Know When You Don't Know

A Robust Deep Learning Approach in the Presence of Unknown Phenotypes

SBI^2 High Content 2018

Elvis Murina, Oliver Dürr, Daniel Siegismund, Vasily Tolkachev, Stephan Steigele, and Beate Sick

Institute of Data Analysis and Process Design

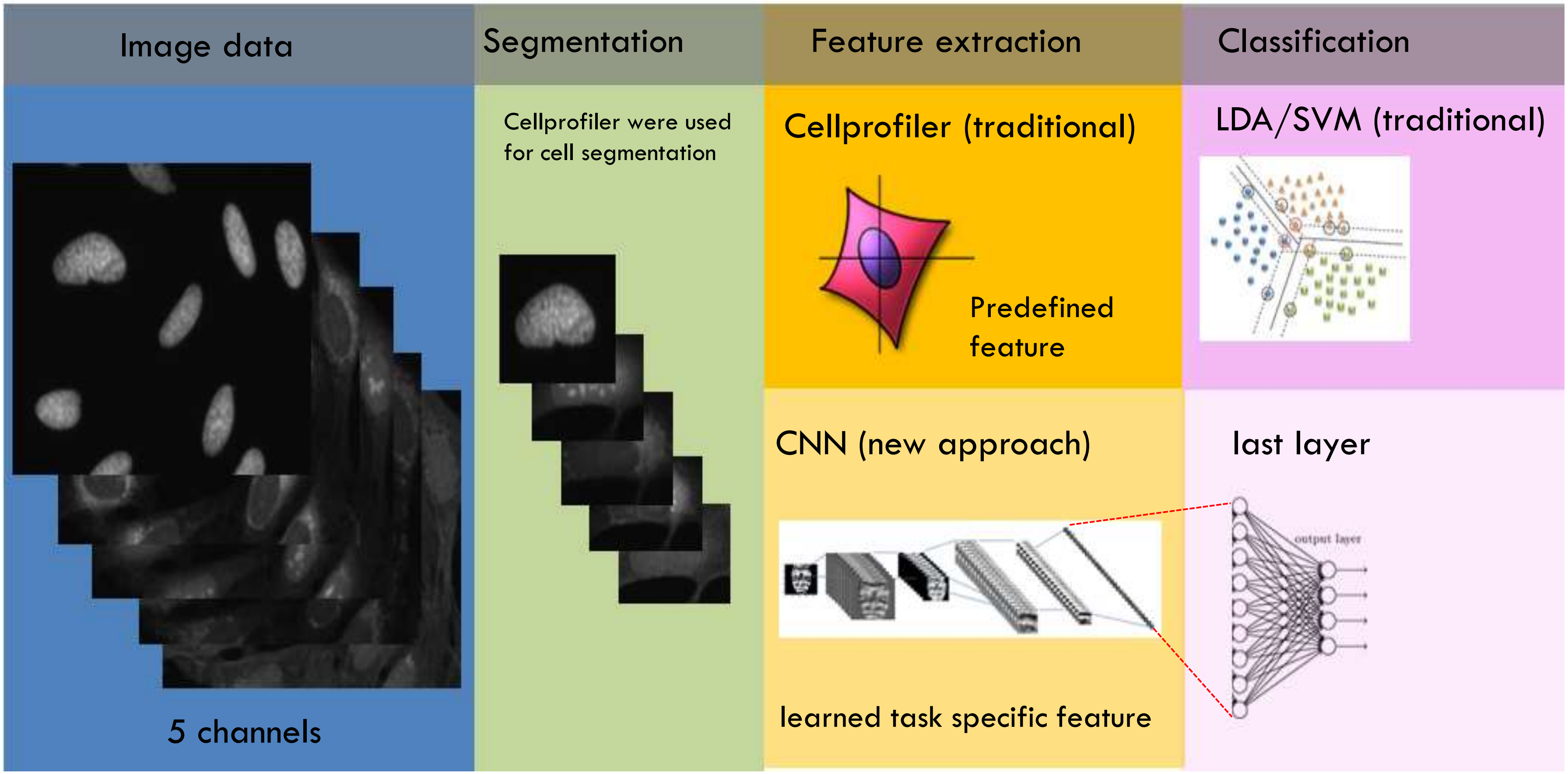
Zurich University of Applied Sciences

20.09.2018

Outline

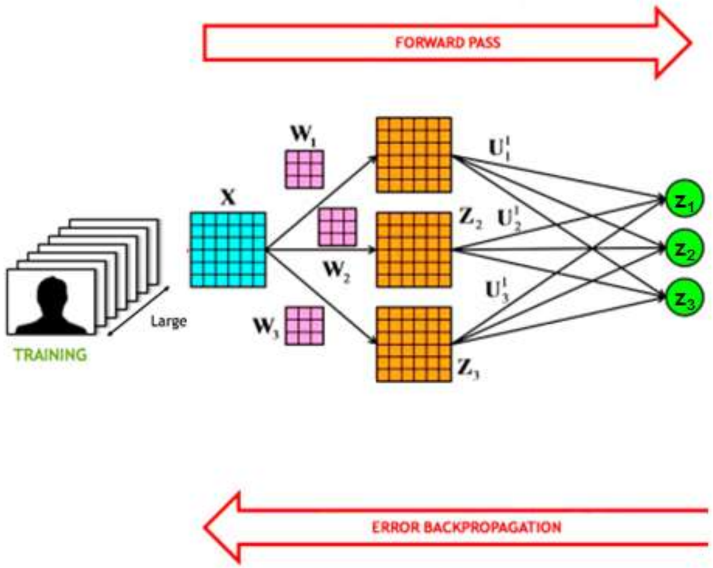
- Why deep learning for HCS?
- How to identy novel phenotypes
- Results / Conclusion

Workflow



Dürr, O., and Sick, B. Single-cell phenotype classification using deep convolutional neural networks. Journal of biomolecular screening 21, 9 (2016), 998-1003

Training of a CNN is based on gradient backpropagation



e^{z_1} P1 0
e^{z_2} P2 0
e^{z_3} P3 1

Backpropagation="chain rule"

Update weights:

$$w_i^{(t)} = w_i^{(t-1)} - l^{(t)} \left. \frac{\partial L(w)}{\partial w_i} \right|_{w_i = w_i^{(t-1)}}$$

Loss-function:

L=distance(predictions, truth)

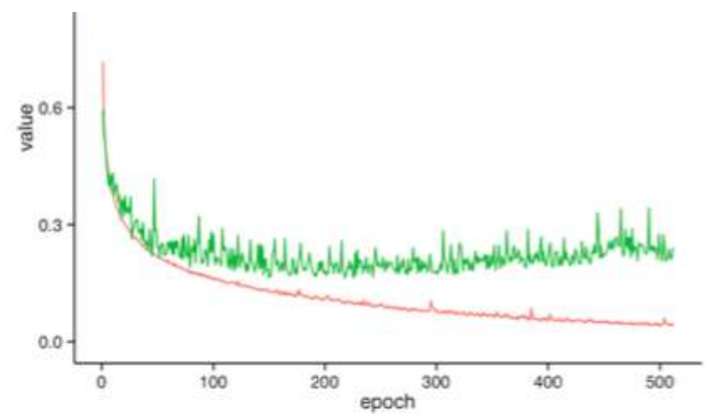
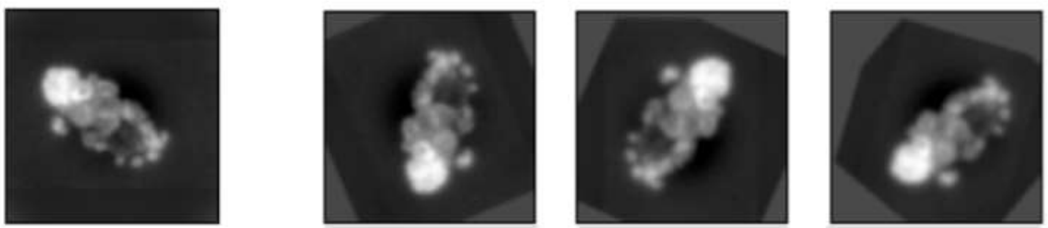


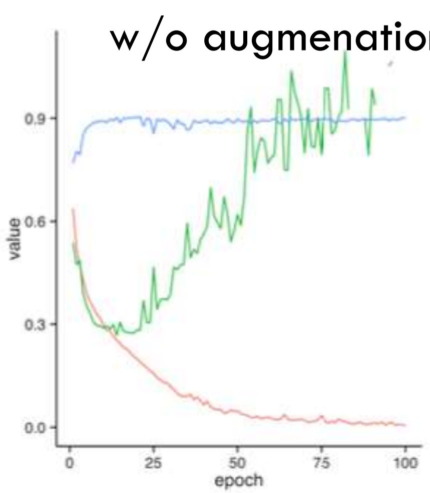
Image credits: Audhkhasi et al., Neural Networks 78 (2016) 15–23

