Analysing the Impact of Mislabeled Data on Training Neural Networks

M.Sc. Bioinformatics

Fabian Fliedner

July 23, 2021

Time of the Internship:	08/2020 – 10/2020	
Instructor:	Dr. Nadine Flinner	Prof. Dr. Enrico Schleiff

Contents

List of Figures

List of Tables

Machine learning as an analytic tool is used in multiple disciplines and fields. From playing games to applications in the financial world and medicine. One of the medical applications is detection and classification of different cancer (sub-)types like breast cancer. In this lab, we investigate how incorrect or incomplete data labels affect the quality of training and classification of a neural network, with respect to the analysis of gastric cancer. This study used training and mislabeling images of animals obtained from the ImageNet database as a proof of concept, which in further studies can be expanded and tested on medical image data. We could show, that training with larger datasets could improve the classification of the neural network and more reliably classify images with the wrong labels.

1 Introduction

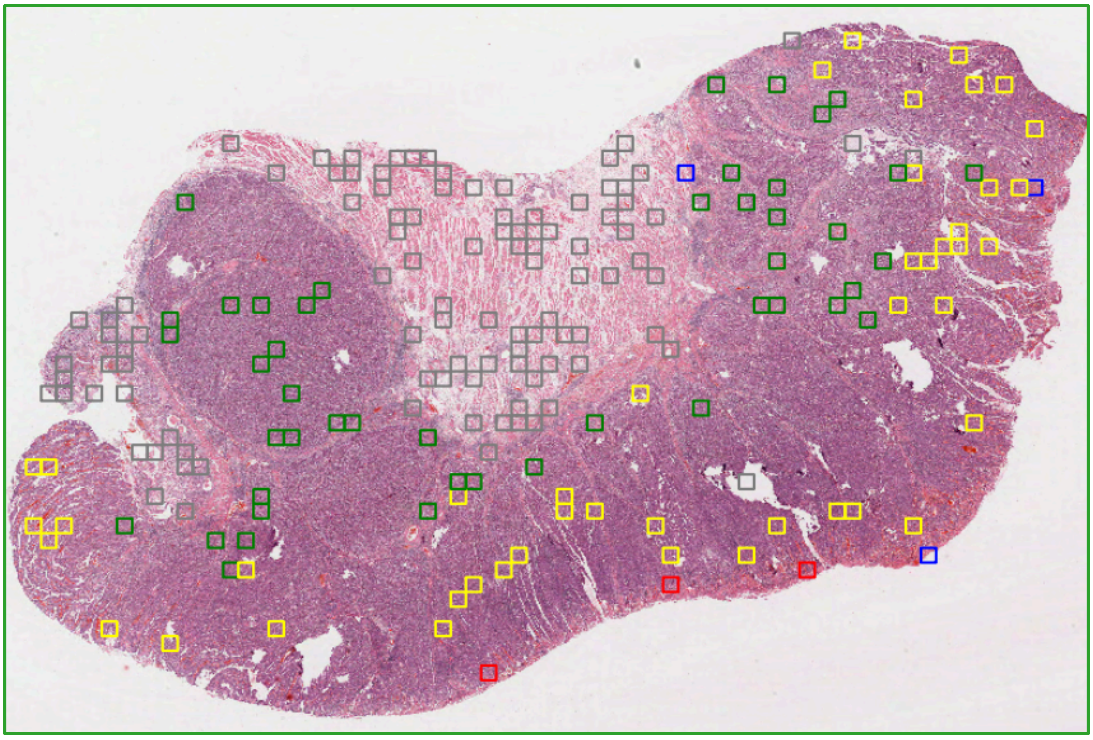


Figure 1: Tissue section of gastric cancer tissue. Annotations for the different molecular subtypes.

Machine learning as an analytical tool has various fields in which it can be implemented. From applications in games such as chess Lai (2015) and Go (AlphaGo) Wu and Baldi (2007), to analyses in the financial world Dixon et al. (2020), as well as in medicine Deo (2015), e.g. for the classification of cancer types on the basis of tissue sections Amrane et al. (2018).

The group of Dr. Nadine Flinner (?) focuses on the use of computer vision to identify and classify gastric cancer. Gastric cancer cases are divided into multiple molecular subtypes Bass et al. (2014): Epstein–Barr virus (EBV)-positive, microsatellite instability (MSI), genomically stable (GS) and chromosomal instability (CIN).

For training a neural network to classify these subtypes, tissue sections are assigned labels corresponding to the subtype obtained from genomic data. However, cancer tissue is not homogeneous de Aretxabala et al. (1989) (see Figure 1), so tissue sections may contain areas where the label does not match the molecular subtype.

