Experimental Results

The current project aims to classify images from the **"10 Monkey Species" dataset**, in **10 classes** corresponding to their species. Consequently, the task is a **multi-class image classification** problem. The problem is solved using an Artificial Neural Network, more precisely a Deep Feed Forward Network.

Performed experiments include the variation of number of layers, layers sizes and learning rate. For each experiment, a **k-fold cross-validation** was used.

Activation functions, regardless of the number or size or layers, are set according to:

- Last layer: **SoftMax** activation; given that we perform multi-class classification, the last layer should output probabilities for each class, where the sum of the probabilities is always 1.
- Other layers: **ReLU** activation

To evaluate performance, **accuracy** is the central metric, but it is associated with **confusion matrices**, **precision**, **recall** and **f-score** for a better understanding of the results. Furthermore, to have a statistical analysis of the accuracy measurements across the cross-validation iterations, we compute **95% confidence interval**.

Experimental setup

Experiments focus on 2 types of variations in **network architecture**: width and depth. For each architecture, experiments include variations in:

- Learning rate
- Number of iterations: early stopping is also implemented, with parametrized patience

For all the experiments, we use **Stochastic Gradient Descent** (i.e., batch size = dataset size).

The dataset has around 1368 images, with the following distribution of images across classes:

Class 0	Class 1	Class 2	Class3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
131	139	137	152	131	141	132	142	133	132

For k-fold cross validation, k is set to 8. Given the fact that we do not have lots of images per class, for k=10 variations from fold to folds were quite large. For k = 5, the number of train images was too small. Thus, k was to set to 8 for all the experiments.

The following metrics are computed to evaluate the network performance:

- Accuracy: Per cross-validation iteration and average
- Confusion matrix: Per cross-validation iteration and average

- Precision: Per cross-validation iteration and average
- Recall: Per cross-validation iteration and average
- F-Score: Per cross-validation iteration and average
- 95% confidence interval for the accuracy of the k cross-validation iterations

The following formulas were used for the metrics:

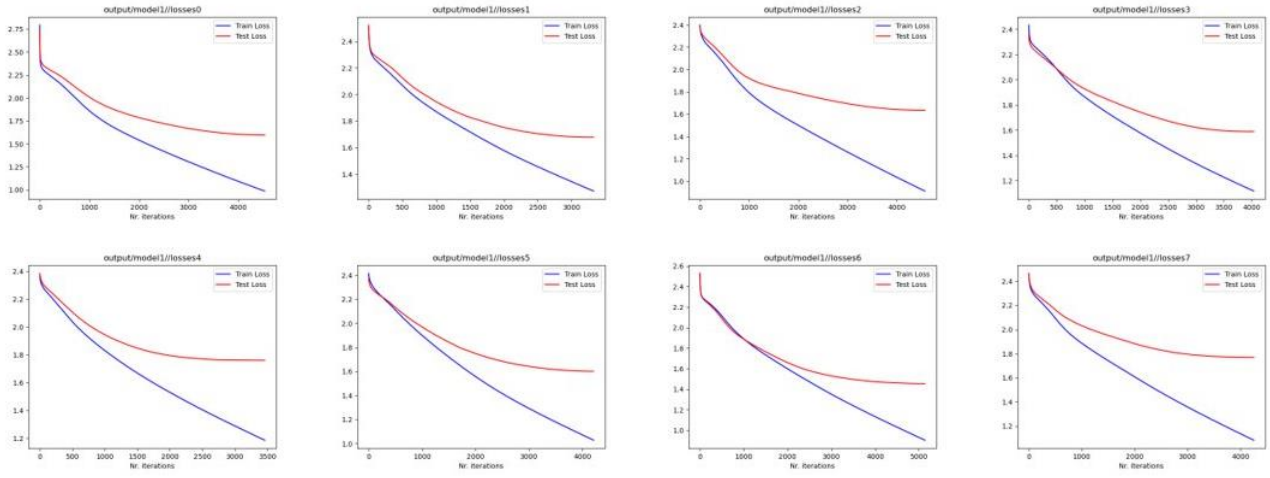
- Accuracy = (TP + TN) / Total
- Confusion matrix: rows are true classes, columns are predictions
- Precision = TP / (TP + FP), in a one vs. all manner
- Recall = TP / (TP + FN), in a one vs. all manner
- 10 folds 95% confidence interval: 1.96 * std / sqrt(sample_size)

The table below shows the setup for all the experiments performed:

