Know When You Don't Know: A Robust Deep Learning Approach in the Presence of Unknown Phenotypes

Oliver Dürr,[1,2] Elvis Murina,[1] Daniel Siegismund,[3]
Vasily Tolkachev,[1] Stephan Steigele,[3] and Beate Sick[1,4]

[1]Institute of Data Analysis and Process Design, ZHAW Winterthur, Winterthur, Switzerland.
[2]Institute for Optical Systems, HTWG Konstanz, Konstanz, Germany.
[3]Genedata AG, Basel, Switzerland.
[4]EBPI, University of Zurich, Zurich, Switzerland.

ABSTRACT

Deep convolutional neural networks show outstanding performance in image-based phenotype classification given that all existing phenotypes are presented during the training of the network. However, in real-world high-content screening (HCS) experiments, it is often impossible to know all phenotypes in advance. Moreover, novel phenotype discovery itself can be an HCS outcome of interest. This aspect of HCS is not yet covered by classical deep learning approaches. When presenting an image with a novel phenotype to a trained network, it fails to indicate a novelty discovery but assigns the image to a wrong phenotype. To tackle this problem and address the need for novelty detection, we use a recently developed Bayesian approach for deep neural networks called Monte Carlo (MC) dropout to define different uncertainty measures for each phenotype prediction. With real HCS data, we show that these uncertainty measures allow us to identify novel or unclear phenotypes. In addition, we also found that the MC dropout method results in a significant improvement of classification accuracy. The proposed procedure used in our HCS case study can be easily transferred to any existing network architecture and will be beneficial in terms of accuracy and novelty detection.

Keywords: screening, classification, deep learning, imaging

INTRODUCTION

Early drug research increasingly relies on complex phenotypic assays as biologically relevant model systems. The goal of the bioassays is a pharmacological assessment of wanted effects from chemical molecules on living objects, for example, cells or higher level structural aggregates. Application areas range from cell segmentation[1] to the classification of dozens of cellular phenotypes.[2–6] From a screening application viewpoint, image-based drug discovery often relies on *a priori* knowledge (*e.g.*, manifested by respective positive and/or negative control compounds with known mode-of-action) to find substances that induce a certain visible effect (phenotype) and thereafter quantify the observed response by using multiple doses of a compound.[7,8] This detection and quantification task can be tackled on different levels: (1) on a single-cell basis[2,4,5] or (2) considering multiple cells/patches[6] up to the whole well (field) images.[3] The applied convolutional neural networks (CNNs) are known for their outstanding performance in computer vision tasks[9] and are seen as the most successful candidate to support modern high-content screening (HCS) in multiple flavors.[10] A typical workflow consists of curation of the training data (distinct phenotype classes), training of a corresponding CNN and its application on production image data.[10]

In many HCS application scenarios, the possible phenotypic endpoints are unknown. In practice, a trained scientist would manually detect and define a certain number of phenotypes, which in a machine learning setup would define training classes on which a CNN is trained for production application in drug discovery campaigns. In an ideal world, the defined training classes resemble all phenotypes present in the production setting, which is almost never true in practice. There are many reasons why phenotypes and their number may be different in assay development and in production analysis. Subtle changes in experimental protocols may trigger slightly altered differentiation of cells, different cell lines might be used, or just a larger chemical space in use can trigger phenotypes that could not be discovered with tool compound sets alone. The situation gets even more complex when a cell painting assay is analyzed, not only working with few target-specific labels but with a large number of compartment-specific labeled proteins as well. Hence, in this setting, the number of detectable possible phenotypic endpoints is especially large and often not known beforehand.