This lab is focused on an analysis of the impact of mislabeled datasets on the training of a neural network. For this purpose, instead of medical data, images of animals from the ImageNet database Deng et al. (2009) are used, since a neural network can be trained quickly on the basis of the different features.

1.1 Definitions

Definition 1.1 (Observable Error)**.** The error calculated by the neural network, comparing the prediction made for an image of the validation set with the label with which the neural network was trained (containing purposefully mislabeled images). This describes the error rates which can be directly observed and calculated by the neural network.

Definition 1.2 (True Error)**.** A second error calculated by comparing the prediction made by the neural network to the true label of the images in the validation set (which is known beforehand). This error can't be obtained in real data. It has to be calculated by purposefully introducing errors into the data-sets and comparing the prediction of the neural network with the label assigned to the data and the true label, which is hidden from the network.

2 Materials and Method

2.1 ImageNet data

To train the neural network and have a sizeable amount of images for each of the classes, we used pictures of *dogs*, *cats* and later *mice* and small *birds*, downloaded from the ImageNet database.

Disclaimer regarding validation and test data

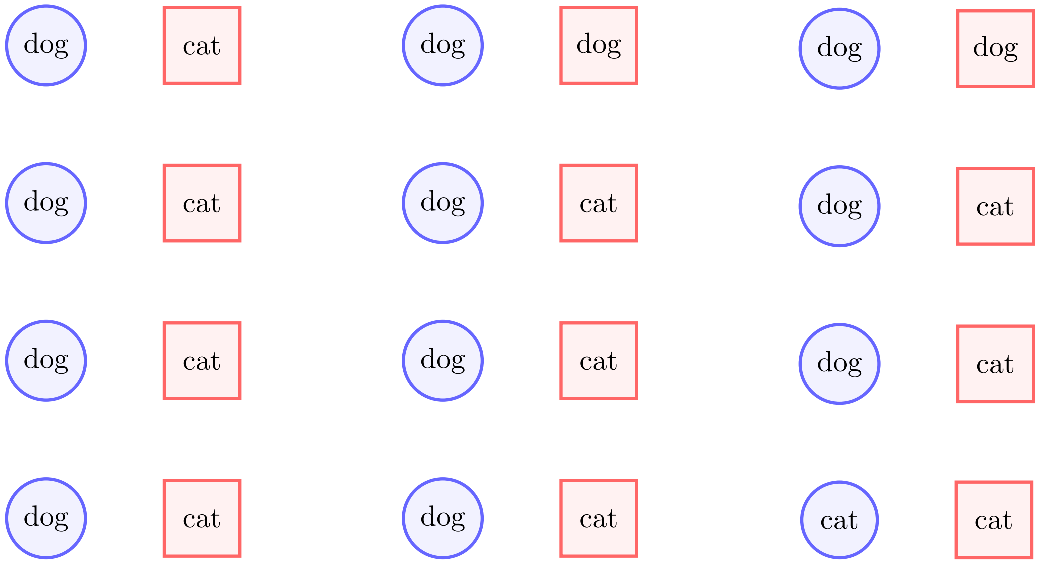
While ideally the test data would be fully independent from the training and validation data, in this study all of the images were obtained from ImageNet and split into the three groups prior to training the network.

2.2 Creation of dataframes and introduction of labeling errors

A Python script is used to generate a dataframe for the training of the neural network. The dataframe contains information about the filenames and two different labels. One is the label used in training the neural network (called 'label') while the other (called 'true_label') is used for the ”manual” calculation of the *True Error*. The true label is obtained through the filename (ImageNet provides a document which can be used to map the filenames to a human readable label), while the label used in training the network is ’randomly’ assigned. Following these labels are five columns containing information if the image is used in training or validation of the neural network. Each image is used four times in training and once for the validation [five-fold cross-validation]. An example is shown in Table 1. A schematic for the introduction of labeling errors is shown in Figure 2.

file	label	true_label	validation [0-4]
`Filename.jpg`	`label`	`label`	`True/False`
⋮			

Table 1: An example for the used dataframes. Validation [0-4] combines five columns used for the five-fold cross-validation



a) Training without any labeling errors.

b) Training with an image showing a dog falsely labeled as a cat.

c) Training with labeling errors in both subclasses.

Figure 2: Schematic of how the labeling errors are introduced. Blue circles are images showing dogs, red squares are images of cats, with the label corresponding to the label used for training the network.

2.3 Training the network

Using the dataframes we trained multiple neural networks with different starting conditions. One of the variables was the amount of data used. We started with a low number of images, to quickly obtain a base line on which we could calculate the different errors. We then increased the amount of data used for training and validation by a factor of 10 and 100 in comparison to our first runs. Another variation was the number of classes, as well as how the labeling errors are introduced into the data.

