Machine Learning - Assignment 3

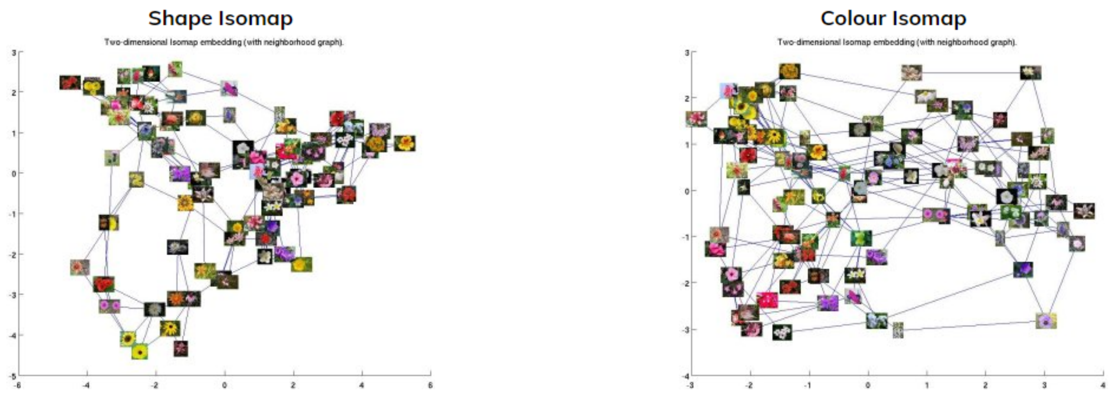
Maor Sagi

Introduction

In this assignment we requested to practice the usage of Convolutional Neural Network (CNN) and Transfer Learning. The task is flower classification by using 2 different pre-trained models.

Data and Preprocessing

The data used for this assignment is the flowers dataset ([LINK]), containing 102 different types and a total number of 8189 images. The maximum number of images in a class is 258, the minimum number is 40 and The average number of images in each class is 80. The flowers are characterized by different colors and different shapes.



An Isomap of categories in the dataset, randomly sampled from the category. Image taken from [here].

All the images resized to 256 x 256 pixels. To improve generalization of the model, I applied several augmentation techniques - rescaling, zooming, rotations, width and heights shifts, horizontal and vertical flipping and shearing (distorted along an axis). The augmentations adjustments generated randomly per image, using Image Data Generator.
In this assignment we requested to split the data to 50% training set, 25% validation and 25% test set. I splitted the data balance split,meaning, first I categorized all images by class and then I sampled from each class randomly.

Architecture and Pretrained Models

After Trying different pre-train models with different configurations, the models performed best are - MobileNet and InceptionV3 using ImageNet weights. I wrapped these models with extra hidden layers, for each model I tested different variations till proceeding to the best results.

1st model - based on Inception V3 pretrained model:

1. Inception V3 layers
2. Global Average Pooling 2D
3. Batch Normalization
4. Dropout (0.3 rate)
5. Dense - 256 and ReLU
6. Batch Normalization
7. Dense - 102 and Softmax

2nd model - based on Mobile Net pretrained model:

1. Mobile Net layers
2. Global Average Pooling 2D
3. Dense - 2048 and ReLU
4. Batch Normalization
5. Dense - 1024 and ReLU
6. Batch Normalization
7. Dense - 1024 and ReLU
8. Batch Normalization
9. Dropout (0.2 rate)
10. Dense - 512 and ReLU
11. Dropout (0.2 rate)
12. Batch Normalization
13. Dropout (0.1 rate)
14. Dense - 256 and ReLU
15. Batch Normalization
16. Dense - 102 and Softmax

Both models compiled with Adam optimizer, learning rate of 0.001. The loss function was defined as categorical cross entropy.

First The models trained while all the pretrained models' weights were frozen and later I commited fine tuning by unfreezing the weights of all layers, except Batch Normalization who commonly recommended to be kept on inference mode to avoid unexpected performance on the validation or test set, as described in Fahdi Kanavati et al. (2021). Later I train the model for another session. In the second train, the models compiled with a smaller learning rate, 1e-4.

Results and Discussion

The 1st and 2nd models experiments configurations:

Initial training:

1. Frozen weights for the pretrained model layers.
2. 30 epochs.
3. Best model checkpoint callback, monitored by minimum validation loss.
4. Split of train-test was according to the given split, otherwise by stratified split while test size is 20%.
5. Reduce learning rate callback when validation loss stops improving with patience factor of 2 epochs.

