

למידה עמוקה – מעבדה – ניסוי 2

חלק א'

1. בקוד.
2. מספרי הערוצים בשכבות הם :

התמונה (קלט) – 3 ערוצי RGB.

הבלוקים שלאחר מכן: לראשון 64, לשני 128, לשלישי 256, לרביעי 512, לחמישי 512.

לבלוק הפלט – מספר הערוצים כמספר הקטגוריות באימון (5 או 10).

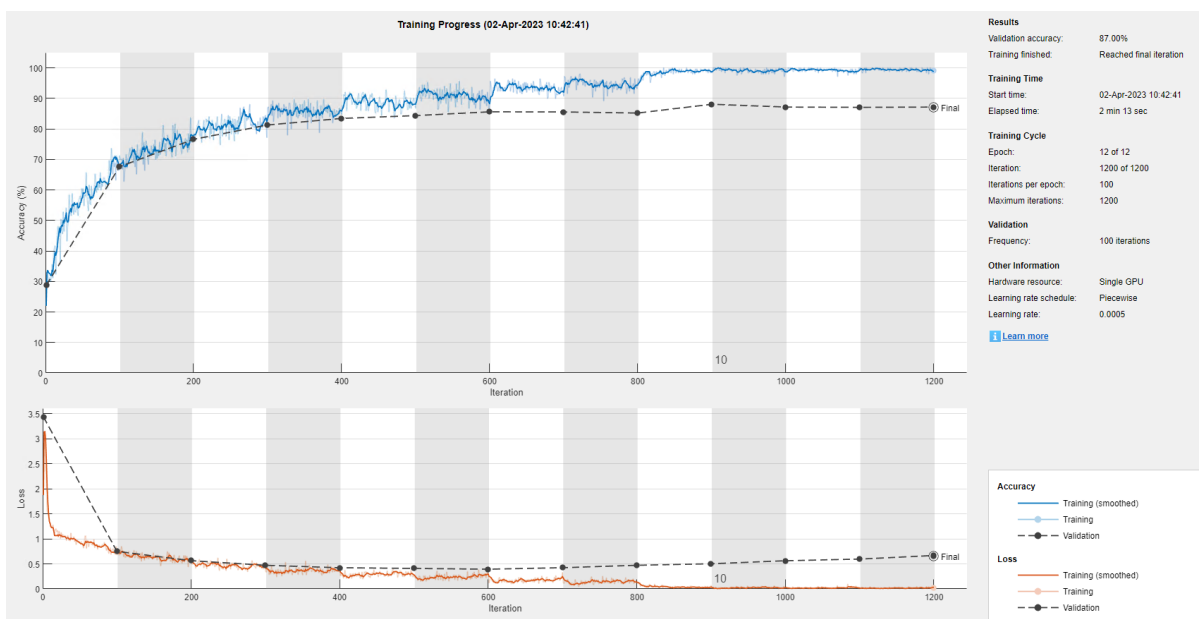
בתוך שכבות הקונבולוציה ממדי רוחב וגובה התמונה לא משתנים, ובכל שכבת Pooling ממדים אלו קטנים פי 2. ממדי העומק הם מספר הערוצים ומשתנים לפי מספר הפילטרים. לאחר ה-FC הממדים הם $1 \times 1 \times \text{numClasses}$.

3. כן, בחירת המחלקות תשפיע על איכות הסיווג. מצד אחד, נרצה לבחור מחלקות בעלות מאפיינים דומים, כדי שהמערכת תדע להבדיל בין הדקויות שביניהן. מצד שני, נרצה לבחור מחלקות בעלות שוני בין המאפיינים כדי שהמערכת תוכל ללמוד מגוון רחב יותר של מאפיינים ואת היכולת להבדיל ביניהם. מכיוון שאנו משתמשים ב-5 מחלקות כדי לקבוע את הפרמטרים, נעדיף תמונות בעלות דמיון, כדי שהסיווג יושפע באופן נרחב מההיפר-פרמטרים ולא מהשוני במשקלים הנדרשים על מנת לסווג תמונות שונות לגמרי. בחרנו בשתי תתי קבוצות שאיבריהן קרובים אחד לשני: מכונית, משאית, חתול, כלב, סוס.

4. א. לאחר 8 אפוקים ניתן לראות כי שגיאת הולידציה מתחילה לעלות. הסיבה לכך היא overfitting, כלומר התאמה של הפרמטרים הנלמדים לסט האימון על חשבון סט הולידציה. ואכן, ניתן לראות שלאחר 8 אפוקים שגיאת האימון יורדת קרוב ל-0 – כלומר התאמה כמעט מדויקת, ושגיאת הולידציה מתחילה לעלות.

ב. הוספת רגולריזציה אכן יכולה לעזור, מכיוון שתפקידן של שיטות רגולריזציה הוא למנוע בדיוק את המצב המדובר (overfitting).

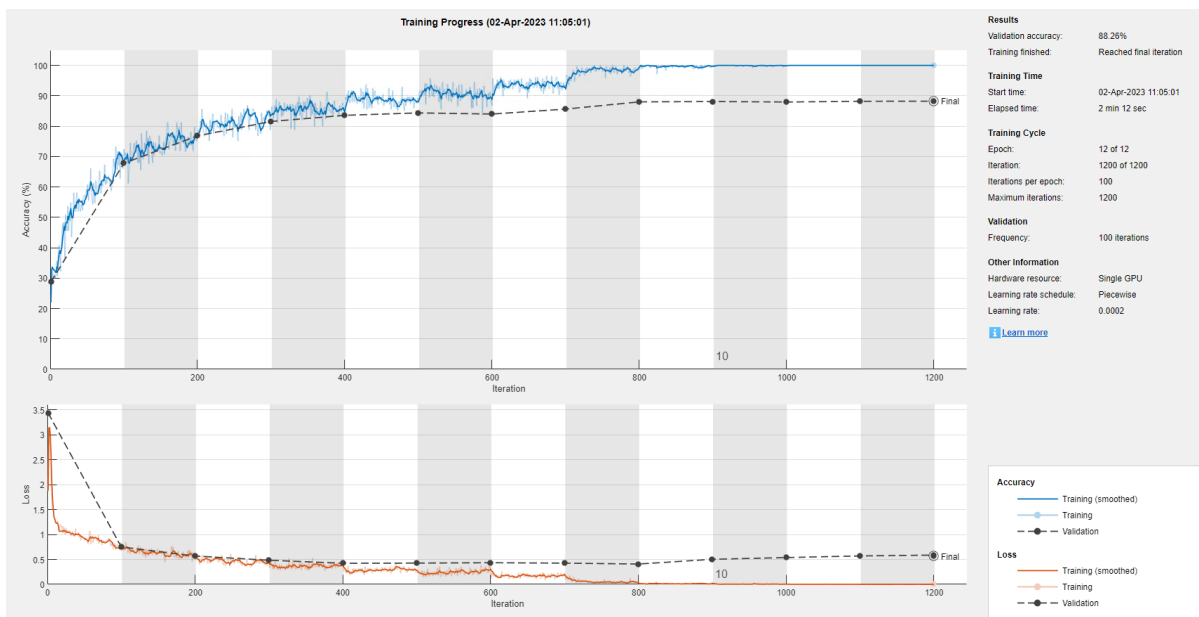
תוצאות ההרצה עבור הפרמטרים הראשוניים :



	0.47		0.12		85.20%		95.60%		800		8	
	0.50		0.06		88.08%		98.40%		900		9	
	0.56		0.01		87.16%		99.60%		1000		10	
	0.60		0.01		87.08%		100.00%		1100		11	
	0.67		0.02		87.16%		99.20%		1200		12	
	0.67		0.02		87.16%		99.20%		1200		12	