3 Results

In all of the tests (studies?) we could observe increasing error rates with higher amounts of data containing labeling errors

3.1 Training a network with two classes with labeling errors in one of the classes.

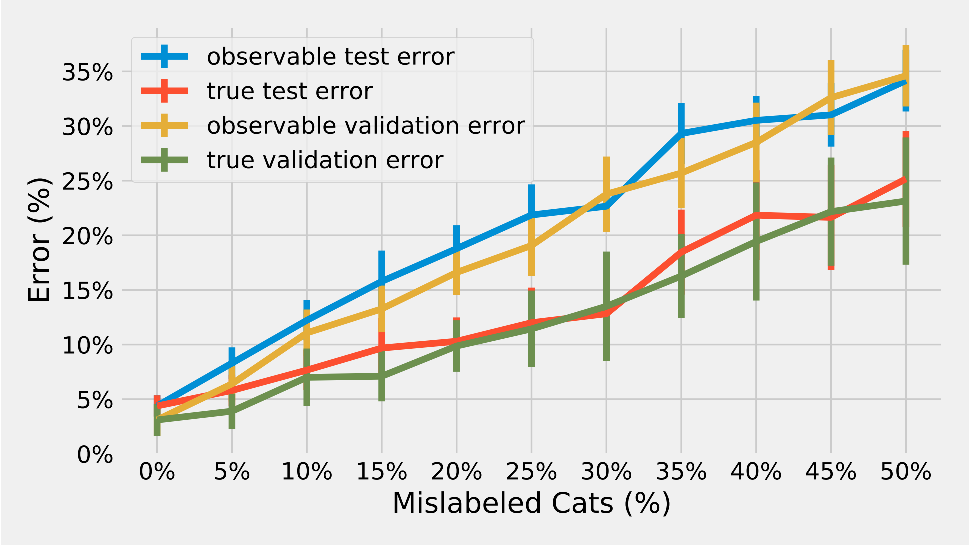
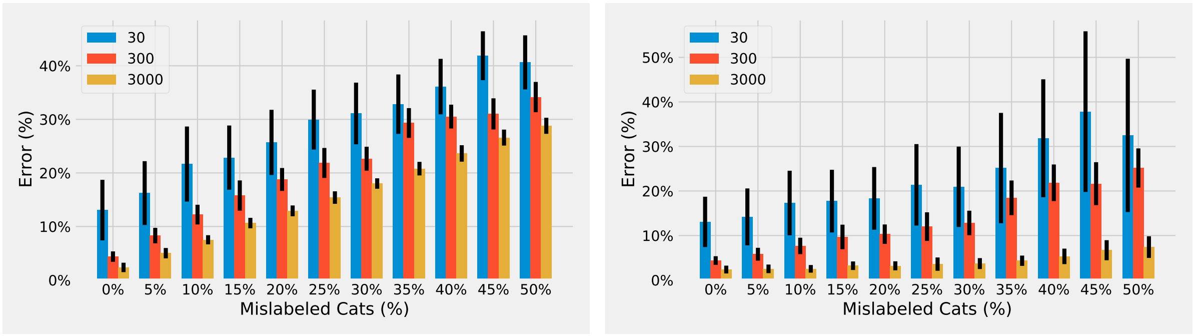


Figure 3: Comparing the development of different error types for training with 300 images in each of the two classes. Labeling errors only occurring in one of the classes.

