IN5400 / IN9400 – Machine Learning for Image Analysis

Lecture 17

Performance estimation and generalisation

12th of May 2021

Andreas Kleppe

UiO : Department of Informatics
University of Oslo

What is the purpose?

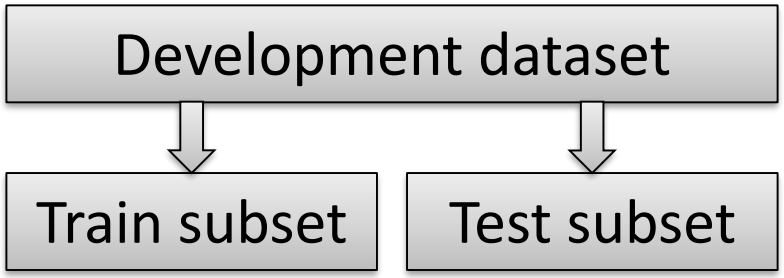
- The ultimate goal of developing neural networks models is usually to enable automatic prediction in a particular task.
- In image classification, the goal is typically to obtain models that could automatically identify the correct output class for *new* images.
- For the users of a model, it usually *does not* matter how the model was trained.
  - But it affects the performance, so model developers should certainly care.
- What matters is how the model performs when applied in practice.
  - Some users would also like an understanding of how the predictions are obtained.
- Todays topics relates to how to estimate the performance and how the *generalisation* to new data might be improved.
  - A model which performs similarly on all new data relevant for the intended application as on the training data has good generalisation.
  - Good performance on new data is sometimes referred to as good generalisation.

Simplest approach: Resubstitution

- Estimate the performance using the set of data that was used to train the model.
- Very efficient use of data.
  - Allows all data to be used for both training and testing.
- Often provides severely *overoptimistic estimates*.
  - That is, provides performance estimates which suggest that the model performs much better than it would do on new data.
- Training longer usually makes the resubstitution estimates more overoptimistic.
  - Training for too many epochs will facilitate the learning of relations between input data and target output that does not generalise to new data.
    - Because this will often reduce the loss used during training.
    - This is called *overfitting* (to the training data).
  - Overfitting might decrease the performance on new data.
- Should *not* be used to estimate how the model performs on new data!
- Could be used to compare with other performance estimates to give an impression of the degree of overfitting to the training data.
- Could also be used to indicate that the training progresses reasonably.
  - Using the loss averaged across some mini-batches is an alternative which requires much less additional computation.

Train and test subsets

- Common to randomly partition the development data into two disjoint subsets:



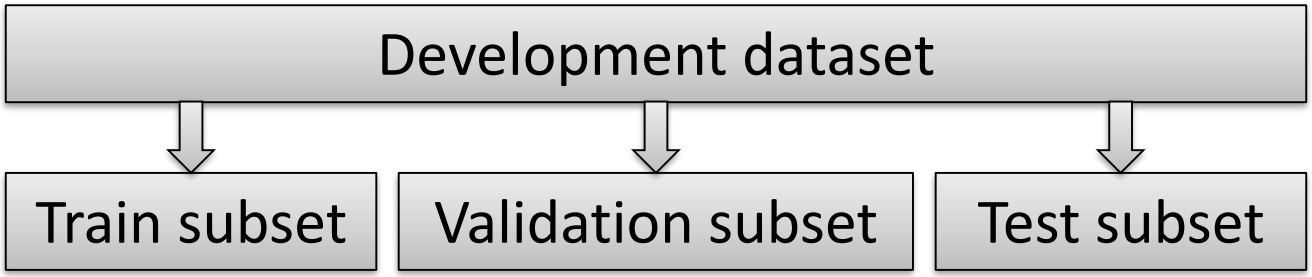
- Use the train subset to train models.
- Use the test subset to evaluate the *final* model.
- "Ideally, the test set should be kept in a "vault", and be brought out only at the end of the data analysis. Suppose instead that we use the test-set repeatedly, choosing the model with smallest test-set error. Then the test set error of the final chosen model will underestimate the true test error, sometimes substantially."
  - Page 222 in the classical text book by Hastie, T., Tibshirani, R. & Friedman, J. *The Elements of Statistical Learning* 2nd edn (Springer- Verlag, 2009).

