The power of ensembles for active learning in image classification Supplementary material

William H. Beluch
BCAI*

Tim Genewein
BCAI

Andreas Nürnberger
University Magdeburg

Jan M. Köhler
BCAI

The supplementary material provides details, additional results, and further comparisons.

Shaded areas in the plots denote $\pm$ one standard deviation.

1. Additional results for AL on image data

In Figure A1 we compare all of the acquisition functions used in [16] when using uncertainties from either MC dropout, an ensemble, or a single network. For both MC dropout and ENS, and on both MNIST and CIFAR-10, Variation Ratio performs best. The differences between the acquisition functions are more pronounced in MNIST, especially in the beginning stages. In Figure A1c and A1f we compare the geometric approaches [52] and [61] with the corresponding baselines: random selection, and the entropy of the softmax outputs of a single network. Both geometric approaches on MNIST perform similar to random. On CIFAR-10, both geometric functions perform worse than random, however the representativeness approach is consistently a few percentage points more accurate than the coreset approach.

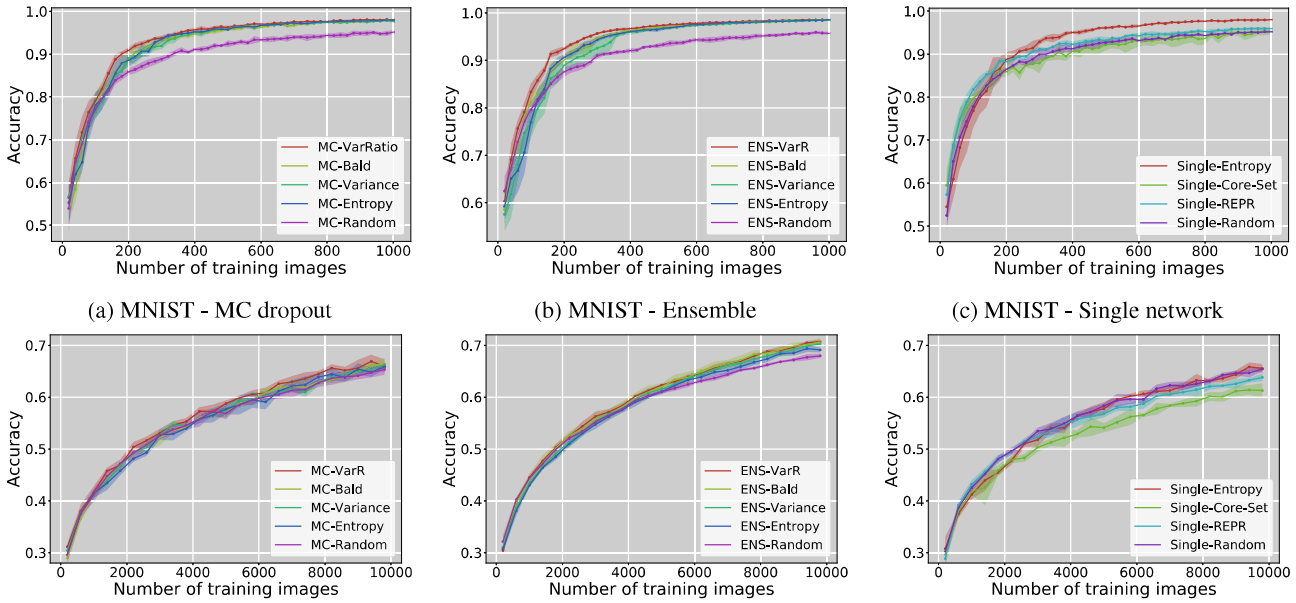
We show a comparison of the uncertainty methods and acquisition functions also on CIFAR-10 with the K-CNN architecture in Figure A2. The results are qualitatively the same as for CIFAR-10 trained on the DenseNet (Figure 1b).