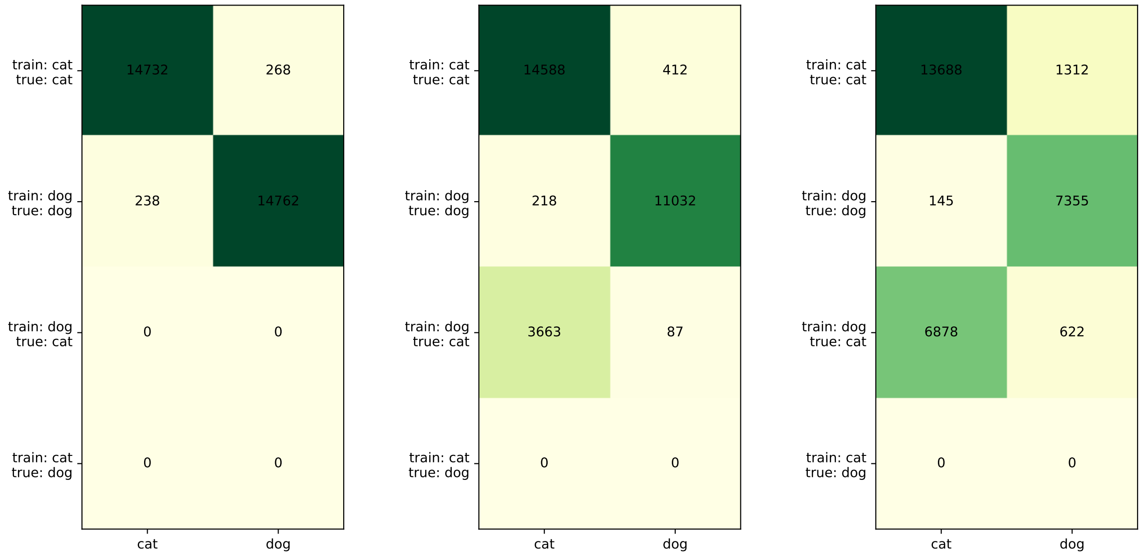
When introducing the labeling error into one of the classes (in this case having images of *cats* labeled as *dogs*, shown schematically in 2b) we noticed a higher *observable error* compared to the *true error* in both validation and testing dataset for percentage of labeling errors ≥ 0%, as shown in Figure 3. At 0% labeling error every image the network evaluates as incorrect is correctly identified as incorrect, since all of the assigned labels are the true labels. Another observation is, that error rates calculated from the test dataset is slightly higher than the ones calculated from the validation dataset. When comparing the error rates for different amounts of data-points (see Figure 4) it shows a lower error rates (both *observable* and *true error*) for higher amounts of images. Looking at the combined confusion matrices in Figure 5 (specifically 5b & 5c) over all runs a majority of the images (> 90%) with an incorrect label are predicted correctly, according to the true label. while this cannot be observed directly, it shows that in our dataset the network can still reliably match the test images even with labeling errors that have a significant proportion (25%) of the training dataset.



a) Observable error

b) True error

Figure 4: Comparing the observable and true error rates for training with different amounts of data.

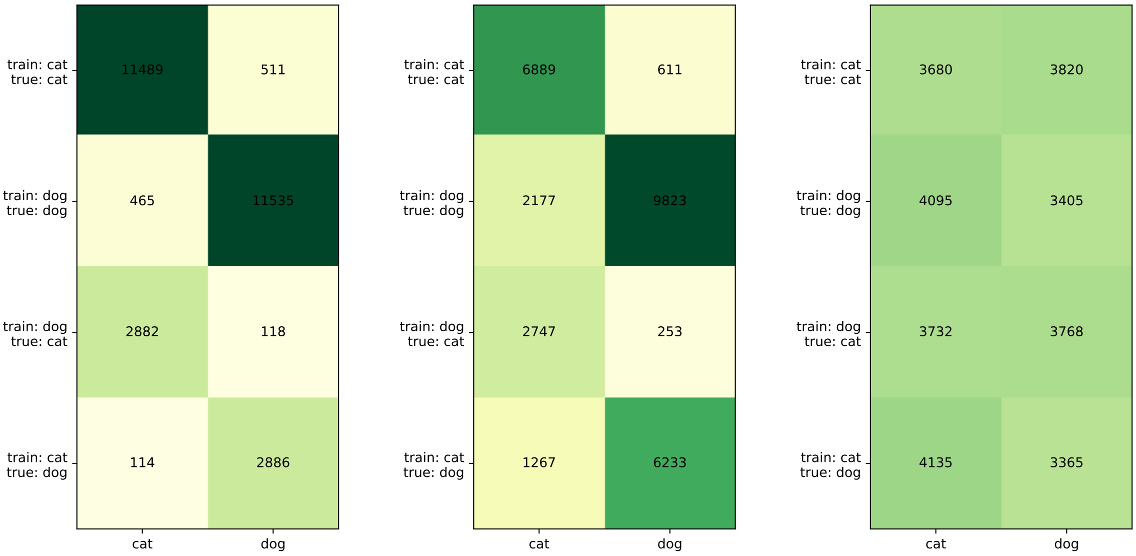


a) no mislabeled training data **b)** 25% of cats are labeled as dogs **c)** 50% of cats are labeled as dogs

Figure 5: Confusion matrices for training with labeling errors occurring in one of the two classes.