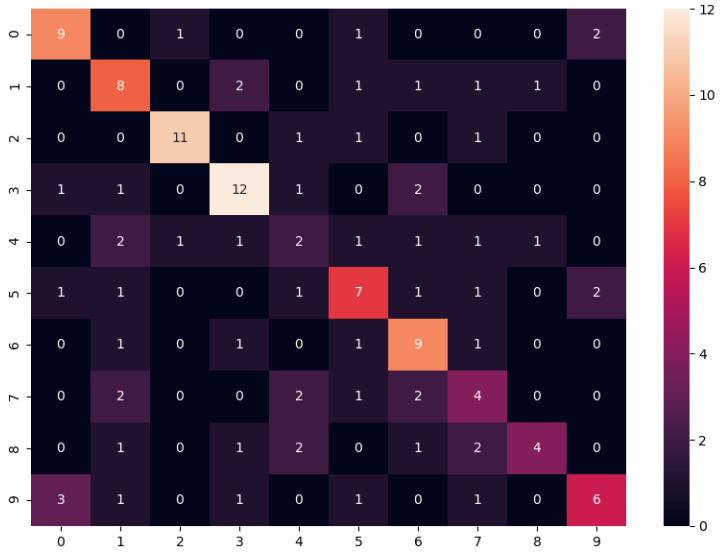
ID	Layer dimensions	Input size	Learning rate	# folds	Early stopping patience
1	[12288, 50, 20, 10]	64x64x3	3e-3	8	50
2	[12288, 100, 50, 20, 15, 10]	64x64x3	1e-3	8	50

Results

Experiment 1



Train and validation losses for every of the 8 cross-validation iterations



Average confusion matrix

Classes	Precision	Recall	F-score
0	0.529	0.581	0.553
1	0.402	0.475	0.433
2	0.631	0.64	0.631
3	0.562	0.651	0.601
4	0.193	0.168	0.176
5	0.415	0.411	0.41
6	0.438	0.56	0.49
7	0.309	0.268	0.286
8	0.389	0.24	0.29
9	0.474	0.408	0.431

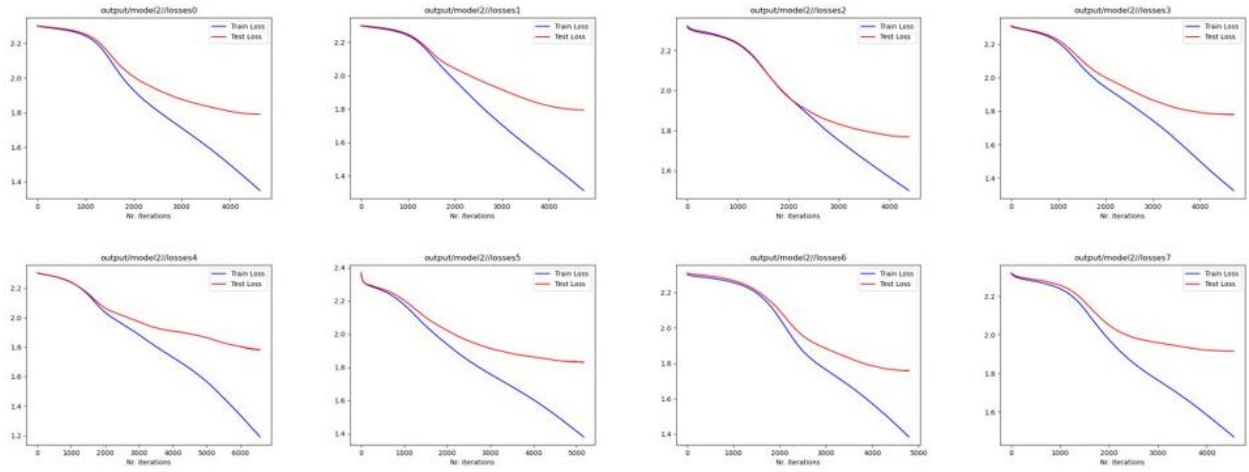
Average precision, recall and F-score

# fold	Test accuracy
0	**46.6%**

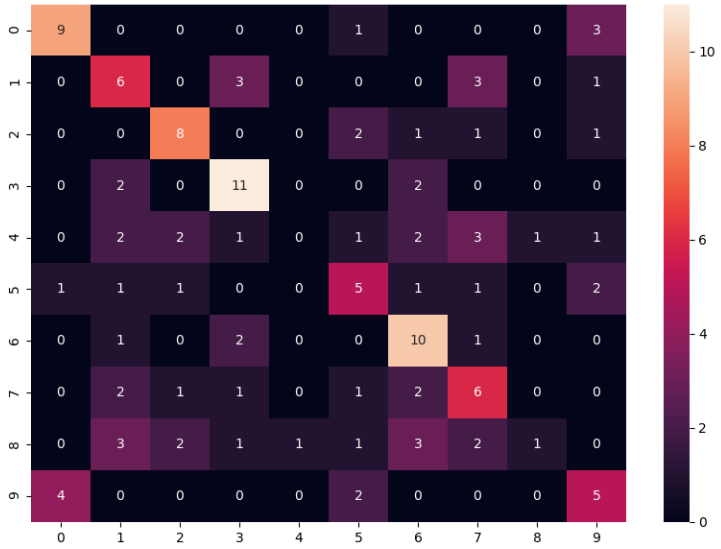
1	**38.09%**
2	**43.1%**
3	**46.8%**
4	**41.8%**
5	**45.2%**
6	**52.7%**
7	**39.5%**
Average	**44.3%**
95% confidence interval	**41.4 - 47.2%**

