Appendix: When Deep Classifiers Agree: Analyzing Correlations between Learning Order and Image Statistics

Iuliia Pliushch[1], Martin Mundt[2], Nicolas Lupp[1], and Visvanathan Ramesh[1]

[1] Goethe University Frankfurt, Germany
{pliushch,vramesh}@em.uni-frankfurt.de
[2] TU Darmstadt and hessian.AI, Germany
martin.mundt@tu-darmstadt.de

In this supplementary material we provide additional details, experimental setup and descriptions for the employed methodology of the main body. The structure is as follows:

A. Experimental setup and training hyper-parameters.
B. Additional plots for initial experiments on different batch-sizes and architectures, as well as expected random agreement and a discussion of an alternative agreement definition, as well as importance of our findings in the light of data shuffling during training.
C. Additional discussion of the reasons for the weakness of correlations for CIFAR10, as well as experimental results for Pascal trained on ResNet, omitted in the main body. In this context also an explanation of the relationship between Pearson correlation coefficient and the p-value.
D. More precise description of the computed dataset metrics, as well as additional visualization thereof.
E. Visualization of dataset metrics histograms
F. Discussion on correlation vs. causation

A Experimental details

Since our aim is to analyze the training process on the original images, we did not use data augmentation techniques, apart from random cropping for train and center cropping for test images on Pascal, ImageNet and KTH-TIPS2b due to the difference in size between images in these datasets. For Pascal and ImageNet, we resize the smaller size to 256 and randomly crop to obtain patches of width and height 224 pixels [6, 9]. For KTH-TIPS2b, in analogy we resize to 200 and then randomly crop to the size 190 pixels. Note that we perform dataset metrics computation on the original non-cropped (training) images. Only for ImageNet's entropy and frequency calculation we downsample the images to 128x128 for computational reasons.

For **CIFAR10**, we trained (5 times) LeNet5 (with added batch normalization after each layer), VGG16, ResNet50 and DenseNet121 on original labels using SGD with momentum 0.9 for 60 epochs with batch-size 128, batch-normalization

10^{-5} and weight-decay $5 * 10^{-4}$, cosine annealing scheduler [7] with initial learning rate 0.1 and minimal learning rate $5 * 10^{-4}$, which lowers the learning rate from the initial to the minimal one over the training epochs (without warm restarts). For the random label experiment, we have lowered the initial learning rate to 0.001 to ensure a quicker convergence. We use Kaiming normal weight initialization [3] for all experiments.

For **KTH-TIPS2b**, we used the *sample a* of each class for testing and the rest for training. We trained DenseNet121 for 60 epochs with batch-size 64, Adam with momentum 0.9, batch-norm 10^{-5}, weight-decay 10^{-5} and a one cycle learning rate scheduler [10, 11] in which the learning rate first increases from a minimal one to a maximal one of 10^{-4} and then decreases over the rest of epochs to an even lower minimum. Standard Pytorch implementation parameters for OneCycleLR have been used to determine the initial and final learning rate.

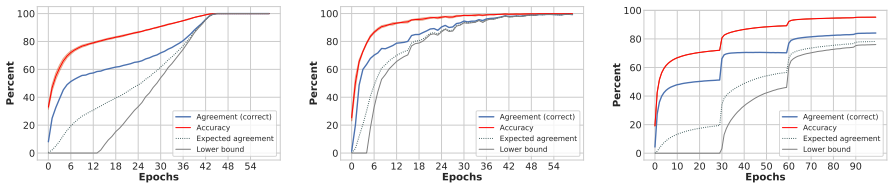
For **Pascal**, we used train and validation splits of 2007 and 2012 for training and 2007 test split (in which we disregarded difficult label instances) for testing. We trained DenseNet121 and ResNet50 for 150 epochs with batch-size 128, SGD with momentum 0.9, batch-norm 10^{-5}, weight-decay $5 * 10^{-4}$ and a step learning rate scheduler [7] which lowers the initial learning rate of 0.1 every 50 steps by a factor of 0.2. For **ImageNet**, we trained DenseNet121 for 100 epochs with batch-size 128, SGD with momentum 0.9, batch-norm 10^{-5}, weight-decay 10^{-5} and a step learning rate scheduler which lowers the initial learning rate of 0.1 every 30 steps by a factor of 0.1. The training procedure for Pascal and ImageNet is inspired by Huang *et al.*[4]. We used single NVIDIA A100 GPU to run Pascal/ImageNet style experiments with DenseNet or ResNet.