Train and test subsets

- If the test subset is used for only a single analysis, then the resulting estimate is expected to be unbiased for the development data.
  - At least if the test subset is a random sample of the development data.
  - "Single analysis" refers to calculating a particular performance metric for a particular model using a particular set of data.
- "if you don't like the results, you have to obtain, and lock away, a completely new test set if you want to go back and find a better hypothesis."
  - Page 709 in the classical text book by Russell, S. & Norvig, P. *Artificial Intelligence: A Modern Approach* 3rd edn (Prentice Hall, 2010).
- Does not use the data as efficiently as resubstitution.
  - Less data will be used for both training and testing.
  - However, the reliability of the performance estimation is substantially increased.

Validation subset

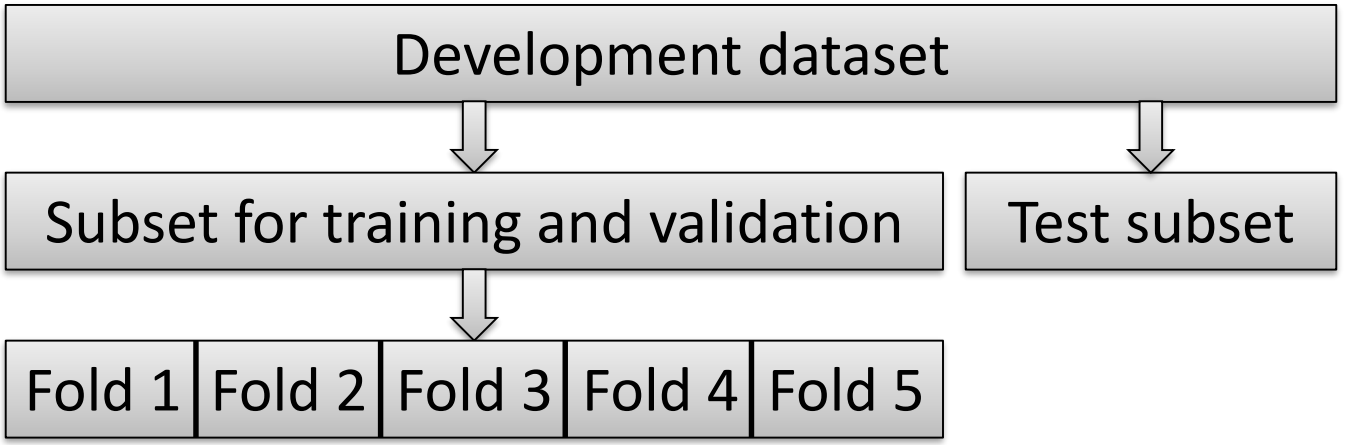
- Usually need a third subset to evaluate different modelling choices.
  - "Modelling choices" refer to selection of hyperparameters in a broad sense, including not only selection of values not optimised during training, but also e.g. choice of neural network and optimisation method.
  - Should not use the training subset because of overfitting.
  - Should not use the test subset because that will bias performance estimation based on the test subset.
- Could make a third random subset (disjoint from the other two subsets):



- Note: In medical statistics, "validation" usually refers to testing.
  - The subset used to determine hyperparameters may be called "tune subset", but this type of "tuning" should not be confused with fine-tuning (i.e. training from an initialisation that applies weights from a pre-trained model).

Validation by resampling

- There are multiple alternatives to applying a fixed validation subset.
- *n*-fold cross-validation:
  - Randomly partition the non-test subset into *n* folds of equal size.
  - Repeat *n* time:
    Use the n^{th} fold for validation, and the other *n*-1 folds for training.
  - Use the average of the performance estimates obtain for the validation folds.
    - Could also calculate the standard deviation or other measures of uncertainty.



Validation by resampling

- Instead of applying a fixed random partitioning into folds, repeatedly partition the non-test subset into a train subset and a validation subset.
- Another alternative is to apply bootstrapping,
  e.g. repeat the following 10 times:
  - Sample with replacement from the non-test data to obtain a train subset with the same size as the non-test subset.
    - Each train subset will on average contain 63.2% of the non-test data but will be of the same size as the non-test subset due to duplicates.
  - Train using the train subset (with duplicates), and use the other non-test data for validation.
- All these approaches are called resampling techniques.
  - In each approach, the average of the performance estimates obtained for the validation subsets are used as the final performance estimate.
  - They also allow estimation of the uncertainty of the performance estimate.

Validation

- Using resampling techniques for validation will use the data more efficiently than applying a fixed validation subset.
  - Because all non-test data will be used for both training and validation.
- Evaluating many modelling choices might result in overfitting.
  - With a fixed validation subset, the choice that appears to provide best performance might do so only for the particular validation subset.
  - When using a resampling technique, the choice that appears to provide best performance might do so only for the non-test data, or only for all development data.
- If data is limited, applying a resampling technique (e.g. cross-validation) is usually preferable if the added computational expense is acceptable.

