Deep Learning Lab 3
Convolutional Neural Networks

Garben Tanghe
Olivier Van den Nest
Group 03

March 6, 2020

We use 5000 samples as the validation set, so that it is exactly $10\,\%$ of the complete training set. We also used 16384 samples for the smaller training set, so that the number of samples in the training set is a power of 2.

1 SimpleNet

Our first powerful model is based on SimpleNet [1]. We changed the kernel initialization scheme for 2D convolutions to 'He Uniform' instead of 'Glorot Normal', because that works well with ReLU activations. We used the variant without dropout to see whether or not the model is powerful enough.

The best learning rate for the Adam optimizer and a batch size of 256 (exactly 64 batches) is found to be 0.001. The validation curve for training converges just before 25 epochs. This can be seen in Figure 1. There can be seen that the model is now complex enough and is capable of capturing the ground truth very well. The training set accuracy is $100\,\%$ and the train loss is 0.0023, while the validation set accuracy is $58\,\%$ and the validation loss is 2.3360.

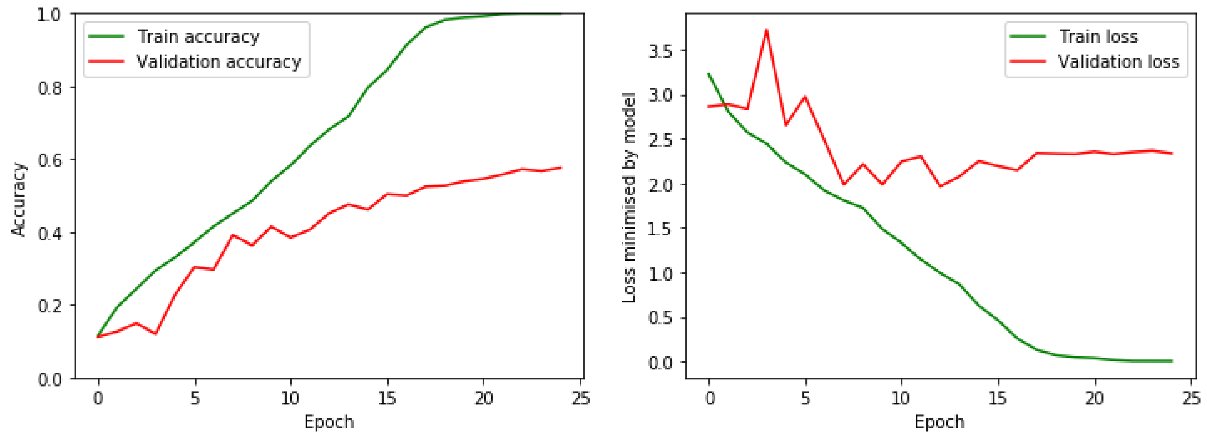


Figure 1: Validation curves for the SimpleNet variant

Now that we have low bias, regularization is needed to reduce the variance. We tried out different regularization techniques, but none of them significantly improved the validation curve on the validation set.

2 ConvNet

Here we decided to go with a much simpler convolutional network. The architecture is the following: A block of 2 convolutional layers and a max pooling layer is repeated 3 times. The first 2 conv layers have 64 output channels and they are doubled after each max pooling layer. All convolutional layers have a receptive field of 3x3, have a He Uniform initialization scheme, make use of Batch Normalization with a momentum of 0.95, and end with ReLU activation. All max pooling layers have a pool size of 2x2 and strides 2. The network ends with flattening the 4x4 feature maps into a vector of length 4096 and a Dense layer with 20 nodes for the classification. This network has only 1.2M parameters instead of 5.5M in the SimpleNet variant.

The best learning rate for the Adam optimizer and a batch size of 256 (exactly 64 batches) is again found to be 0.001. The validation curve for training converges much earlier, after about 10 epochs. This can be seen in Figure 2. There can be seen that the model is still complex enough and is capable of capturing the ground truth. The training set accuracy is $100\,\%$ and the train loss is 0.0031, while the validation set accuracy is $54\,\%$ and the validation loss is 1.8873.

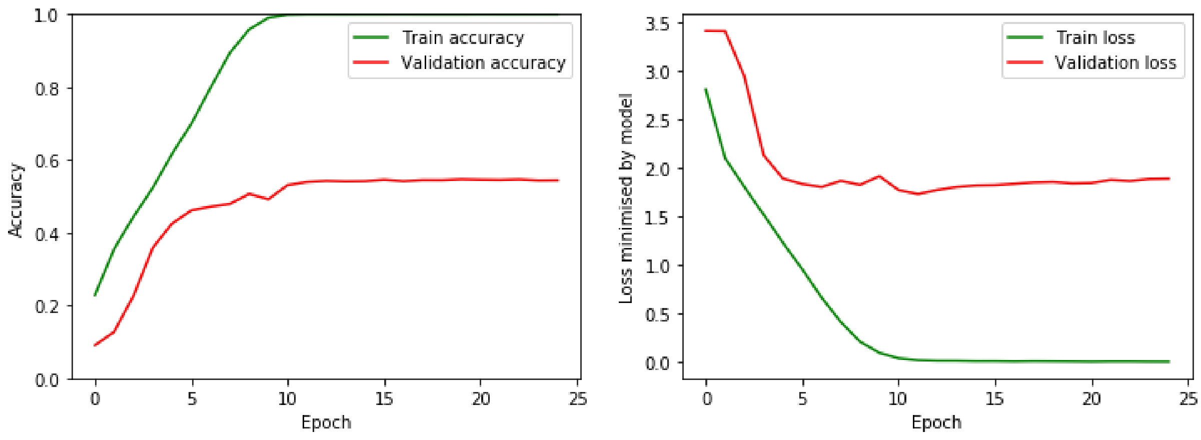


Figure 2: Validation curves for ConvNet

We regularized the smaller model in multiple ways. First of all, $20\,\%$ Dropout is added before the final, Dense layer. Secondly, Early Stopping is added on the validation accuracy with a patience of 5. A third way of achieving regularization is added by putting a constraint on the weights of the convolutional kernels. We make use of MaxNorm with its default parameters. And last but not least, there is made use of Data Augmentation. The image data generator will now randomly rotate images with a maximum of 22.5 degrees. Shifting and zooming happens in a range of $20\,\%$. The images can also be flipped horizontally, but not vertically, since for most classes, that would not make sense. Most objects in the scene will always appear up-right.