B Initial experiments: additional plots

First and foremost, let us add a few word about why our findings are important in the light of the stochasticity of gradient descent. During training, train instances are usually shuffeled (once per epoch) and gradient descent is performed in mini-batches. An epoch is defined as a period during which the learner has seen all train instances once. So, in every epoch the order, in which instances are presented, is not fixed (due to this shuffling process). Minibatch-size defines how many instances are seen before a gradient update is performed and, hence, when the information about this instances influences the gradients. If the order, in which instances are presented, changes, but the same instances are classified correctly by all networks, it means that the information contained in this instance was more important to be learned than in others at that stage of the learning process. To find out, which information it is, we chose the dataset metrics to correlate agreement with.

In addition to the lower bound presented in the paper, we also computed *expected random agreement* by multiplying the network accuracies (divided by 100 to the range between 0 and 1) in a given epoch. This gives us an assessment of how probable it is that networks randomly agree on dataset instances which they

classify correctly, assuming that they classify dataset instances independently. We observe in fig. 1 that expected random agreement is higher than our lower bound, but still lower than the actual agreement.



(a) **CIFAR10** (b) **KTH-TIPS2b** (c) **ImageNet**

Fig. 1: Ablation study: Computing *expected random agreement*, in comparison to agreement and lower bound. Expected random agreement is higher than the lower bound, but still lower than agreement. For Pascal we see in fig. 2 that the deviation on agreement and the expected random agreement is rather small too.

As mentioned in the main body, we also conducted an experiment to calculate the standard deviation on agreement, similarly to the way we computed the deviation on accuracy. For Pascal, we ran the experimental setup 5 times, hence training 25 neural networks in total, to be able to calculate the deviation on agreement (and the lower bound). fig. 2 visualizes that it is quite small.

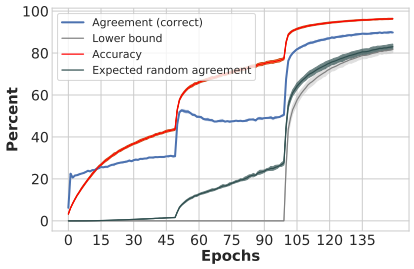
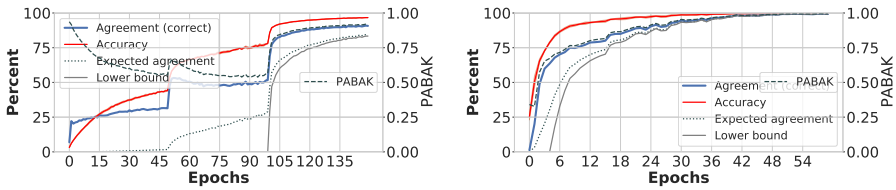


Fig. 2: **Pascal DenseNet**: Agreement visualization on *train set*, with expected random agreement, as well as standard deviation on agreement, expected random agreement and the lower bound.

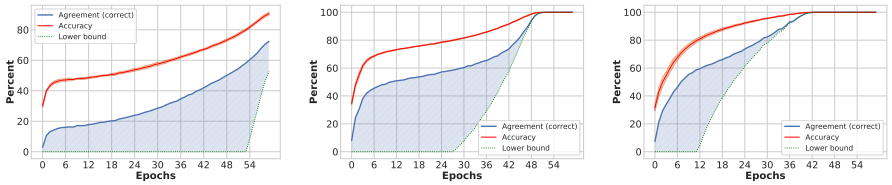
Second, let us strenghten the argument in favor of our definition of agreement even further. In the main body we have mentioned Cohen's kappa and PABAK as measures for the reliability of agreement. Usually, both operate on the notion of observed agreement, which considers not only true positives, but



(a) **Pascal DenseNet** (b) **KTH-TIPS2b DenseNet**

Fig. 3: Visualization of **PABAK** on *train set* for a 2-class scenario (correctly classified vs. incorrectly classified). PABAK 's range is between -1 and 1. PABAK measure is 0 when observed agreement is 50%.