Accuracy over all folds

Experiment 2



Train and validation losses for every of the 8 folds



Average confusion matrix

Classes	Precision	Recall	F-score
0	0.575	0.58	0.574
1	0.296	0.373	0.328
2	0.43	0.487	0.449
3	0.481	0.586	0.525
4	0.089	0.046	0.058
5	0.298	0.309	0.3
6	0.404	0.606	0.482
7	0.301	0.351	0.316
8	0.339	0.061	0.088
9	0.344	0.352	0.344

Average precision, recall and F-score

Scope	**Test accuracy**

0	**41.5%**
1	**41.1%**
2	**35.6%**
3	**40.9%**
4	**39.4%**
5	**33.9%**
6	**37.1%**
7	**32.9%**
Average	**37.8%**
95% confidence interval	**35.6 - 40%**

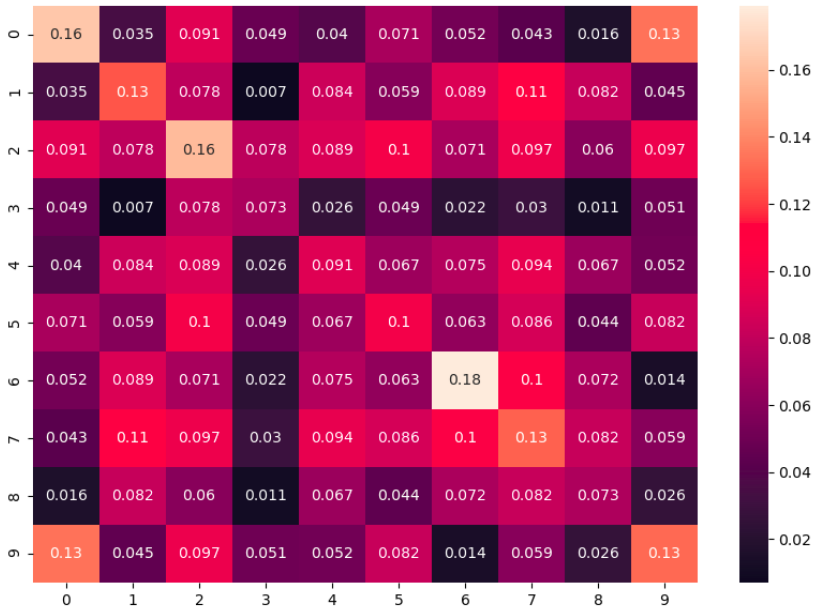
Accuracy over all folds

Discussion

Two experiments we performed, the main difference between them being network architecture. In Experiment 2, both the network width and depth were increased. Despite increasing the network complexity, the performance did not improve. I hypothesize that the biggest challenge for the network is the lack of translation or projection invariance, which cannot be compensated for by adding more layers or increasing their sizes.

Then, we will take a closer look at the results obtained for experiment 1. The average accuracy is 44.3%, with a 95% confidence interval of 41.4-47.2%. The variation between cross-validation iterations is around 4%, which might be easily explained by the reduced number of images in the dataset.

One important remark is the close connection between the confusion matrixes and the correlation matrix. In the problem analysis document, a correlation matrix is presented. We computed the correlation between all the images in class *i* and all the images in class *j*, where *i = [1, 10], j = [1, 10]*. To obtain a single value for the correlation, we average the correlation between all pairs of images. The correlation matrix is displayed below:



By evaluating the correlation and confusion matrices, we notice that classes with smaller in-class correlation are often wrongly predicted. Furthermore, these classes are often confused with classes to which they are strongly correlated. This close connection between correlation and confusion matrices shows that, by nature, ANNs struggle to find features with strong semantics in images. This is a consequence of the large input dimensionality, along with the fact that fully connected layers are not position invariant nor do they perform convolutional operations.

The other computed metrics (precision, recall and F-score) reflect the fact that a significant decrease in accuracy comes from a reduced set of classes (*e.g.,* classes 4 and 8).

Based on all these observations, further experiments might include:

- Data augmentation: the number of images in the dataset might be increased by augmentation via rotations, translations, projections, brightness adjustments
- Increasing the input image resolution