The AL setup used in this study has many associated hyperparameters. Generally, the hyperparameters chosen were based on previous work, or chosen based on early qualitative results or computational efficiency. To show that tuning most hyperparameters (within reasonable range) has little effect, we run the AL experiment on CIFAR-10 using the K-CNN architecture for multiple settings (Figure A3). The hyperparameters explored are: number of images per acquisition step (with the subset pool size always being 10 times the acquisition size), number of images in the subset pool, number of forward passes in the MC dropout setting, number of networks in the ensemble, and dropout rates for the MC dropout setting. Most tested hyperparameters have little effect, with the exception of using only 2 or 5 forward passes for MC-Dropout, only 3 classifiers in an ensemble, or deviating strongly from the dropout rate of 0.25 / 0.5 after convolution / dense layers. In these cases the performance gets slightly worse.

In Figure A4 we additionally show results for Monte-Carlo Dropout with increased capacity on CIFAR-10 with the K-CNN network. The number of filters for the convolutional layers and the number of neurons for the dense layers is increased so after applying dropout with rate of 0.25 and 0.5, respectively, the same number of activations are present compared to a network used in the ensemble-based approach. For MC-VarR there is no benefit, though for MC-dropout with entropy as an acquisition function there is a small benefit. The performance of the ENS-VarR is not reached, though, MC dropout has 25 forward passes and on average the same amount of activations as the ensemble-based approach with 5 members.

*Bosch Center for Artificial Intelligence. First name.last name@de.bosch.com

*Bosch Center for Artificial Intelligence. First name.last name@de.bosch.com



(a) MNIST - MC dropout (b) MNIST - Ensemble (c) MNIST - Single network

(d) CIFAR - MC dropout (e) CIFAR - Ensemble (f) CIFAR - Single network

Figure A1: Test accuracy as a function of acquired images using different acquisition functions on MNIST (panels a, b, c. S-CNN architecture) and CIFAR-10 (panels d, e, f. K-CNN architecture). See main paper for a detailed description of the acquisition functions, "Random" corresponds to random selection of data-points which does not depend on the predictive uncertainty. In (panels a, d) we compare different acquisition functions for MC Dropout uncertainties with 25 forward passes. In (panels b, e) same as before but with uncertainties provided by an ensemble of five networks. In (panels c, f) the softmax-entropy of a single network is used as an uncertainty measure. Solid lines show results averaged over five repetitions. The shaded area indicates the standard-deviation band across these five runs ($\pm$ one standard-deviation).