DOI: 10.1089/adt.2018.859

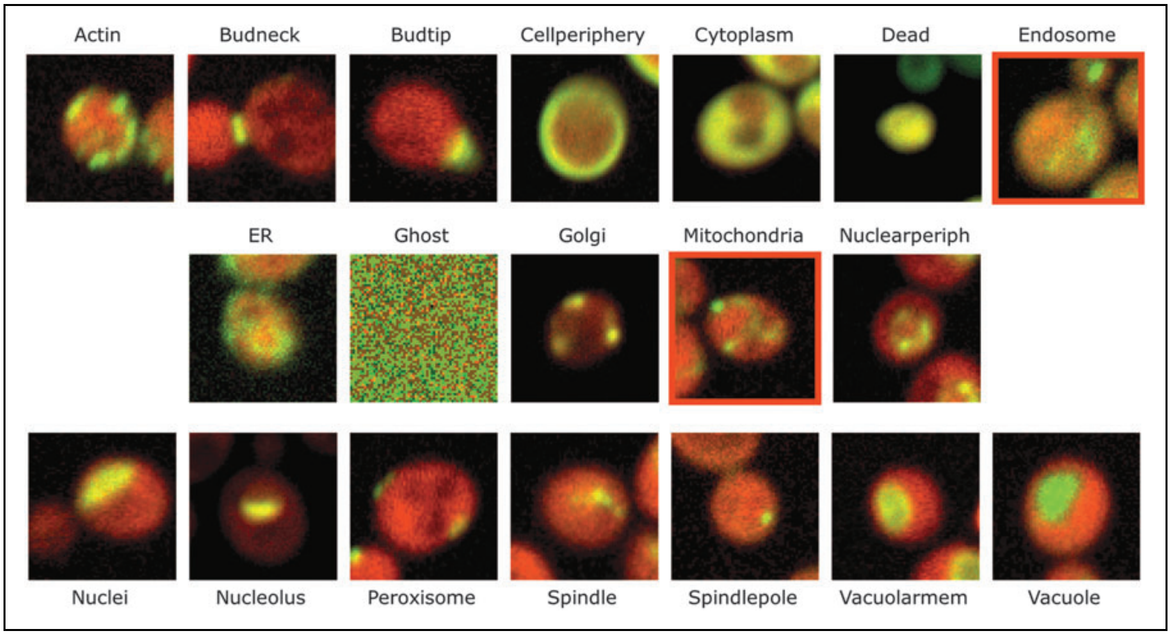

Fig. 1. Exemplary images of yeast cells with localized GFP-tagged proteins for the 19 subcellular compartments. The compartment localization classes "mitochondria" and "endosome" are *highlighted*. GFP, green fluorescent protein.

The problem of not knowing that one misses important phenotypes can become a real challenge for black box procedures such as deep learning. In this study, a CNN trained for classification usually uses a so-called softmax function at the last layer,[9] whose operation provides a probability value p_k for each *known* class label that adds up to one over all labels. In contrast to the popular belief, these probabilities usually cannot be interpreted as a measure of uncertainty. Especially in cases where a classified image does not correspond to a known training class, the network cannot predict the correct class by definition (because it is unknown) and must assign the probabilities to wrong class labels. Astonishingly, even if classifying an image of an unknown class, one of the known classes often obtains a prediction probability close to one.

To overcome this particularly difficult challenge, one has to introduce a measure of uncertainty to indicate if a classified image has an unexpected structure and should potentially not be assigned to any of the available class labels. There are many examples for such uncertainty measures: the length of a 95% confidence interval (statistical modeling) or the 95% region of highest posterior density (in Bayesian statistics). However, in case of standard neural networks, such as CNNs, we have no uncertainty information of the predicted probability by default. We therefore propose to use a recently introduced Monte Carlo (MC) dropout method[11] developed as a practical, useful, and performant approximation of Bayesian neural networks, which naturally provide uncertainty information. The basic idea is to sample from a posterior distribution of the network weights and thus to construct a distribution of predicted probabilities. A broad and flat distribution across all classes indicates a general, unspecific class assignment, which hints that a classified object is not covered by the known training classes. In contrast, a sharp distribution can increase the certainty for the correct class assignment. In this article, we evaluate the correctness of a probability estimate of our network and introduce novelty detection in the context of image-based analysis (HCS and medical screening). We introduce the principle that enables quantifying uncertainty measures of a

Table 1. Experiment Settings

	Total no. of images	Mitochondria	Experiment w/o mitochondria	Endosome	Experiment w/o mitochondria and endosome
Training	21,882	1,500	20,382	1,500	18,882
Validation	4,491	642	3,849	220	3,629
Test	4,516	643	4,516	222	4,516