The validation curves for training the smaller model on the small training set for 50 epochs and learning rate kept at 0.001, is found in Figure 3. From this figure, there can be concluded that the learning rate can be lowered. This should stop the validation curve on the validation set to jump up and down. There can also be seen that the model is still overfitting quite a bit, so even more regularization can be added. The training set accuracy is $97\,\%$, the loss is 0.1111. On the validation set, the accuracy and loss are $61\,\%$ and 1.8118 respectively.

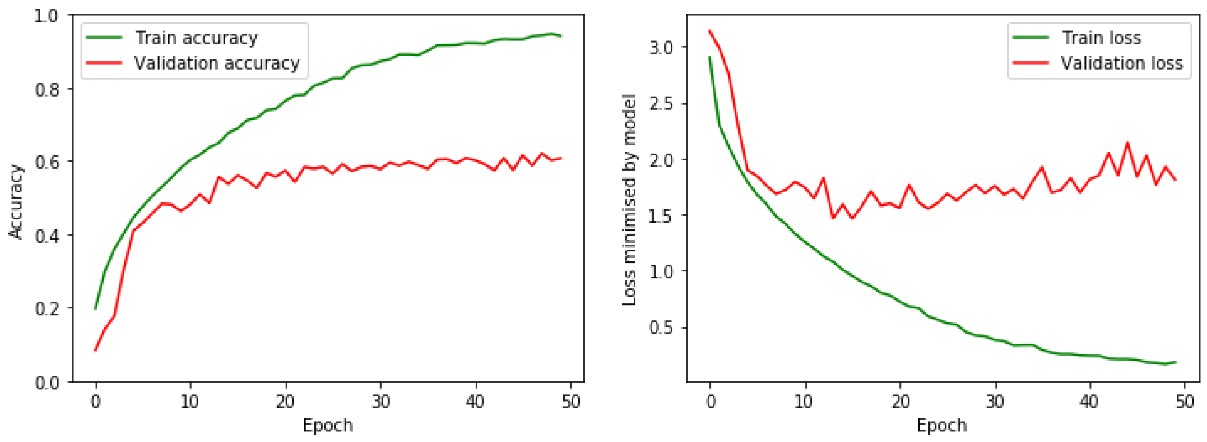


Figure 3: Validation curves of the regularized ConvNet

Now, the model is trained on 45000 training samples with a learning rate of 0.0005, dropout rate of 0.25 and zoom range of 0.25. The resulting curves are found in Figure 4. The training set accuracy is lowered to $89\,\%$ and the loss is 0.3169. The validation set accuracy is now up to $69\,\%$ and the loss is down to 1.1649.

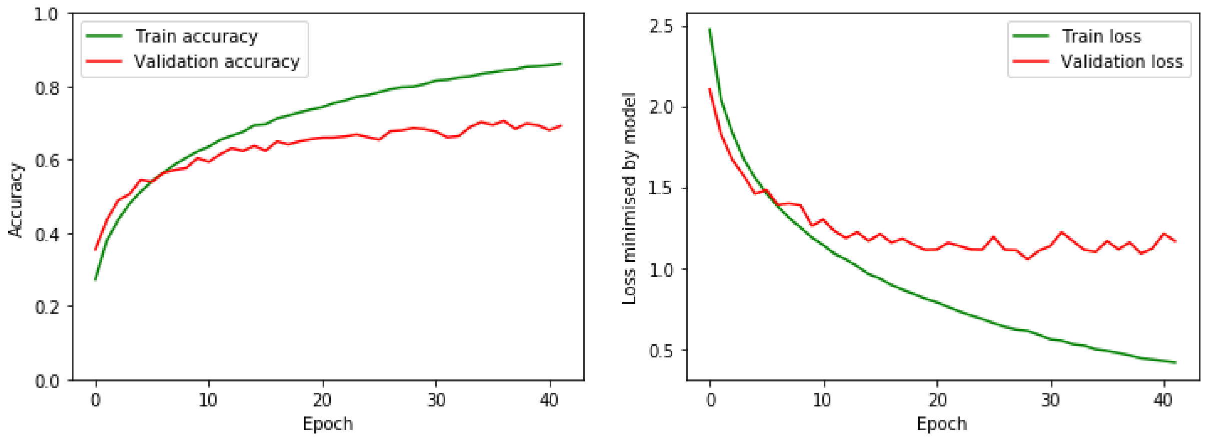


Figure 4: Validation curves for ConvNet with even more regularization

The results for retraining ConvNet one last time on all 50000 samples are seen in Figure 5.

The final model has a total accuracy of $71.31\,\%$ on the test set. The precision, recall and F1 scores for each class can be found in Figure 6. The confusion matrix and its normalized version are provided in Figure 7 an Figure 8 respectively.