3.2 Training a network with two classes with labeling errors in both classes.

In this setup we introduced the labeling errors into both classes the network is training with, as schematically shown in Figure 2c. Even with the introduction of a labeling error of 50% in one class and 20% in another, the true error is less than 15% Figure 6a and 6b. With labeling errors in both classes of 40-50%, the network's predictions become a coin flip, as reflected in Figure 6c. Figure 7 shows that the effect of the labeling errors is symmetrical with both classes. There is a noticeable difference in the development of the observable versus true error. While the observable error behaves almost linearly, the effect of the labeling errors on the true error is very low until the percentage of labeling errors approaches 50%. Figure 7c shows the differences between observable and true error. Since for no labeling errors both describe the same error, the difference is 0, which is also the case for 50% labeling errors in both classes, where the network has the exact same amount of images showing cats being labeled as dogs and vice versa.



a) 20% of each class labeled as the other

b) 20% of cats labeled as dogs, 50% of dogs labeled as cats

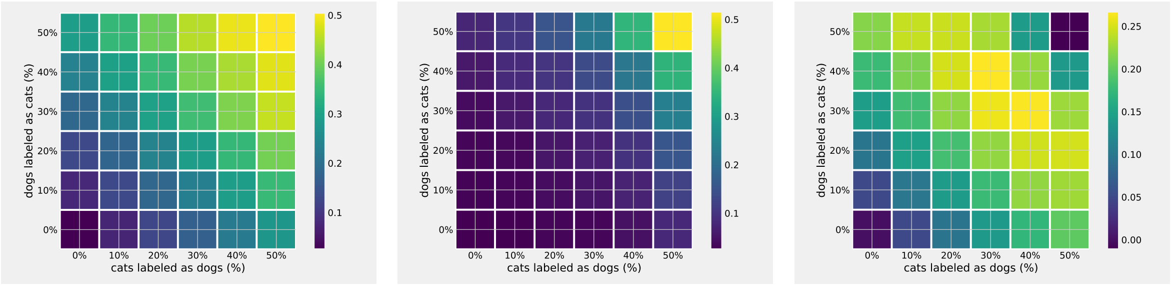
c) 50% of each class labeled as the other

Figure 6: Confusion matrices for training with labeling errors occurring in both classes.

3.3 Training a network with four classes. One class mislabeled and evenly distributed.

For this study we added two additional classes *birds* and *mice*. We introduced labeling error occurs symmetrically into all of the non-*cat*-classes, which means for every class of images not showing cats we added a percentage of images showing cats with the label of the class.

Comparing the development of the error rates with four classes Figure 8 to the study with



a) Development of the observable error for differing amounts of labeling errors in both classes.

b) Development of the true error for differing amounts of labeling errors in both classes.

c) difference between the observable and true error.

Figure 7: Comparing the development of the observable and true error when labeling errors occur in both of the classes.

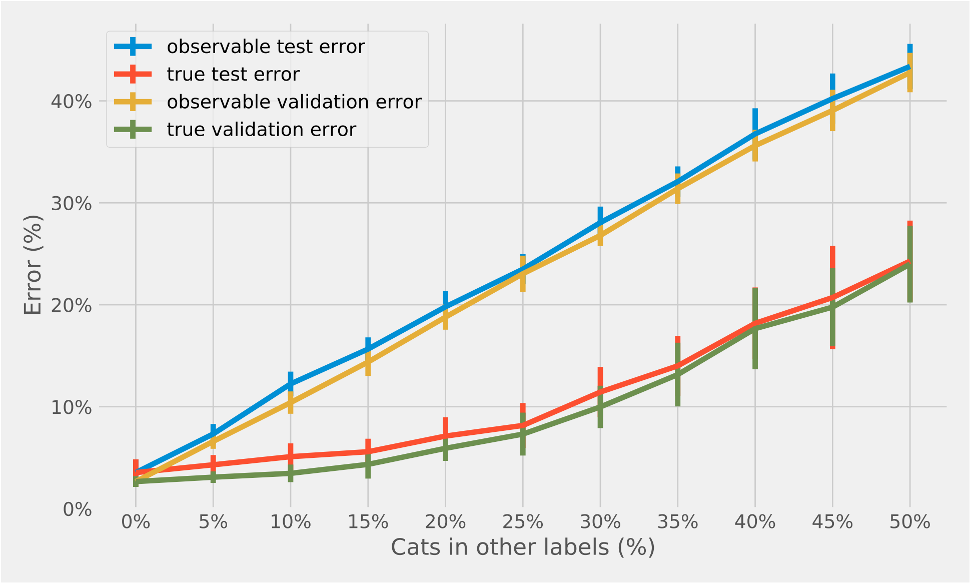
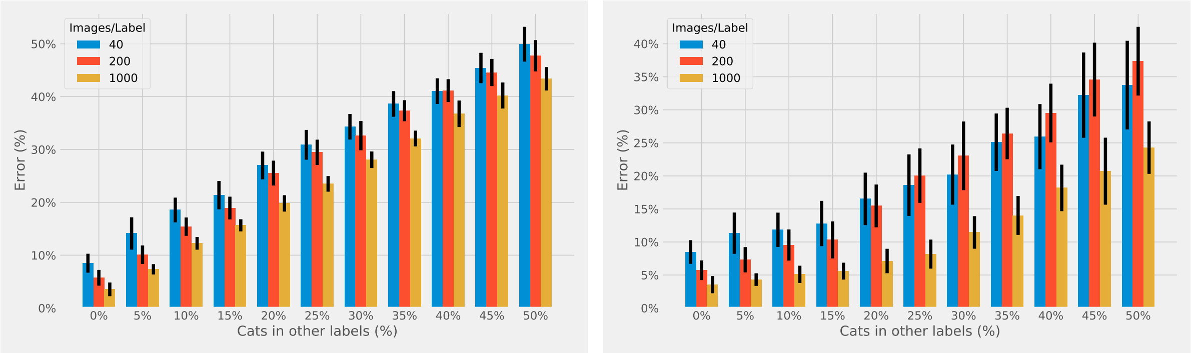


