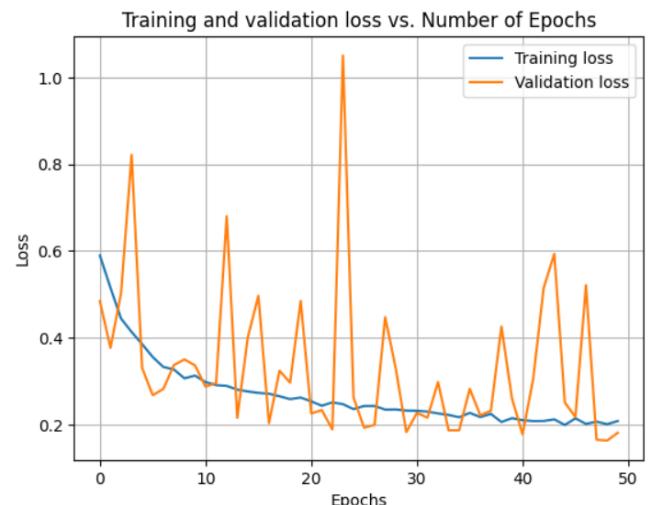
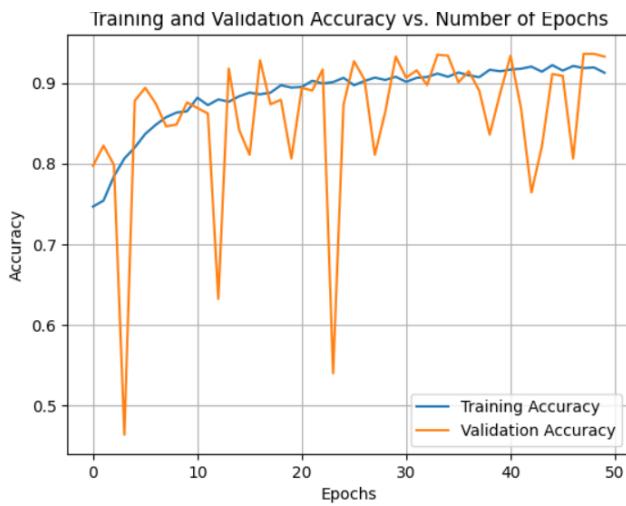


Test accuracy: 0.9300341010093689

Test loss: 0.193874329328537

:RMSprop

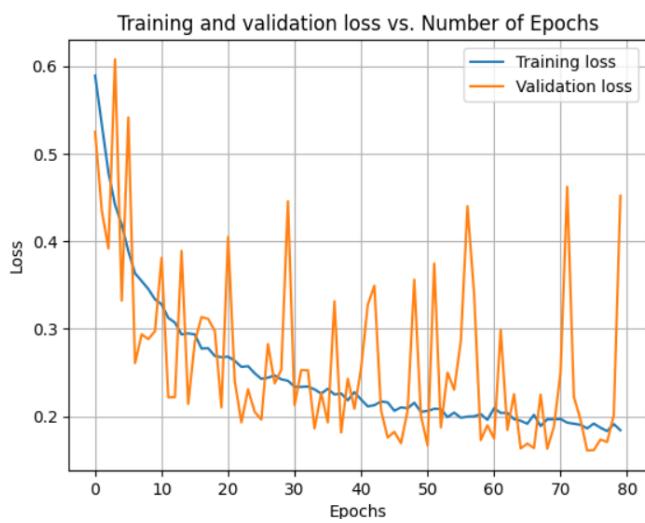
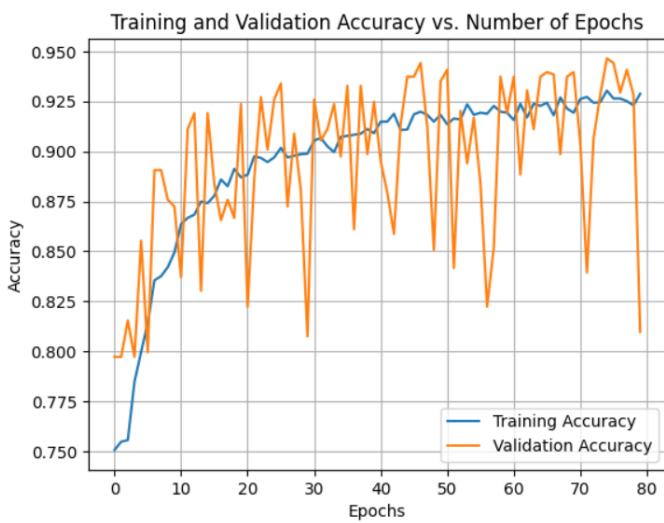
R=0.001, EPOCHS=50