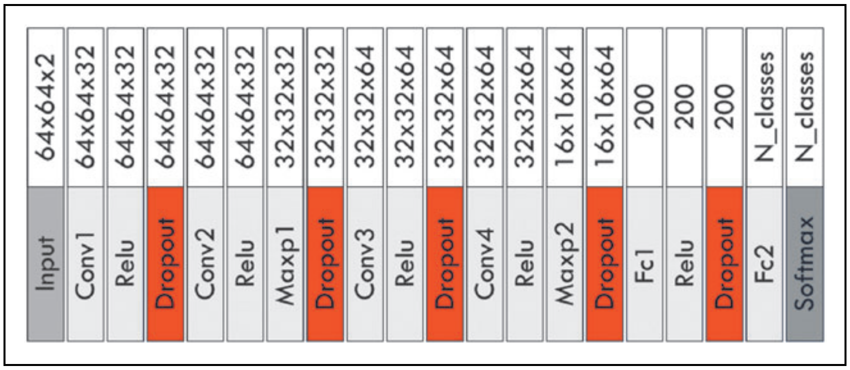

Fig. 2. CNN architecture. CNN, convolutional neural network.

CNN-based prediction and illustrate how to utilize them in a real-world HCS setting.

MATERIALS AND METHODS

Data Set

Budding yeast is a well-studied model system for protein expression and localization experiments. Genetically, yeast is especially simple since it has only 5,797 protein coding genes. We work with a public image data set of budding yeast known as open reading frame-green fluorescent protein (GFP) fusion collection consisting of 4,156 (out of 5,797 possible) GFP-tagged protein strains.[12] In each strain a different gene is fused to a green fluorescent protein plasmid, and hence, the corresponding protein is tagged with GFP allowing to detect its position in the cell via fluorescent imaging. The used images have two channels—the green channel shows the tagged proteins, and the red channel shows the cell body. Huh *et al.*[13] classified ~75% of the yeast proteome into 22 distinct subcellular localization categories. We used a data set that Kraus *et al.*[6] have utilized for other deep learning approaches and provided as a ZIP container.* This data set is only a subset of derived single-cell images and relabeled by Kraus *et al.* to achieve 19 localization classes (ACTIN, BUDNECK, BUDTIP, CELLPERIPHERY, CYTOPLASM, DEAD, ENDOSOME, ER, GHOST, GOLGI, MITOCHONDRIA, NUCLEARPERIPHERY, NUCLEI, NUCLEOLUS, PEROXISOME, SPINDLE, SPINDLEPOLE, VACUOLARMEMBRANE, VACUOLE). We treat the 19 localization classes as surrogates for different phenotypes. *Figure 1* shows example images of the yeast cells for all 19 protein localization classes. The data set has 21,882, 4,491, and 4,516 images for the training, validation, and test set, respectively.

Since we want to demonstrate an approach for phenotype novelty detection, we conducted two experiments. In the first, we removed all images with the localization class label "MITOCHONDRIA" from the original training and validation set (*Table 1*). In the second experiment, we removed in addition to "MITOCHONDRIA" images, all images of the class "ENDOSOME" from the training and validation set (*Table 1*). The test set in both experiments was left unchanged. *Table 1* below summarizes the data set sizes. We have picked these two phenotypes since they are represented with a reasonable high number allowing for a reliable evaluation of our methods. In a leave-one-out experiment, we repeated the experiment for each of the 19 phenotypes, each acting once as an unknown class (Supplementary Data; Supplementary Data are available online at www.liebertpub.com/adt).

CNN Model

We use a rather shallow network with ~3 million weights (*Fig. 2*). The network is constructed from convolutional blocks consisting of two convolutional layers followed by a max pooling layer. We stack two such building blocks, the first with 32 3×3 filters and the second with 64 3×3 filters. After these convolutional blocks, we add a first dense layer with 200 neurons and a second dense layer with 18 or 17 neurons depending on the number of classes in the experiment.

Rectified linear units (ReLu)[14] are used as activation functions. Dropout with a rate of 0.3 is applied within the convolutional and fully connected part of the network (*Fig. 2*). The network is built using Keras and available as IPython notebook.† To reduce overfitting, we augmented the data set by randomly applying rotations, shifts, and flips to the training images during training. Besides normalizing the pixel values to be between zero and one, no preprocessing has been performed. The final network is trained for 500 epochs. In both experiments, after around 500 epochs, a slight increase in the validation loss heralds the onset of overfitting.