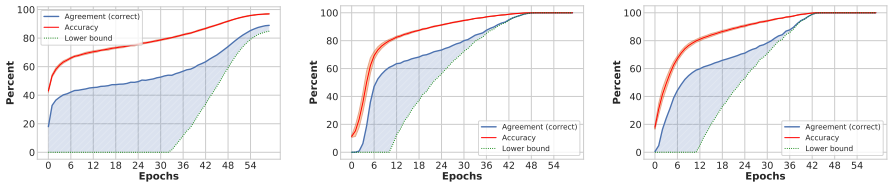
also true negatives. We focus only on true positives, because already taking into account true negatives makes the analysis more complex, since several trends are evaluated simultaneously. In addition, Cohen's kappa and PABAK operate over only 2 estimators. Since we have 5 networks, we have to either choose another measure, or to compute the average over all pairs of estimators. One measure for more than 2 estimators is *Fleiss kappa*, but it assumes that instead of a fixed number of estimators, estimators are sampled from a larger pool such that it is not the case that every dataset instance is classified by the same estimators.



(a) Batch-size 16 (b) Batch-size 64 (c) Batch-size 256

Fig. 4: Ablation study on **CIFAR10**: training with different batch-sizes

To get a feel for observed agreement, let us consider a simplified scenario in which there are 2 classes - correctly classified and wrongly classified. We can then sum instances both estimators classify correctly and incorrectly, normalize by the total number of instances. We then linearly transform it to counteract the prevalence bias as described in [1] and average over pairs of estimators. We see in fig. 3a that if the accuracy grows slowly, we get a U-shape. First, PABAK is high due to the number of true negatives - it is the case when both estimators classify wrongly,- then it gets higher due to the number of instances pairs of estimators classify correctly. In fig. 3b we can see that if accuracy grows fast,



(a) LeNet5, batch-size 128 (b) VGG16, batch-size 128 (c) ResNet50, batch-size 128

Fig. 5: Ablation study on **CIFAR10**: training with differnt architectures

PABAK curve resembles the true positive agreement we defined. However, the exact values of agreement we defined and PABAK cannot be compared as easily, because PABAK ranges between -1 and 1 and is 0 when observed agreement (which incorporates true positives and negatives) is 50%. Note that the 2 class scenario is a crude simplification, as we would actually want to know in a multi-scenario, whether estimators missclassify *in the same way* (into the same wrong class).

Third, we further conducted agreement experiments for CIFAR10 on DenseNet for several batch-sizes (5 networks for every batch-size, in analogy to the main body experiments), see fig. 4. We also trained 5 networks each for CIFAR10 on LeNet5, VGG16 and ResNet50, in addition to DenseNet, see fig. 5. Comparison of both figures shows that agreement is present for different batch-sizes and architectures, as well as that the agreement curve changes similarly for growing batch-sizes and architecture complexity.

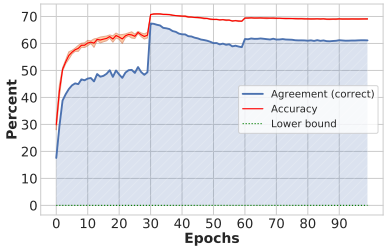
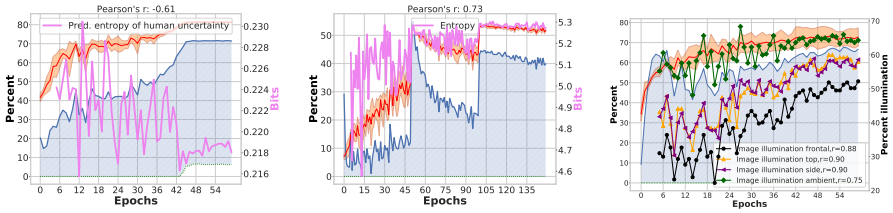


Fig. 6: **ImageNet DenseNet**: Agreement visualization on *test set*

In fig. 7, we exemplary visualize the test agreement for the three datasets CIFAR10, Pascal and KTH-TIPS2b and in fig. 6 for ImageNet. We observe that for all four datasets there is sufficiently high agreement on the test set. Not surprisingly, the standard deviation of the accuracy is higher than for all train sets. In analogy to Pascal train set results, we see jumps in accuracy and agreement



(a) **CIFAR10** Uncertainty (b) **Pascal** Image Entropy (c) **KTH-TIPS2b** Illumination

Fig. 7: Agreement and selected dataset metrics on the *test sets* of CIFAR10, Pascal, and KTH-TIPS2b, based on DenseNet. Metric values are shown in **purple** (right y-axis), in correlation to accuracy (**red**), agreement (**blue** curve) and its difference to lower-bound (shaded **blue** area) (left y-axis).