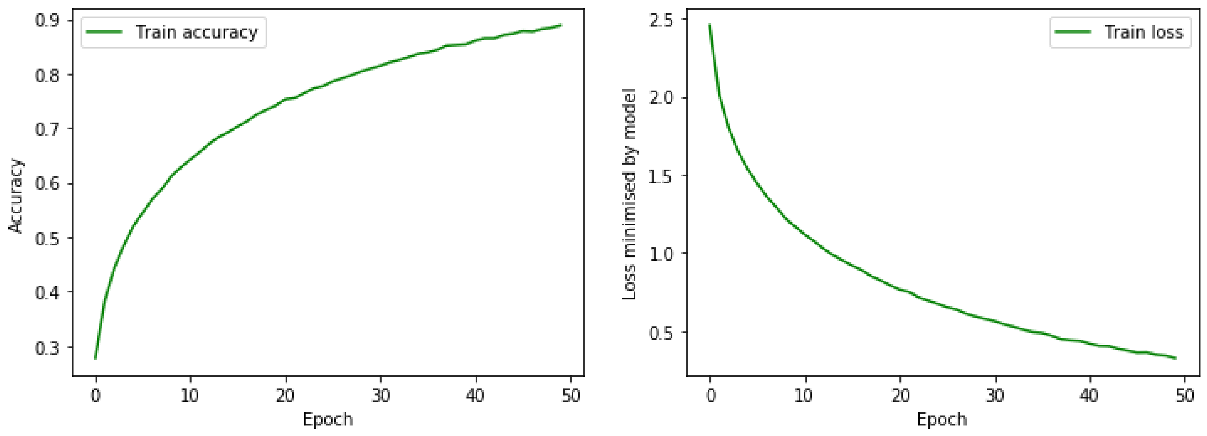


Figure 5: Validation curves for ConvNet, trained on the complete train set

3 ResNet

We also experimented with SkipConnections in a residual neural network. The architecture is the following: First, the input is fed through 1 convolutional layer to get a first tensor with 128 channels. Then, a block of 2 convolutional layers, an addition layer (for the skip connections) and a max pooling layer is repeated 3 times. All convolutional layers have 128 output channels, a receptive field of 3x3, have a He Uniform initialization scheme, make use of Batch Normalization with a momentum of 0.95, and end with ELU activation (vs. ReLU in ConvNet). We make use of MaxNorm with its default parameters in each convolutional layer. All max pooling layers have a pool size of 2x2 and strides 2. Next, the network flattens the 4x4 feature maps into a vector of length 2048. Subsequently, two Dense layers with 512 and 256 nodes. A Dropout layer with dropout rate of 0.25 is applied after the flattened layer. They are also initialized with the 'He Uniform' weights and constrained by MaxNorm with default parameters. Additionally, L2 regularization with a regularization factor of 0.001, as well as a MaxNorm constraint is applied on these layers. Finally, a last Dense layer with 20 nodes and softmax activation ends the network. This network is slightly bigger (around 2.1M parameters).

Now, the model is trained on 45000 training samples with a learning rate of 0.0005, a zoom range of 0.25, and for 50 epochs. The training set accuracy is lowered to $88\,\%$ and the loss is 0.6269. The validation set accuracy is now up to $72\,\%$ and the loss is down to 1.1838.

The validation curves for retraining ResNet on the complete train set for 35 epochs are seen in Figure 9.

ResNet scores $72.16\,\%$ on the test set. The precision, recall and F1 scores for each class can be found in Figure 10. The confusion matrix and its normalized version are provided in Figure 11 an Figure 12 respectively. When comparing the results of ConvNet and ResNet, we can make the following conclusions:

- The predictions for each class of ResNet are slightly more accurate than those of ConvNet. All classes have more correctly predicted samples and fewer wrongly predicted samples.
- The same classes are easy to classify and the 2 networks make similar mistakes.