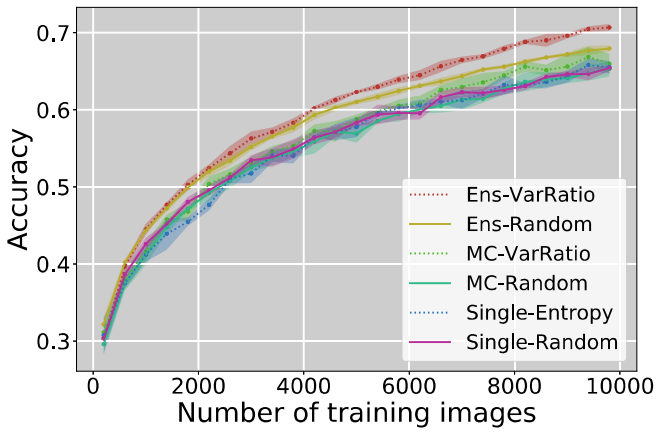
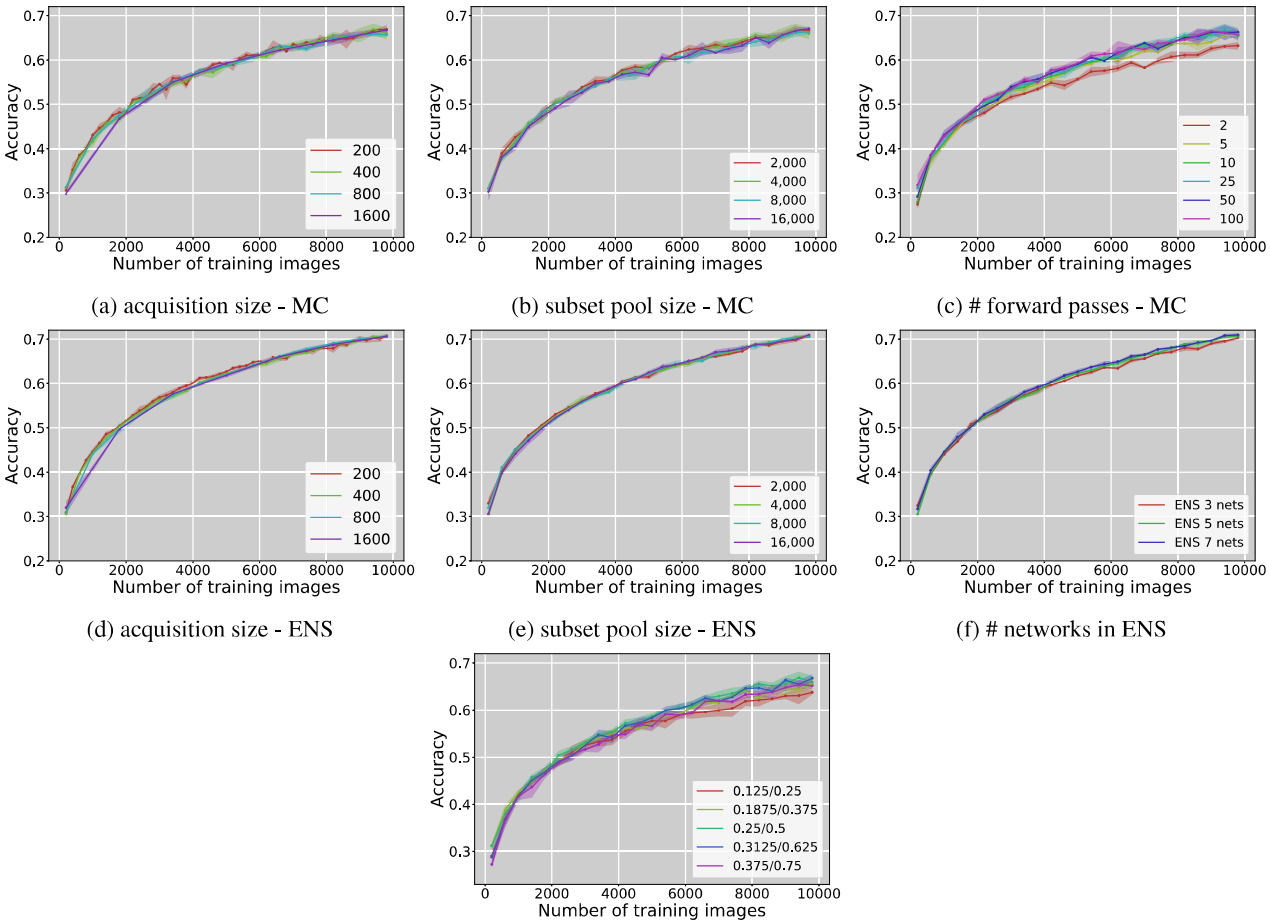


Figure A2: Test accuracy as a function of acquired images. Variation Ratio for MC dropout, Variance for the ensemble, and two density based acquisition functions are compared to its respective random acquisition function with the simple K-CNN architecture for CIFAR-10.



(a) acquisition size - MC

(b) subset pool size - MC

(c) # forward passes - MC

(d) acquisition size - ENS

(e) subset pool size - ENS

(f) # networks in ENS

(g) dropout rates

Figure A3: Test accuracy over acquired images for different hyperparameter settings on CIFAR-10 using the simple K-CNN architecture. Unless otherwise noted, 25 forward passes are used for MC-Dropout, and 5 classifiers for the ENS approaches. Variation Ratio is the acquisition function used for all experiments.