We use the same trained network and operate it in two different prediction modes to do predictions on new images. In the first prediction mode, we do classical predictions by freezing the weights (no dropout during test time) and use the maximal predicted softmax probability as the probability of the predicted class. In the second prediction mode, we do MC dropout during test time, meaning that we use different dropout versions of the trained model on the same image and aggregate the received predictions (for details see below).

*http://spidey.ccbr.utoronto.ca/~okraus/DeepLoc_full_datasets.zip

†https://github.com/tensorchiefs/hcs_uncertainty

Point Estimates and Uncertainty Measures

In general, probabilistic classifiers are able to predict a probability for an input belonging to a certain class rather than just providing a binary decision. For a standard neural network architecture with a softmax activation in the final layer, the output p_k is interpreted as the probability that the input belongs to class k. If p_k is an unbiased estimator of the proportion with which we observe the predicted class, we call the model calibrated. The quality of the calibration can be investigated using calibration plots (Supplementary Data). To derive a classification decision, the input image is usually classified into the class with the highest p_k. However, our main focus here is to examine the ability of the network to signal if an image class has not been seen in the training set and should therefore not be classified to any of the available classes.

In addition to standard use case of dropout only during training of the network, we additionally use dropout during the test phase (MC dropout) where we pass the same image through different dropout-sparsified versions of the trained CNN, where in each version another random set of nodes is deleted or dropped. This procedure is theoretically rooted in the Bayesian neural network framework and was shown to approximately correspond to sampling from the posterior distribution of the network weights.[11] Using the MC dropout during test time, we obtain $i=1,\ldots, N_{dropout}$ samples of probability values p^*_{ik} for the $k=1,\ldots, K_{training}$ classes used in the training. Note that the p^*_{ik} are compositional data in the sense that their sum $\sum_k p^*_{ik}=1$ for each prediction run i. From the obtained sample of predictions we derive different measures. Specifically, we distinguish between estimates for the probabilities predicted for each class and for quantifying the uncertainty of the predicted probabilities.

Probability estimates. For a prediction in the standard (no MC dropout) use case, we freeze the learned weights and pass the image through the network only once. The resulting softmax output p_k is used as a probability estimate for the class k.

For prediction with MC dropout, we need to aggregate the received sample of MC predictions. In this study, we investigate two different aggregation methods. The first probability estimate is the mean over all predicted MC probabilities for class k.

$$p^*_k = \frac{1}{N_{dropout}} \sum_i p^*_{ik} \tag{1}$$

The second probability estimate is count based. We consider MC dropout evaluations as an ensemble of CNN classifiers and N_k counts in how many dropout variants the class k had the highest probability.

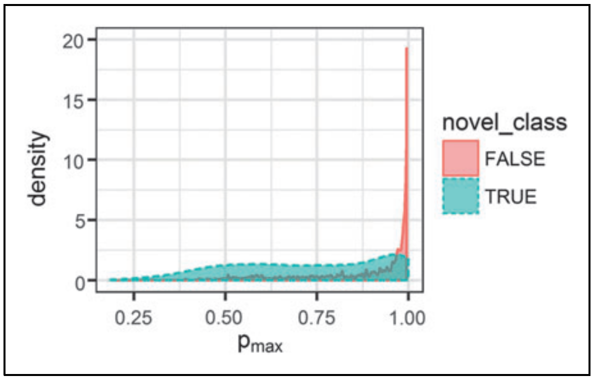

Fig. 3. Density estimation of classical p_{max} predictions for phenotypes present in the training set (novel_class = FALSE) and phenotypes not included in the training set (novel_class = TRUE).

$$f^*_k = \frac{N_k}{N_{dropout}} \tag{2}$$

In addition, we experimented with the multivariate maximum *a posteriori* (MAP) estimate calculated using kernel density estimators. However, first experiments showed inferior results and thus the MAP was not used in this study.

Uncertainty estimates. When we are interested in the overall confidence of a classification decision, we can use the probability of the predicted class as measure for the classification certainty:

$$p_{max} = max(p_k), \quad p^*_{max} = max(p^*_k), \quad \text{and} \quad f^*_{max} = max(f^*_k) \tag{3}$$

In addition, we can use estimates operating on all p_{ik} like the total standard deviation of the observed probabilities over all MC runs:

$$\sigma^* = \sqrt{\sum_k \frac{1}{N_{dropout}-1} \sum_i \left(p^*_{ik} - p^*_k\right)^2} \tag{4}$$

Another commonly used uncertainty measure is the (information) entropy of the distribution of probabilities p^*_k

$$PE^* = - \sum_k p^*_k \log_2\left(p^*_k + \epsilon\right) \tag{5}$$

where a small number $\epsilon = 10^{-14}$ has been added for numerical stability.