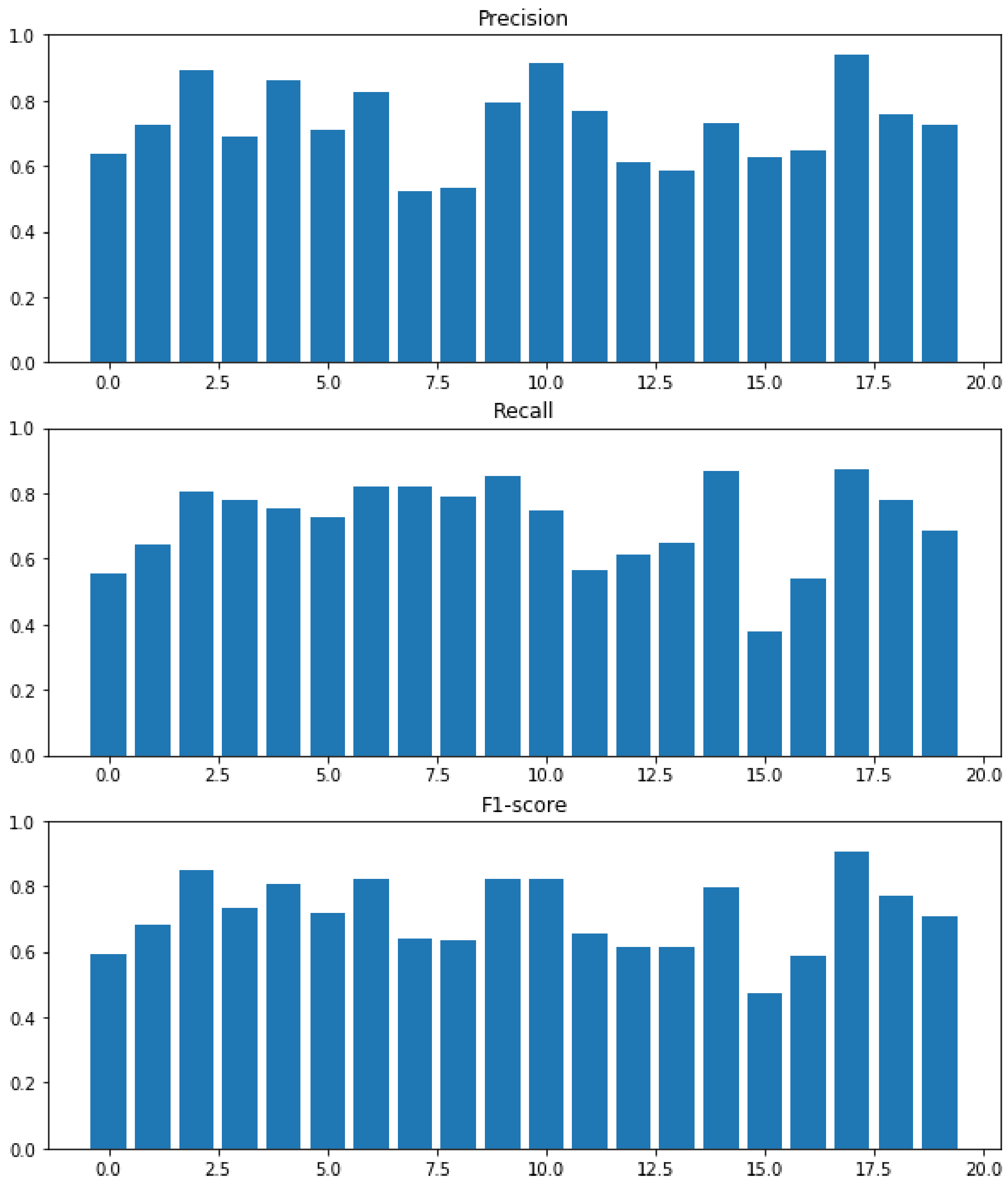
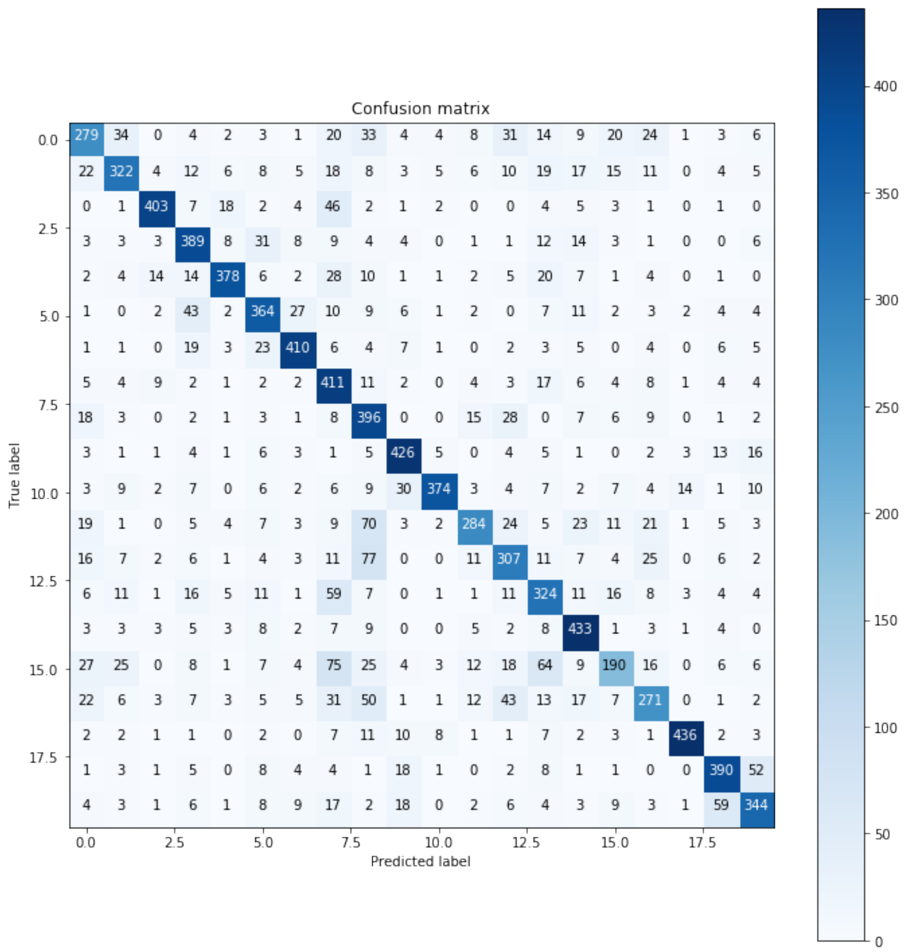
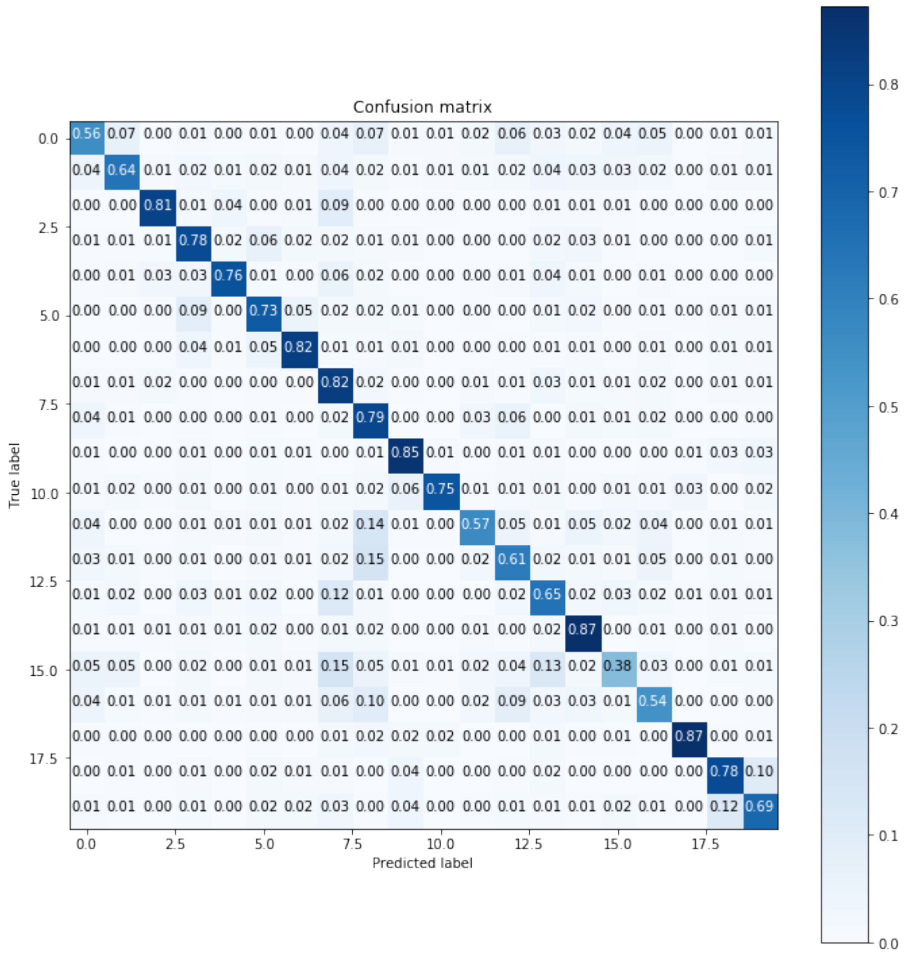