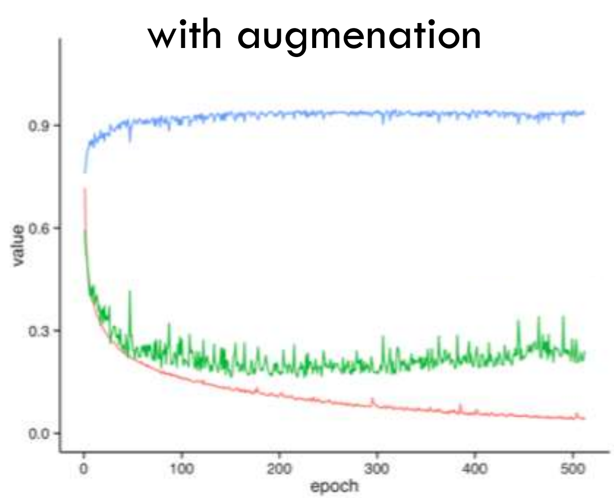
Fighting Overfitting

Data Augmentation

- Generate additional training data by rotating, zooming, shearing, flipping…

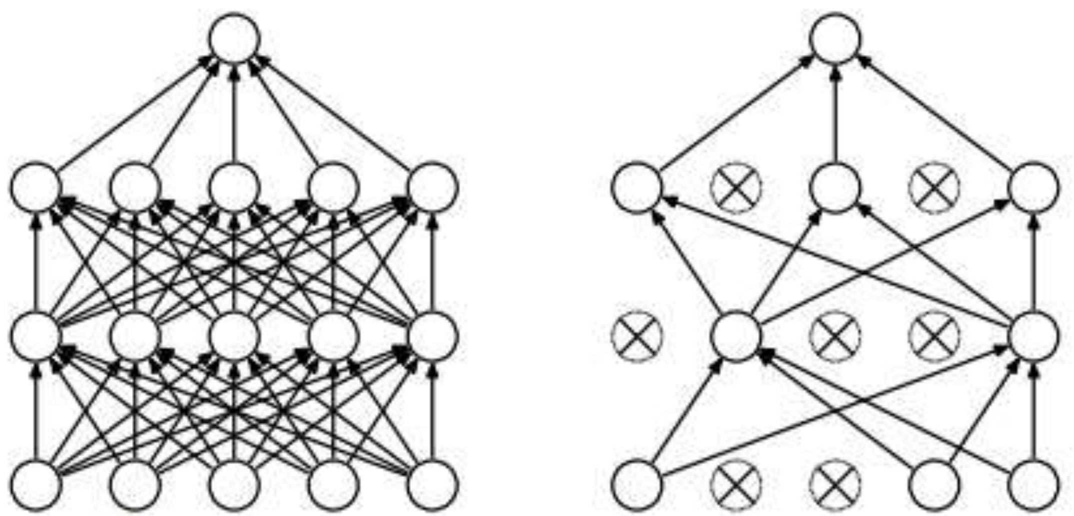






Dürr, O., and Sick, B. Single-cell phenotype classification using deep convolutional neural networks. Journal of biomolecular screening 21, 9 (2016), 998-1003

Fighting Overfitting



(a) Standard Neural Net (b) After applying dropout.

- In each training step we train another sparse NN
- Dropout prevents co-adaptation and overfitting

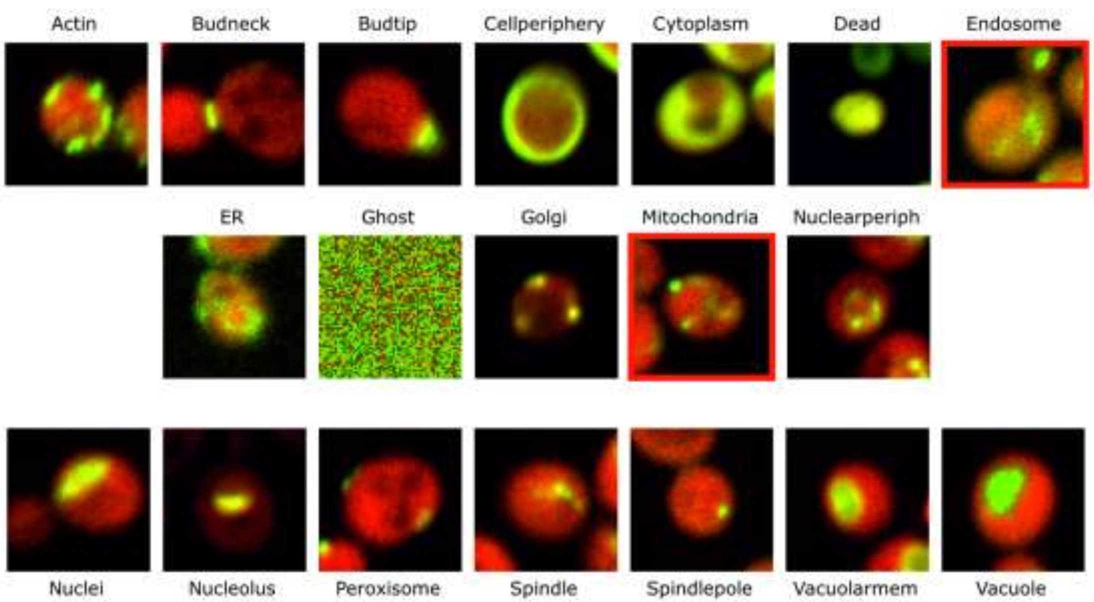
Srivastava et al., *Journal of Machine Learning Research* 15 (2014), Image Martin Görner

HCS yeast protein localization data set

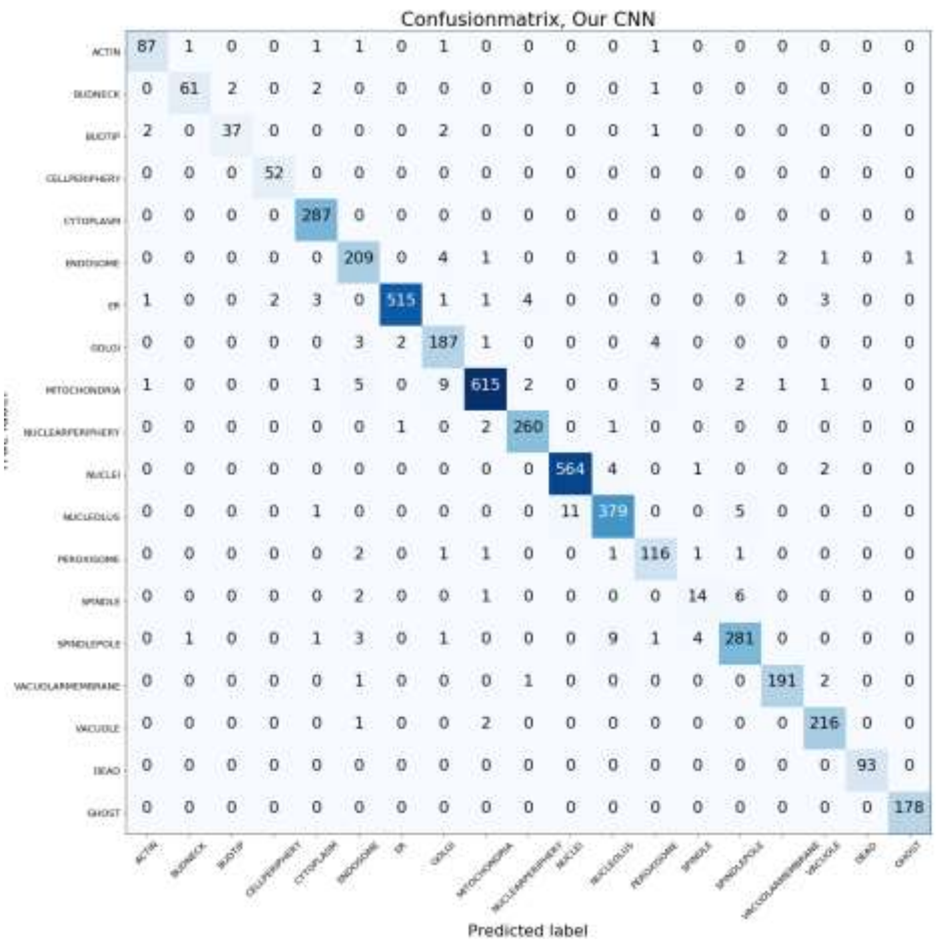
Data from budding yeast protein localizations

(all available from https://github.com/okraus/DeepLoc)

- 19 Classes
- 2 Channels
- 21882 64x64x2 segmented images in training set
- 4491 validation and 4516 for testing



Classification performance



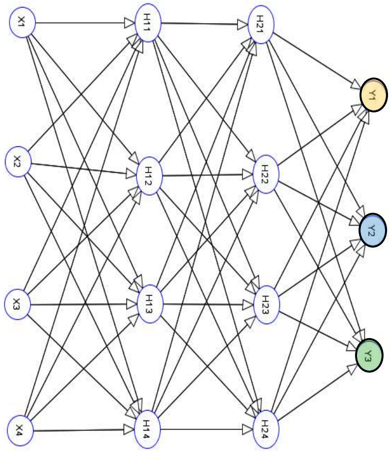
Overall test acc: 96.3% [95.7%,96.8%]