Secondary training:

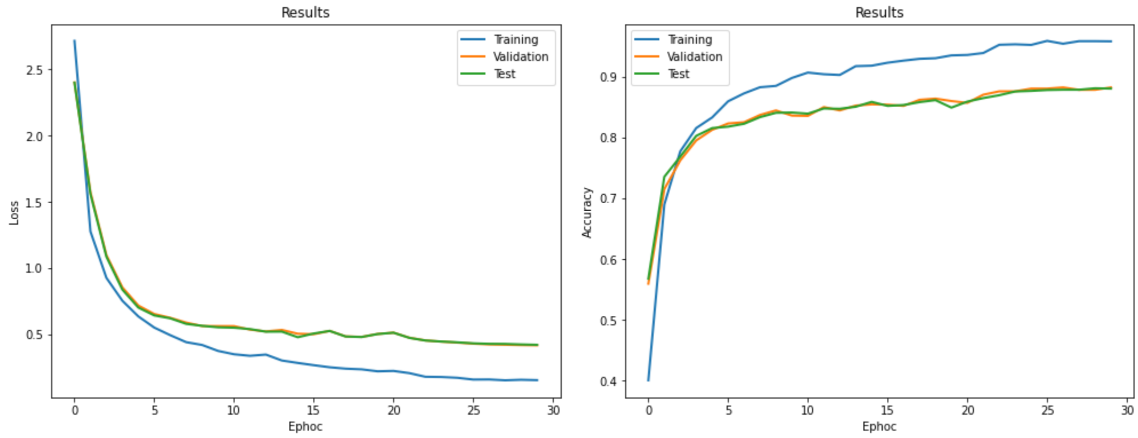
1. Unfreeze frozen weights except Batch Normalization layers.
2. 60 epochs.
3. Best model checkpoint callback, monitored by minimum validation loss.
4. Early stopping callback, monitored by minimum validation loss with patience factor of 4 epochs.
5. Reduce learning rate callback when validation loss stops improving with patience factor of 2 epochs.

I will discuss the results for each experiment, I splitted the dataset randomly as described above. I executed the experiment with the above configuration twice for each architecture, 4 executions in total, 4 different data separations. I will discuss the results as presented in the figures, the detailed results can be found on the appendices, and attached to the submission file together with the exact data splits.

The initial epoch in the secondary training is defined to be the last result on the initial training, that is explaining the starting point in the secondary loss and accuracy figures.

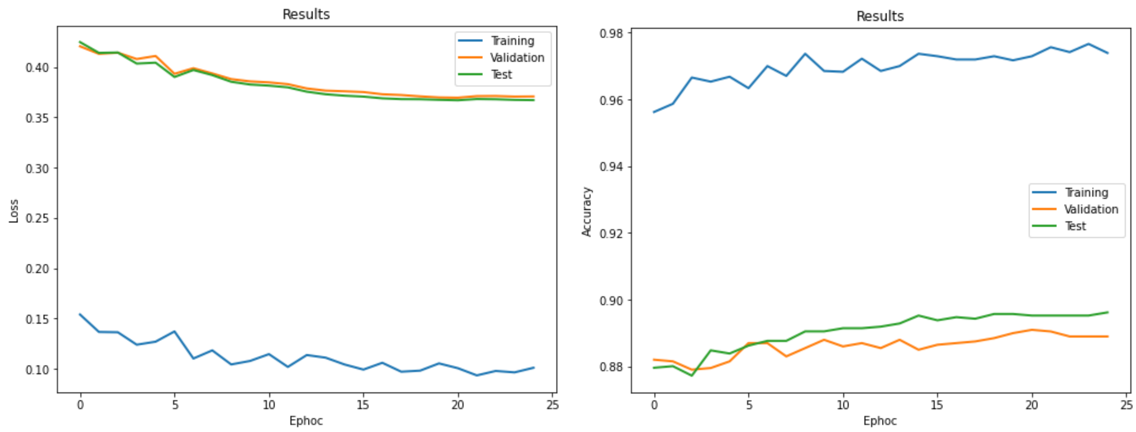
1st experiment - based on Inception V3 pretrained model:

<u>1st execution:</u>



loss: 0.4216
accuracy: 0.8801

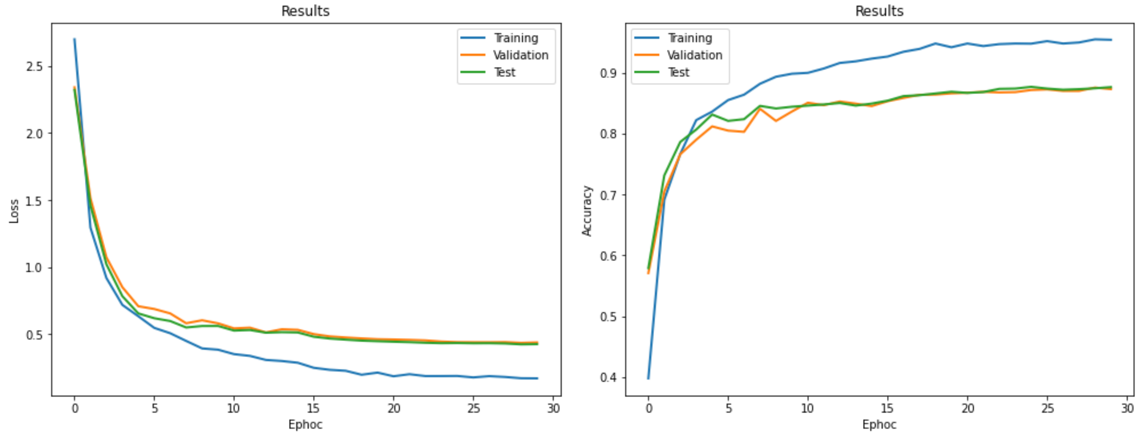
<u>2nd execution:</u>



loss: 0.3672
accuracy: 0.8962

2nd experiment - based on Inception V3 pretrained model:

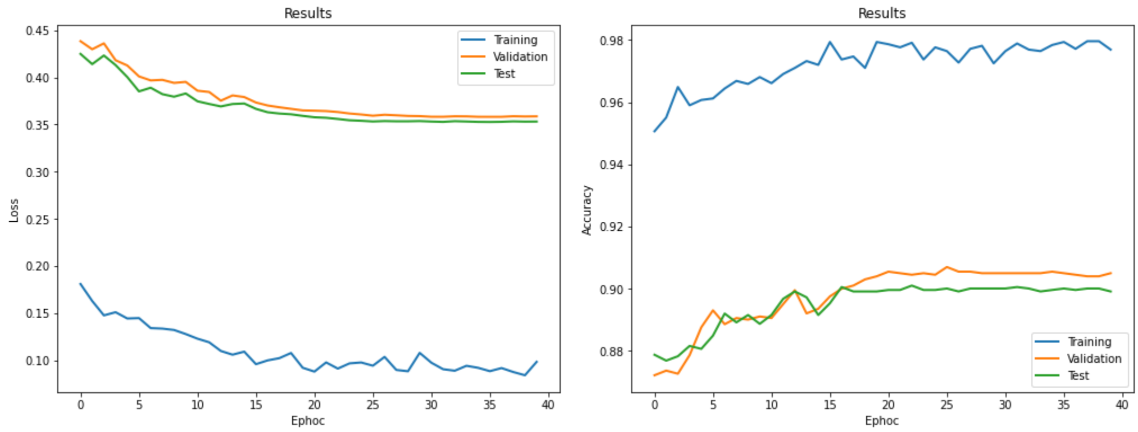
<u>1st execution:</u>



loss: 0.4258

accuracy: 0.8763

<u>2nd execution:</u>

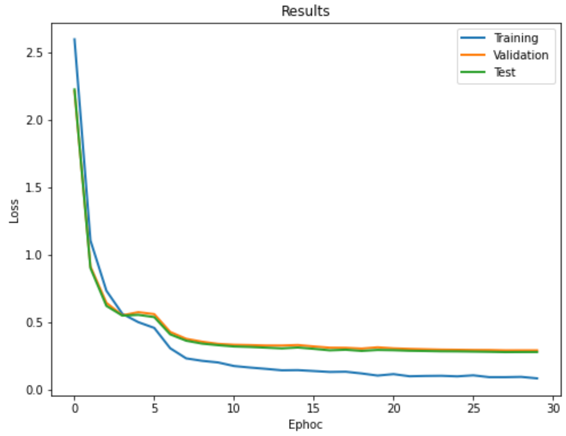


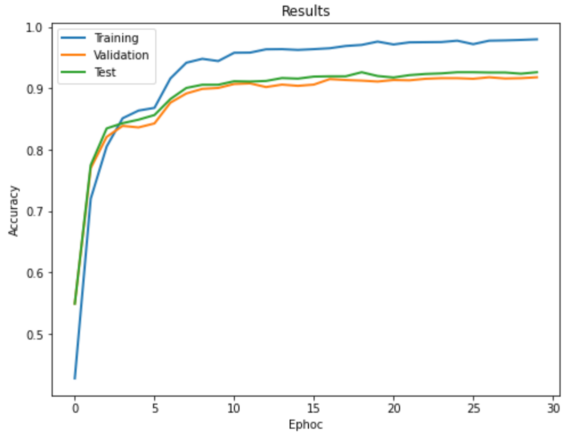
loss: 0.3531

accuracy: 0.8991

3rd experiment - based on Mobile Net pretrained model:

1st execution:

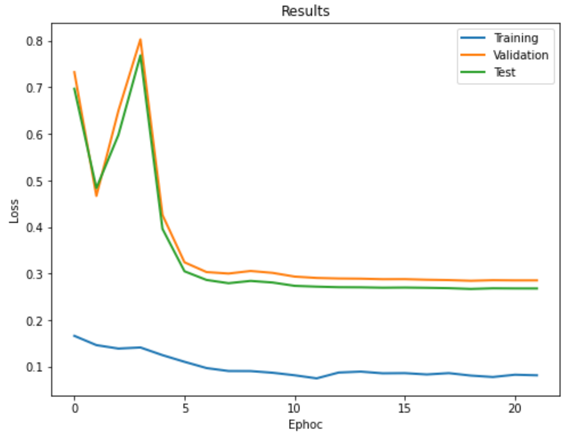


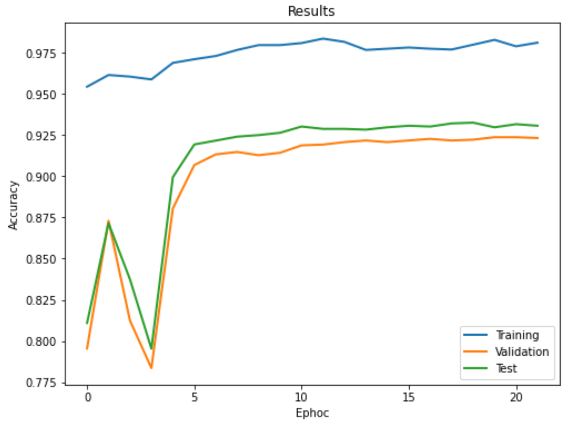


loss: 0.2782

accuracy: 0.9261

2nd execution:



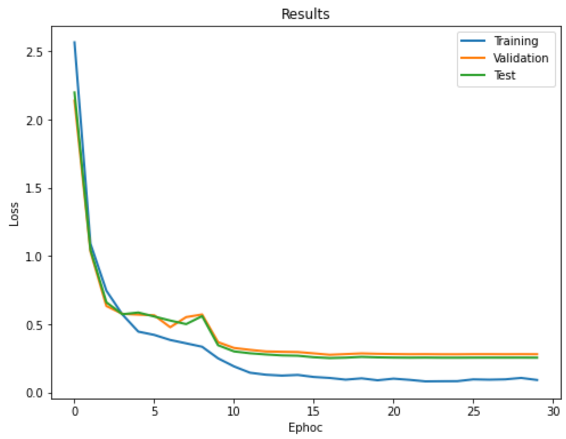


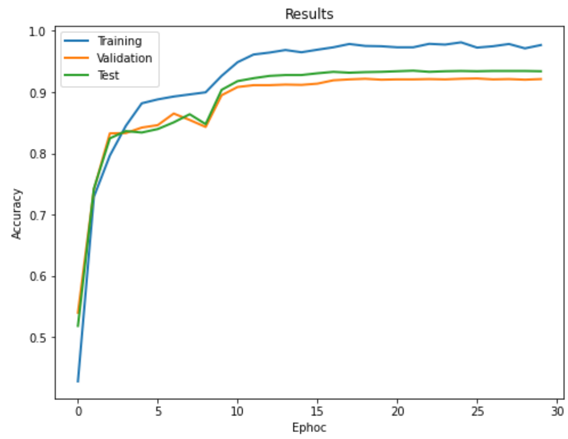
loss: 0.2683

accuracy: 0.9308

4th experiment - based on Mobile Net pretrained model:

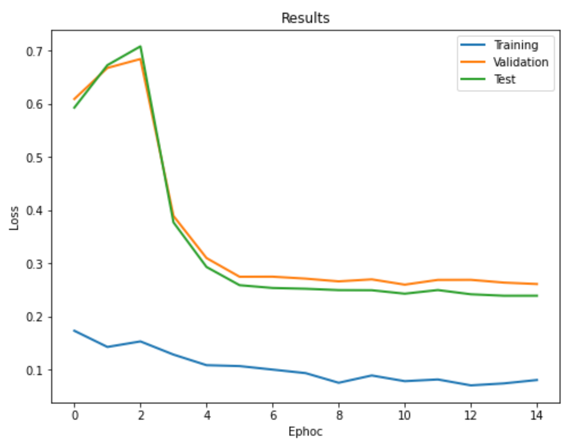
<u>1st execution:</u>

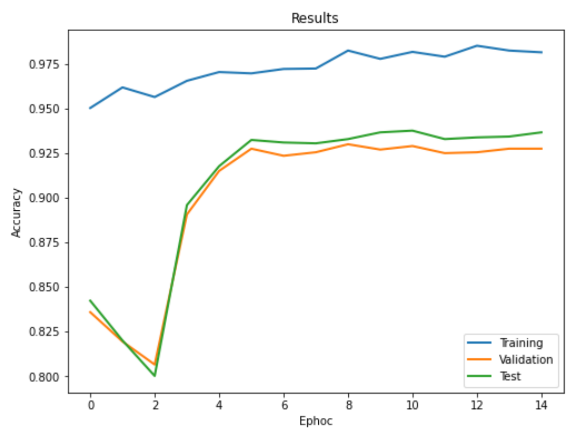




loss: 0.2550
accuracy: 0.9341

<u>2nd execution:</u>





loss: 0.2389
accuracy: 0.9365

We can see for all graphs the validation set and the test set provide similar results. The Mobile Net based model provides best performance of 93.65% accuracy. The Inception V3 accuracy is pretty high as well, 89.9%.