Test accuracy: 0.9283276200294495

Test loss: 0.20689105987548828

R=0.001, EPOCHS=80



Test accuracy: 0.8455631136894226

Test loss: 0.3543850779533386

סיכום התוצאות:

נסכם את התוצאות על סט הבדיקה בטבלאות:

| SGD | |
|--------|---------------|
| Epochs | Learning Rate |
| 50 | 0.834 |
| 80 | 0.883 |

| SGD (With Momentum = 0.9) | |
|---------------------------|---------------|
| Epochs | Learning Rate |
| 50 | 0.897 |
| 80 | 0.820 |

| RMSprop | |
|---------|---------------|
| Epochs | Learning Rate |
| 50 | 0.928 |
| 80 | 0.845 |

| Adam | |
|--------|---------------|
| Epochs | Learning Rate |
| 50 | 0.888 |
| 80 | 0.930 |

- ניתן לראות כי עבור האלגוריתם Adam עם שיעור למידה של 0.001 – 80 אפוקס קיבלנו את התוצאה הטובה ביותר עם 93% דיוק על סט הבדיקה.
- נבחן את טיב התוצאות שקיבלונו עבור אלגוריתם זה עם מנגנון עצירה מוקדם – Early Stopping.

: EARLY STOPPING

```

Epoch 1/80
191/191 19s 54ms/step - accuracy: 0.7595 - loss: 0.5628 - val_accuracy: 0.7973 - val_loss: 0.5118
Epoch 2/80
191/191 2s 12ms/step - accuracy: 0.7513 - loss: 0.5153 - val_accuracy: 0.8531 - val_loss: 0.3372
Epoch 3/80
191/191 2s 11ms/step - accuracy: 0.8113 - loss: 0.4038 - val_accuracy: 0.8804 - val_loss: 0.3014
Epoch 4/80
191/191 2s 12ms/step - accuracy: 0.8191 - loss: 0.3832 - val_accuracy: 0.8770 - val_loss: 0.2738
Epoch 5/80
191/191 2s 12ms/step - accuracy: 0.8595 - loss: 0.3117 - val_accuracy: 0.9055 - val_loss: 0.2146
Epoch 6/80
191/191 2s 12ms/step - accuracy: 0.8607 - loss: 0.3088 - val_accuracy: 0.9089 - val_loss: 0.2035
Epoch 7/80
191/191 2s 12ms/step - accuracy: 0.8816 - loss: 0.2641 - val_accuracy: 0.9123 - val_loss: 0.1902
Epoch 8/80
191/191 2s 11ms/step - accuracy: 0.8748 - loss: 0.2980 - val_accuracy: 0.9169 - val_loss: 0.1911
Epoch 9/80
191/191 2s 11ms/step - accuracy: 0.9002 - loss: 0.2386 - val_accuracy: 0.9134 - val_loss: 0.2048
Epoch 10/80
191/191 2s 12ms/step - accuracy: 0.8851 - loss: 0.3021 - val_accuracy: 0.8189 - val_loss: 0.4047