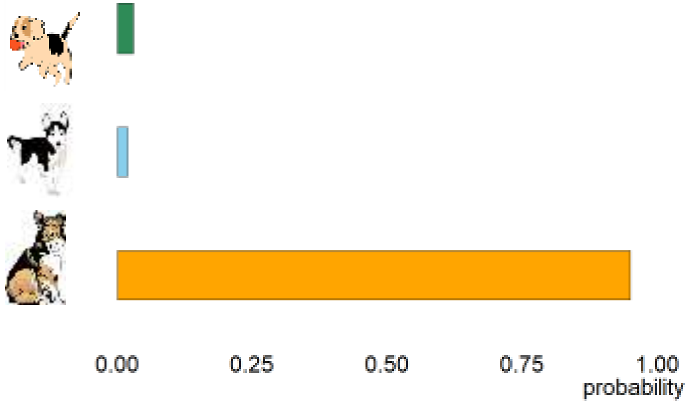
	Our_CNN	#Cells	Label
0	0.9457	92.0	ACTIN
1	0.9242	66.0	BUDNECK
2	0.881	42.0	BUDTIP
3	1.0	52.0	CELLPERIPHERY
4	1.0	287.0	CYTOPLASM
5	0.95	220.0	ENDOSOME
6	0.9717	530.0	ER
7	0.9492	197.0	GOLGI
8	0.9579	642.0	MITOCHONDRIA
9	0.9848	264.0	NUCLEARPERIPHERY
10	0.9877	571.0	NUCLEI
11	0.9571	396.0	NUCLEOLUS
12	0.9431	123.0	PEROXISOME
13	0.6087	23.0	SPINDLE
14	0.9336	301.0	SPINDLEPOLE
15	0.9795	195.0	VACUOLARMEMBRANE
16	0.9863	219.0	VACUOLE
17	1.0	93.0	DEAD
18	1.0	178.0	GHOST

Why dont we stop here?

A first thought experiment

Suppose you train a classifier on dogs and show it a new dog that was in the training set.

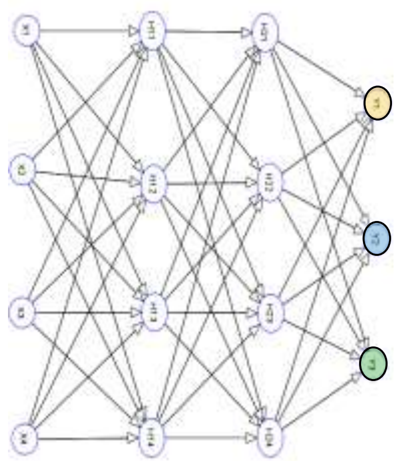


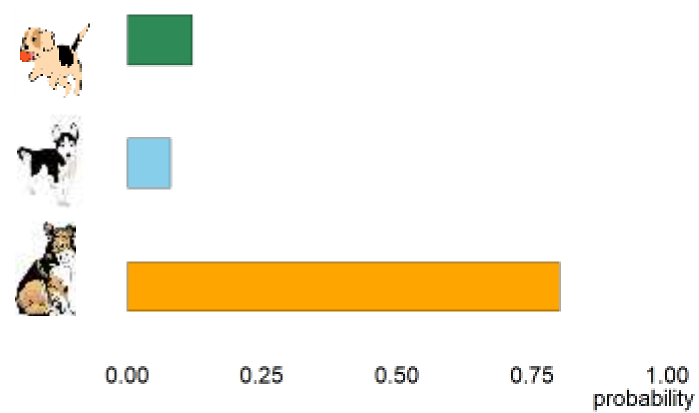


A first thought experiment

Suppose you show the same classifier a cat

What will be the result?





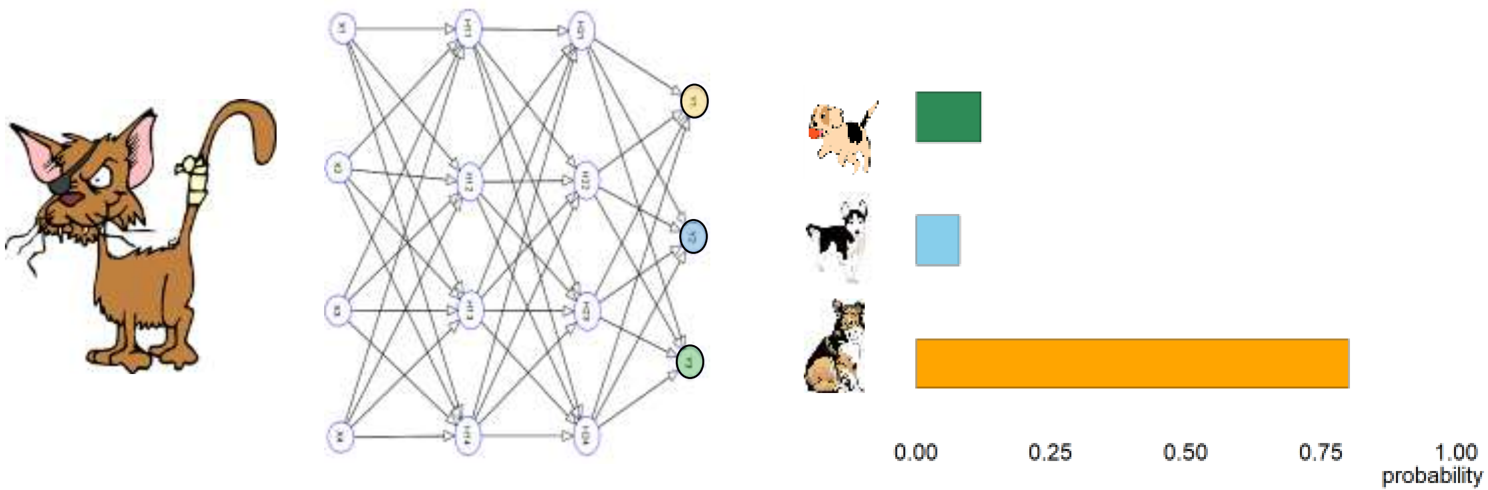
How can that be?

- Forced to classify it into one of the dogs.
- If it’s a dog, than most probably a collie
- No confidence of the prediction given

A first thought experiment

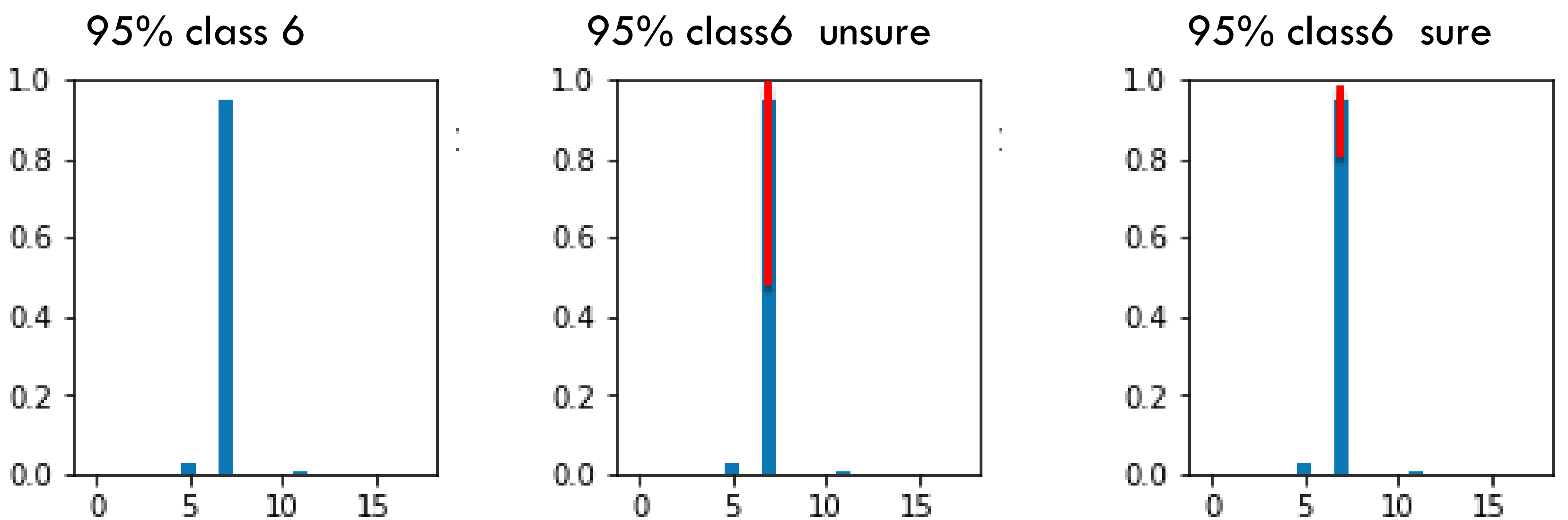
Suppose you show the same classifier a cat

What will be the result?



We need uncertainty measures to our predictions!