RESULTS AND DISCUSSION

We started our work by proving that the networks are calibrated and in both prediction modes—the classical and the MC dropout—we see quite calibrated predictions, provided

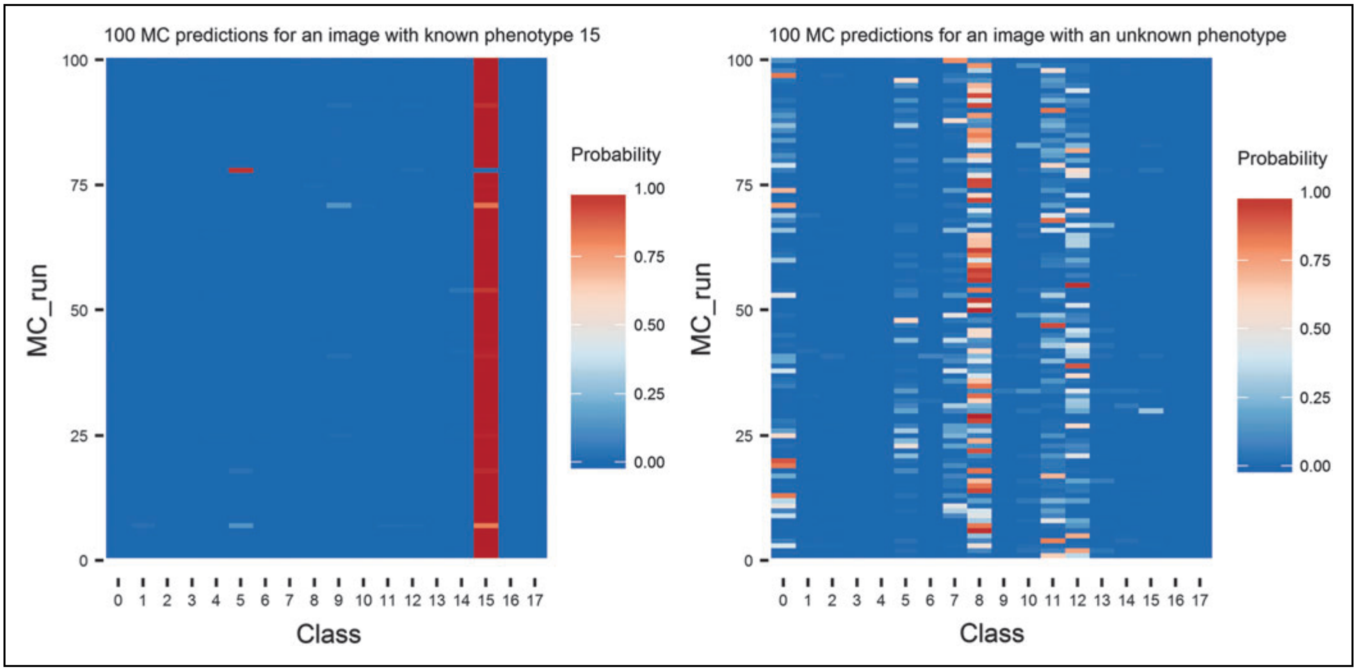

Fig. 4. The heatmaps visualize the probabilities for each of the class labels (*columns*) received from 100 MC runs (*rows*) of two test images in the experiment with "mitochondria" class left out. *Left*: the test image corresponds to the class that was represented in the train set, *right*: test images correspond to a novel class. MC, Monte Carlo.

that the input image belongs to a known class (Supplementary Data for details).

We now investigate how the models react when an unknown class not included in the training set is presented. We focus on the experiment in which the class "mitochondria" is not contained in the training set.