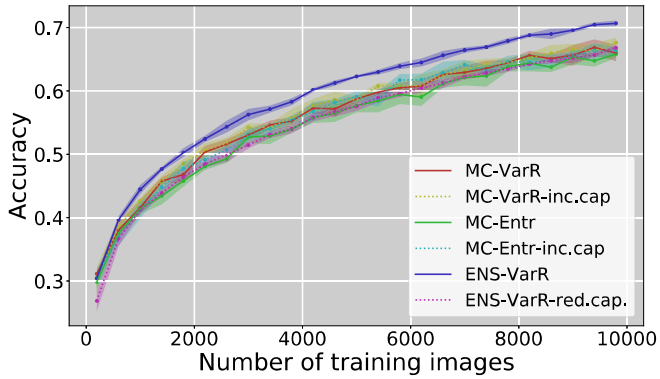


Figure A4: Increasing capacity for Monte-Carlo Dropout in isolation. Plots show test accuracy on CIFAR-10 with K-CNN using Variation Ratio and Entropy as an acquisition function. MC-VarR-inc.cap: MC dropout with Variation Ratio as acquisition function. The number of filters / neurons (conv / dense layer) was increased so the same number of activations are present after dropout rate of 0.25 / 0.5. There is a small benefit for MC-Entropy, especially for little data, though no benefit is visible for VarR.

2. Implicit ensembling

We evaluate the use of three implicit ensemble methods described in the literature as opposed to using a full ensemble of classifiers for uncertainty estimation. The approaches used are:

Snapshot ensembling proposed by [28] is a method to train an implicit ensemble using a cyclic learning rate schedule to converge to different local minima. The **diversity encouraging ensemble (DEE)** by [60] uses a base network trained for a small number of epochs as the initialization for n different networks, each trained using different dropout rates to encourage diversity. In the **splithead approach** by [48] each member of the ensemble shares the same base model, with only a final dense layer being unique to each classifier.

The different implicit ensemble methods were implemented with little deviation from the original specifications. For the Snapshot ensembles, we use the cyclic learning rate schedule based on the shifted cosine function, with an initial learning rate of 0.1, and stochastic gradient descent. Each Snapshot is trained for the total number of epochs / N, where N is the desired number of classifiers. To be consistent with our previous experiments, N is equal to 5, and the total number of epochs is 150. Details of the shifted cosine function and the rationale of the method can be found in [28].

For the diversity encouraging ensembles (DEE), we again match the total number of epochs and classifiers to our previous experiments. The network is first trained for 25 epochs with stochastic gradient descent, with an initial learning rate of 0.01, and momentum of 0.9. A cyclic learning rate schedule using the shifted cosine function is again used, and 5 networks are trained initialized with the weights from the first training phase (the 25 epochs), and the 0.01 learning rate. The different dropout rates for each network are: $(0.25, 0.25, 0.5)$, $(0.25, 0.5, 0.5)$, $(0.25, 0.25, 0.25)$, $(0.25, 0.3, 0.4)$, $(0.25, 0.5, 0.25)$, $(0.2, 0.4, 0.4)$, with the first in the list being the rates used in the initial implementation and the first training phase. The dropout schedules were chosen empirically, as there is no heuristic provided in the paper. More details can be found in the original publication [60].

The split-head ensemble approach is inspired by [48]. We share the weights of every layer except for the final fully-connected layer before the output layer. All other training specifications do not change from the baseline experiments. The paper also explores the idea of bootstrapping, however found that in practice it is best to provide each head of the network with as much data as possible - considering that we use small amounts of data in our AL experiments, we opt to not use any bootstrapping.

All experiments on implicit ensembles were run for 5 repetitions.