Figure 6: Precision, recall, and F1 score for predictions of ConvNet



279	34	0	4	2	3	1	20	33	4	4	8	31	14	9	20	24	1	3	6
22	322	4	12	6	8	5	18	8	3	5	6	10	19	17	15	11	0	4	5
0	1	403	7	18	2	4	46	2	1	2	0	0	4	5	3	1	0	1	0
3	3	3	389	8	31	8	9	4	4	0	1	1	12	14	3	1	0	0	6
2	4	14	14	378	6	2	28	10	1	1	2	5	20	7	1	4	0	1	0
1	0	2	43	2	364	27	10	9	6	1	2	0	7	11	2	3	2	4	4
1	1	0	19	3	23	410	6	4	7	1	0	2	3	5	0	4	0	6	5
5	4	9	2	1	2	2	411	11	2	0	4	3	17	6	4	8	1	4	4
18	3	0	2	1	3	1	8	396	0	0	15	28	0	7	6	9	0	1	2
3	1	1	4	1	6	3	1	5	426	5	0	4	5	1	0	2	3	13	16
3	9	2	7	0	6	2	6	9	30	374	3	4	7	2	7	4	14	1	10
19	1	0	5	4	7	3	9	70	3	2	284	24	5	23	11	21	1	5	3
16	7	2	6	1	4	3	11	77	0	0	11	307	11	7	4	25	0	6	2
6	11	1	16	5	11	1	59	7	0	1	1	11	324	11	16	8	3	4	4
3	3	3	5	3	8	2	7	9	0	0	5	2	8	433	1	3	1	4	0
27	25	0	8	1	7	4	75	25	4	3	12	18	64	9	190	16	0	6	6
22	6	3	7	3	5	5	31	50	1	1	12	43	13	17	7	271	0	1	2
2	2	1	1	0	2	0	7	11	10	8	1	1	7	2	3	1	436	2	3
1	3	1	5	0	8	4	4	1	18	1	0	2	8	1	1	0	0	390	52
4	3	1	6	1	8	9	17	2	18	0	2	6	4	3	9	3	1	59	344

Figure 7: Confusion matrix for predictions of ConvNet



0.56	0.07	0.00	0.01	0.00	0.01	0.00	0.04	0.07	0.01	0.01	0.02	0.06	0.03	0.02	0.04	0.05	0.00	0.01	0.01
0.04	0.64	0.01	0.02	0.01	0.02	0.01	0.04	0.02	0.01	0.01	0.01	0.02	0.04	0.03	0.03	0.02	0.00	0.01	0.01
0.00	0.00	0.81	0.01	0.04	0.00	0.01	0.09	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00
0.01	0.01	0.01	0.78	0.02	0.06	0.02	0.02	0.01	0.01	0.00	0.00	0.00	0.02	0.03	0.01	0.00	0.00	0.00	0.01
0.00	0.01	0.03	0.03	0.76	0.01	0.00	0.06	0.02	0.00	0.00	0.00	0.01	0.04	0.01	0.00	0.01	0.00	0.00	0.00
0.00	0.00	0.00	0.09	0.00	0.73	0.05	0.02	0.02	0.01	0.00	0.00	0.00	0.01	0.02	0.00	0.01	0.00	0.01	0.01
0.00	0.00	0.00	0.04	0.01	0.05	0.82	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.01	0.01
0.01	0.01	0.02	0.00	0.00	0.00	0.00	0.82	0.02	0.00	0.00	0.01	0.01	0.03	0.01	0.01	0.02	0.00	0.01	0.01
0.04	0.01	0.00	0.00	0.00	0.01	0.00	0.02	0.79	0.00	0.00	0.03	0.06	0.00	0.01	0.01	0.02	0.00	0.00	0.00
0.01	0.00	0.00	0.01	0.00	0.01	0.01	0.00	0.01	0.85	0.01	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.03	0.03
0.01	0.02	0.00	0.01	0.00	0.01	0.00	0.01	0.02	0.06	0.75	0.01	0.01	0.01	0.00	0.01	0.01	0.03	0.00	0.02
0.04	0.00	0.00	0.01	0.01	0.01	0.01	0.02	0.14	0.01	0.00	0.57	0.05	0.01	0.05	0.02	0.04	0.00	0.01	0.01
0.03	0.01	0.00	0.01	0.00	0.01	0.01	0.02	0.15	0.00	0.00	0.02	0.61	0.02	0.01	0.01	0.05	0.00	0.01	0.00
0.01	0.02	0.00	0.03	0.01	0.02	0.00	0.12	0.01	0.00	0.00	0.00	0.02	0.65	0.02	0.03	0.02	0.01	0.01	0.01
0.01	0.01	0.01	0.01	0.01	0.02	0.00	0.01	0.02	0.00	0.00	0.01	0.00	0.02	0.87	0.00	0.01	0.00	0.01	0.00
0.05	0.05	0.00	0.02	0.00	0.01	0.01	0.15	0.05	0.01	0.01	0.02	0.04	0.13	0.02	0.38	0.03	0.00	0.01	0.01
0.04	0.01	0.01	0.01	0.01	0.01	0.01	0.06	0.10	0.00	0.00	0.02	0.09	0.03	0.03	0.01	0.54	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.02	0.02	0.00	0.00	0.01	0.00	0.01	0.00	0.87	0.00	0.01
0.00	0.01	0.00	0.01	0.00	0.02	0.01	0.01	0.00	0.04	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.78	0.10
0.01	0.01	0.00	0.01	0.00	0.02	0.02	0.03	0.00	0.04	0.00	0.00	0.01	0.01	0.01	0.02	0.01	0.00	0.12	0.69