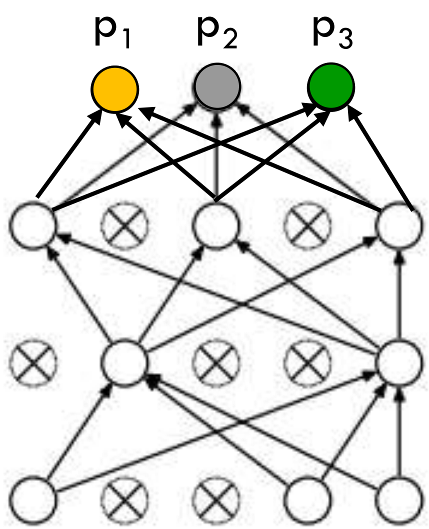
We want error bars (or even better a distribution)



How to get error bars?

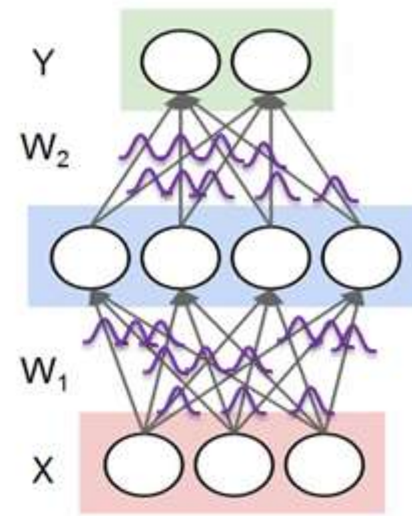
- **Experimenter:** Go in lab and repeat!
- **kaggle kid:** Spin up 100 AWS instances and repeat (train and predict)
- **Machine Learner / Statistician:** …

MC Dropout and Bayesian Neural Networks



Equivalence

Yarin Gal* (2015)



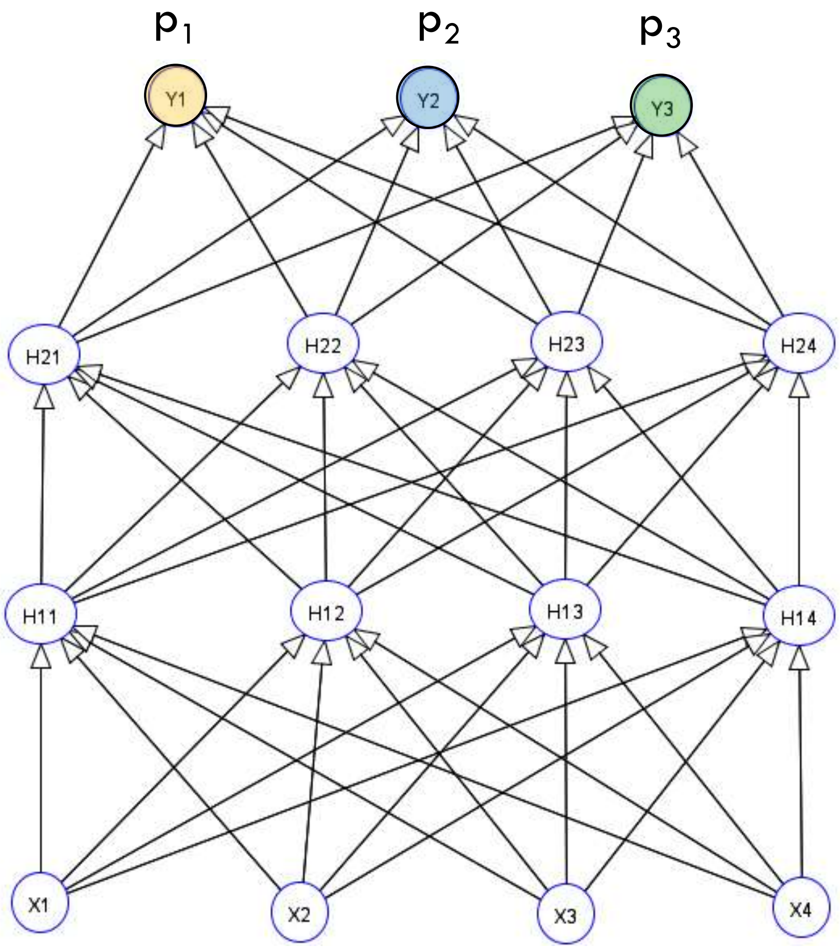
MC Dropout
At each training and testing step we remove random nodes with a probability p

Bayesian Neural Networks
Provides predictive probability distribution

*Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning https://arxiv.org/abs/1506.02142