where the learning rate has been lowered in steps. Tentatively, for CIFAR10, Pascal and KTH-TIPS we visualize some dataset metric correlations on the test set too. For CIFAR10, we visualize the entropy of the soft labels as a metric. It has been computed by Peterson *et al.*[8] only for the test set. We see a slight downward tendency such that the entropy of soft labels decreases over the course of training. For Pascal, we see that similarly to the train entropy in fig. 4 of the main body, there is a correlation present for the test entropy. Even more interesting is the correlation of illumination on the KTH-TIPS2b dataset. Further, in fig. 5e of the main body we have seen that frontal illumination is learned slower than other kinds of illumination on the train set, in fig. 7 we see that for the test set this tendency is even more nuanced such that agreement is highest on the ambient illumination type and lowest on the frontal illumination type. A thourough analysis though, when dataset metric correlations are present/absent on the test data and how well they correlate with those on the train data is left for future work.

C Additional evaluation of correlations for CIFAR10 and Pascal

As mentioned in the main section, the range of fluctuations of CIFAR10 dataset metrics is negligible and therefore it is hard to judge the correlations between agreement and dataset metrics. To elaborate, the entropy is almost the same around 6.5, while sum of edge strenghts, segment count and percentage of DCT coefficients decrease slightly. The CIFAR10 distributions of dataset metrics indicate that for entropy, uncertainty and segment count, the distribution of values centers on a couple of values and, hence, there is no diversity, which can be reflected in agreement correlations. Further, since the dataset contains highly downsampled images, neither the presence of high frequencies, nor meaningful

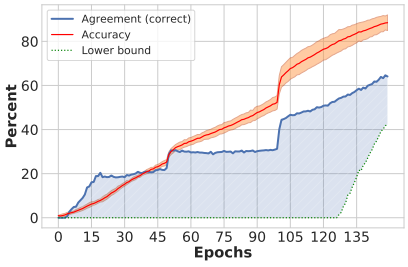


Fig. 8: **Pascal ResNet**: Agreement visualization on *train set*

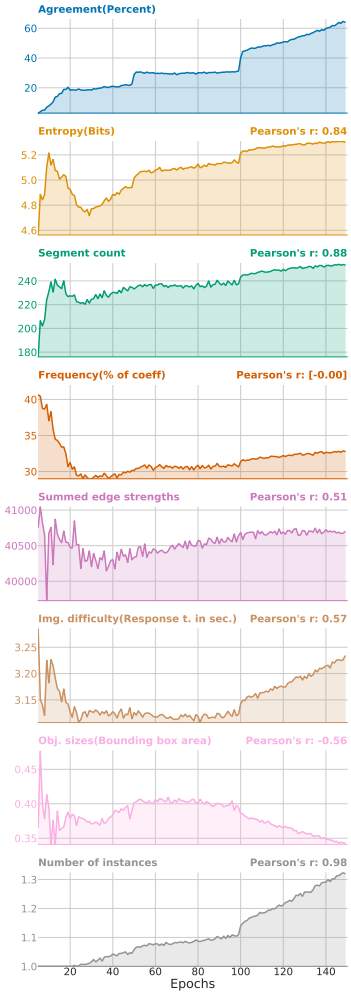


Fig. 9: **Pascal ResNet**: Dataset metrics correlations on *train set*

Fig. 10: **Pascal ResNet**: Agreement and dataset metrics correlations

edge strengths are expected. Hence, the direction of correlations is the same as for the texture dataset KTH-TIPS2b and opposite of Pascal, which also contains objects as CIFAR10 does.

To further support our results we, in addition to DenseNet, trained 5 ResNet50 networks on Pascal and computed with the same experimental setup the agreement (see fig. 8), as well as the dataset metrics correlations (see fig. 9). We see that ResNet learns more slowly than DenseNet (with the experimental setup chosen for DenseNet), but the general metrics tendency, when agreement ap-

proximately reaches 20%, remains the same as for DenseNet in fig. 4 of the main body. The strength of the correlation, measured by the Pearson correlation coefficient, is not as high for ResNet, as for DenseNet. For the frequency dataset metric it is even absent. The value is in brackets, because the corresponding *2-tailed p-value* is bigger than 0.001. For Pearson correlation coefficient between agreement and the given dataset metric, the null hypothesis is that both are uncorrelated. The higher the p-value, the more the null hypothesis is supported, The lower p-value supports the presence of a correlation.