Validation

- Example from classical machine learning and medicine:
- ≈20 years ago, microarray gene-expression profiling was the new hot.
- Michiels et al. used a resampling technique to re-evaluate publicly available data from seven studies that had applied specific validation subsets to evaluate the performance of trained molecular signatures.
  - The studies were published in Nature, Nature Medicine, Nature Genetics, Lancet, NEJM, PNAS, and Cancer Cell.
- They found that "Five of the seven studies did not classify patients better than chance."

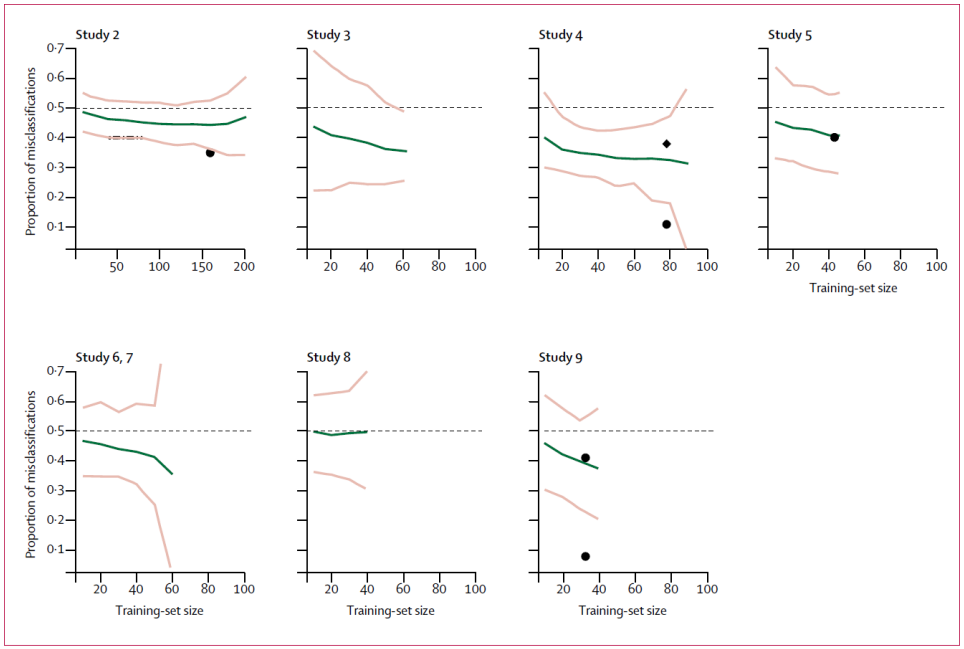
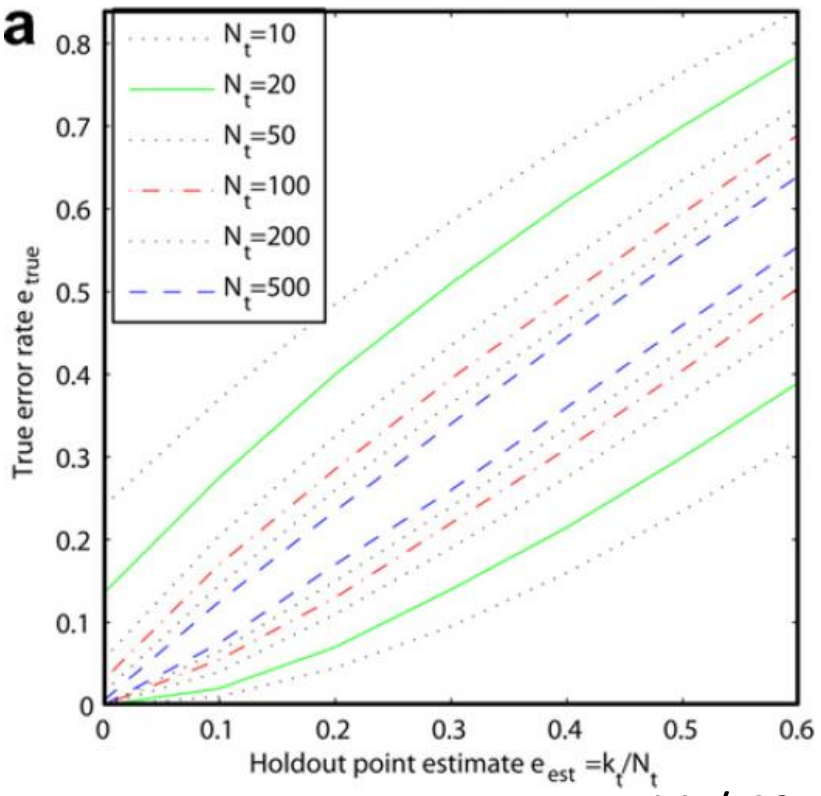