Figure 8: Normalized confusion matrix for predictions of ConvNet

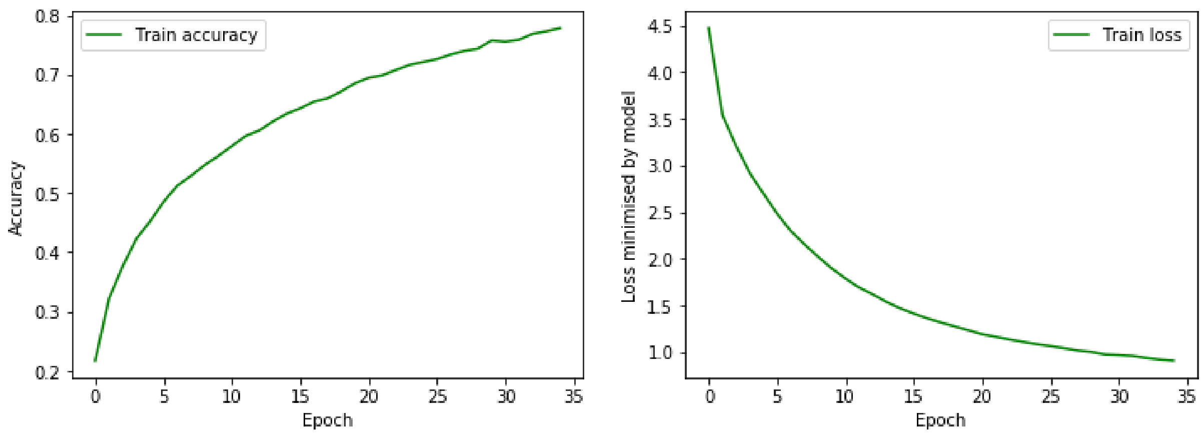


Figure 9: Validation curves for ResNet, trained on the complete train set

This means that combining the models in an ensemble will not help much.

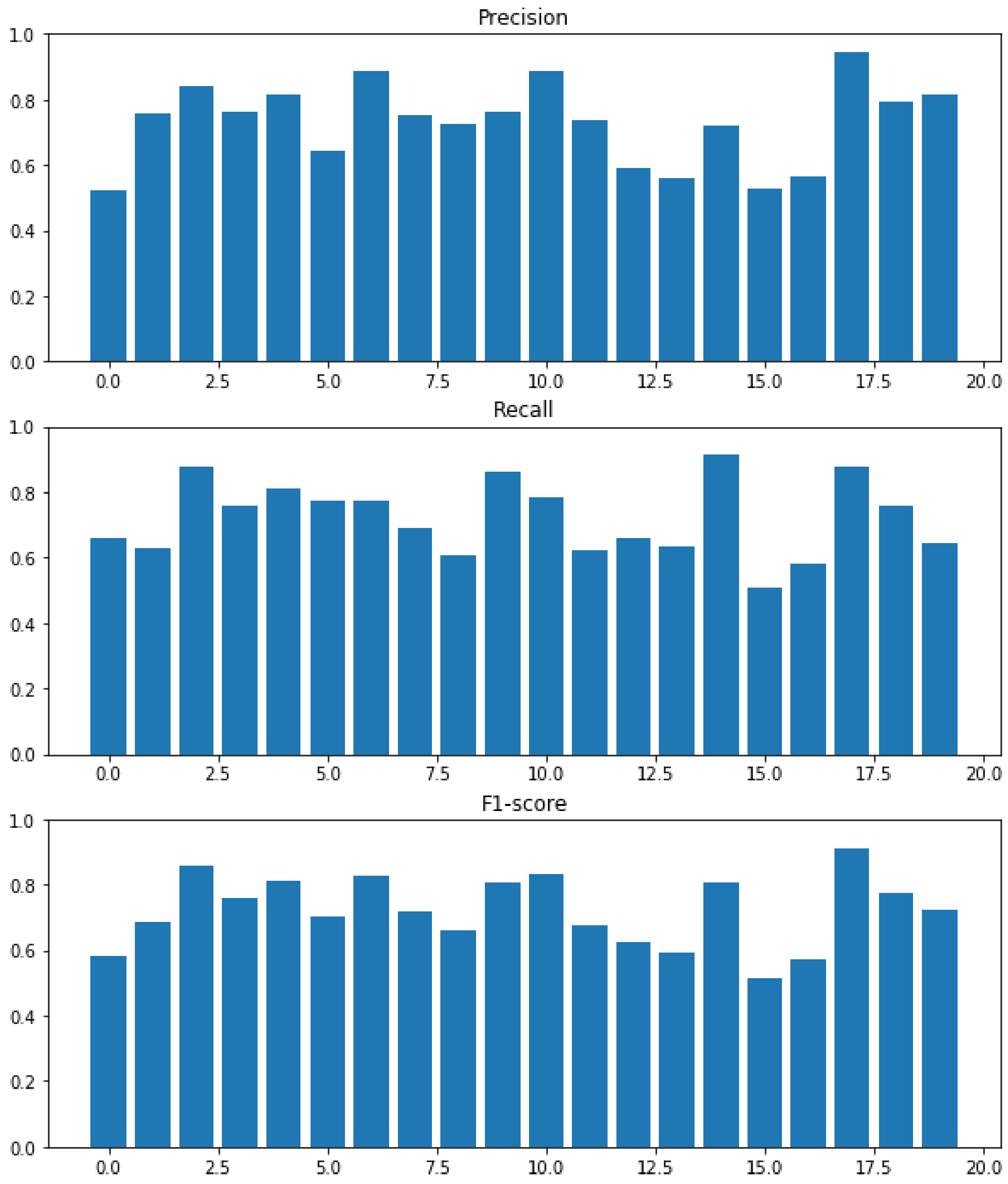
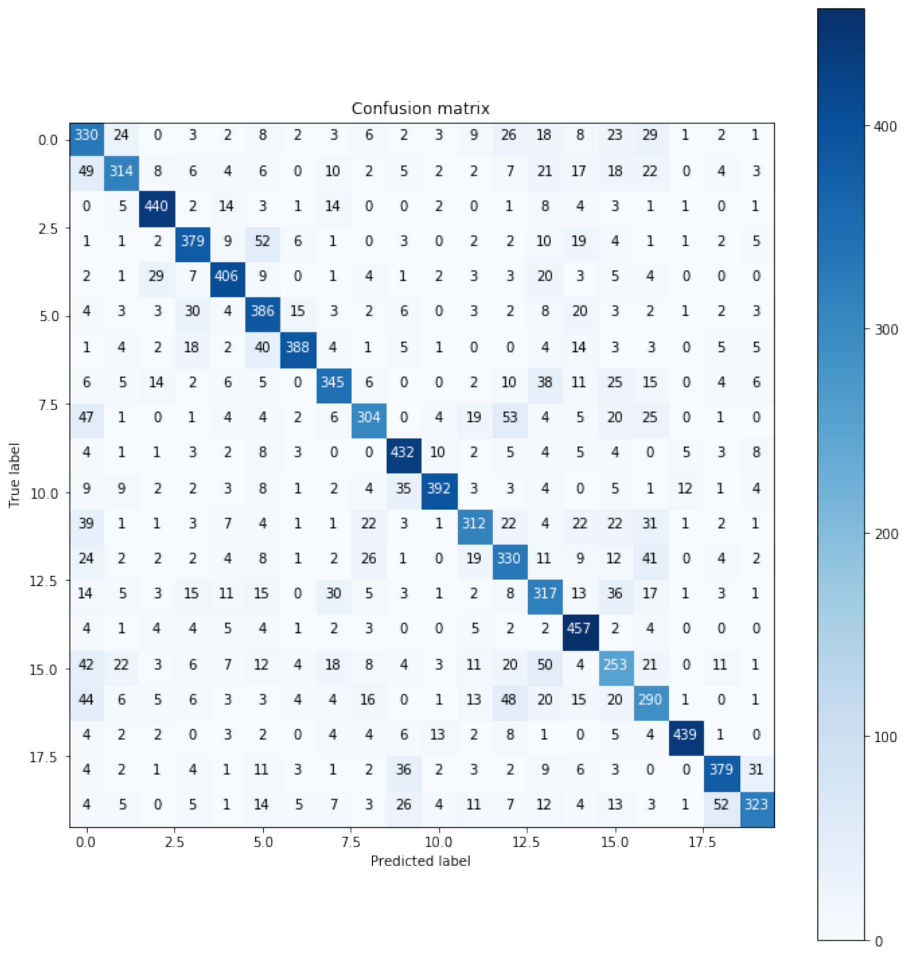