We first consider the case when no dropout has been applied during the test time. This is the current standard setting for classification in deep learning. As described above, usually an image under consideration is assigned to the class with the highest probability p_{max}. First, consider the images in the test set whose phenotype has been also present in the training set. In this case the accuracy is 0.9367 (0.9286, 0.9442).[‡] In *Figure* 3, the density of p_{max} for the phenotypes present during training is shown by a solid line. We see that the network assigns images to a certain class with a high probability. Now looking at the phenotype not present in the training (dotted curve) in *Figure* 3, we observe a much flatter distribution and quite a fraction also has high p_{max}. Note that there is a significant overlap of the two distributions. If we want to use a small p_{max} as an indicator for an unseen phenotype and set the threshold, for example, to $p_{max} = 0.8$, we would falsely call 43% of the novel class (area under the dotted curve for p_{max} between 0.8 and 1) as coming from a known phenotype. We quantify this intuition later using receiver operating characteristic (ROC) analysis and lift charts.

For MC dropout predictions, we now also apply dropout during the test time (MC dropout)—not only during training. Before we investigate the resulting prediction distribution, we have a closer look at the individual results of single dropout runs. *Figure* 4 shows the estimates p_{ik} for $i = 1 \ldots 100$ MC runs—left for an image of a cell phenotype that was seen in the training set and right for an image of a previously unseen phenotype.

If the class of the image is present in the training set, in all, but one, MC runs the highest probability is correctly assigned (left side *Fig.* 4). If the class was not present in the training (right side of *Fig. 4*), there is a significant spread in the predictions between the different MC runs and also the mean value of p_{max} is significantly smaller than 1.

We now systematically investigate the use of MC dropout on the accuracy and to quantify the uncertainty. MC dropout during the test phase allows for the definition of additional point estimates. Equations (3–5) allow to obtain uncertainty estimates for the predicted probabilities Equations (1) and (2). First, we note that using p^*_k given by the mean value of the MC runs significantly enhances the accuracy for images included

[‡] The 95% confidence interval is calculated by using the Clopper and Pearson procedure.[15]

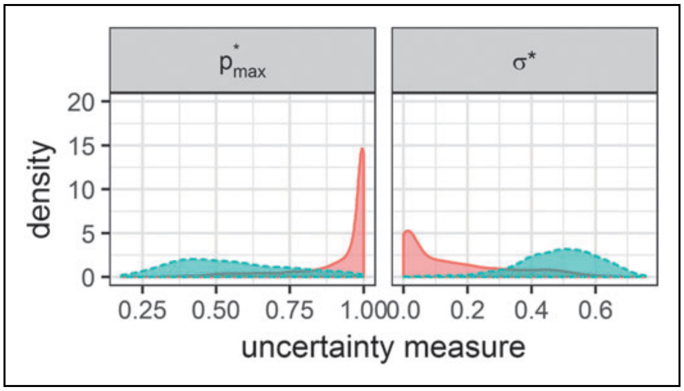

Fig. 5. Density estimation of p^*_{max} (*left*) and σ^* (*right*) for phenotypes present in the training set (*solid*) and phenotypes not included in the training set (*dashed*) using MC dropout.

in the training from 0.9367 (0.9286, 0.9442) to 0.9543 (0.9472, 0.9607). This is consistent with other findings.[16] In *Figure 5*, we compare how well p_{max}, σ from Equations (3) and (4), respectively, are able to detect novel classes.

Compared with the nondropout case (*Fig. 3*), the p^*_{max} shows a higher sensitivity if the class is unknown. This can be seen from the smaller overlap of the distributions.

To quantify which of the uncertainty measures are most sensitive to novel phenotypes, we provide an ROC and lift chart analysis of the various uncertainty measures in *Figure 6*. With the ROC analysis, we are investigating the ability of the different uncertainty measures to discriminate novel phenotypes from known ones. With the lift chart, we study the ability of the uncertainty measure to rank the images so that easy-to-classify images come first.