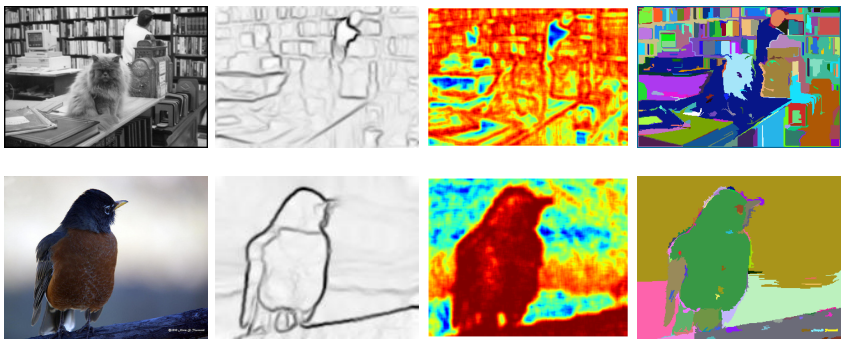
It would be interesting to further study both the initial learning phase when agreement is low, as well as the subsequent learning phase which this paper primarely was focused on.

D Dataset metrics

In this section we first give more details on how exactly we computed the dataset metrics and then visualize them on the example of ImageNet, in addition to the Pascal examples presented in the main paper, as well as visualize the matrices of DCT coefficients for those examples.

Let us start with the computation of the dataset metrics, evaluated in the main paper:

- **Segment count**: Felzenszwalb and Huttenlocher [2] introduce a graph-based image segmentation algorithm into regions, which can be summed up to get a segment count - a numer of segments in the image. First, images are smoothed with a Gaussian kernel of σ standard deviation, then image regions are compared for similarity at a certain scale k and merged if similar, subsequently small regions of size *min* are filtered out. Hence, the most important parameter is the scale k, larger value means preference for larger components. We used default parameters for the segmentation.
- **Sum of edge strengths**: Isola *et al.*[5] compute semantically meaningful boundaries (between objects) in an image based on statistical pixel dependencies (pointwise mutual information). The resulting edge strengths (edge contours) can be summed to get one value characterizing the amount of edges in the image.
- **Mean image intensity entropy**: Image intensity entropy for grayscale images is computed by sliding a window of a certain size k (in our case 10) and then averaging the local entropies. Similar to the case of segment count, the window-size reflects how much noise to ignore in the image.
- **Percentage of important DCT coefficients**: DCT coefficient matrix quantifies the spatical frequency in vertical and horizontal directions. Usually, lower frequency coefficients exhibit greater values. On the basis of this matrix we compute the percentage of DCT coefficients which contain 99.98% of the energy in the image, computed by comparing the norm of the first c sorted absolute values of DCT coefficients against the norm of all coefficients. In other words, this metric calculates how many coefficients are needed to reconstruct the image to a sufficiently high degree.



(a) Image (b) Edge strengths (c) Entropy (d) Segments

Fig. 11: Visualization of metrics on ImageNet

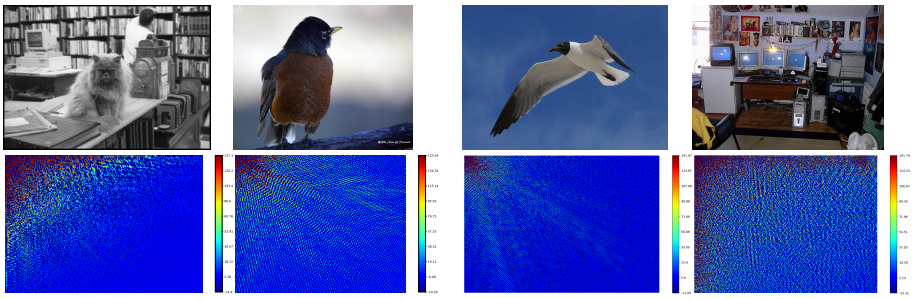


Fig. 12: Visualization of DCT matrix on ImageNet and Pascal examples

Similar to fig. 3 of the main body, which visualizes the computed dataset metrics on two selected images from the Pascal dataset, we also selected an 'easy' and 'difficult' image from ImageNet to visualize the metrics in fig. 11, as well as computed the DCT coefficients matrix for both chosen Pascal and ImageNet examples in fig. 12. What we see is that the more cluttered the image, the more irregular the entropy and segment image becomes. Cluttered images lead to higher amount of edges, but the edge strenghts of non-cluttered ones can be more prominent, which in summation may lead to similar sum of edge strengths. The DCT coefficients in fig. 12 show that the more clutter there is, the higher the coefficients in all directions. With less clutter, but more prominent horizontal or vertical variations in the image, like the wings of the bird, lead to higher values in the DCT coefficient matrix for these horizontal and vertical directions.