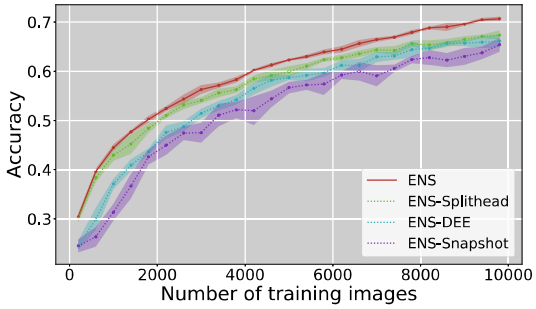


Figure A5: Test accuracy over acquired images using different implicit ensembling techniques (CIFAR with K-CNN).

In figure A5 we compare these implicit ensembling methods against a plain ensemble using Variation Ratio as an acquisition function. All implicit ensemble techniques achieve a significantly lower accuracy compared to the standard ensemble. The question of why implicit ensembling methods perform worse is left for future research.

3. Additional results for uncertainty calibration

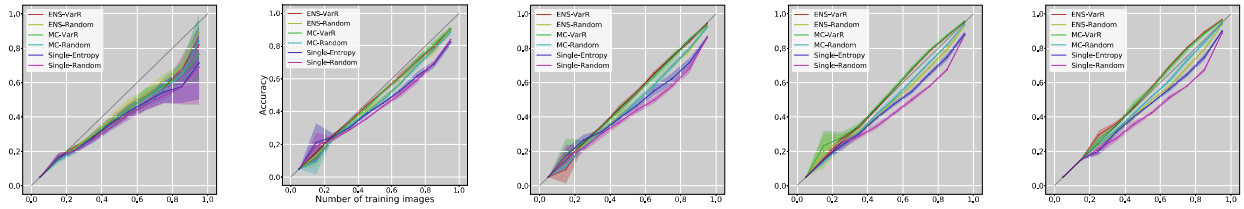
To assess calibration ([7]) quality we determine whether the expected fraction of correct classifications (as predicted by the model confidence, i.e. the uncertainty over predictions) matches the observed fraction of correct classifications. When plotting both values against each other, a well-calibrated model lies close to the diagonal. Results are shown in Figure 3 in the main paper after 3 acquisition steps. Figure A6 shows calibration plots for randomly initialized networks and after 6, 12, 18 and 24 acquisition steps respectively.

The mean-squared-error (MSE) between the main diagonal and the calibration line is used to quantitatively express calibration quality of a model (see Figure 3b in the main paper).

Additional measures are the negative log likelihood (NLL) and the Brier score. Quantitative results for the latter two measures are shown in Table 3 in the main paper for the first three acquisition steps. The results across all acquisition steps are shown in Figure A7.