No MC Dropout

Output: probabilities for each class



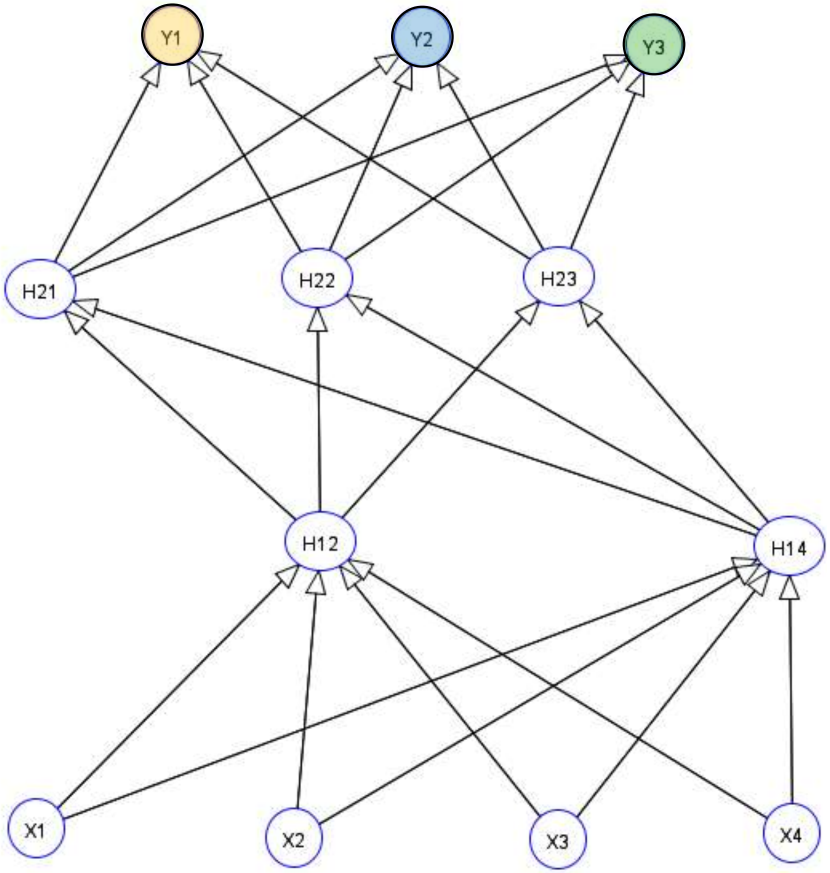
Done in training anyway

Why not use it also at test time

Input: image pixel values

Use dropout at test time: Run 1

p_1=0.08 **p_2=0.89** p_3=0.03



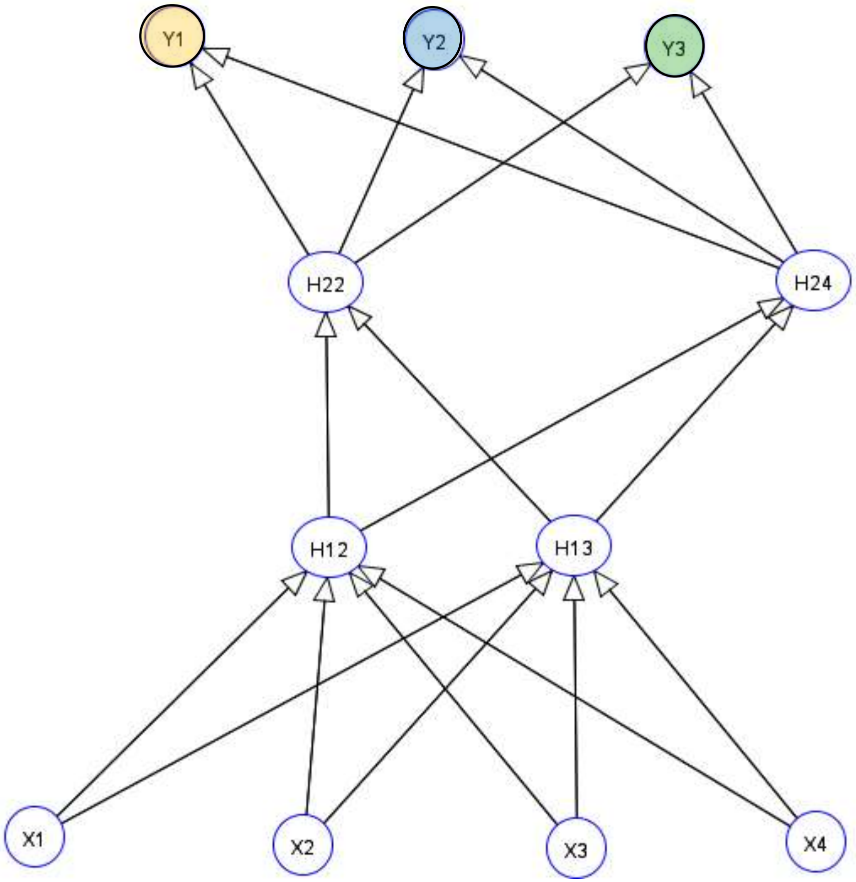
Output depends on dropout

Stochastic dropout of units

Same input image

Use dropout at test time: Run 2

p_1=0.11 **p_2=0.81** p_3=0.08

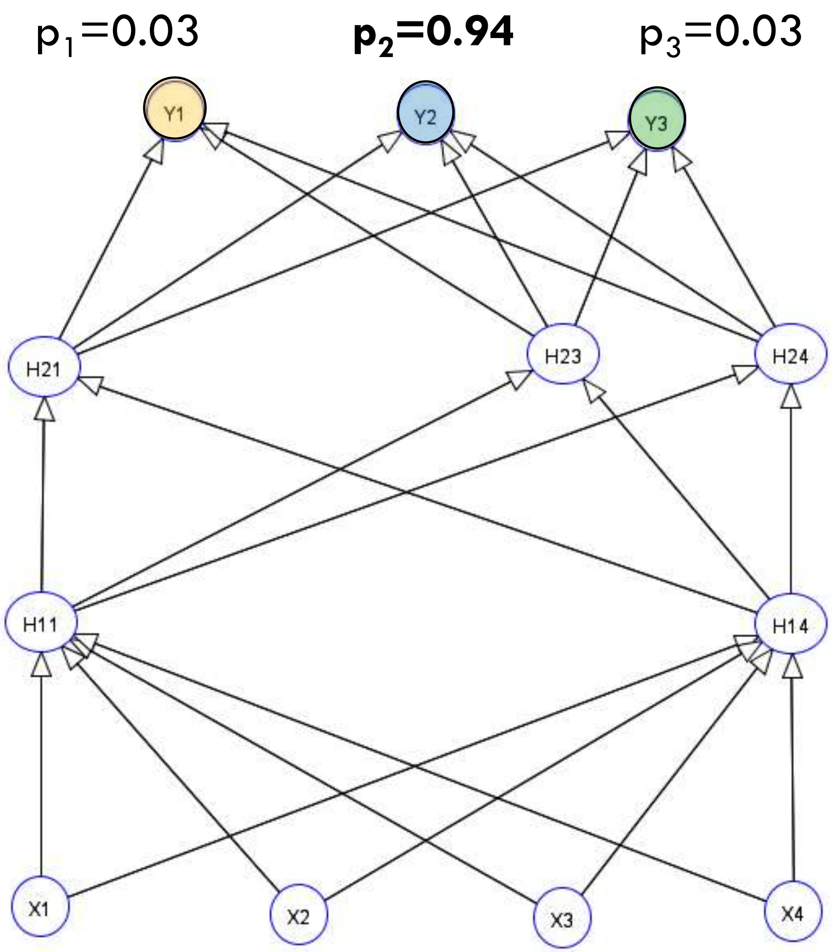


Output depends on dropout

Stochastic dropout of units

Same input image

Use dropout at test time: Run 3



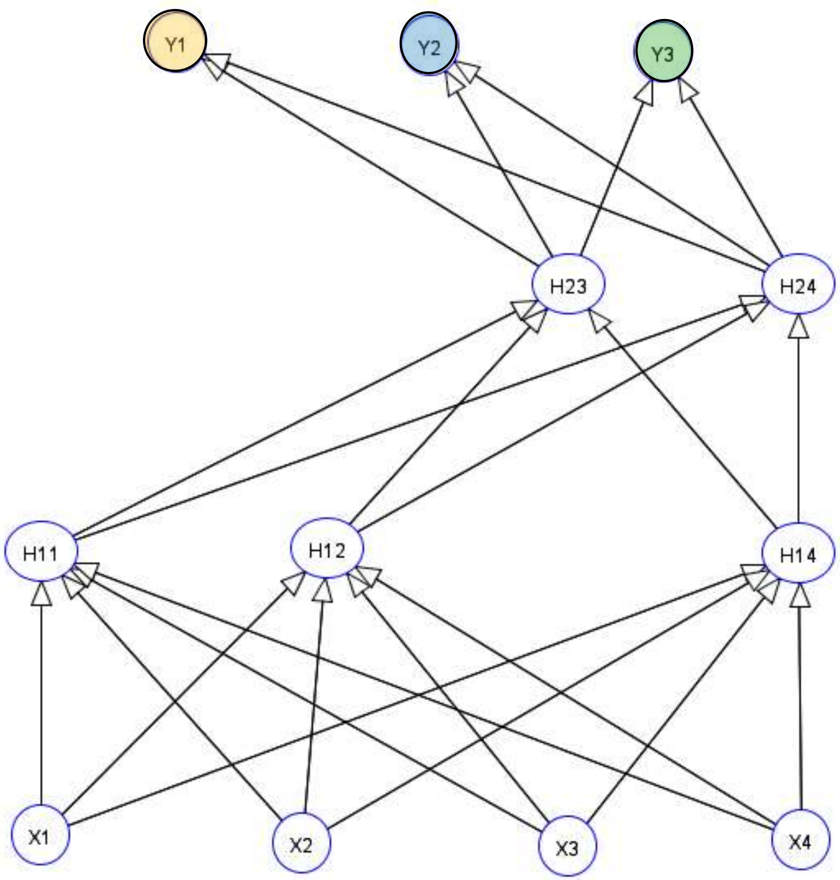
Output depends on dropout

Stochastic dropout of units

Same input image

Use dropout at test time: Run 4

p_1=0.16 **p_2=0.78** p_3=0.06



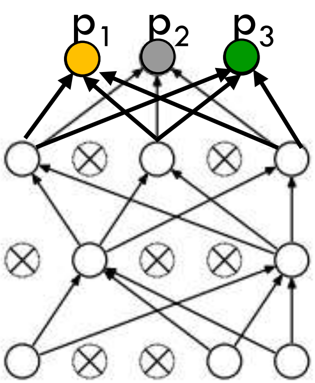
Output depends on dropout

Stochastic dropout of units

Same input image

…Repeat 500 times

Distributions of predicted probabilities



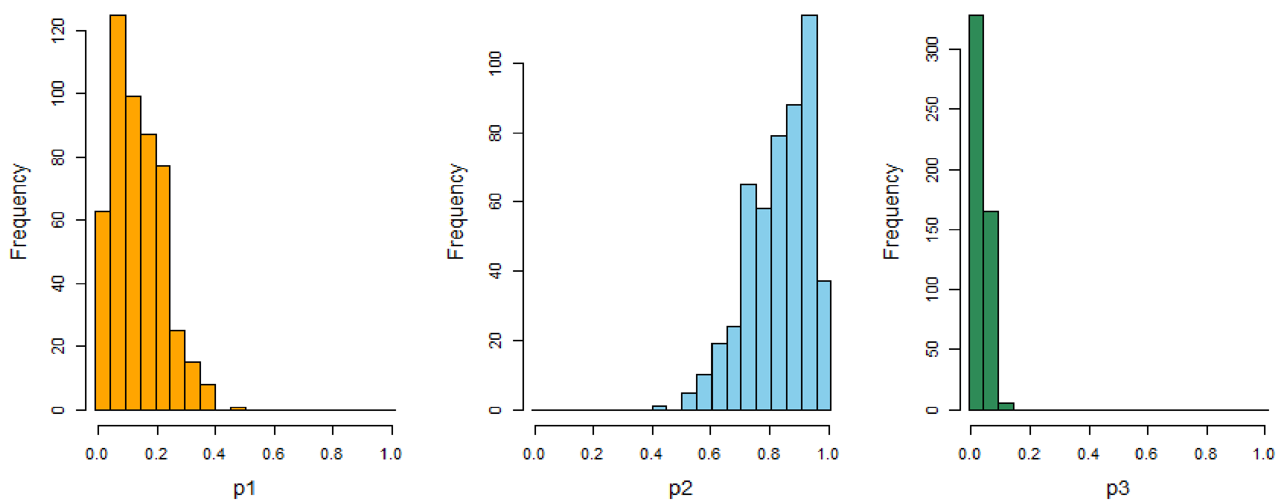
From the predicted distributions we can derive different measures