Epoch 11/80
191/191 2s 11ms/step - accuracy: 0.8796 - loss: 0.3037 - val_accuracy: 0.8394 - val_loss: 0.4196
Epoch 12/80
191/191 2s 12ms/step - accuracy: 0.8946 - loss: 0.2378 - val_accuracy: 0.9248 - val_loss: 0.1732
Epoch 13/80
191/191 2s 12ms/step - accuracy: 0.9021 - loss: 0.2589 - val_accuracy: 0.9362 - val_loss: 0.1655
Epoch 14/80
191/191 2s 12ms/step - accuracy: 0.9203 - loss: 0.2073 - val_accuracy: 0.9305 - val_loss: 0.1624
Epoch 15/80
191/191 2s 11ms/step - accuracy: 0.9117 - loss: 0.2183 - val_accuracy: 0.9180 - val_loss: 0.1921
Epoch 16/80
191/191 2s 12ms/step - accuracy: 0.9006 - loss: 0.2370 - val_accuracy: 0.9385 - val_loss: 0.1556
Epoch 17/80
191/191 2s 12ms/step - accuracy: 0.9230 - loss: 0.1977 - val_accuracy: 0.9431 - val_loss: 0.1520
Epoch 18/80
191/191 2s 12ms/step - accuracy: 0.9221 - loss: 0.2007 - val_accuracy: 0.9431 - val_loss: 0.1479
Epoch 19/80
191/191 2s 11ms/step - accuracy: 0.9164 - loss: 0.2028 - val_accuracy: 0.9317 - val_loss: 0.1515
Epoch 20/80
191/191 2s 11ms/step - accuracy: 0.8910 - loss: 0.2605 - val_accuracy: 0.8690 - val_loss: 0.2729
Epoch 21/80
191/191 2s 12ms/step - accuracy: 0.8797 - loss: 0.2711 - val_accuracy: 0.9351 - val_loss: 0.1479
Epoch 22/80
191/191 2s 11ms/step - accuracy: 0.9165 - loss: 0.2083 - val_accuracy: 0.9157 - val_loss: 0.2031

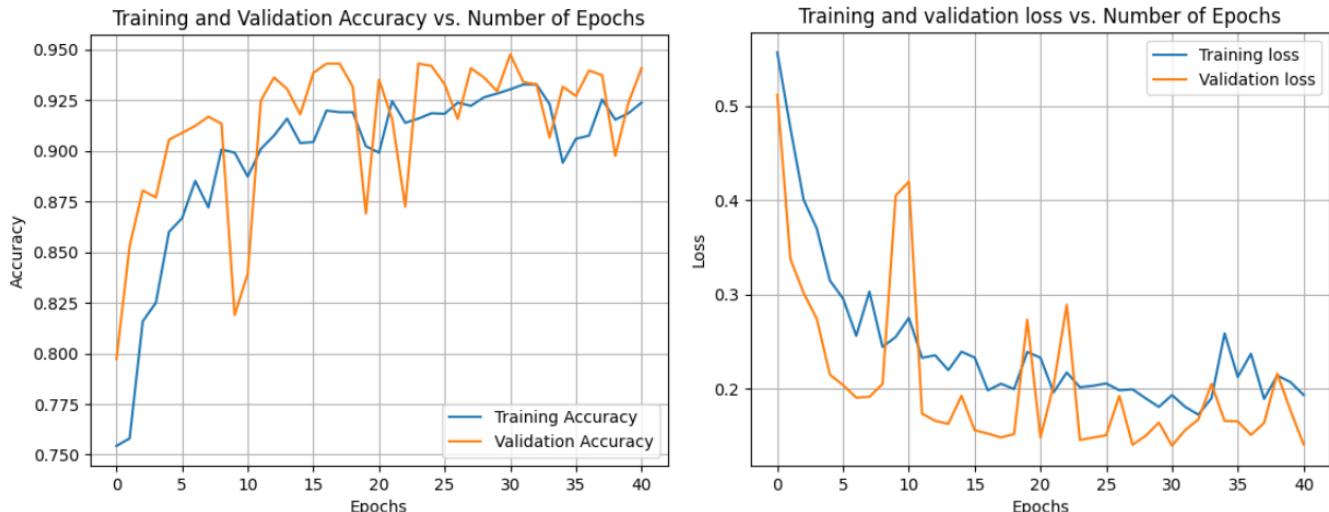
```

```

Epoch 23/80
191/191 2s 11ms/step - accuracy: 0.9087 - loss: 0.2230 - val_accuracy: 0.8724 - val_loss: 0.2889
Epoch 24/80
191/191 2s 12ms/step - accuracy: 0.9152 - loss: 0.2118 - val_accuracy: 0.9431 - val_loss: 0.1453
Epoch 25/80
191/191 2s 11ms/step - accuracy: 0.9262 - loss: 0.1828 - val_accuracy: 0.9419 - val_loss: 0.1479
Epoch 26/80
191/191 2s 11ms/step - accuracy: 0.9111 - loss: 0.2199 - val_accuracy: 0.9328 - val_loss: 0.1502
Epoch 27/80
191/191 2s 11ms/step - accuracy: 0.9212 - loss: 0.1999 - val_accuracy: 0.9157 - val_loss: 0.1921