Validation error: 13.00%

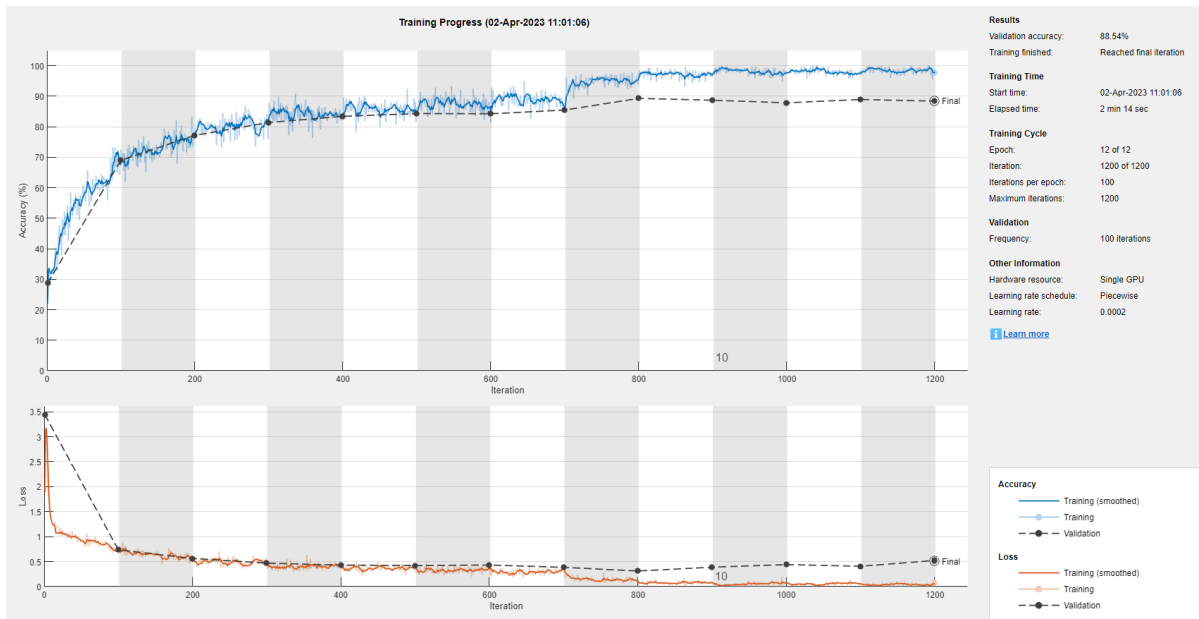
5. הפרמטרים שבחרנו הם $\text{period} = 7$, $\text{drop factor} = 0.2$.
התוצאות:



	0.41		0.02		87.98%		99.60%		800		8	
	0.50		0.04		88.10%		99.60%		900		9	
	0.54		0.00		87.94%		100.00%		1000		10	
	0.57		0.00		88.18%		100.00%		1100		11	
	0.59		0.00		88.22%		100.00%		1200		12	
	0.59		0.00		88.22%		100.00%		1200		12	

Validation error: 11.74%

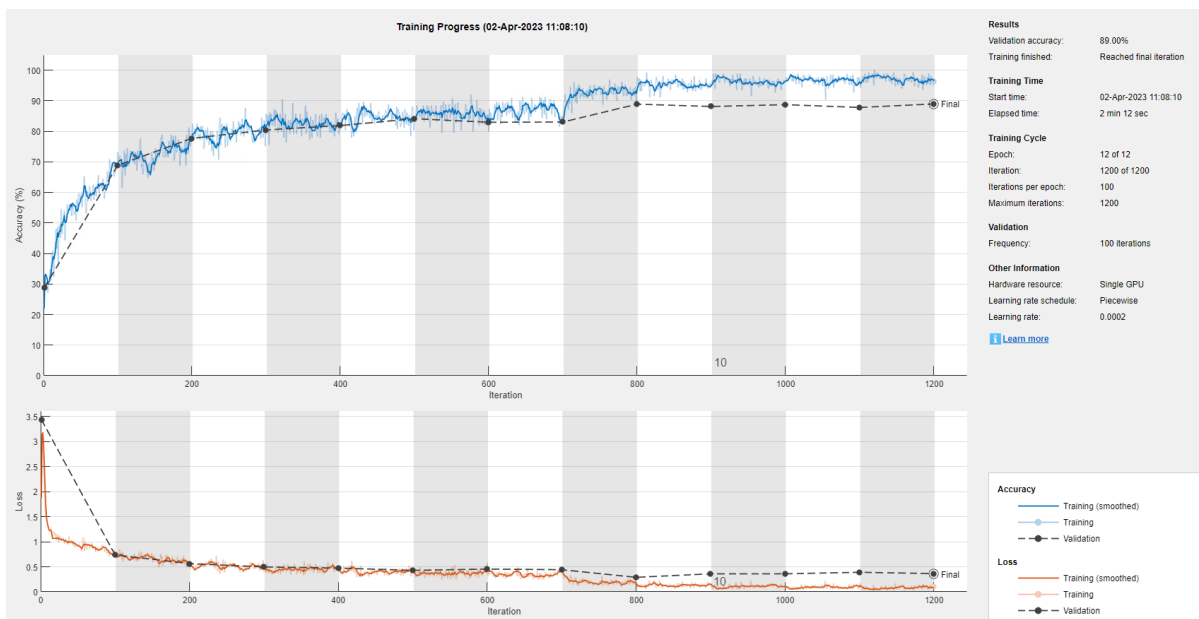
6. עבור 0.002:



	0.31		0.07		89.28%		98.00%		800		8	
	0.39		0.10		88.68%		97.60%		900		9	
	0.44		0.08		87.84%		97.60%		1000		10	
	0.41		0.07		88.92%		96.80%		1100		11	
	0.52		0.07		88.38%		97.60%		1200		12	
	0.52		0.07		88.38%		97.60%		1200		12	

Validation error: 11.46%

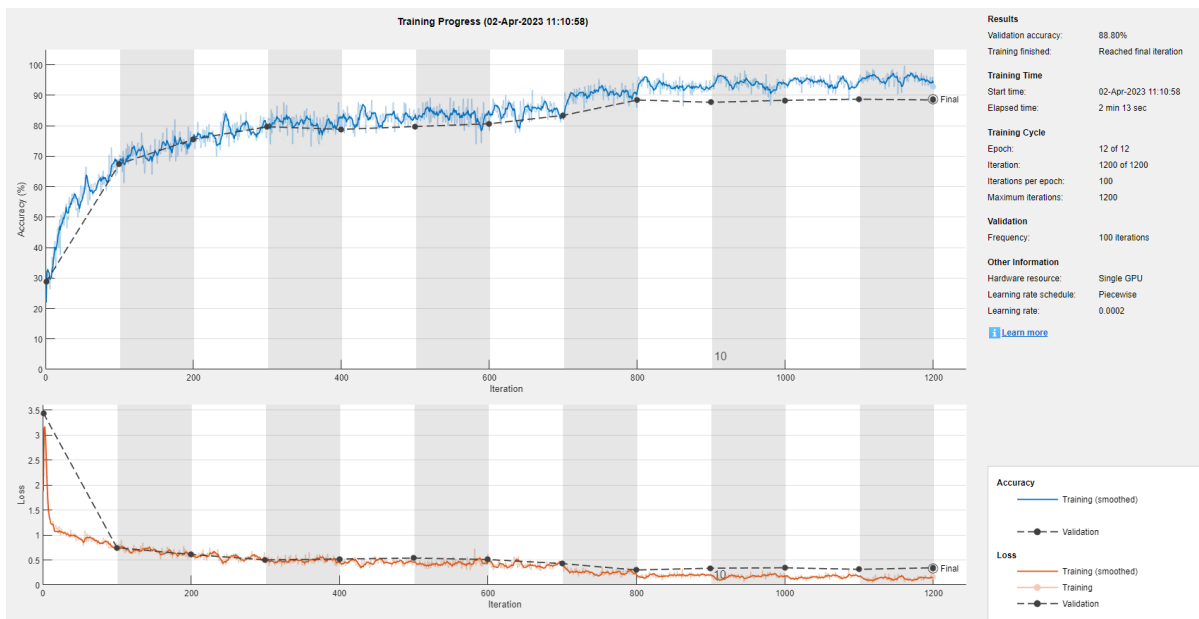
עבור 0.004:



	0.29		0.10		88.88%		97.60%		800		8	
	0.36		0.16		88.12%		95.60%		900		9	
	0.36		0.11		88.72%		96.40%		1000		10	
	0.39		0.08		87.80%		96.80%		1100		11	
	0.36		0.10		88.96%		96.40%		1200		12	
	0.36		0.10		88.96%		96.40%		1200		12	

Validation error: 11.00%

עבור 0.008:



	0.30		0.15		88.44%		94.80%		800		8	
	0.34		0.24		87.68%		92.40%		900		9	
	0.34		0.17		88.34%		93.60%		1000		10	
	0.31		0.16		88.72%		94.00%		1100		11	
	0.34		0.16		88.44%		92.80%		1200		12	
	0.34		0.16		88.44%		92.80%		1200		12	