Probability estimates

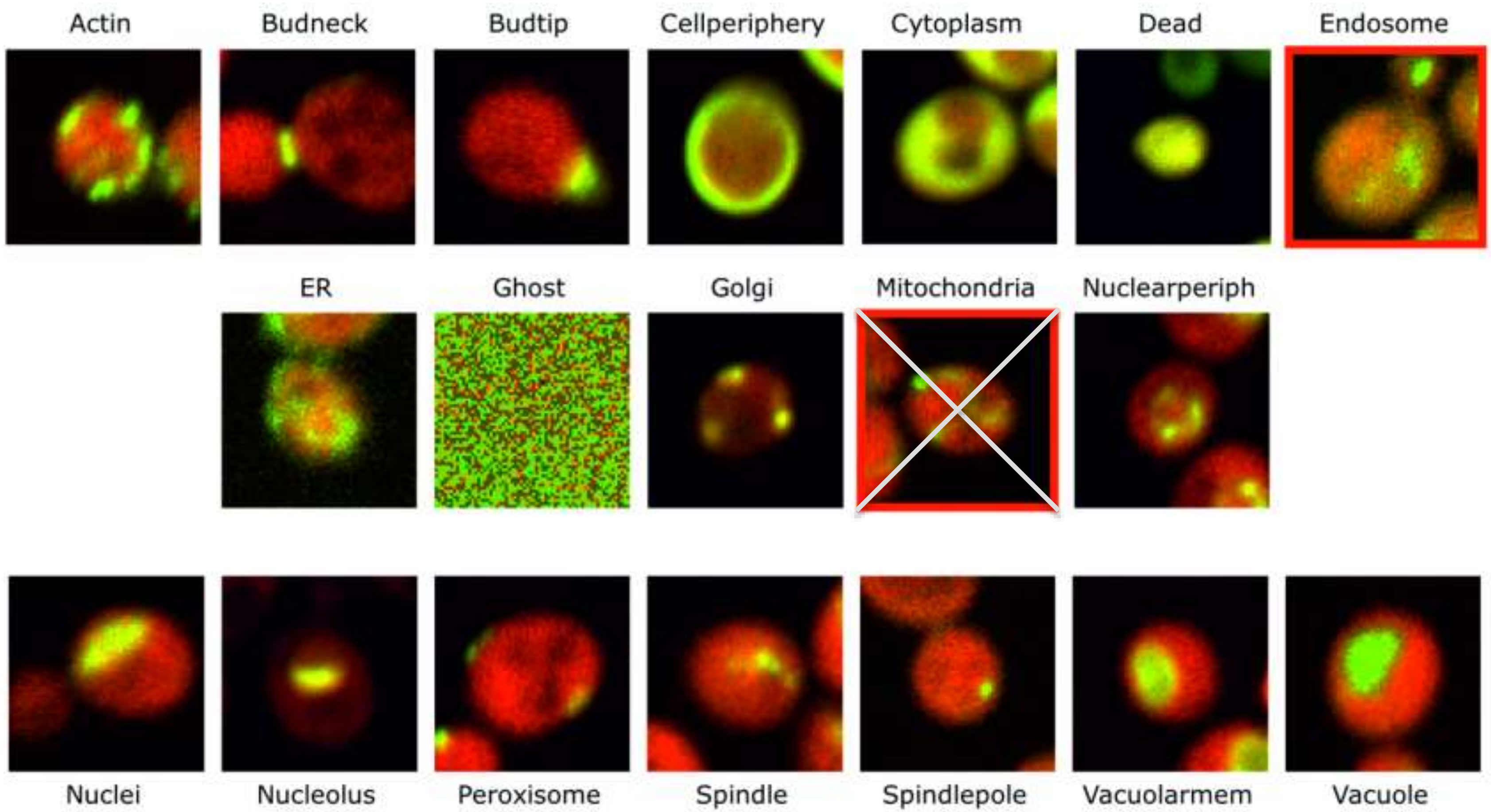
- p_{max} (when **not** using mc dropout)
- p^{*}_{max} (when using mc dropout)

Uncertainty estimates

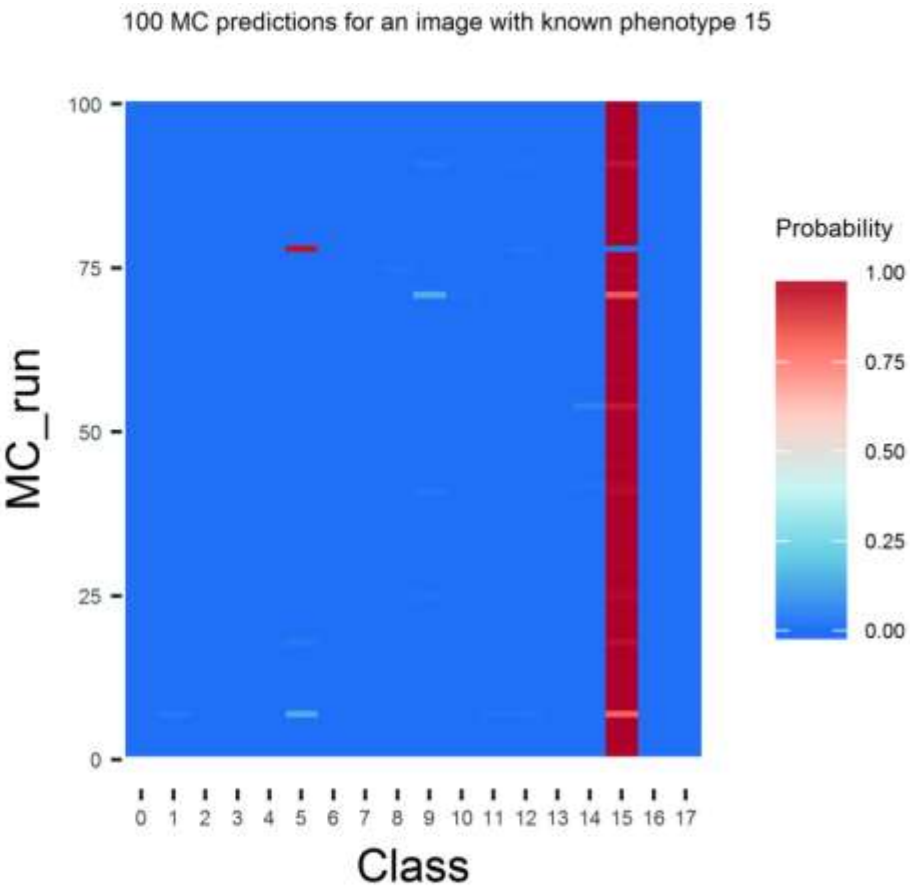
- σ^{*} total standard deviation
- PE^{*} entropy