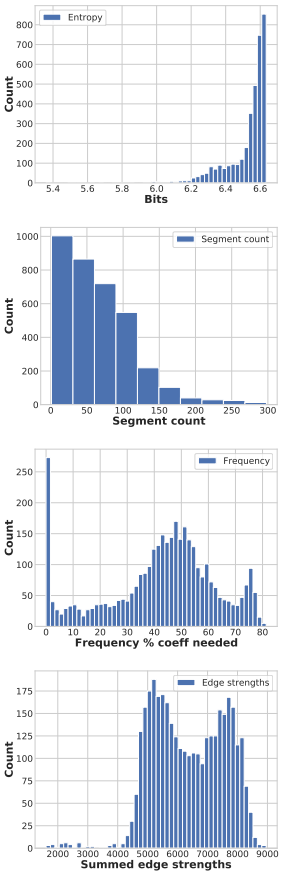


Fig. 13: **KTH-TIPS2b**: Dataset metrics histograms on *train set*

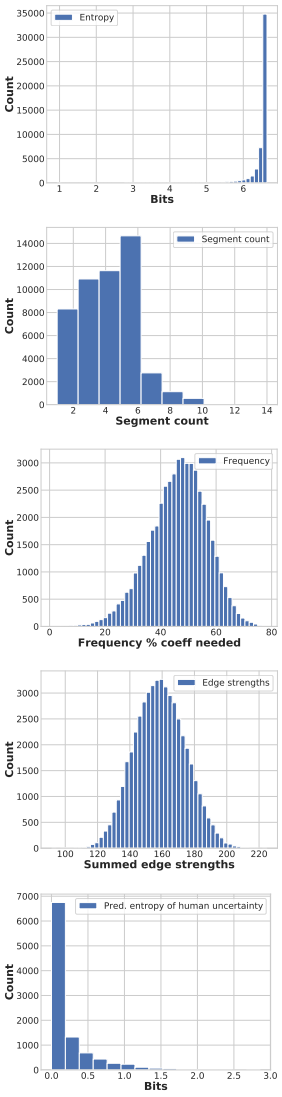


Fig. 14: **CIFAR10**: Dataset metrics histograms

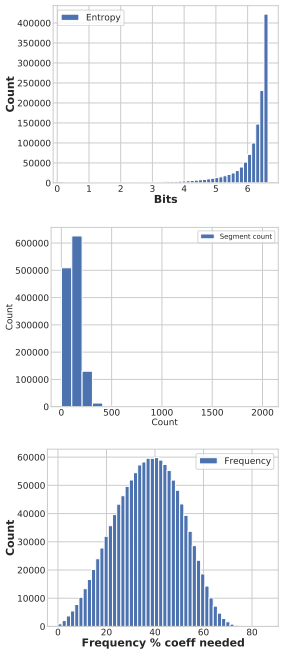


Fig. 15: **ImageNet**: Dataset metrics histograms on *train set*

Fig. 16: Dataset metrics histograms on *train set* for **KTH-TIPS2b**, **CIFAR10** and **ImageNet**, as well as for the *Pred. entropy of human uncertainty* on the *test set* of **CIFAR10**