Appendices

Detailed Results:

1st experiment - based on Inception V3 pretrained model:

1st execution:

epoch	accuracy	loss	val_accuracy	val_loss	test_accuracy	test_loss
0	0.400737	2.712055	0.559482	2.399103	0.567773	2.397126
1	0.689435	1.276012	0.714783	1.571921	0.735071	1.557508
2	0.776904	0.926258	0.762071	1.09801	0.768246	1.085515
3	0.815233	0.754019	0.794923	0.85469	0.80237	0.837704
4	0.832678	0.635891	0.812344	0.717681	0.815166	0.702521
5	0.859214	0.551524	0.822797	0.653773	0.817536	0.642973
6	0.872236	0.494393	0.824788	0.625341	0.822275	0.621462
7	0.88231	0.441687	0.836735	0.589079	0.833175	0.579493
8	0.884521	0.420462	0.844201	0.563071	0.840284	0.564843
9	0.897543	0.375722	0.835739	0.562691	0.840758	0.553542
10	0.906388	0.350401	0.835241	0.562583	0.838863	0.550453
11	0.903686	0.338832	0.849676	0.536213	0.847393	0.540068
12	0.902457	0.348001	0.844201	0.522404	0.846919	0.520517
13	0.916953	0.303356	0.852165	0.533768	0.850237	0.521175
14	0.91769	0.285041	0.854156	0.50505	0.858294	0.478574

15	0.922604	0.26818	0.853659	0.501664	0.851659	0.506993
16	0.926044	0.252452	0.851668	0.526492	0.853081	0.526229
17	0.928993	0.241928	0.861623	0.485115	0.85782	0.484006
18	0.929975	0.236808	0.863614	0.481055	0.861137	0.480294
19	0.934644	0.221979	0.859632	0.503469	0.848815	0.502795
20	0.935381	0.224885	0.856645	0.51304	0.858768	0.512957
21	0.938575	0.208199	0.870085	0.475029	0.864455	0.473584
22	0.952088	0.180675	0.87556	0.454291	0.869194	0.454085
23	0.952826	0.179105	0.87556	0.44556	0.875355	0.446407
24	0.951843	0.173241	0.88004	0.438484	0.876303	0.440486
25	0.958477	0.159819	0.88004	0.430414	0.877725	0.432749
26	0.953808	0.16055	0.882031	0.424478	0.878199	0.428995
27	0.957985	0.154252	0.878049	0.421957	0.878199	0.428403
28	0.957985	0.158308	0.878049	0.419439	0.880569	0.424522
29	0.95774	0.155464	0.882031	0.417957	0.880095	0.421577

2nd execution:

epoch	accuracy	loss	val_accuracy	val_loss	test_accuracy	test_loss
0	0.956265	0.15394	0.882031	0.420655	0.879621	0.424842
1	0.958722	0.136614	0.881533	0.413162	0.880095	0.414122
2	0.966585	0.136309	0.879044	0.414334	0.877251	0.414468
3	0.965356	0.123929	0.879542	0.407992	0.884834	0.403501

4	0.96683	0.126935	0.881533	0.411045	0.883886	0.404393
5	0.963391	0.137124	0.887008	0.393333	0.886256	0.390175
6	0.970025	0.110178	0.887008	0.39889	0.887678	0.397109
7	0.967076	0.118288	0.883026	0.393698	0.887678	0.392049
8	0.97371	0.104289	0.885515	0.388063	0.890521	0.385415
9	0.96855	0.107782	0.888004	0.385784	0.890521	0.382664
10	0.968305	0.114551	0.886013	0.384824	0.891469	0.38157
11	0.972236	0.101831	0.887008	0.383014	0.891469	0.379855
12	0.96855	0.113642	0.885515	0.37877	0.891943	0.37555
13	0.970025	0.111024	0.888004	0.376595	0.892891	0.372999
14	0.97371	0.104242	0.885017	0.375935	0.895261	0.371557
15	0.972973	0.099184	0.886511	0.375186	0.893839	0.370643
16	0.97199	0.10594	0.887008	0.372975	0.894787	0.368932
17	0.97199	0.097085	0.887506	0.372205	0.894313	0.36815
18	0.972973	0.098124	0.888502	0.370892	0.895735	0.368038
19	0.971744	0.105324	0.889995	0.3698	0.895735	0.367447
20	0.972973	0.100557	0.890991	0.369551	0.895261	0.36701
21	0.975676	0.093401	0.890493	0.371117	0.895261	0.368245
22	0.974202	0.097818	0.889	0.371239	0.895261	0.368019
23	0.976658	0.0964	0.889	0.370666	0.895261	0.367392
24	0.973956	0.100991	0.889	0.370798	0.896209	0.367153