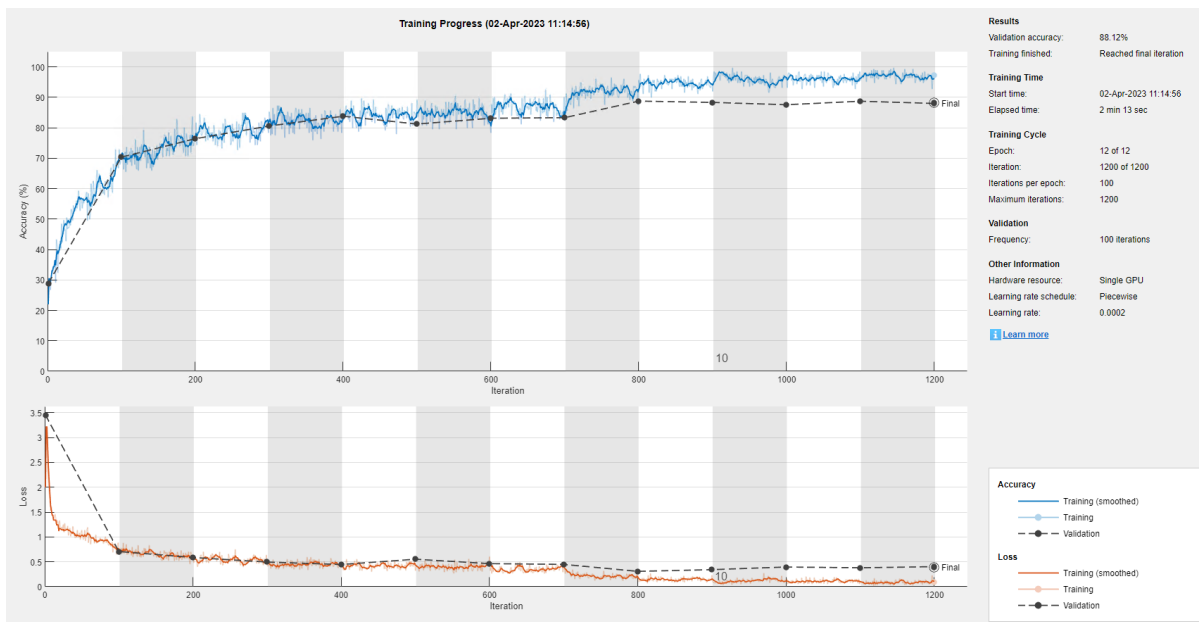
Validation error: 11.20%

עבור 0.5 פונקציית ה-loss תגדל מאוד עבור כל ערך של משקל ששונה מ-0. לכן, המשקלים יונחתו ל-0 ולא תתבצע למידה.

ב. הערך המיטבי הוא 0.004.

7. א. ערך ההסתברות הנ"ל הוא ההסתברות של נזירון כלשהו להיות לא פעיל. ערך ההסתברות ההתחלתי הוא 0.

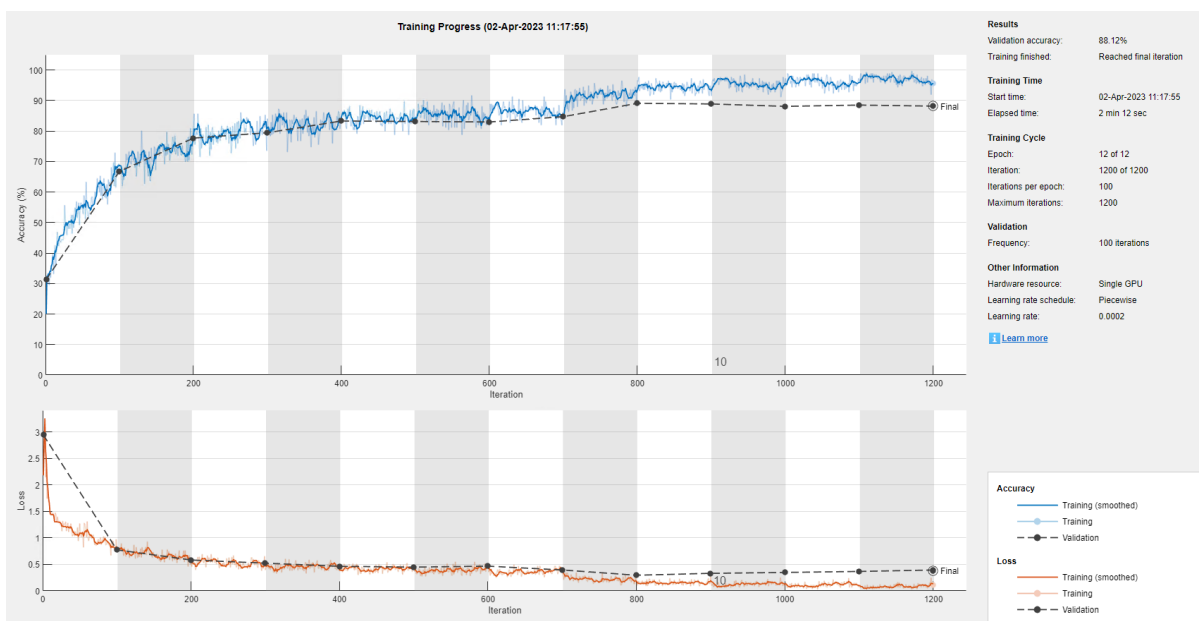
ב. עבור $p=0.1$:



	0.31		0.14		88.76%		95.60%		800		8	
	0.34		0.22		88.32%		92.00%		900		9	
	0.40		0.10		87.56%		96.40%		1000		10	
	0.38		0.11		88.72%		95.60%		1100		11	
	0.40		0.10		87.98%		97.20%		1200		12	
	0.40		0.10		87.98%		97.20%		1200		12	

Validation error: 11.88%

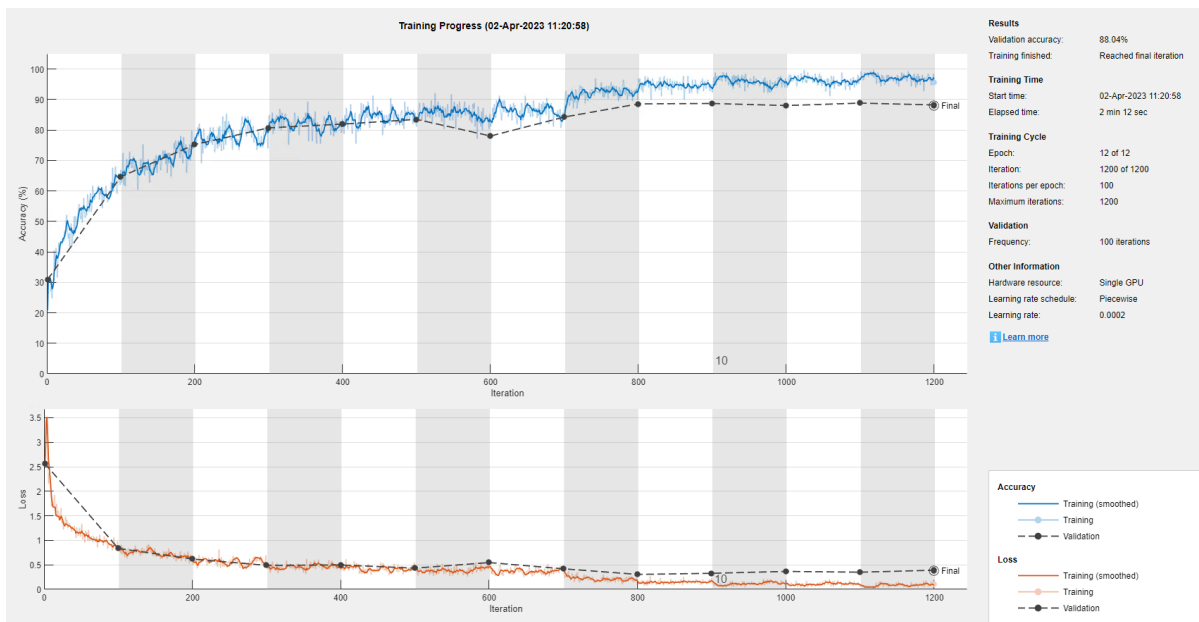
עבור $p=0.2$:



	0.30		0.13		89.08%		96.40%		800		8	
	0.33		0.27		88.86%		91.60%		900		9	
	0.35		0.10		88.08%		97.20%		1000		10	
	0.37		0.09		88.50%		96.40%		1100		11	
	0.39		0.11		88.16%		95.60%		1200		12	
	0.39		0.11		88.16%		95.60%		1200		12	

Validation error: 11.88%

עבור $p=0.4$:



	0.31		0.11		88.56%		96.80%		800		8	
	0.33		0.23		88.70%		92.40%		900		9	
	0.37		0.13		88.02%		95.60%		1000		10	
	0.35		0.08		88.82%		96.40%		1100		11	
	0.39		0.11		88.22%		95.60%		1200		12	
	0.39		0.11		88.22%		95.60%		1200		12	

Validation error: 11.96%

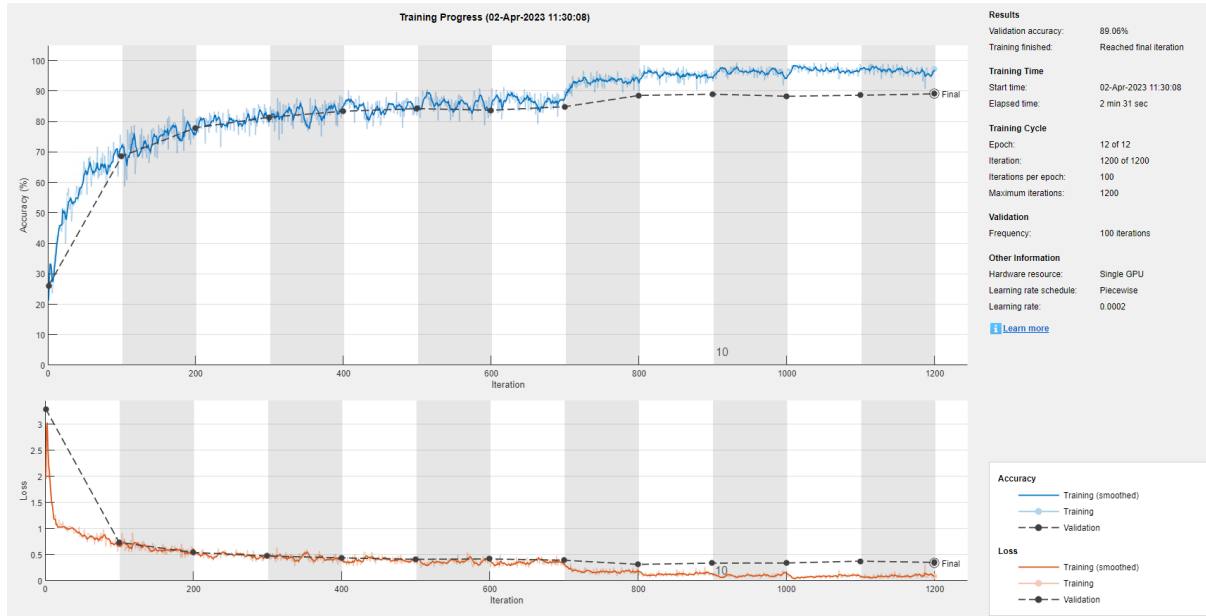
ג. ערך ההסתברות הנבחר משפיע על התוצאות בכך שהוא משנה את כמות הנורונים הלא פעילים. ערך נמוך מדי לא ימנע מספיק את ה-overfitting, וערך גבוה מדי לא יתן למידה מספיק אפקטיבית. מבין הערכים שניסינו לא קיבלנו שיפור, לכן נבחר בערך 0.

8. א. נבחר בסוגי האוגמנטציה המתאימים למאגר:

- שיקוף סביב ציר Y – שכן העצמים המשוקפים נכנסים תחת אותה מחלקה.
- הזזה – נרצה לזהות את האובייקטים בכל מיקום בתמונה, השינוי נחוץ אך יכול להקשות על הלמידה יותר מאשר בשיקוף.
- שינוי בהירות (לא קיים ב-Matlab) – כל עשר המחלקות נבדלות בצורת האובייקטים ולא בצבעם, ולכן נרצה לזהותם בכל רמת בהירות.

- שינוי גודל – האובייקטים צריכים להיות מזוהים בכל גודל, אך ב-32 פיקסלים הדבר עשוי לשבש את הלמידה.
- סיבוב – נרצה לזהות את העצמים בזוויות שונות, אך הדבר פחות חשוב מכיוון שהעצמים לרוב יופיעו באותן הזוויות.

ב. שיקוף :

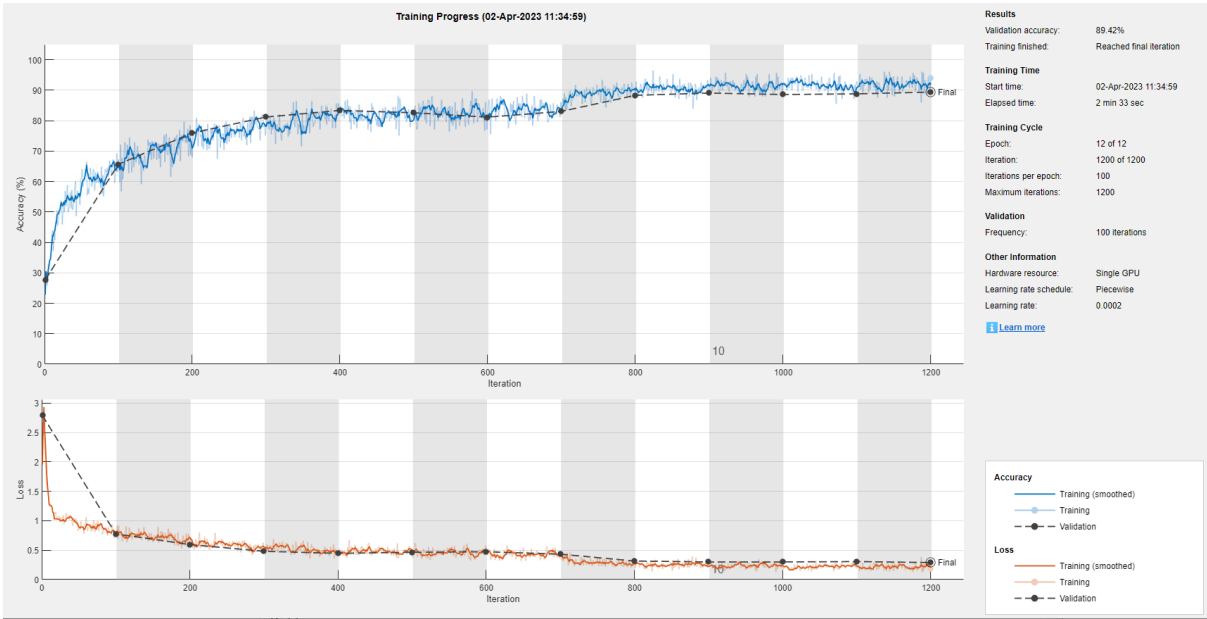


	0.32		0.25		88.54%		90.00%		800		8	
	0.34		0.15		88.96%		93.20%		900		9	
	0.34		0.13		88.26%		96.00%		1000		10	
	0.38		0.05		88.60%		98.40%		1100		11	
	0.35		0.09		89.06%		97.20%		1200		12	
	0.35		0.09		89.06%		97.20%		1200		12	

=====

Validation error: 10.94%

הזזה בתחום [3 -3] בשני הצירים:

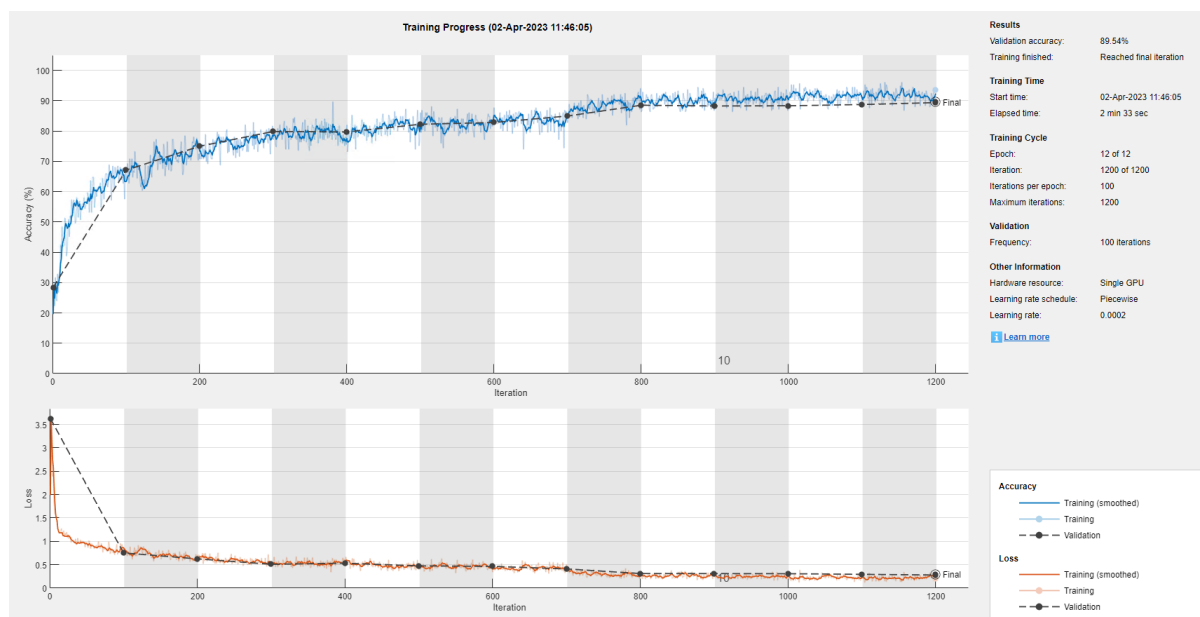


	0.32		0.31		88.32%		87.60%		800		8	
	0.30		0.25		89.06%		89.20%		900		9	
	0.30		0.26		88.58%		92.00%		1000		10	
	0.31		0.20		88.80%		92.80%		1100		11	
	0.29		0.19		89.38%		94.00%		1200		12	
	0.29		0.19		89.38%		94.00%		1200		12	

=====

Validation error: 10.58%

שינוי גודל בתחום [0.9 1.1]:



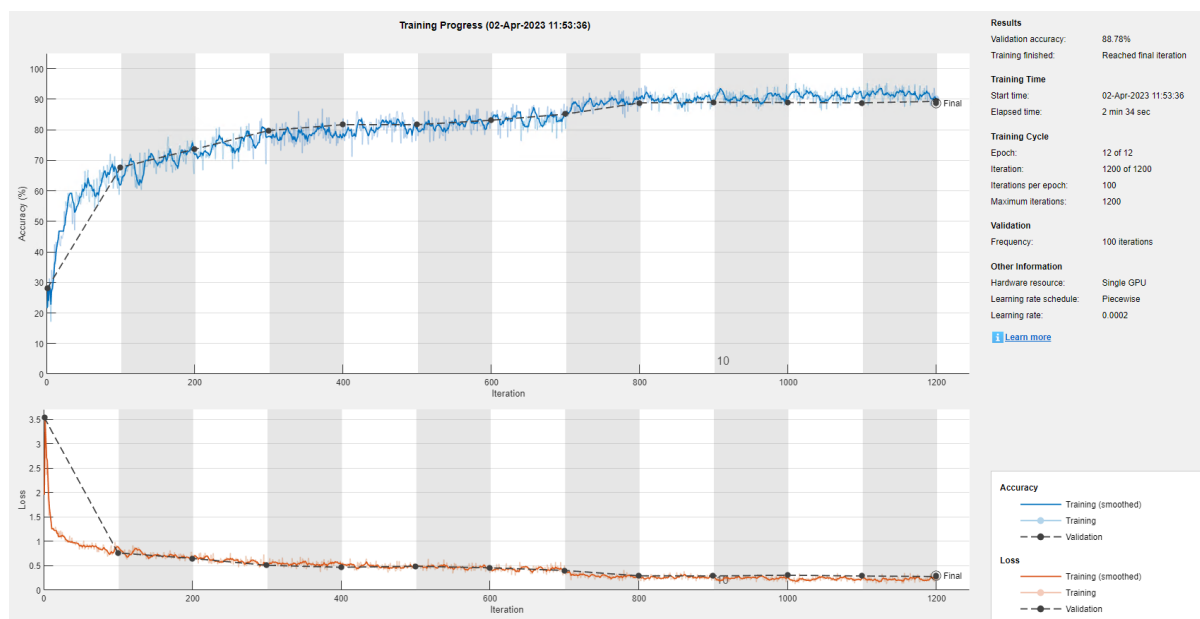
	0.31		0.36		88.50%		86.40%		800		8	
	0.31		0.31		88.24%		90.80%		900		9	
	0.31		0.26		88.32%		90.40%		1000		10	
	0.30		0.31		88.72%		89.20%		1100		11	
	0.28		0.21		89.36%		93.60%		1200		12	
	0.28		0.21		89.36%		93.60%		1200		12	

=====

Validation error: 10.46%

עבור סיבוב בטווח [-5 5]:

הסיבוב לא עזר.



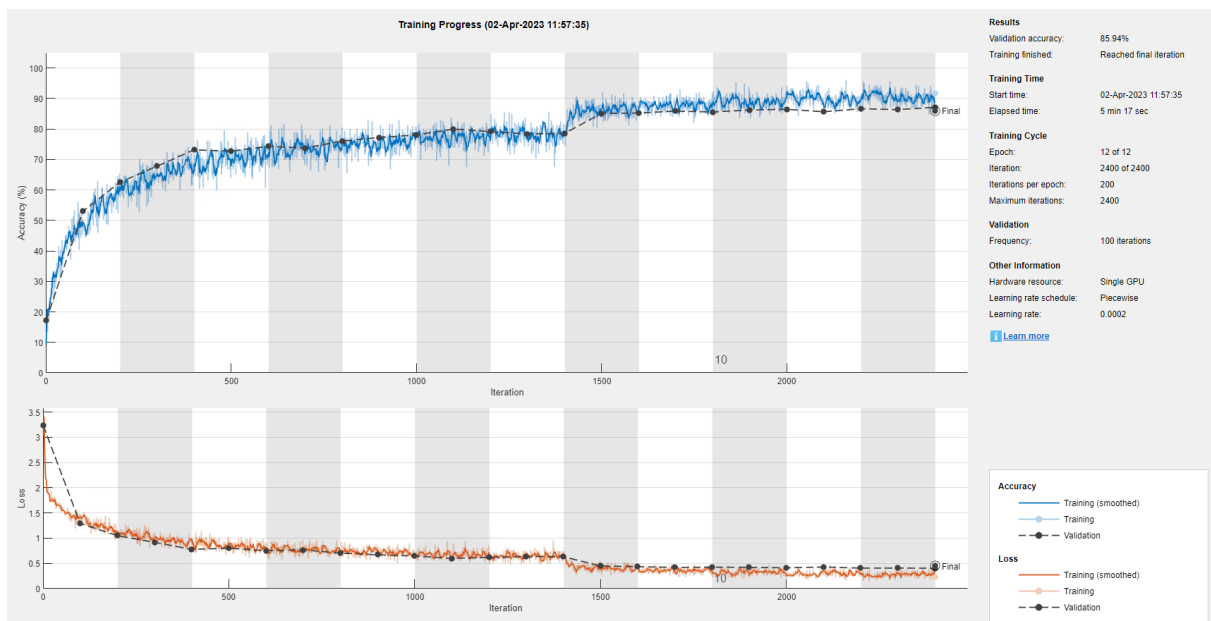
	0.29		0.34		88.80%		89.60%		800		8	
	0.29		0.27		88.98%		91.60%		900		9	
	0.31		0.25		88.90%		90.40%		1000		10	
	0.29		0.31		88.80%		88.80%		1100		11	
	0.28		0.25		89.28%		90.40%		1200		12	
	0.28		0.25		89.28%		90.40%		1200		12	

=====

Validation error: 11.22%

ג. ניתן לראות כי הסוגים שעזרו לנו הם שיקוף סביב ציר ה-Y, הזזה, שינוי גודל.