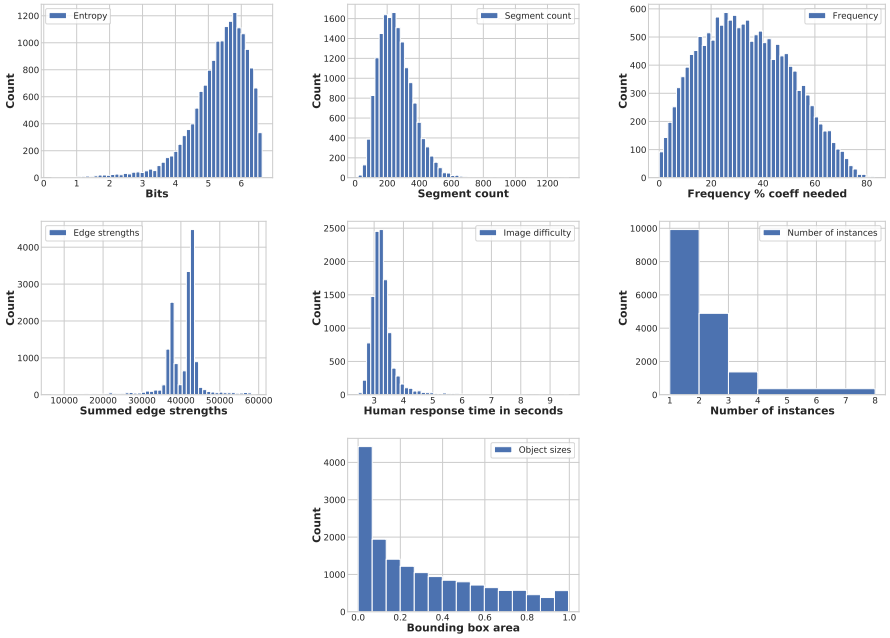


Fig. 17: **Pascal**: Dataset metrics histograms on *train set*

E Dataset metrics histograms

In order to assess the relevance of the results reported in the main body, we computed the histograms of the dataset metrics for CIFAR10 (fig. 14), Pascal (fig. 17), KTH-TIPS2b (fig. 13) and ImageNet (fig. 15) train sets. The **CIFAR10** histograms show that the frequency and edge strengths distributions are Gaussian, while the entropy and segment count are more or less centered on one value. Particularly for the metrics, which do not recognizably follow a certain distribution, the interpretation of the correlations is more difficult. The histogram for the predictive entropy of human uncertainty is not for the train, but for the test set, the corresponding correlation is in fig. 7a. We see that predictive entropy is low for most instances.

The **ImageNet** histograms in fig. 15 resemble those of CIFAR10 more than those of KTH-TIPS2b or Pascal, in that there is no skew of the frequency Gaussian and the segment count distribution is irregular-shaped.

The **KTH-TIPS2b** histograms in fig. 13 are more nuanced. The frequency and edge strengths distributions show several peaks, while the entropy and segment count exhibit an exponential course.

The **Pascal** histograms in fig. 17 are skewed Gaussians for entropy, segment count, frequency and human response time, multi-peak Gaussian for edge strenghts, similar to KTH-TIPS2b, as well as more or less centered around one value for number of instances and bounding box area.

F Discussion on correlation vs. causation

An important extention of the study is identifying causal links instead of mere correlations. Our results in the main body and additional visualizations in the appendix demonstrate agreement during the learning process of neural networks, as well as its correlation to several dataset metrics. Still, there are several differences between chosen datasets, which makes an analysis of why a certain correlation (and in which direction) was present difficult. To control the variation in the data, one could as a next step generate data with specific image statistics, which allows for an intervention into the data-generating process. In this way, one can study the influence of specific data-generating factors on agreement, while keeping all the others constant, in order to understand, which changes in the data *caused* which observed correlations.

References

1. Byrt, T., Bishop, J., Carlin, J.B.: Bias, prevalence and kappa. Journal of Clinical Epidemiology **46**(5), 423–429 (1993)
2. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient Graph-Based Image Segmentation. International Journal of Computer Vision **59**, 167–181 (2004)
3. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. International Conference on Computer Vision (ICCV) (2015)
4. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2261–2269 (2017)
5. Isola, P., Zoran, D., Krishnan, D., Adelson, E.H.: Crisp boundary detection using pointwise mutual information. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **8691 LNCS**(PART 3), 799–814 (2014)
6. Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images. Tech. rep., Toronto (2009)
7. Loshchilov, I., Hutter, F.: SGDR: Stochastic Gradient Descent With Warm Restarts. In: International Conference on Learning Representations (ICLR) (2017)
8. Peterson, J.C., Battleday, R.M., Griffiths, T.L., Russakovsky, O.: Human uncertainty makes classification more robust. International Conference on Computer Vision (ICCV) (2019)
9. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: International Conference on Learning Representations (ICLR) (2015)
10. Smith, L.N.: Cyclical learning rates for training neural networks. In: Winter Conference on Applications of Computer Vision (WACV) (2017)
11. Smith, L.N., Topin, N.: Super-convergence: Very fast training of neural networks using large learning rates. International Conference on Learning Representations (ICLR) (2018)