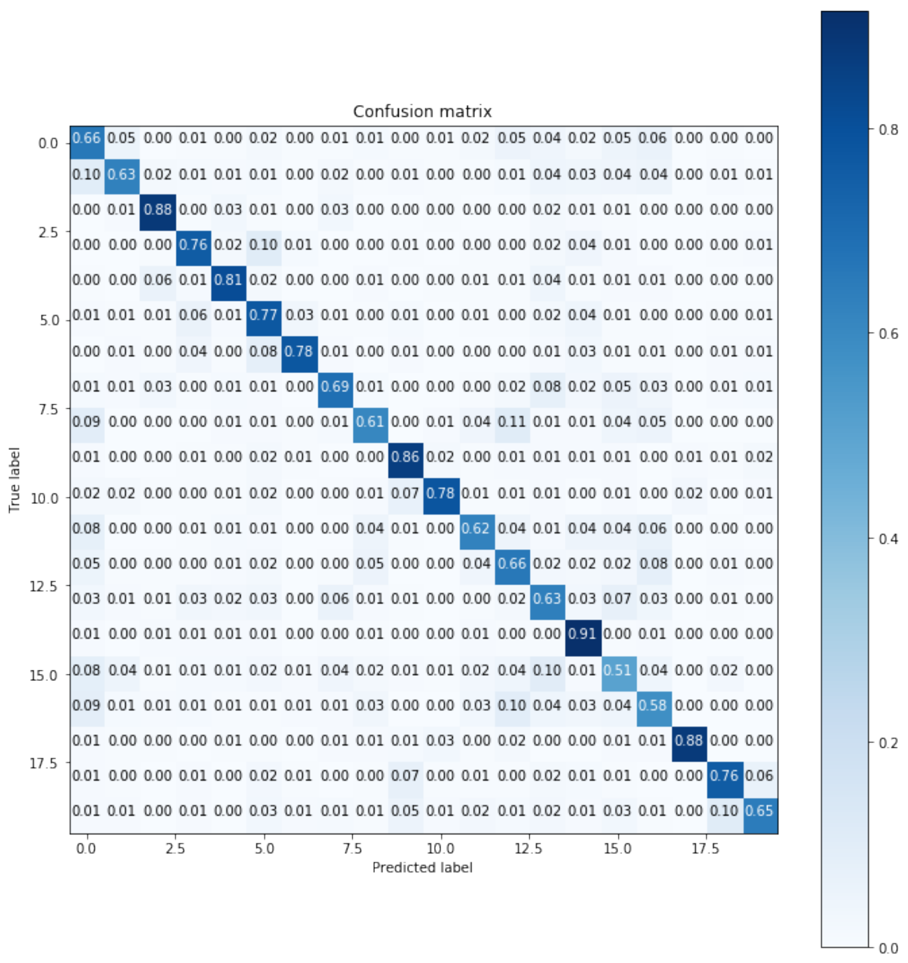
Figure 10: Precision, recall, and F1 score for predictions of ResNet

We conclude that the effect of SkipConnections on a small network does not increase accuracy significantly. It primarily helps learning faster in the first few epochs and was originally proposed to resolve vanishing gradients. To improve upon our networks, a possibility could be to add Inception modules.