2nd experiment - based on Inception V3 pretrained model:

1st execution:

epoch	accuracy	loss	val_accuracy	val_loss	test_accuracy	test_loss
0	0.398034	2.697731	0.570931	2.340134	0.579147	2.32041
1	0.6914	1.294044	0.704828	1.513701	0.73128	1.460068
2	0.766585	0.91877	0.766053	1.07297	0.786256	1.020844
3	0.822113	0.718556	0.789945	0.850488	0.806635	0.783481
4	0.836118	0.633733	0.811847	0.70824	0.83128	0.654625
5	0.855037	0.547039	0.804878	0.687777	0.820853	0.618884
6	0.863882	0.507135	0.802887	0.654819	0.823697	0.598013
7	0.881818	0.449723	0.840717	0.581431	0.845498	0.549997
8	0.893366	0.393901	0.820806	0.603586	0.841232	0.560856
9	0.89828	0.384419	0.836237	0.580729	0.844076	0.561233
10	0.899754	0.351368	0.850672	0.543192	0.845972	0.526958
11	0.90688	0.338053	0.84669	0.548566	0.847867	0.531183
12	0.915971	0.308009	0.852663	0.513298	0.850237	0.511224
13	0.918673	0.300018	0.849179	0.536685	0.845972	0.51511
14	0.923096	0.287055	0.845197	0.532392	0.849289	0.512903
15	0.926536	0.249431	0.853161	0.500541	0.854028	0.481077
16	0.934398	0.234458	0.858636	0.48351	0.861611	0.467439
17	0.939066	0.227147	0.863614	0.475228	0.863033	0.459246

18	0.947912	0.197722	0.864111	0.468399	0.865877	0.45201
19	0.941769	0.213602	0.866103	0.462511	0.86872	0.44769
20	0.947912	0.186321	0.867098	0.460409	0.866825	0.443852
21	0.943735	0.201741	0.868591	0.4574	0.868246	0.4406
22	0.946929	0.187815	0.867596	0.453558	0.87346	0.435994
23	0.947912	0.187961	0.868094	0.444896	0.873934	0.433165
24	0.947666	0.18867	0.871578	0.440665	0.876777	0.43422
25	0.951843	0.17773	0.872573	0.440027	0.873934	0.43238
26	0.947912	0.187371	0.870085	0.439919	0.872038	0.432873
27	0.949631	0.181064	0.870085	0.440817	0.872986	0.43066
28	0.954791	0.171585	0.87556	0.435462	0.874408	0.424191
29	0.954054	0.170666	0.873071	0.438438	0.876303	0.425752

2nd execution:

epoch	accuracy	loss	val_accuracy	val_loss	test_accuracy	test_loss
0	0.950614	0.180745	0.872076	0.438356	0.878673	0.424863
1	0.955037	0.162729	0.873569	0.429798	0.876777	0.414013
2	0.964865	0.147428	0.872573	0.436069	0.878199	0.423203
3	0.958968	0.150808	0.878547	0.418137	0.881517	0.413023
4	0.960688	0.144146	0.887506	0.412594	0.880569	0.400308
5	0.961179	0.144569	0.892982	0.400996	0.884834	0.385117
6	0.964373	0.133944	0.888502	0.396743	0.891943	0.388983

7	0.96683	0.133429	0.890493	0.397336	0.8891	0.382167
8	0.965848	0.131953	0.889995	0.394107	0.891469	0.379447
9	0.968059	0.127609	0.890991	0.395203	0.888626	0.382902
10	0.966093	0.122718	0.890493	0.385839	0.891469	0.374617
11	0.969042	0.118892	0.894973	0.384485	0.896682	0.371746
12	0.971007	0.109666	0.899452	0.375245	0.899052	0.369128
13	0.973219	0.105736	0.891986	0.380811	0.897156	0.371704
14	0.97199	0.10897	0.893479	0.379089	0.891469	0.372226
15	0.979361	0.095657	0.897461	0.373304	0.895261	0.36658
16	0.97371	0.099673	0.89995	0.370123	0.900474	0.362973
17	0.974693	0.102036	0.900946	0.368233	0.899052	0.361586
18	0.971007	0.107542	0.902937	0.366593	0.899052	0.360801
19	0.979361	0.091848	0.903932	0.364927	0.899052	0.359124
20	0.978624	0.087784	0.905426	0.364628	0.899526	0.35766
21	0.977641	0.097547	0.904928	0.364219	0.899526	0.357098
22	0.979115	0.090945	0.90443	0.363205	0.900948	0.355847
23	0.97371	0.096511	0.904928	0.361618	0.899526	0.354448
24	0.977641	0.097472	0.90443	0.360587	0.899526	0.353919
25	0.976413	0.094028	0.906919	0.359291	0.9	0.353168
26	0.972727	0.103399	0.905426	0.360301	0.899052	0.35362
27	0.97715	0.089547	0.905426	0.359721	0.9	0.353362
28	0.978133	0.088137	0.904928	0.359051	0.9	0.353358
29	0.972482	0.107668	0.904928	0.358862	0.9	0.353613