Figure 2: **Proportion of misclassifications in validation sets as a function of corresponding training-set sizes in the seven studies**[2–9]
Green lines=mean proportion of misclassifications obtained from 500 random training-validation sets as a function of the training-set size. Pale red lines=95% CIs. Dots=misclassification rates in original publications. Iizuka and colleagues[9] published two misclassification rates by two different methods on the same validation set. Diamond=second misclassification rate on a larger independent validation set[16] from the van 't Veer study.[4]

Michiels, S., Koscielny, S. & Hill, C. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet* **365**, 488–492 (2005).

Subset sizes

- Want as much data as possible for training, validation, and testing.
- Common to randomly partition the development data according to some percentages, e.g. 70% for training, 15% for validation, and 15% for testing.
- Resampling techniques might allow use of less validation data.
- Size of test subset should be considered in relation to the uncertainty of the resulting estimates.
  - If performance differences of <1 percentage point should be considered, then thousands of test samples are usually needed.



The lines depict the range of 95% Bayesian confidence intervals.

Figure 4a by Isaksson, A., Wallman, M., Göransson, H. & Gustafsson, M. G. Cross-validation and bootstrapping are unreliable in small sample classification. *Pattern Recognit. Lett.* **29**, 1960–1965 (2008).

Subset sizes

- The relation between performance and size of training data appears to be linear on log-log scale.
  - Thus, the performance gained from increasing the size of the train subset is initially large and then quickly becomes smaller.
  - With too small train subset, the model struggle to learn.
  - With an excessive train subset, adding more training data might not increase the performance.
- One might attempt to estimate the power-law exponent and coefficient from estimates of the performance for different sizes of the train subset.
  - Could also be useful if considering to obtain a larger development set (if possible, this usually requires some time and resources).

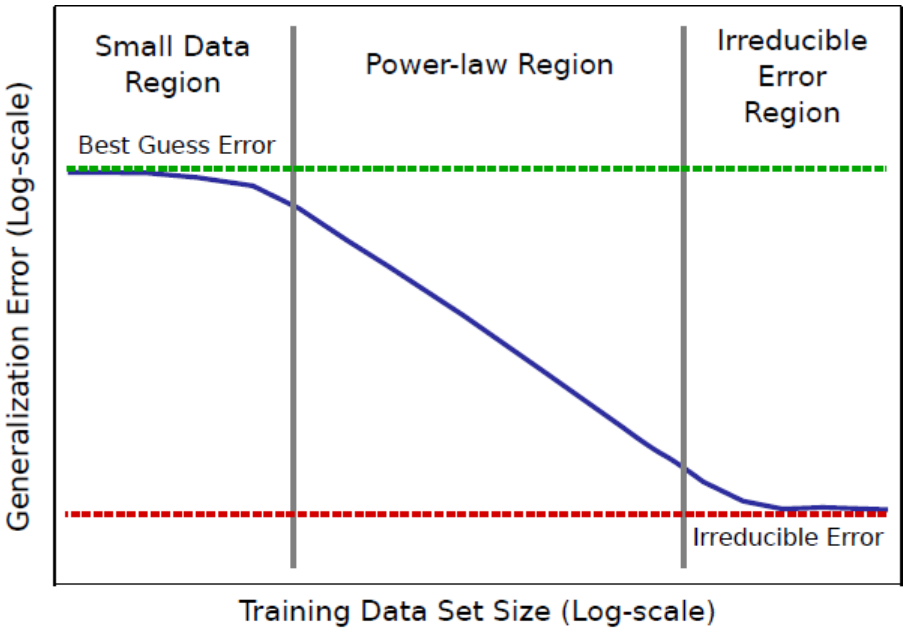
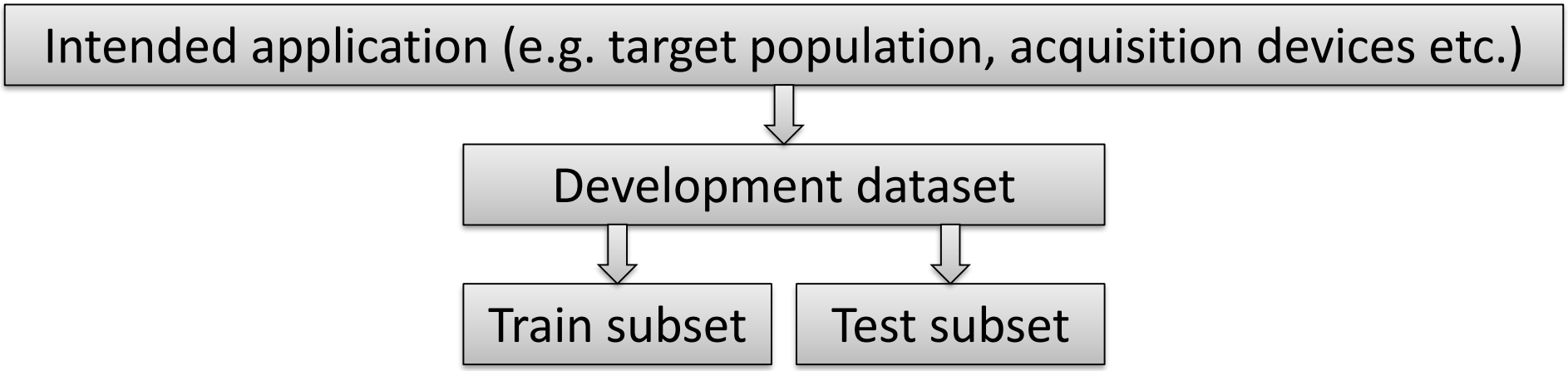