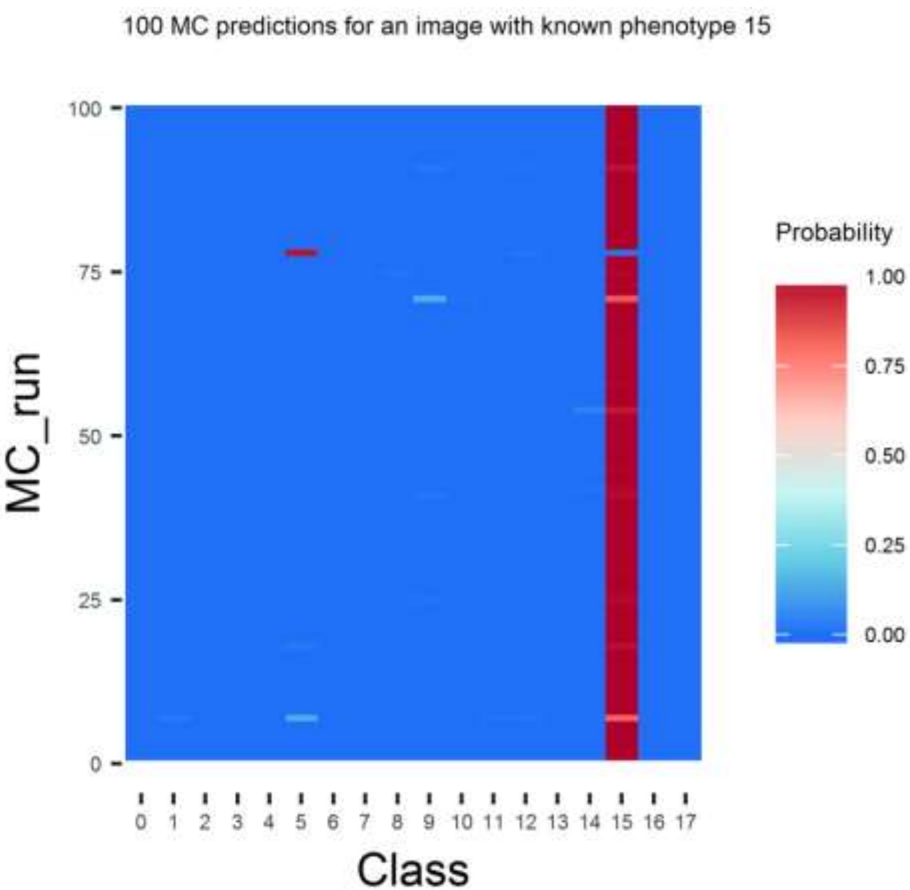
Experiment with unknown phenotype

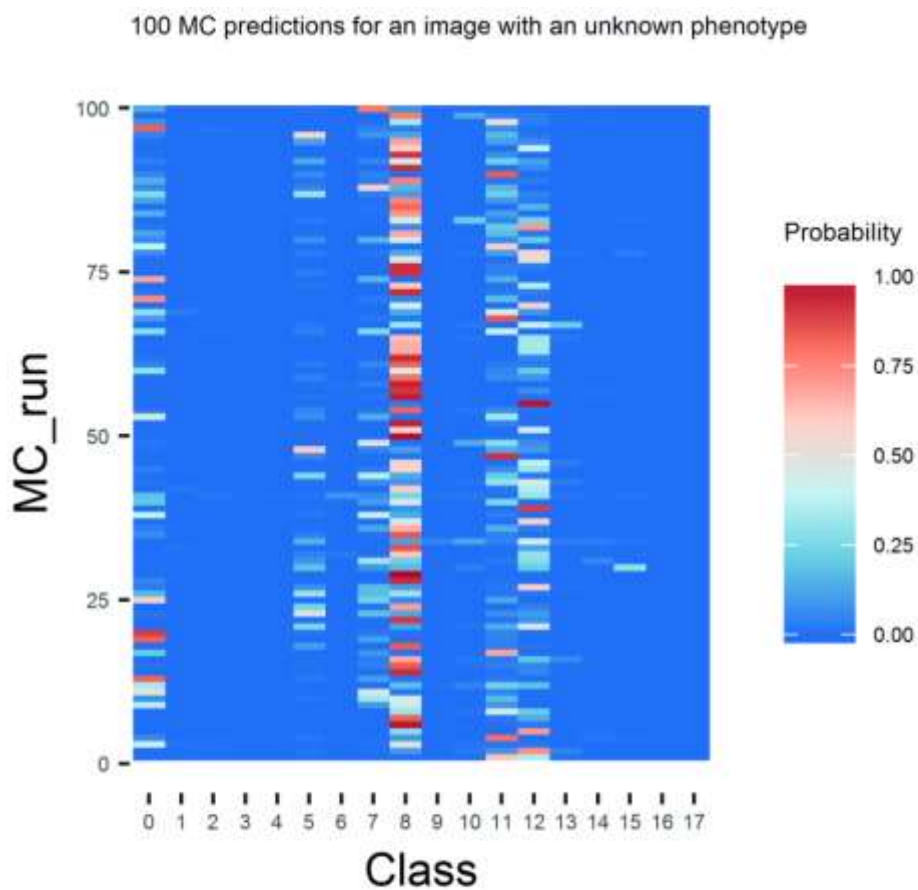


Experiment with unknown phenotype



Experiment with unknown phenotype

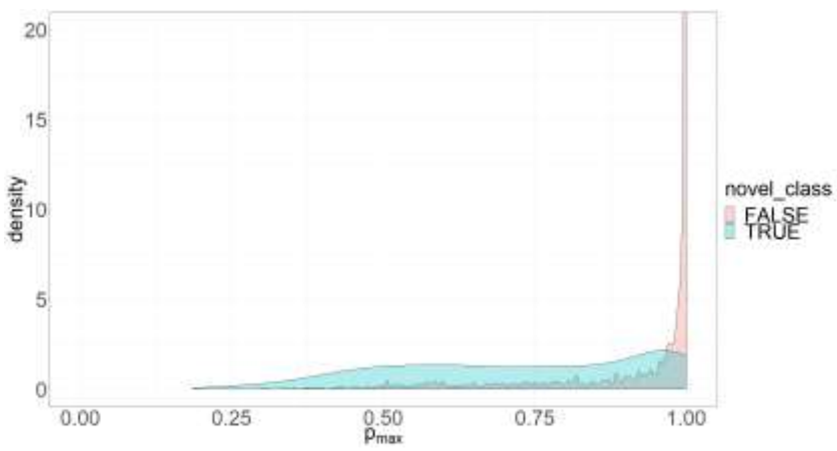




Experiment with unknown phenotype

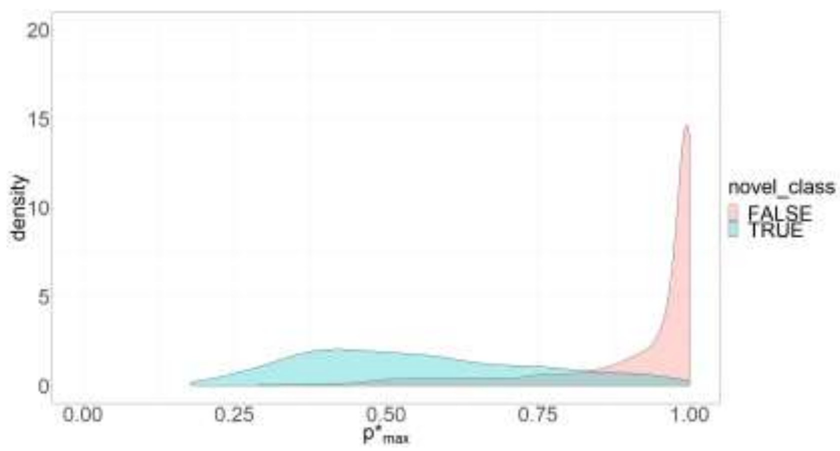
No MC Dropout (classical)
Probability estimate p_{max}

- accuracy (only not novel classes considered):
- **0.9367 [0.9286, 0.9442]**



MC Dropout
Probability estimate p^*_{max}

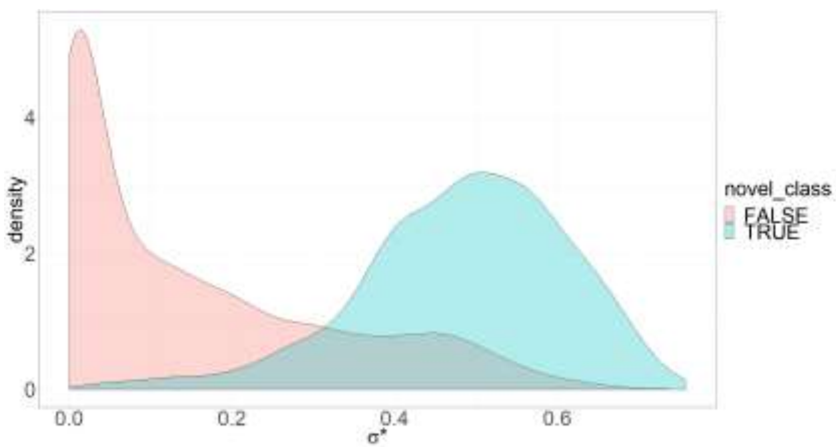
- accuracy (only not novel classes considered):
- **0.9543 [0.9472, 0.9607]**



Experiment with unknown phenotype

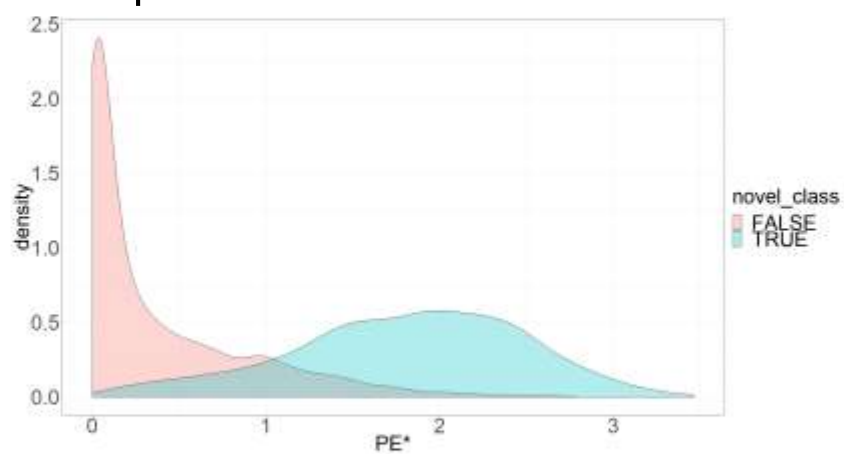
MC Dropout
Uncertainty estimate σ^*

- High σ^* indicates a class not seen in during traning
- Low σ^* indicates a confident prediction

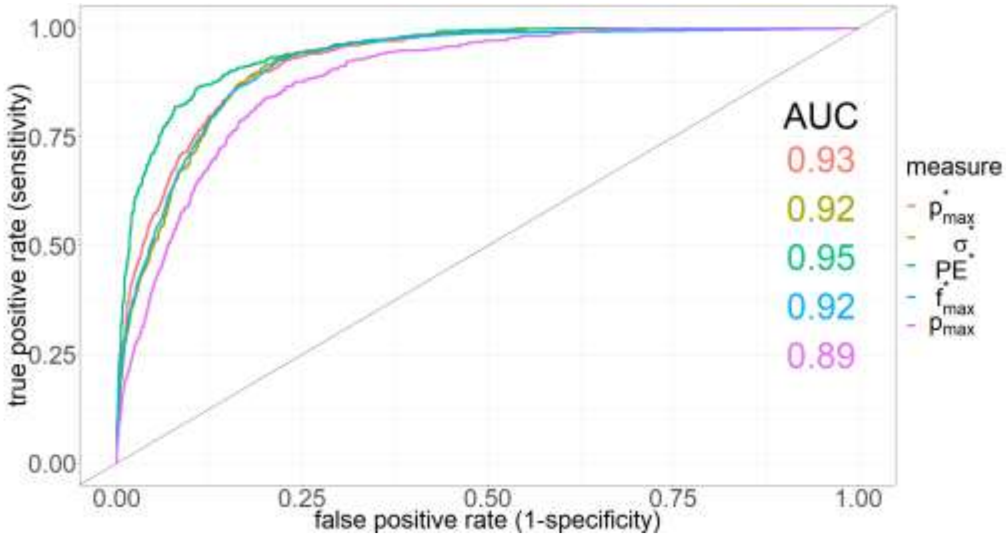


MC Dropout
Uncertainty estimate PE^*

- High PE^* indicates a class not seen in during traning
- Low PE^* indicates a confident prediction