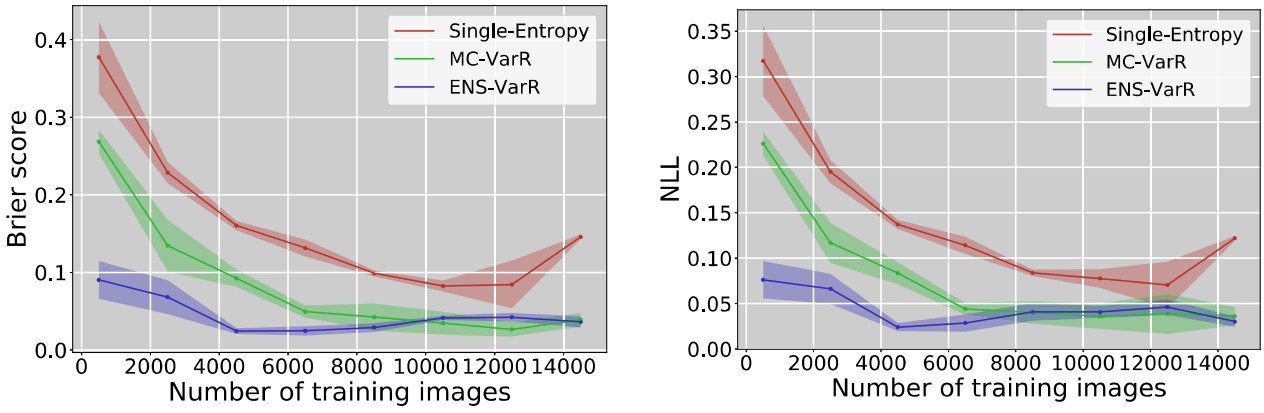
4. Uncertainty Decomposition

Predictive uncertainty can be decomposed over a label into data-dependent aleatoric uncertainty (which is intrinsic to the data and cannot be resolved, even in the limit of infinite training data) and epistemic uncertainty (uncertainty over the correct model-parameters that goes to zero in the limit of infinite data). To incorporate the separation between aleatoric and epistemic uncertainty into AL, input-dependent variance (over predictions) is learned as an extra output that splits off of the network at the final layer before the prediction, as first introduced in [47]. This technique has recently been implemented for the regression case in



(a) Initialization (b) After 6 steps (c) After 12 steps (d) After 18 steps) (e) After 24 steps)

Figure A6: Calibration plots at different acquisition steps (200-2,600-5,000-7,400-9.800 images) for different models. "Initialization" refers to fully trained networks using the initial pool of labeled data (drawn by random selection). MC-VarR: MC Dropout with Variation Ratio acquisition. ENS-VarR: Ensembles with Variation Ratio acquisition. Single-Entropy: single network with softmax-entropy for the acquisition. The lines labeled with "-Random" correspond to random acquisition. Perfectly calibrated models would lie on the main diagonal (dashed line). Results show DenseNet on K-CNN.



(a) Brier score (b) NLL

Figure A7: Negative Log Likelihood (NLL) and Brier score for different acquisition steps for a single network with softmax-entropy acquisition, MC Dropout and an ensemble (the latter both with Variation Ratio acquisition). Particularly in the low-data regime, ensembles lead to better uncertainties. After a sufficient amount of training MC Dropout and ensembles perform equally well. Uncertainties of a single network are consistently worse. These results are consistent with the calibration score (MSE) shown in the main paper (Figure 3b). Results are shown for DenseNet on CIFAR-10.

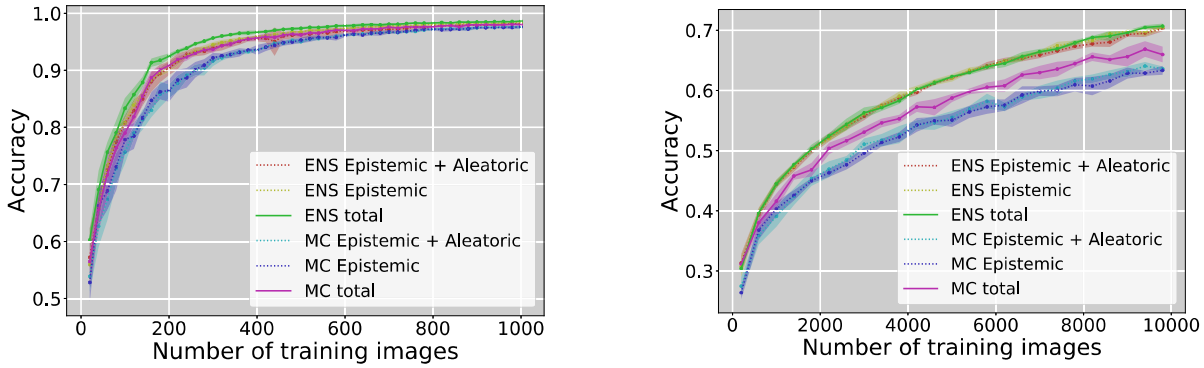
[36] using MC Dropout for estimation of the total predictive uncertainty, as well as for ensemble-based uncertainty estimation in [40].