Epoch 28/80
191/191 2s 12ms/step - accuracy: 0.9204 - loss: 0.1968 - val_accuracy: 0.9408 - val_loss: 0.1402
Epoch 29/80
191/191 2s 11ms/step - accuracy: 0.9178 - loss: 0.2079 - val_accuracy: 0.9362 - val_loss: 0.1496
Epoch 30/80
191/191 2s 11ms/step - accuracy: 0.9356 - loss: 0.1671 - val_accuracy: 0.9294 - val_loss: 0.1637
Epoch 31/80
191/191 2s 12ms/step - accuracy: 0.9314 - loss: 0.1959 - val_accuracy: 0.9476 - val_loss: 0.1391
Epoch 32/80
191/191 2s 12ms/step - accuracy: 0.9333 - loss: 0.1735 - val_accuracy: 0.9339 - val_loss: 0.1558
Epoch 33/80
191/191 2s 11ms/step - accuracy: 0.9344 - loss: 0.1604 - val_accuracy: 0.9328 - val_loss: 0.1669
Epoch 34/80
191/191 2s 11ms/step - accuracy: 0.9335 - loss: 0.1739 - val_accuracy: 0.9066 - val_loss: 0.2049
Epoch 35/80
191/191 2s 11ms/step - accuracy: 0.8977 - loss: 0.2451 - val_accuracy: 0.9317 - val_loss: 0.1654
Epoch 36/80
191/191 2s 11ms/step - accuracy: 0.9043 - loss: 0.2218 - val_accuracy: 0.9271 - val_loss: 0.1650
Epoch 37/80
191/191 2s 11ms/step - accuracy: 0.9117 - loss: 0.2296 - val_accuracy: 0.9396 - val_loss: 0.1507
Epoch 38/80
191/191 2s 11ms/step - accuracy: 0.9309 - loss: 0.1871 - val_accuracy: 0.9374 - val_loss: 0.1633
Epoch 39/80
191/191 2s 11ms/step - accuracy: 0.9106 - loss: 0.2228 - val_accuracy: 0.8975 - val_loss: 0.2157
Epoch 40/80
191/191 2s 11ms/step - accuracy: 0.9137 - loss: 0.2097 - val_accuracy: 0.9237 - val_loss: 0.1768
Epoch 41/80
191/191 2s 11ms/step - accuracy: 0.9324 - loss: 0.1723 - val_accuracy: 0.9408 - val_loss: 0.1406
Epoch 41: early stopping
Restoring model weights from the end of the best epoch: 31.

```

- ניתן לראות שקיבלו את מנגנון העצירה עבור אפוק 41.



Test accuracy: 0.9325938820838928

Test loss: 0.1971578299999927

- ניתן לראות כי מנגנון העצירה המוקדמת שיפר כמעט את הדיק של הרשות וביקת הדיווק על סט הבדיקה עומד על 93.259% הנחשב לביצועים טובים יותר.