30	0.976413	0.09721	0.904928	0.358191	0.9	0.353089
31	0.97887	0.090313	0.904928	0.358169	0.900474	0.352732
32	0.976904	0.088705	0.904928	0.358664	0.9	0.353499
33	0.976413	0.093961	0.904928	0.358583	0.899052	0.35313
34	0.978378	0.091782	0.905426	0.358147	0.899526	0.352738
35	0.979361	0.088228	0.904928	0.358121	0.9	0.352641
36	0.97715	0.091532	0.90443	0.358134	0.899526	0.352809
37	0.979607	0.087371	0.903932	0.358716	0.9	0.353281
38	0.979607	0.083903	0.903932	0.35847	0.9	0.353016
39	0.976904	0.09816	0.904928	0.358605	0.899052	0.353091

3rd experiment - based on Mobile Net pretrained model:

1st execution:

epoch	accuracy	loss	val_accuracy	val_loss	test_accuracy	test_loss
0	0.427764	2.594738	0.549527	2.211029	0.549289	2.223706
1	0.720147	1.104411	0.770035	0.911722	0.774882	0.900108
2	0.80516	0.732204	0.820806	0.642602	0.834597	0.619907
3	0.851106	0.561439	0.838726	0.549873	0.843128	0.548008
4	0.863636	0.499356	0.836237	0.573468	0.848815	0.553569
5	0.868059	0.456784	0.842708	0.558858	0.856398	0.53687
6	0.915971	0.306097	0.876556	0.426757	0.882464	0.40922

7	0.941523	0.231155	0.891488	0.376043	0.900474	0.362085
8	0.947912	0.21297	0.898955	0.354533	0.905687	0.340478
9	0.944226	0.200864	0.900448	0.338591	0.905687	0.328887
10	0.95774	0.174518	0.906919	0.331984	0.911374	0.319302
11	0.957985	0.163201	0.907914	0.329612	0.9109	0.31576
12	0.963391	0.153504	0.901941	0.326451	0.911848	0.310473
13	0.963636	0.142722	0.905923	0.326242	0.916588	0.30509
14	0.962408	0.14394	0.903932	0.330031	0.91564	0.311547
15	0.963636	0.137386	0.905923	0.319529	0.918957	0.301883
16	0.965111	0.130488	0.914883	0.310144	0.919431	0.290691
17	0.968796	0.132232	0.91339	0.309271	0.919431	0.295132
18	0.970516	0.119454	0.912394	0.303623	0.926066	0.286247
19	0.975921	0.103723	0.910901	0.313061	0.919905	0.293547
20	0.971253	0.114423	0.91339	0.305364	0.917536	0.291587
21	0.974693	0.098606	0.912892	0.301413	0.921327	0.287478
22	0.974939	0.100967	0.915381	0.29853	0.923223	0.285722
23	0.975184	0.102131	0.916376	0.295765	0.924171	0.28357
24	0.977396	0.097898	0.916376	0.294649	0.926066	0.282866
25	0.971744	0.105573	0.915381	0.293086	0.926066	0.281197
26	0.977396	0.092013	0.91787	0.292731	0.925592	0.279634
27	0.977887	0.09191	0.915879	0.290309	0.925592	0.277458
28	0.978624	0.094112	0.916376	0.290493	0.923697	0.277892
29	0.979607	0.083019	0.91787	0.290392	0.926066	0.278156

2nd execution:

epoch	accuracy	loss	val_accuracy	val_loss	test_accuracy	test_loss
0	0.954545	0.166609	0.795421	0.732725	0.8109	0.696968
1	0.961671	0.146709	0.873071	0.466926	0.871564	0.48382
2	0.960688	0.139322	0.812344	0.649756	0.837441	0.597874
3	0.958968	0.141669	0.783474	0.80308	0.795261	0.768735
4	0.969042	0.125206	0.880538	0.426077	0.899526	0.396283
5	0.971253	0.110899	0.906919	0.324464	0.919431	0.305203
6	0.973219	0.097459	0.91339	0.303493	0.921801	0.286592
7	0.976904	0.091137	0.914883	0.300341	0.924171	0.279597
8	0.979853	0.091004	0.912892	0.305845	0.925119	0.284547
9	0.979853	0.08742	0.914385	0.301803	0.92654	0.281088
10	0.981081	0.081998	0.918865	0.293805	0.930332	0.273967
11	0.983784	0.075356	0.919363	0.290893	0.92891	0.272229
12	0.981818	0.087872	0.920856	0.289748	0.92891	0.271024
13	0.976904	0.08986	0.921852	0.289327	0.928436	0.270781
14	0.977641	0.086209	0.920856	0.288218	0.929858	0.26996
15	0.978378	0.086572	0.921852	0.288327	0.930806	0.270317
16	0.977641	0.083753	0.922847	0.287045	0.930332	0.269729
17	0.97715	0.086586	0.921852	0.286293	0.932227	0.268968