Decomposing the uncertainty output from the neural network potentially is useful for active learning: points with a high epistemic uncertainty should be selected, as this type of uncertainty can be reduced with more data, while points with high aleatoric uncertainty should be avoided, as this uncertainty stemming from the input dependent noise will not decrease with more data. To practically incorporate the aleatoric uncertainty into an acquisition function, we tried two approaches, both of which involve learning the variance of the inputs with the split-head architecture initially proposed in [47], using the adaptation to the classification case and the associated loss function (eq. 9) proposed in [36]. The first method learns the aleatoric variance, but still only uses the epistemic component of the uncertainty to select new points (corresponding to the previous acquisition functions used). The second method is a hybrid approach in which first the N most epistemically uncertain points (based on of the softmax outputs) are chosen, and from this subset the $N/4$ points with the lowest aleatoric uncertainty are selected.

. For regression, we use the loss function from [36] with $s_i = log(\hat{\sigma_i^2})$ being the learned variance (as opposed to using a fixed variance in traditional learning)

$$L(\theta) = \sum_i \frac{1}{2}\exp(-s_i)||\mathbf{y_i} - \hat{\mathbf{y}}_\mathbf{i}||^2 + \frac{1}{2}s_i \qquad (8)$$

For classification, we also use the implementation in [36]. The logit output $\mathbf{f}_i$ (input to the softmax activation) is corrupted with Gaussian noise with variance $\sigma_i^{\mathbf{W}}$ and fed to the softmax activation S times. Intuitively, both of these loss functions encourage the model to increase the input-dependent, learned variance when the model predicts incorrectly with high confidence, and to decrease the variance



(a) MNIST (b) CIFAR

Figure A8: Test accuracy as a function of acquired images using acquisition functions that incorporate a decomposition of the total uncertainty into its aleatoric and epistemic components. *Total* refers to the original acquisition function, in which the network does not learn any variance, and the uncertainty is based on the average softmax outputs. *Epistemic* refers to learning the variance as a separate parameter, but only using the softmax based uncertainty in the acquisition function as in the previous method. *Epistemic + Aleatoric* refers to the hybrid approach described in the text. All functions use the Variation Ratio as the method for measuring epistemic uncertainty. Results are shown for MNIST on S-CNN and CIFAR-10 on K-CNN.

when the model predicts correctly with high confidence.

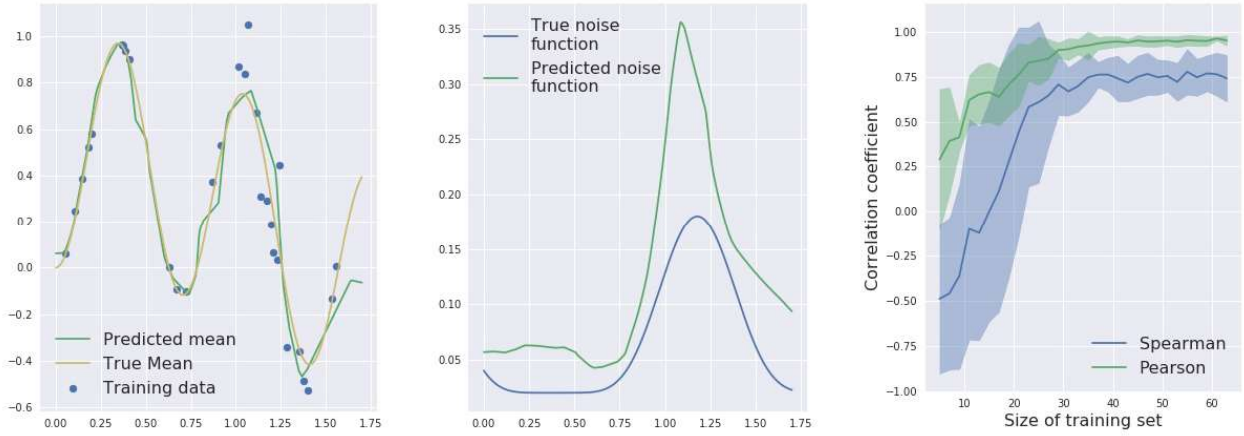
$$\mathbf{x}_{i,t} = \mathbf{f}_i^{\mathbf{W}} + \sigma_i^{\mathbf{W}} \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,1)$$
$$L_x = -\sum_i \log \frac{1}{S} \sum_s \exp(\hat{x}_{i,s,c} - \log \sum_{c'} \exp \hat{x}_{i,s,c'}) \tag{9}$$