Figure 6: Sketch of power-law learning curves

Hestness, J. et al. Deep learning scaling is predictable, empirically. https://arxiv.org/pdf/1712.00409.pdf (2017).

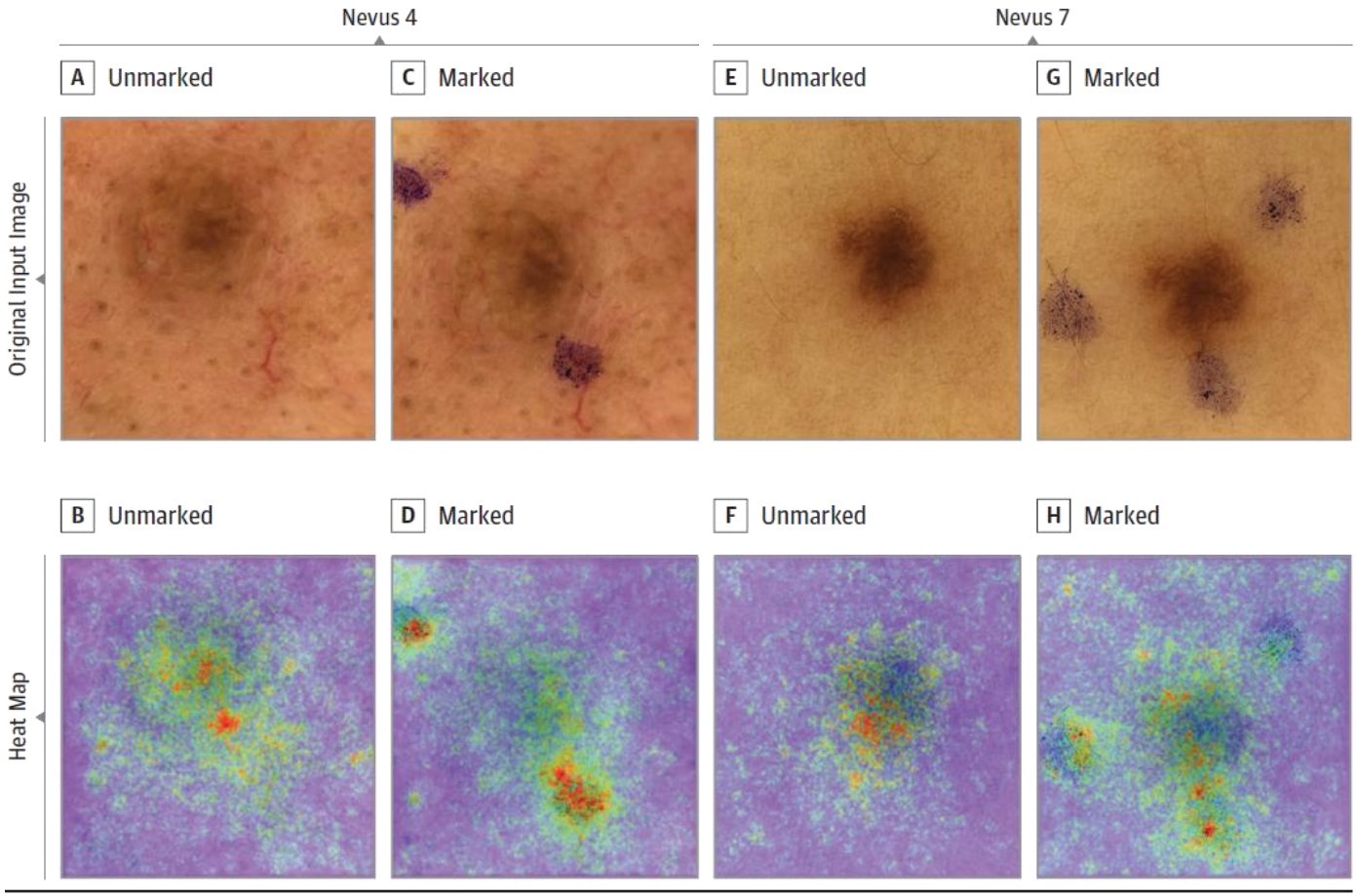
Representability

- Recap: If a random test subset is used for only a single analysis, then the resulting estimate is expected to be unbiased for the development data.
- It is often assumed that the development dataset is a set of realisations of independent and identically distributed random variables.
- If that is true and the distribution of the random variables is that of the intended application, then the single analysis of the random test subset would be unbiased for the intended application.
- But should be expect the development data to be representative of intended application?



Representability

Figure 4. Heat Maps of 2 Benign Nevi With Major Increase in Melanoma Probability Scores After Addition of In Vivo Skin Markings



The heat maps were created by vanilla (meaning basic) gradient descent backpropagation. A and E, Unmarked input images. B and F, Heat maps reveal relevant pixels for the convolutional neural network's (CNN's) prediction of benign nevi. C and G, Marked input images. D and H, Heat maps reveal that skin markings are of high relevance for CNN's changed prediction of malignant melanomas, while the nevus itself is mostly ignored.

Winkler, J. K. et al. Association between surgical skin markings in dermoscopic images and diagnostic performance of a deep learning convolutional neural network for melanoma recognition. *JAMA Dermatol.* **155**, 1135–1141 (2019).