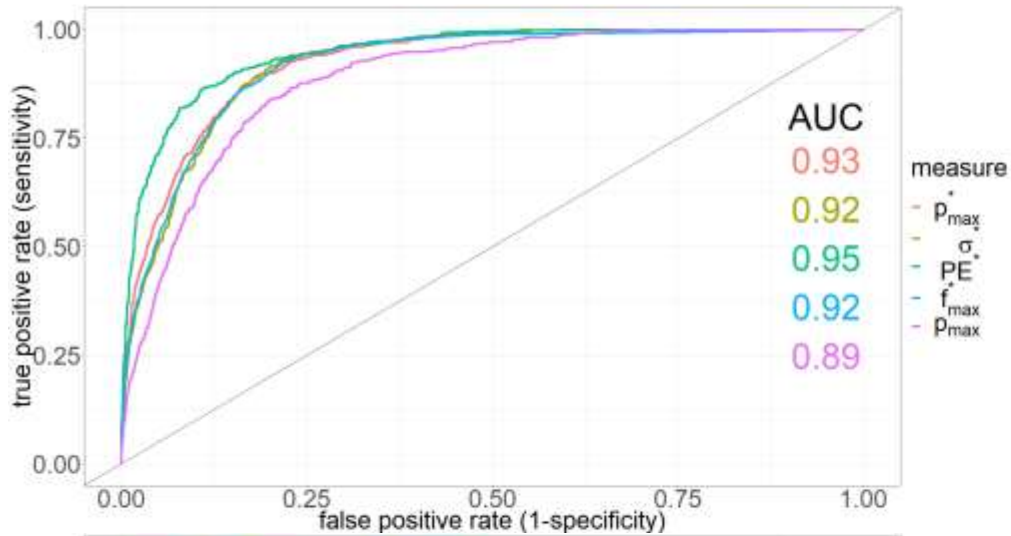
Experiment with unknown phenotypes

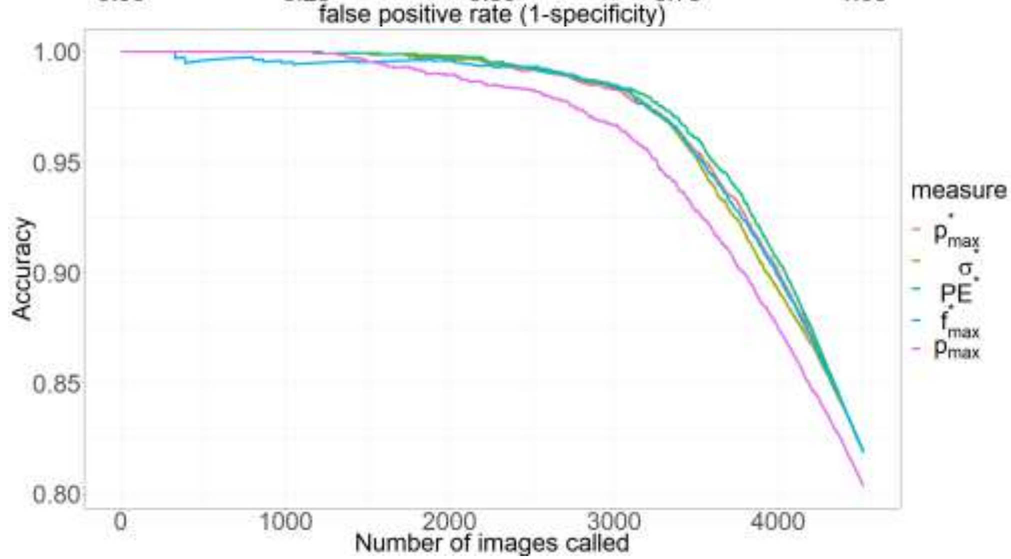


Discriminative power between known and unkown class

- All **MC Dropout based appoaches are superior** to the non-MC based approaches

Experiment with unknown phenotypes





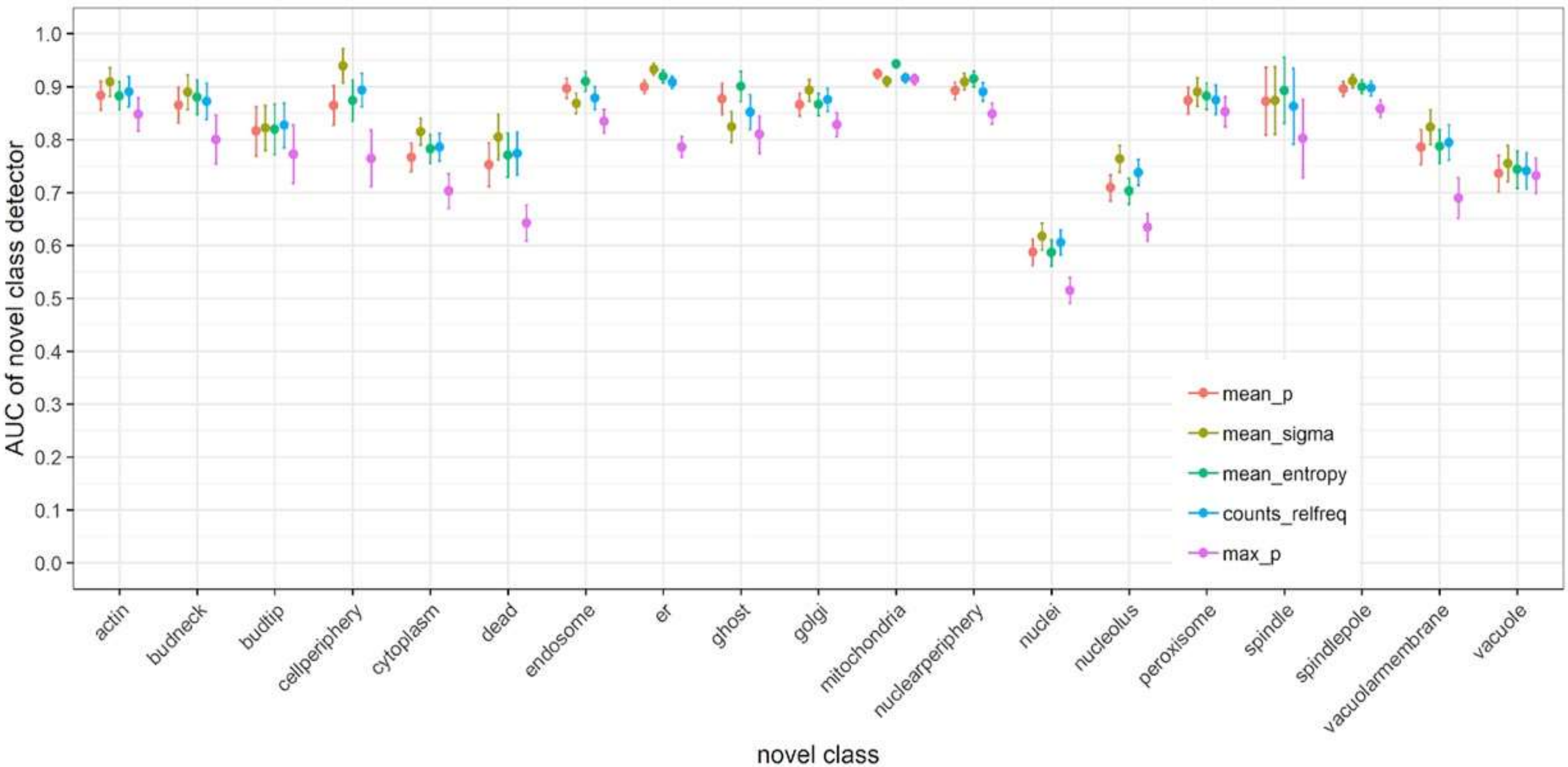
Discriminative power between known and unkown class

- All **MC Dropout based appoaches are superior** to the non-MC based approaches

Filter uncertain predictions

- In the lift chart, we order the classified images according to the certainty of their call
- p_{max} which corresponds to the classical probability without MC dropout, is clearly worse compared with the MC dropout approaches

MC dropout in leave-one-out experiments



Conclusion

- Deep Learning works for single cell phenotype classification
- No hand crafting of features needed
- MC Dropout yields to model uncertainty and boosts performance
- Novel phenotypes can be identified

Thank you!

Beate Sick

Oliver Dürr

Vasily Tolkachev

Stephan Steigele

Daniel Siegismund

This work has been partly funded by the CTI grant: “DeepCells”

Thank you, Questions?