We show results in Figure A8. The addition of the aleatoric uncertainty, in both methods described above, led to slightly worse results for both ENS and MC-Dropout for MNIST. On CIFAR-10, the addition in the ENS case results in virtually identical performance, while for MC-Dropout there is a notable drop in performance. Some potential reasons for this performance are: the combination of the epistemic and aleatoric uncertainty in the actual acquisition function is not adequate, the datasets used (MNIST and CIFAR-10) inherently do not contain much input noise, and thus the aleatoric measure adds no information (or inaccurate information), and the aleatoric approximation is poor, especially in the earlier iterations with little data.

To investigate the third option, we perform an experiment with the toy regression function used in [47]. A similar regression example was shown recently in [9] (albeit with a different underlying function), wherein the use of the decomposition of uncertainty for AL was also posited. We present an example toy regression function with a specific sampling regime for the training data, and indeed, the decomposition would work well for the situation provided. However in the early stages of AL, the function approximation can be poor (unlike the example just discussed). To investigate this, we start with five random points as the training data, and randomly add two points each iteration. We find that in the early stages of AL, in which the function approximation is poor or very generalized, the learned variance is similarly not very accurate, as this is also a learned parameter (see Figure A9). As in AL we are mainly concerned with relative uncertainties, and not absolute uncertainties, the Spearman rank correlation between the two curves is, therefore, informative; at low amounts of data, this correlation is spurious and noisy. Extrapolating to the classification case, a similar situation could occur; even though the loss function is rather different, the aleatoric variance is still a learned parameter.

5. AL for diagnosis of diabetic retinopathy

The retinal images shown in Figure A10 show images containing lesion areas which indicate diabetic retinopathy (images (a) and (b)). Different lesion types, like small red dots, micro-aneurysms, hemorrhages are responsible for diabetic retinopathy. For a layman it is hard to spot these lesions, especially when the disease is at an early stage, as the lesions tend to grow the longer diabetic retinopathy stays untreated.



(a) (b) (c)

Figure A9: Toy regression example, averaged over 10 repetitions with different data subsets. **a)** The underlying sinusoid function, and the predicted mean based on the training data seen at a specific iteration (62 data-points). **b)** The predicted variance function compared to the true variance function at 62 data-points. **c)** The correlation between the predicted and true variance function, as a function of the number of points in the training data (randomly selected).

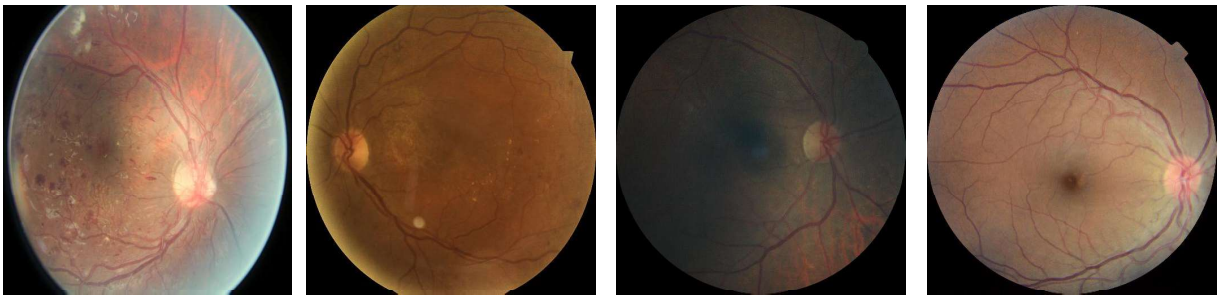


Figure A10: Eye fundus example images from [11] which were used in a 2015 Kaggle challenge. The two images on the left show signs of diabetic retinopathy, whereas the other two images are healthy. For training the images were cropped and scaled to 512x512 pixel size.