9. לאחר ההרצה הראשונה קיבלנו :



0.41	0.25	86.45%	94.00%	2000	10
0.43	0.39	85.64%	85.60%	2100	11
0.41	0.30	86.63%	89.60%	2200	11
0.41	0.26	86.34%	89.60%	2300	12
0.40	0.24	87.10%	91.60%	2400	12
0.40	0.24	87.10%	91.60%	2400	12

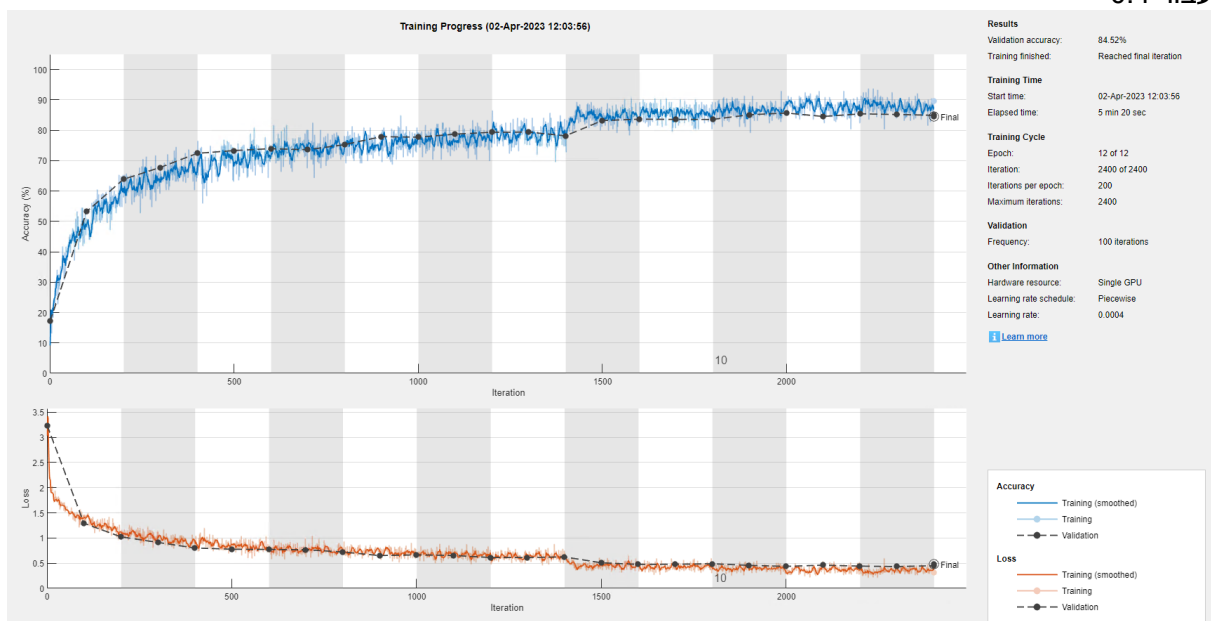
=====

|=====

Validation error: 14.06%

אימון הרשת עם 10 מחלקות היה מוצלח מאשר עם 5 מחלקות.
 הרצנו פעמיים נוספות עם ערכי LearnRateDropFactor שונים:

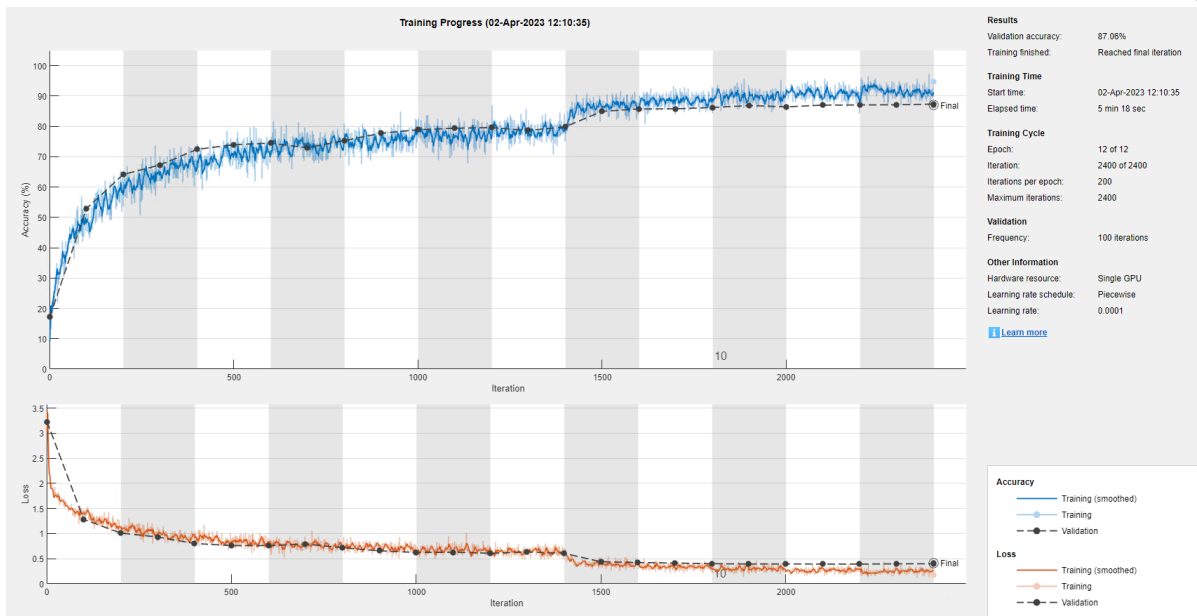
עבור 0.4



	0.44		0.30		85.65%		88.00%		2000		10	
	0.46		0.53		84.53%		83.60%		2100		11	
	0.44		0.45		85.39%		82.80%		2200		11	
	0.43		0.31		85.19%		87.20%		2300		12	
	0.45		0.32		84.91%		89.60%		2400		12	
	0.45		0.32		84.91%		89.60%		2400		12	

Validation error: 15.48%

עבור 0.1



	0.40		0.22		86.44%		92.80%		2000		10	
	0.39		0.29		86.99%		89.20%		2100		11	
	0.39		0.33		87.07%		87.60%		2200		11	
	0.40		0.28		87.11%		90.00%		2300		12	
	0.40		0.18		87.34%		94.80%		2400		12	
	0.40		0.18		87.34%		94.80%		2400		12	

Validation error: 12.94%

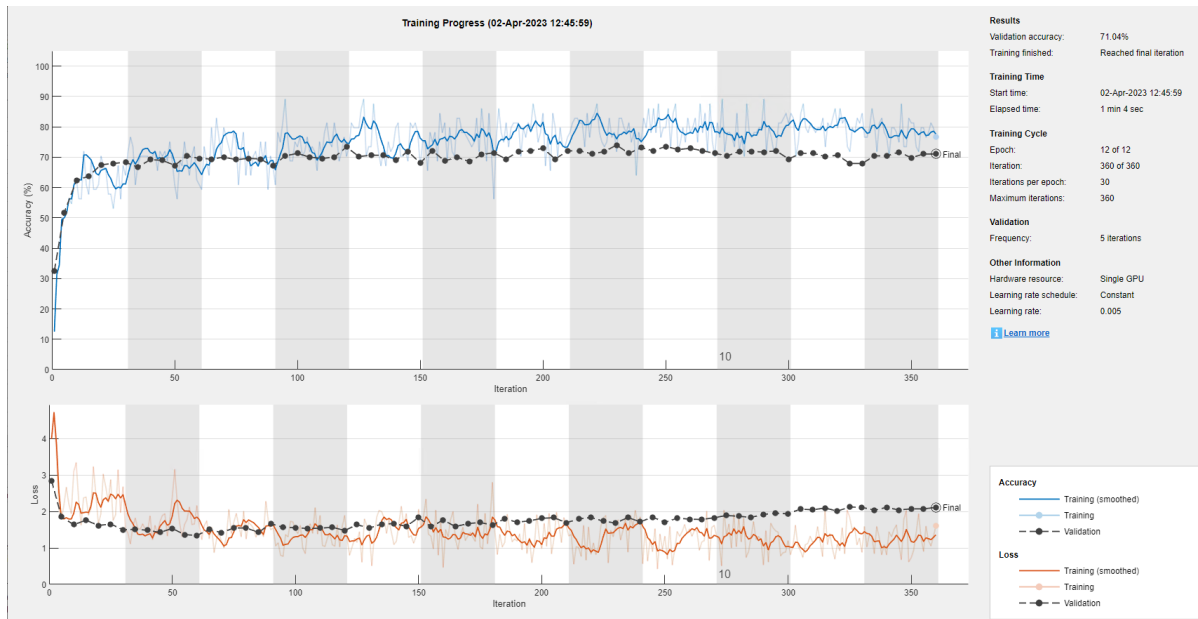
התוצאה עבור 0.1 טובה בהרבה ולכן נישאר עם פרמטר זה.