Figure 8: Comparing the development of different error types for training with 300 images in each of the four classes. Labeling errors are introduced by having images of cats among all of the other classes.



a) Development of the observable error with different amounts of images

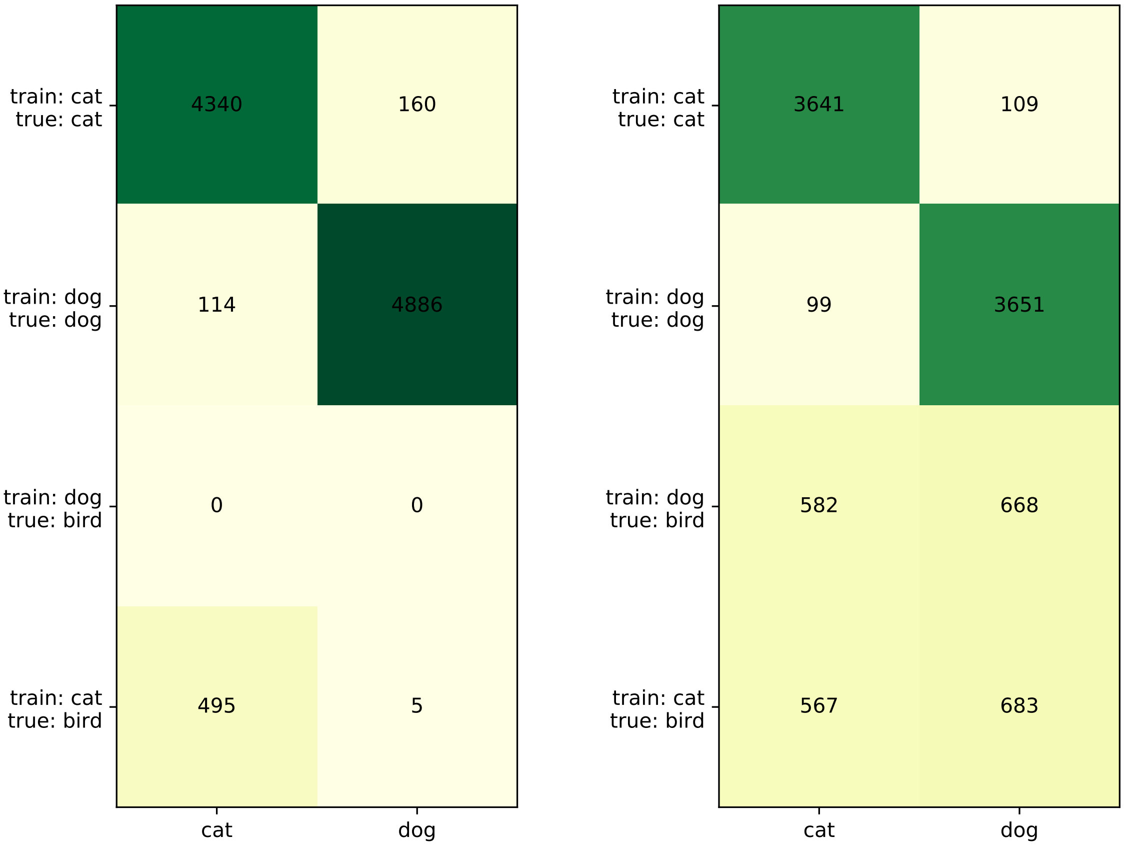
b) Development of the true error with different amounts of images

Figure 9: Comparing the observable and true error rates for training with different amounts of data. Labeling errors of up to 50% occurring in three of four labels.

only two classes Figure 3 shows similar trends, with the observable error rates being significantly higher than the true error and the error rates calculated from the validation dataset slightly lower than the ones obtained from the test datasets. The conclusion that we can mitigate labeling errors by increasing the amount of data used for training the neural network, with the higher true error in the run with 200 images per label (Figure 9b) can be explained through the limited amount of repetitions.