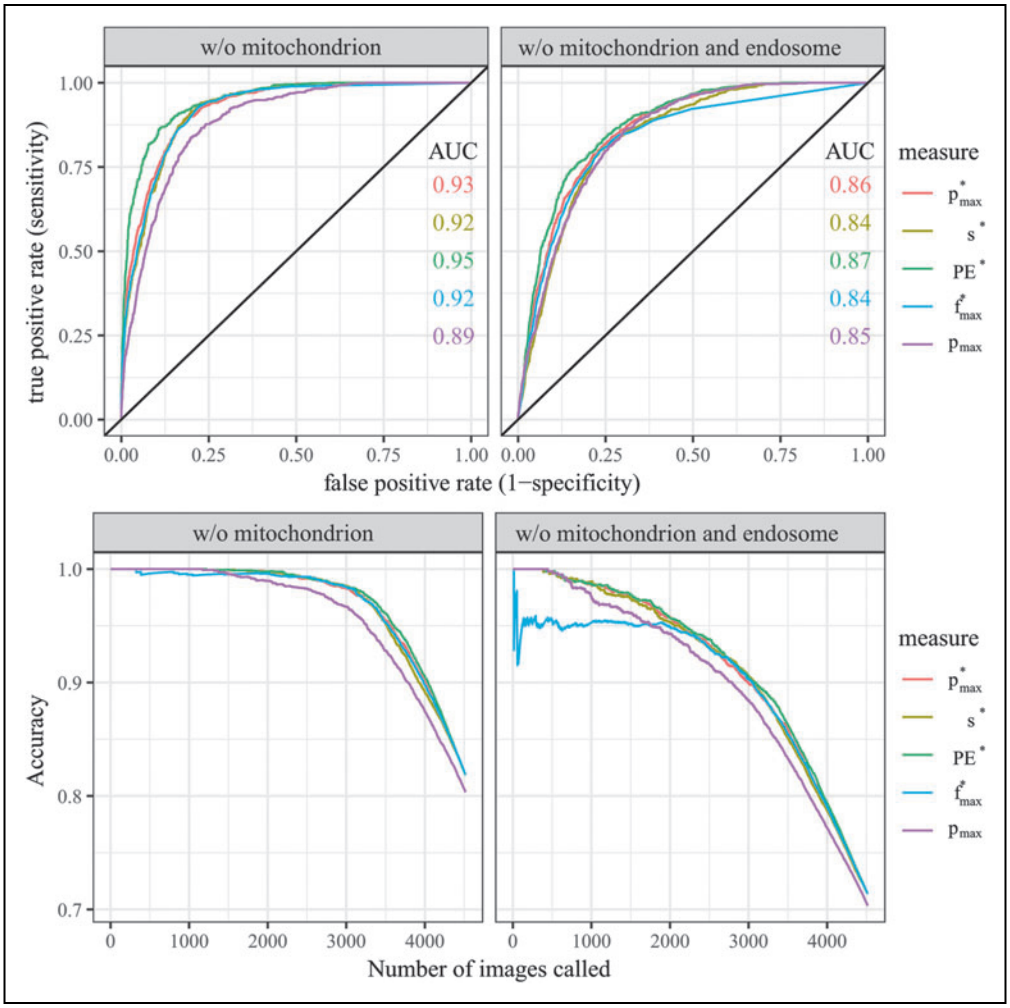

Fig. 6. ROC curves with corresponding AUC (*upper panel*) and lift charts (*lower panel*) for various uncertainty measures. AUC, area under curve; ROC, receiver operating characteristic.

For the lift chart, we order the classified cell images according to the certainty of their call. For p_{max}, p^*_{max}, f^*_{max} high values come first, while for σ^* and PE^* low values indicate certainty, and hence, we order them in a descending manner. The class assignment for the p_{max}, p^*_{max}, and f^*_{max} is done into the class with the maximal value. For the entropy and variance-based uncertainties, we use the maximal value of p^*_k to assign the class. In the beginning, we only call the phenotype with the largest certainty and achieve an accuracy of 100%, regardless which uncertainty measure is used. However, after the first 500 to 1,000 most certain images are classified, the measure p_{max}, which corresponds to the classical probability predictions without MC dropout, clearly yields inferior accuracy compared with the uncertainty measures derived from the predicted distribution in the MC dropout evaluation procedure.

The differences between different MC dropout-based approaches are negligible. Also for the ROC analysis, all MC-based approaches (except the count-based method) are clearly superior to the non-MC-based approaches. In both experiments under consideration, the entropy has the largest AUC value.