Representability

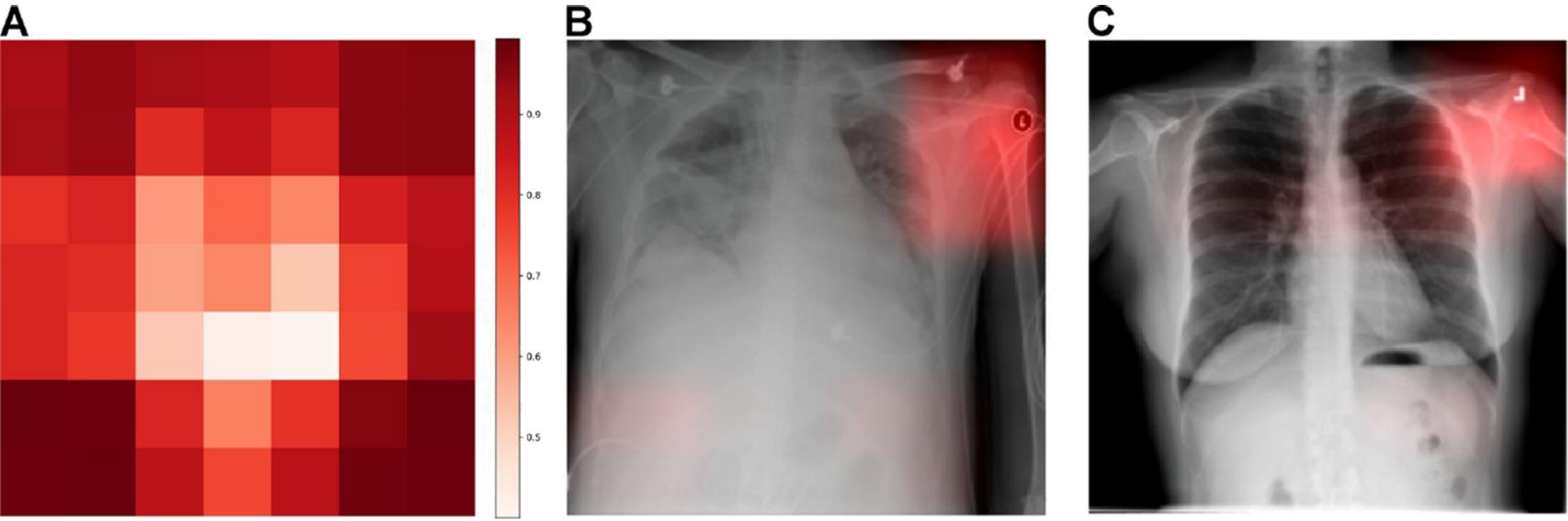


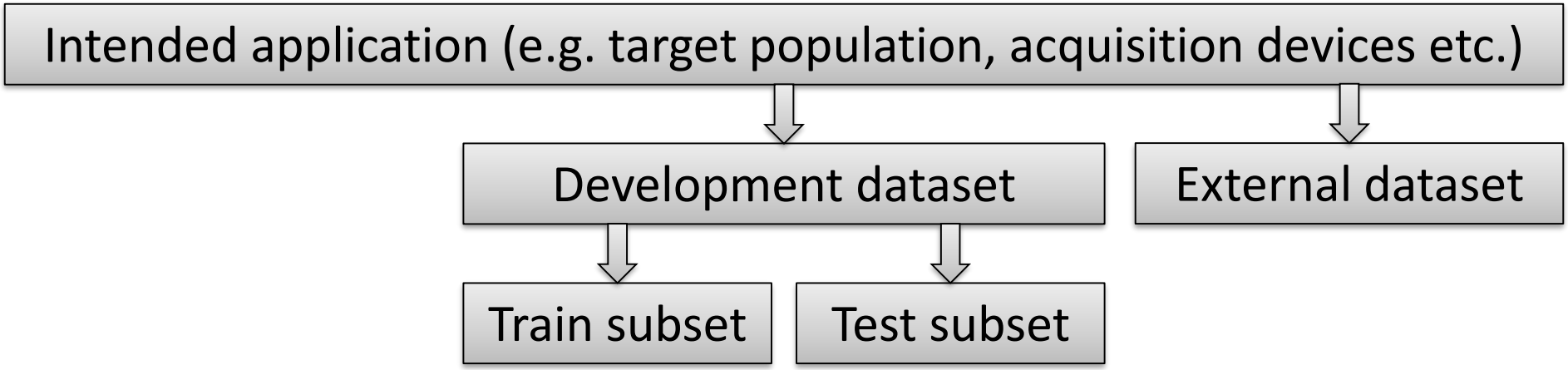
Fig 2. CNN to predict hospital system detects both general and specific image features. (A) We obtained activation heatmaps from our trained model and averaged over a sample of images to reveal which subregions tended to contribute to a hospital system classification decision. Many different subregions strongly predicted the correct hospital system, with especially strong contributions from image corners. (B-C) On individual images, which have been normalized to highlight only the most influential regions and not all those that contributed to a positive classification, we note that the CNN has learned to detect a metal token that radiology technicians place on the patient in the corner of the image field of view at the time they capture the image. When these strong features are correlated with disease prevalence, models can leverage them to indirectly predict disease. CNN, convolutional neural network.

Zech, J. R. et al. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross- sectional study. *PLoS Med.* **15**, e1002683 (2018).

Representability

- Random test subset are similar to the training data.
- This could cause the performance to be overestimated e.g.:
  - because there exists input data features that correlate with the target outcome only in the development data, or
  - because important predictive features are not adequately represented in the development data.
- Thus, data external to the development data are needed to obtain more realistic performance estimates.
  - What is considered “external” varies a bit, but it at least includes data from different geographical locations than the development data.
  - Note: External data is a luxury that is too seldom available.

External datasets



- External dataset representative of the intended application may be used for unbiased performance estimation.
- In practice, often not representative of the entire intended application, but it could be representative of a part of it.
- Good performance on an external dataset often indicates:
  - good generalisability to one of multiple intended settings, and
  - suggests that the model would also generalise to other settings.

Use of external datasets

- Performance estimated using an external dataset becomes overoptimistic if multiple models are evaluated and the best one selected.
  - Analogous to multiple testing on a test subset.