330	24	0	3	2	8	2	3	6	2	3	9	26	18	8	23	29	1	2	1
49	314	8	6	4	6	0	10	2	5	2	2	7	21	17	18	22	0	4	3
0	5	440	2	14	3	1	14	0	0	2	0	1	8	4	3	1	1	0	1
1	1	2	379	9	52	6	1	0	3	0	2	2	10	19	4	1	1	2	5
2	1	29	7	406	9	0	1	4	1	2	3	3	20	3	5	4	0	0	0
4	3	3	30	4	386	15	3	2	6	0	3	2	8	20	3	2	1	2	3
1	4	2	18	2	40	388	4	1	5	1	0	0	4	14	3	3	0	5	5
6	5	14	2	6	5	0	345	6	0	0	2	10	38	11	25	15	0	4	6
47	1	0	1	4	4	2	6	304	0	4	19	53	4	5	20	25	0	1	0
4	1	1	3	2	8	3	0	0	432	10	2	5	4	5	4	0	5	3	8
9	9	2	2	3	8	1	2	4	35	392	3	3	4	0	5	1	12	1	4
39	1	1	3	7	4	1	1	22	3	1	312	22	4	22	22	31	1	2	1
24	2	2	2	4	8	1	2	26	1	0	19	330	11	9	12	41	0	4	2
14	5	3	15	11	15	0	30	5	3	1	2	8	317	13	36	17	1	3	1
4	1	4	4	5	4	1	2	3	0	0	5	2	2	457	2	4	0	0	0
42	22	3	6	7	12	4	18	8	4	3	11	20	50	4	253	21	0	11	1
44	6	5	6	3	3	4	4	16	0	1	13	48	20	15	20	290	1	0	1
4	2	2	0	3	2	0	4	4	6	13	2	8	1	0	5	4	439	1	0
4	2	1	4	1	11	3	1	2	36	2	3	2	9	6	3	0	0	379	31
4	5	0	5	1	14	5	7	3	26	4	11	7	12	4	13	3	1	52	323

Figure 11: Confusion matrix for predictions of ResNet



0.66	0.05	0.00	0.01	0.00	0.02	0.00	0.01	0.01	0.00	0.01	0.02	0.05	0.04	0.02	0.05	0.06	0.00	0.00	0.00
0.10	0.63	0.02	0.01	0.01	0.01	0.00	0.02	0.00	0.01	0.00	0.00	0.01	0.04	0.03	0.04	0.04	0.00	0.01	0.01
0.00	0.01	0.88	0.00	0.03	0.01	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.01	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.76	0.02	0.10	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.02	0.04	0.01	0.00	0.00	0.00	0.01
0.00	0.00	0.06	0.01	0.81	0.02	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.04	0.01	0.01	0.01	0.00	0.00	0.00
0.01	0.01	0.01	0.06	0.01	0.77	0.03	0.01	0.00	0.01	0.00	0.01	0.00	0.02	0.04	0.01	0.00	0.00	0.00	0.01
0.00	0.01	0.00	0.04	0.00	0.08	0.78	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.03	0.01	0.01	0.00	0.01	0.01
0.01	0.01	0.03	0.00	0.01	0.01	0.00	0.69	0.01	0.00	0.00	0.00	0.02	0.08	0.02	0.05	0.03	0.00	0.01	0.01
0.09	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.61	0.00	0.01	0.04	0.11	0.01	0.01	0.04	0.05	0.00	0.00	0.00
0.01	0.00	0.00	0.01	0.00	0.02	0.01	0.00	0.00	0.86	0.02	0.00	0.01	0.01	0.01	0.01	0.00	0.01	0.01	0.02
0.02	0.02	0.00	0.00	0.01	0.02	0.00	0.00	0.01	0.07	0.78	0.01	0.01	0.01	0.00	0.01	0.00	0.02	0.00	0.01
0.08	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.04	0.01	0.00	0.62	0.04	0.01	0.04	0.04	0.06	0.00	0.00	0.00
0.05	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.05	0.00	0.00	0.04	0.66	0.02	0.02	0.02	0.08	0.00	0.01	0.00
0.03	0.01	0.01	0.03	0.02	0.03	0.00	0.06	0.01	0.01	0.00	0.00	0.02	0.63	0.03	0.07	0.03	0.00	0.01	0.00
0.01	0.00	0.01	0.01	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.91	0.00	0.01	0.00	0.00	0.00
0.08	0.04	0.01	0.01	0.01	0.02	0.01	0.04	0.02	0.01	0.01	0.02	0.04	0.10	0.01	0.51	0.04	0.00	0.02	0.00
0.09	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.03	0.00	0.00	0.03	0.10	0.04	0.03	0.04	0.58	0.00	0.00	0.00
0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.01	0.03	0.00	0.02	0.00	0.00	0.01	0.01	0.88	0.00	0.00
0.01	0.00	0.00	0.01	0.00	0.02	0.01	0.00	0.00	0.07	0.00	0.01	0.00	0.02	0.01	0.01	0.00	0.00	0.76	0.06
0.01	0.01	0.00	0.01	0.00	0.03	0.01	0.01	0.01	0.05	0.01	0.02	0.01	0.02	0.01	0.03	0.01	0.00	0.10	0.65

Figure 12: Normalized confusion matrix for predictions of ResNet

References

[1] S. H. HasanPour, M. Rouhani, M. Fayyaz, and M. Sabokrou, “Lets keep it simple, using simple architectures to outperform deeper and more complex architectures”, *CoRR*, vol. abs/1608.06037, 2016. arXiv: `1608.06037`. [Online]. Available: `http://arxiv.org/abs/1608.06037`.