CONCLUSION

Besides reliable phenotype predictions, an improved detection of undiscovered phenotypes is of crucial importance in HCS and other imaging applications. While standard deep learning approaches show state-of-the-art prediction performance for known phenotypes, they fail to indicate if novel phenotypes are present. So far, this problem is widely ignored or tackled by a two-step approach where the classification model is preceded by a separate novelty detection model.[17]

To solve this problem in a one-step procedure, we used MC dropout during test time, allowing to determine several uncertainty estimators that can be used to discover a novel class not present during training. In our case study we investigated four uncertainty estimators derived from MC dropout predictions and showed that three of them were suitable to detect unseen classes. Using the classically predicted class, probability as an uncertainty estimate was consistently inferior to the other three estimators. MC dropout predictions were also shown to increase the overall accuracy of the model. These findings seem to generalize: first, we tested the method by means of leave-one-out experiments on all 19 classes of the analyzed yeast data set (shown in Supplementary Data) and second, we applied our approach on three other data sets (two widely used image data benchmark sets, *i.e.*, CIFAR10[18] and MNIST,[19] and third a nucleolar translocation high-content assay data set, data not shown). We achieved in all experiments a good discrimination between known and unknown classes during test time as long as the novel class is not extremely similar to a known class. After filtering out uncertain and potentially unknown cases, we achieved consistently a higher accuracy for the classified images. We therefore advocate utilizing MC dropout during test time for phenotype classification based on the MC dropout predictions as well as for novelty detection based on the uncertainty measures.

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

1. Ronneberger O, Fischer P, Brox T: U-Net: convolutional networks for biomedical image segmentation. *arXiv* 2015:150504597.
2. Kraus O, Grys B, Ba J, *et al.*: Automated analysis of high-content microscopy data with deep learning. *Mol Syst Biol* 2017;13:924.
3. Godinez WJ, Hossain I, Lazic SE, *et al.*: A multi-scale convolutional neural network for phenotyping high-content cellular images. *Bioinformatics* 2017;33:2010–2019.
4. Ando M, McLean C, Berndl M: Improving phenotypic measurements in high-content imaging screens. *bioRxiv* 2017;161422.
5. Dürr O, Sick B: Single-cell phenotype classification using deep convolutional neural networks. *J Biomol Screen* 2016;21:998–1003.
6. Kraus O, Ba J, Fery B: Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics* 2016;32:i52–i59.
7. Nichols A: High content screening as a screening tool in drug discovery. *Methods Mol Biol* 2007;356:379–387.
8. Caicedo J, Cooper S, Heigwer F, *et al.*: Data-analysis strategies for image-based cell profiling. *Nat Methods* 2017;14:849.
9. LeCun Y, Bengio Y, Hinton G: Deep learning. *Nature* 2015;521:436–444.
10. Siegismund D, Tolkachev V, Heyse S, *et al.*: Developing deep learning applications for life science and pharma industry. *Drug Res* 2018;68:305–310.
11. Gal Y: *Uncertainty in Deep Learning.* PhD thesis, University of Cambridge. 2016.
12. Bray M, Singh S, Han H, *et al.*: Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nat Protoc* 2016;11:1757.
13. Huh WK, Falvo J, Gerke L, *et al.*: Global analysis of protein localization in budding yeast. *Nature* 2003;425:686–691.
14. Nair V, Hinton G: Rectified linear units improve restricted Boltzmann machines. In: *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.
15. Clopper C, Pearson E: The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* 1934;26:404–413.
16. McClure P, Kriegeskorte N: Robustly representing uncertainty through sampling in deep neural networks. *arXiv* 2016:1611.01639.
17. Sommer C, Hoefler R, Samwer M, *et al.*: A deep learning and novelty detection framework for rapid phenotyping in high-content screening. *Mol Biol Cell* 2017; 28:3428–3436.
18. Krizhevsky A: Learning multiple layers of features from tiny images. Technical Report, University of Toronto, 2009.
19. LeCun Y, Bottou L, Bengio Y, *et al.*: Gradient-based learning applied to document recognition. *Proc IEEE* 1998;86:2278–2324.

Address correspondence to:
Oliver Dürr, PhD
Institute for Optical Systems
HTWG Konstanz
Alfred-Wachtel-Strasse 8
Konstanz 78462
Germany

E-mail: oliver.duerr@htwg-konstanz.de

Beate Sick, PhD
Institute of Data Analysis and Process Design
ZHAW Winterthur
Rosenstrasse 3
Winterthur 8400
Switzerland

E-mail: beate.sick@zhaw.ch

Abbreviations Used

CNN = convolutional neural network
GFP = green fluorescent protein
HCS = high-content screening
MAP = maximum a posteriori
MC = Monte Carlo
ROC = receiver operating characteristic