חלק ב'

1. השיטה שימושית כאשר בידינו מגוון מצומצם יחסית של דוגמאות, בכמות שאינה מספיקה לאימון הולם של הרשת מאפס. לכן, ניתן להסתמך על דמיון צפוי של השכבות הראשונות בין הרשת הרצויה לרשת קיימת אחרת ומאומנת. היתרונות הם הגעה לרמה גבוהה של דיוק על סמך דוגמאות מועטות, וזמן למידה קצר יותר.

2. ב transfer-learning אנחנו מצפים שהשגיאה תקטן מהר יחסית בתחילת הריצה מכיוון שחלק גדול מהרשת כבר מאומן. בנוסף לכך, בגלל שמאמנים מספר קטן יחסית של שכבות (במקרה הזה שכבה בודדת) קשה הרבה יותר להגיע למצב של overfitting ונצפה שיהיה מרחק קטן יותר בין שגיאת הולידציה לשגיאת האימון.

3.

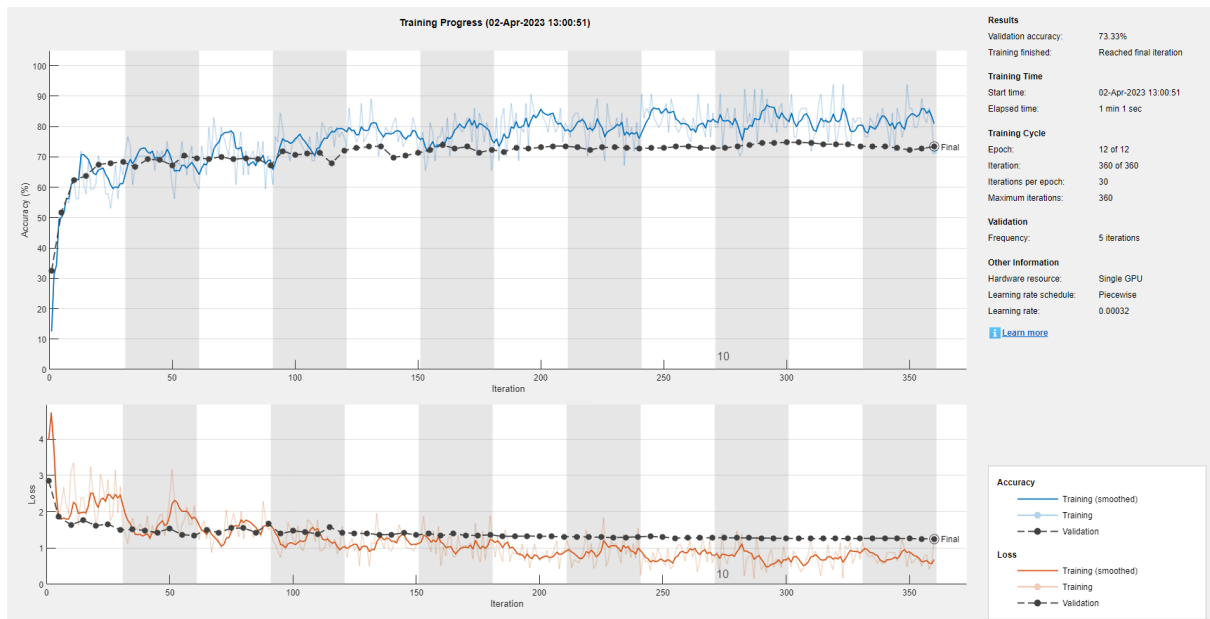


	2.01		1.10		70.63%		79.69%		320		11	
	2.12		0.89		67.92%		84.38%		330		11	
	2.12		0.82		70.42%		76.56%		340		12	
	2.08		0.73		69.79%		81.25%		350		12	
	2.11		1.61		71.04%		76.56%		360		12	
	2.11		1.61		71.04%		76.56%		360		12	

=====|
|=====

Validation error: 28.96%

4. תחילה, הוספנו Scheduler עם פרמטרים $\text{period}=3$, $\text{factor}=0.4$

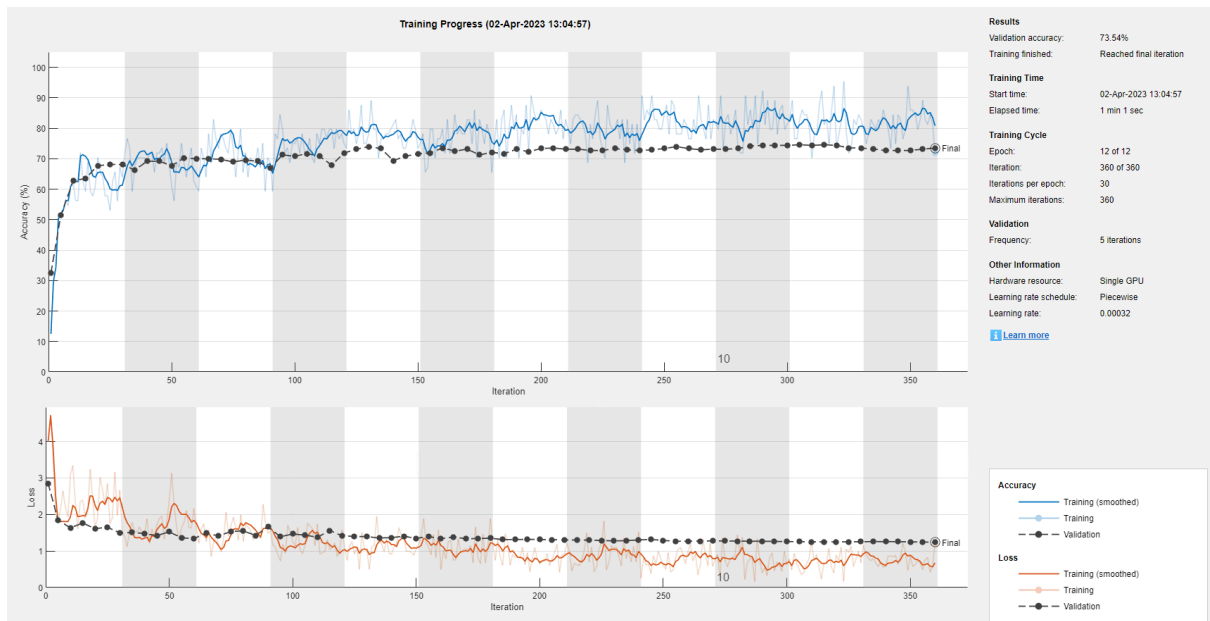


	1.26		0.90		74.17%		73.44%		320		11	
	1.27		0.71		73.54%		82.81%		330		11	
	1.27		0.60		73.33%		82.81%		340		12	
	1.26		0.85		72.29%		85.94%		350		12	
	1.25		1.22		73.33%		71.88%		360		12	
	1.25		1.22		73.33%		71.88%		360		12	

=====

Validation error: 26.67%

כעת, הוספנו regularization בעל פרמטר 0.002:



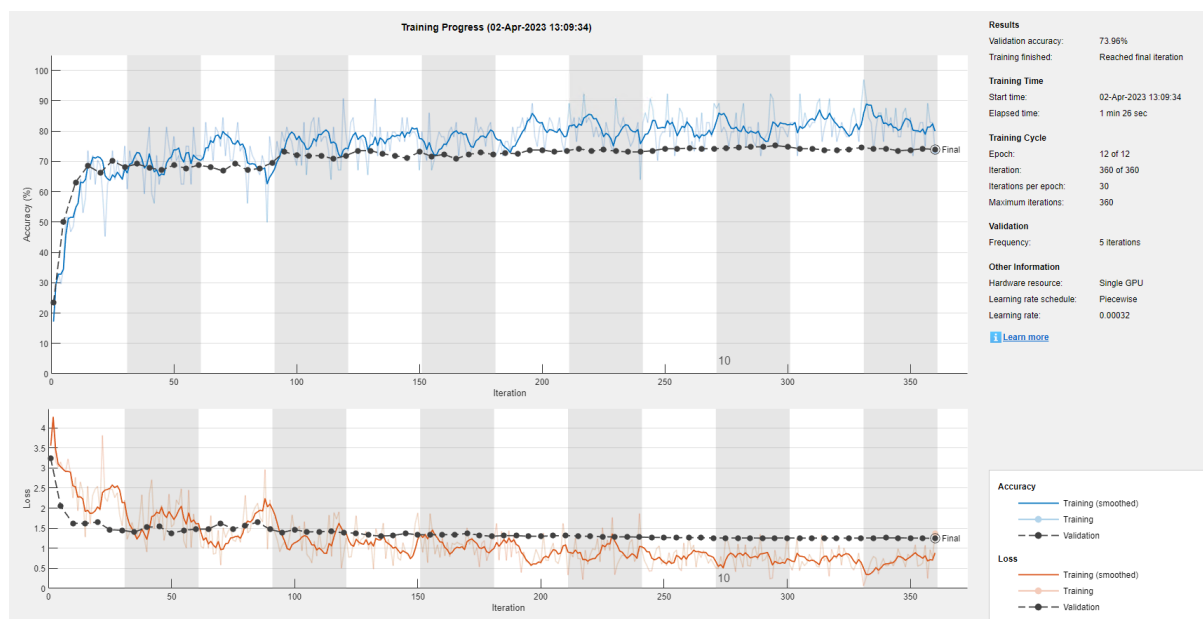
	1.25		0.88		74.38%		73.44%		320		11	
	1.26		0.69		73.54%		82.81%		330		11	
	1.26		0.58		72.71%		82.81%		340		12	
	1.25		0.85		72.71%		85.94%		350		12	
	1.24		1.21		73.54%		71.88%		360		12	
	1.24		1.21		73.54%		71.88%		360		12	

=====

Validation error: 26.46%

כעת ננסה לשלב אוגמנטציות:

שיקוף סביב ציר Y –



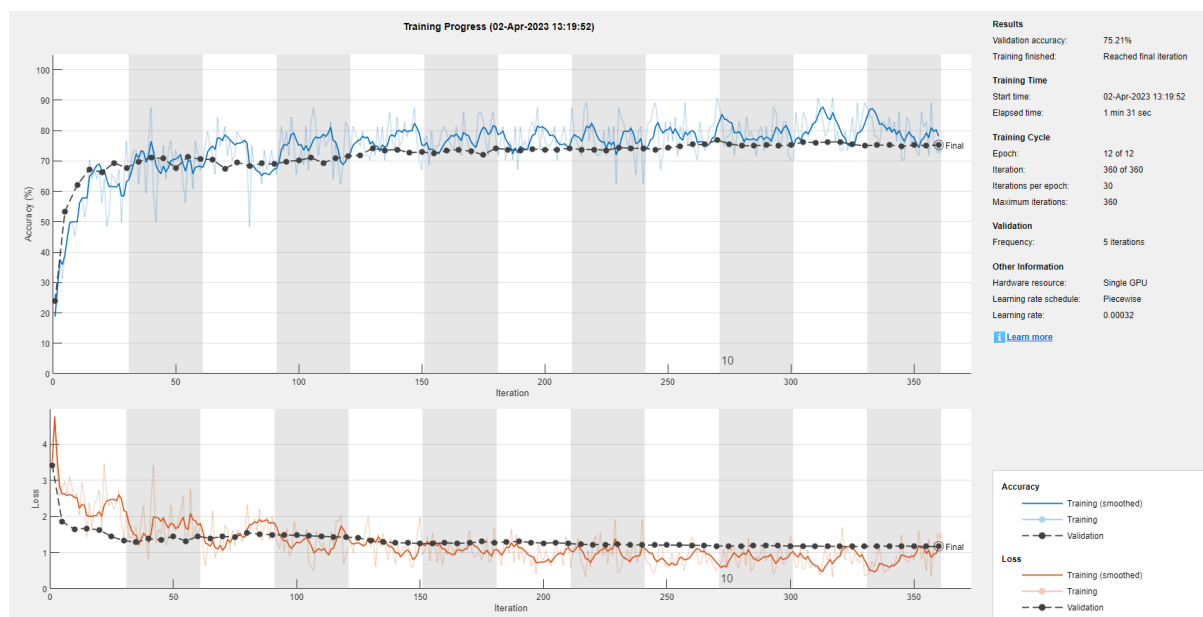
	1.26		0.75		73.75%		76.56%		320		11	
	1.25		0.47		74.58%		89.06%		330		11	
	1.26		0.57		74.17%		87.50%		340		12	
	1.26		0.79		73.75%		84.38%		350		12	
	1.25		1.37		73.96%		73.44%		360		12	
	1.25		1.37		73.96%		73.44%		360		12	

=====

Validation error: 26.04%

ניסינו להוסיף גם scaling אך התוצאות היו פחות טובות ולכן לא השתמשנו לבסוף.

כעת נוסיף הזזות בשני הצירים –

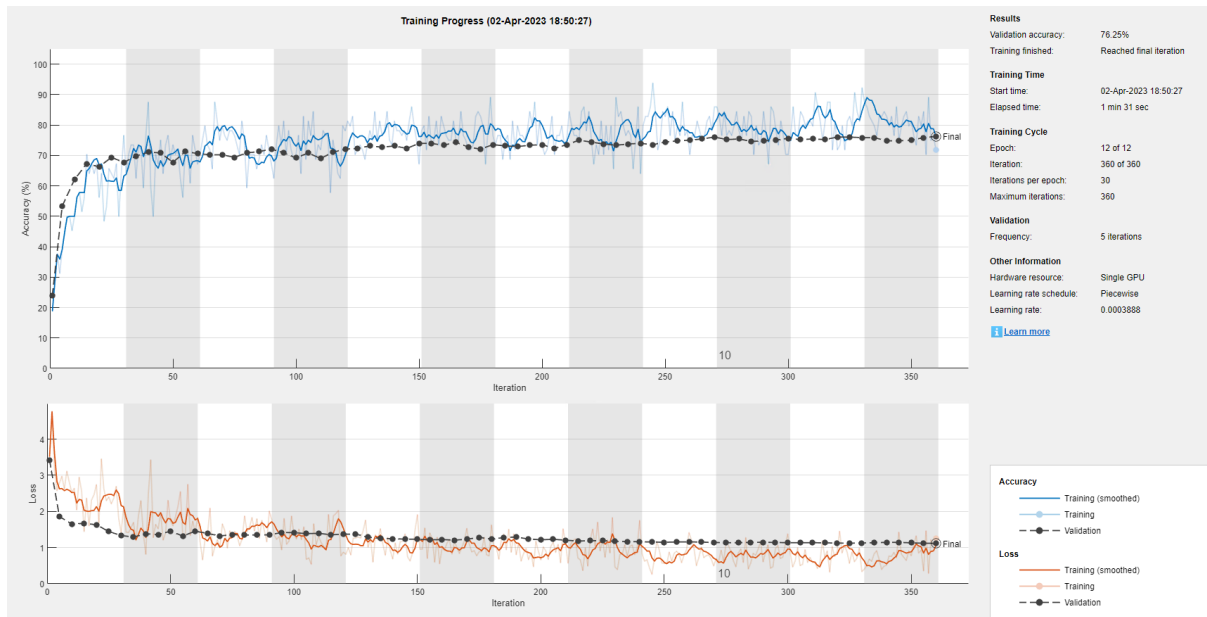


1.17	0.97	76.25%	73.44%	320	11
1.17	0.38	75.00%	89.06%	330	11
1.18	0.53	75.21%	84.38%	340	12
1.18	0.80	75.21%	78.13%	350	12
1.17	1.46	75.21%	73.44%	360	12
1.17	1.46	75.21%	73.44%	360	12

=====

Validation error: 24.79%

לאחר כמה נסיונות נוספים החלטנו לשנות את ה $\text{LearnRateDropFactor}$ ל 0.6 ואת $\text{LearnRateDropPeriod}$ ל 2 ואילו התוצאות:



11	320	71.88%	76.04%	0.89	1.12	
11	330	92.19%	75.83%	0.34	1.13	
12	340	82.81%	74.79%	0.51	1.14	
12	350	81.25%	75.00%	0.78	1.13	
12	360	71.88%	76.25%	1.25	1.12	
12	360	71.88%	76.25%	1.25	1.12	

|=====|

Validation error: 23.75%

לסיכום הגענו לשגיאת ולידציה של 23.75%.