3.4 Training a network with two known classes and introducing a hidden/unknown third class.

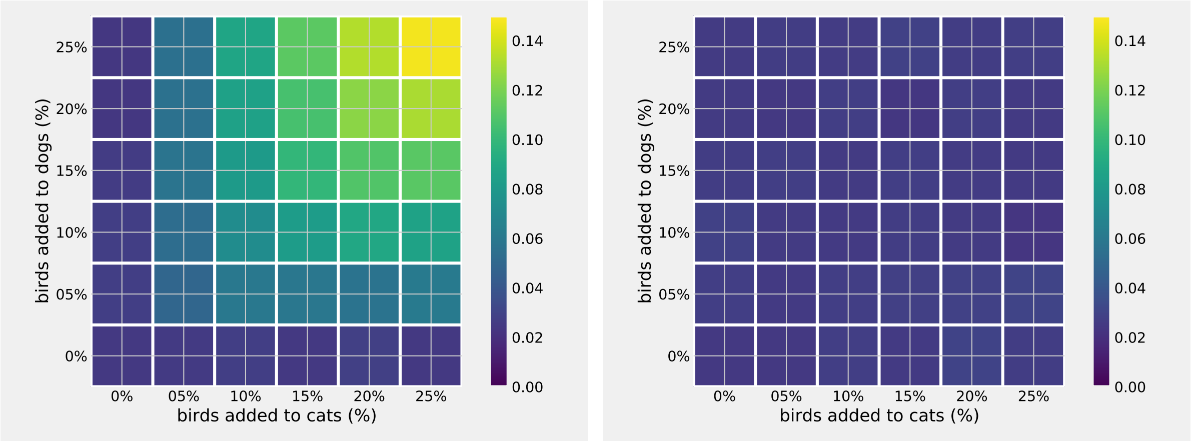
For this study, images of birds were inserted into the training datasets as dogs as well as cats. The amount of inserted mislabeled images, was increased in five percent increments up to 25% and could be different for the two classes. Since the hidden class *birds* cannot be detected by the Neural Network, we adjusted the definition of the *true error*. Because the network continues to train on the two classes *cats* and *dogs* all images of birds assigned to either class would count towards the True Error. This would result in the True Error being equal to the number of images inserted. For this reason, we calculate the True Error only for images that are not part of the hidden subclass.



a) 10% of the images labeled as cats are showing birds

b) 25% of the images in both classes are showing birds, labeled as either dogs or cats.

Figure 10: Confusion matrices for a hidden class which is spread between the other classes.



a) Observable error development

b) True error development

Figure 11: Comparing the development of the observable and true error when introducing a hidden subclass into one or both classes.

label		**predictions**							
train	**true**	cat	dog	cat	dog	cat	dog	cat	dog
cat	cat	145	6	140	4	152	1	138	7
dog	dog	3	147	5	141	4	142	2	140
dog	bird	33	17	24	30	25	29	28	30
cat	bird	33	17	21	35	30	17	25	30
		run 3		**run 6**		**run 12**		**run 21**	

Table 2: excerpt of confusion matrices from different runs, with 25% of both labels being mislabeled birds

3.4.1 Conclusions for training with a hidden subclass

Since we changed the calculation of the true error for this analysis, the effect of introducing a hidden class is nonexistant, since all of the analyzed classes are very distinct from each other.

The effect on the observable error depends on how the unknown subclass is spread throughout the training and validation data-sets. If the subclass is mostly contained in one of the trained subclasses the observable doesn't change significantly compared to a network trained only on two classes, since most of the images are predicted to be of the label they are wrongly associated with (see Figure 10a, only 1% of the images containing birds, labeled as cats were predicted to be dogs). This is also reflected in Figure 11, showing a homogenous distribution of the true error, regardless of the amount of images showing birds added to both classes.

When the hidden subclass is spread more or less equally between the other two labels it turns into a coin flip which label is assigned to images of birds. While Figure 10b shows a slight bias towards dogs, this seems to be a product of adding up all of the runs. Looking at Table 2 also shows examples, where the Network has a bias towards predicting birds as cats.