18	0.980098	0.081358	0.922349	0.284838	0.932701	0.267314
19	0.983047	0.078286	0.923843	0.286084	0.929858	0.268677
20	0.979115	0.083173	0.923843	0.285705	0.931754	0.268388
21	0.981327	0.081948	0.923345	0.285718	0.930806	0.2683

4th experiment - based on Mobile Net pretrained model:

1st execution:

epoch	accuracy	loss	val_accuracy	val_loss	test_accuracy	test_loss
0	0.42801	2.567291	0.54007	2.14001	0.518483	2.199684
1	0.729238	1.091192	0.741165	1.033818	0.742654	1.045747
2	0.79656	0.745772	0.832753	0.633602	0.824645	0.660998
3	0.844963	0.571905	0.83325	0.576488	0.836493	0.573463
4	0.881818	0.445941	0.84221	0.570823	0.834123	0.586053
5	0.888206	0.421996	0.846192	0.565698	0.83981	0.556919
6	0.892875	0.384817	0.865107	0.47815	0.850711	0.52739
7	0.896315	0.360183	0.854654	0.552471	0.863981	0.501155
8	0.899754	0.335123	0.843206	0.571839	0.847867	0.559652
9	0.926536	0.250141	0.894973	0.368924	0.903791	0.34626
10	0.948894	0.191008	0.908412	0.326267	0.918009	0.301005
11	0.961425	0.144661	0.911399	0.312437	0.922749	0.287304
12	0.964619	0.129996	0.911399	0.300833	0.92654	0.278502

13	0.968796	0.123654	0.912394	0.298248	0.927962	0.271554
14	0.965111	0.128485	0.911896	0.29639	0.927962	0.269185
15	0.969287	0.11351	0.913888	0.286661	0.930806	0.258198
16	0.973219	0.106336	0.919363	0.276382	0.933175	0.252221
17	0.978624	0.093853	0.920856	0.281348	0.931754	0.255086
18	0.97543	0.103523	0.921852	0.286381	0.932701	0.26084
19	0.974939	0.089214	0.920358	0.283408	0.933175	0.257314
20	0.973219	0.101287	0.920856	0.281725	0.934123	0.255629
21	0.973219	0.092435	0.920856	0.28056	0.935071	0.254871
22	0.97887	0.081278	0.921354	0.280984	0.933175	0.255506
23	0.977641	0.082202	0.920856	0.280139	0.934123	0.25464
24	0.981327	0.082743	0.921852	0.28002	0.934597	0.254617
25	0.972727	0.095937	0.922349	0.280911	0.934123	0.255149
26	0.974939	0.093663	0.920856	0.280933	0.934597	0.255342
27	0.978624	0.096065	0.921354	0.280483	0.934597	0.255211
28	0.971499	0.106588	0.920358	0.280906	0.934597	0.255345
29	0.976904	0.091402	0.921354	0.280541	0.934123	0.254973

2nd execution:

epoch	accuracy	loss	val_accuracy	val_loss	test_accuracy	test_loss
0	0.950123	0.172917	0.835739	0.609395	0.84218	0.593124
1	0.961671	0.142407	0.819313	0.667908	0.819905	0.673135

2	0.956265	0.152871	0.806371	0.684719	0.8	0.708566
3	0.965356	0.12806	0.890493	0.389155	0.895735	0.377055
4	0.97027	0.1081	0.914883	0.309831	0.917536	0.293023
5	0.969533	0.106403	0.927327	0.274624	0.932227	0.258735
6	0.97199	0.099774	0.923345	0.27476	0.930806	0.253478
7	0.972236	0.093134	0.925336	0.271096	0.930332	0.252017
8	0.98231	0.074955	0.929816	0.265922	0.932701	0.249452
9	0.977641	0.088721	0.926829	0.26971	0.936493	0.249249
10	0.981573	0.077985	0.92882	0.259752	0.937441	0.24273
11	0.97887	0.081218	0.924838	0.268699	0.932701	0.249586
12	0.985012	0.070082	0.925336	0.268831	0.933649	0.241635
13	0.98231	0.073777	0.927327	0.263614	0.934123	0.238867
14	0.981327	0.080186	0.927327	0.260941	0.936493	0.23891