3.5 Comparing the development of the error rates in dependance of training size

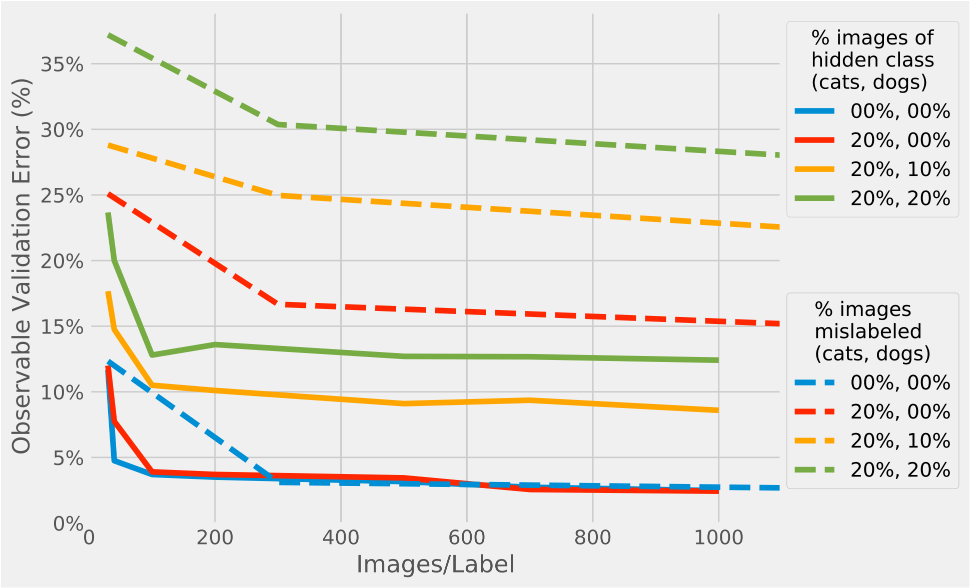


Figure 12: Comparing the change of the observable error between the runs with a hidden subclass and labeling errors in both classes, when training with different amounts of data.

Unsurprisingly, if the data contains more labeling errors, the error rates (true and especially observable) of the predictions increase. Comparing the true errors for the validation data-set with an increasing number of images, we see a decrease in the true error rate for a similar percentage of labeling errors – with an exception for a high percentages of mislabeled data (50% mislabeled cats ≡ 25% of overall Data mislabeled) Figure 4 and Figure 9. The difference, however, is within one standard deviation, so we can attribute this to a small number of 25 runs, five runs with a five-fold cross-validation each.

Comparing this figure, where the amount of data increases by an order of magnitude in between runs, to the previous one, the error rate between the different runs decreases significantly. This leads to the assumption that we can mitigate the effect of false labels in training data by increasing the amount of data used in training significantly (10×/100×).

4 Discussion and Outlook

Since this study has only worked with ImageNet data so far, the next logical step is to apply the method to medical images of gastric cancer to see if more heterogeneous images behave similarly to the very distinct ones used in this study. The ImageNet data and classes used are highly distinctive and have already been used by FastAI to train the neural networks.

In addition, it is possible to investigate whether can be detected if the data carries a wrong label or there is a class in the data that was not known before, either by looking at the loss or comparing the error-rates.

In the current study, only the error rate was used to compare how the errors affect the training. For subsequent analyses, other criteria (e.g. loss) should be used for further comparison.

In addition, it can be analyzed how certain images are predicted in different runs during training. For example, in the context of this lab, one could look at whether images that have cats on them are predicted as *cats* more often than expected by chance, regardless of the assigned label.

References

Amrane, M., Oukid, S., Gagaoua, I., and Ensari, T. (2018). Breast cancer classification using machine learning. In *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, pages 1–4. IEEE.

Bass, A. J., Thorsson, V., Shmulevich, I., Reynolds, S. M., Miller, M., Bernard, B., Hinoue, T., Laird, P. W., Curtis, C., Shen, H., et al. (2014). Comprehensive molecular characterization of gastric adenocarcinoma. *Nature*, 513(7517):202.

de Aretxabala, X., Yonemura, Y., Sugiyama, K., Hirose, N., Kumaki, T., Fushida, S., Miwa, K., and Miyazaki, I. (1989). Gastric cancer heterogeneity. *Cancer*, 63(4):791–798.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Deo, R. C. (2015). Machine learning in medicine. *Circulation*, 132(20):1920–1930.

Dixon, M. F., Halperin, I., and Bilokon, P. (2020). *Machine Learning in Finance*. Springer.

Lai, M. (2015). Giraffe: Using deep reinforcement learning to play chess. *arXiv preprint arXiv:1509.01549*.

Wu, L. and Baldi, P. (2007). A scalable machine learning approach to go. *Advances in Neural Information Processing Systems*, 19:1521.