- The external dataset should therefore ideally be used only once to evaluate the final model.
  - Details related to the evaluation should ideally also be pre-specified, e.g. which subjects, what data, and which performance metric.
  - Any post-hoc adjustments might bias the performance estimation.

Use of external datasets

- In the most influential deep learning studies, evaluation on external dataset is becoming common.
  - In particular in studies published in journals with impact factor ≥10.
  - This is at least evident in the field of cancer diagnostics:

"Influential" is here defined as ≥20 citations per year or published in journals with impact factor ≥10.

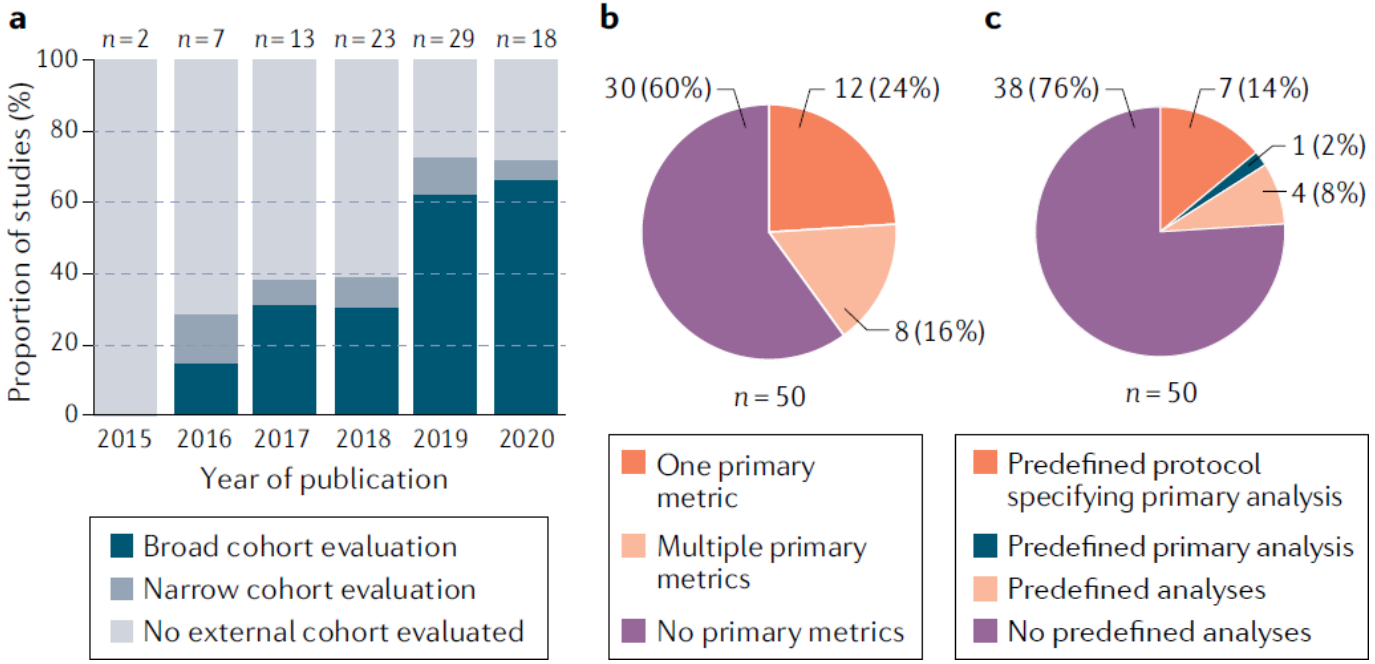


Figure 1 in Kleppe, A. et al. Designing deep learning studies in cancer diagnostics. *Nat. Rev. Cancer* **21**, 199–211 (2021).

- Evaluations of external datasets are rarely reported to be predefined.
  - Except in clinical trial studies, where pre-specification is usually required.

Facilitating generalisation

- Control the network's capacity.
  - Reduce the number of adjustable parameters.
    - Stacking convolutional layers with small kernel sizes.
    - Depthwise separable convolutions.
    - Reduce width (e.g. number of channels in convolutional layers) and depth (number of layers).
      - Should be scaled appropriately with input size (e.g. image resolution), e.g. as in EfficientNet.
  - Dropout.
  - Weight decay.

Facilitating generalisation

- Control the network's capacity.
- Facilitate learning.
  - Residual connections.
  - Batch normalisation.
  - Transfer learning.
  - Learning rate schedule.
  - Optimisation method.

Facilitating generalisation

- Control the network's capacity.
- Facilitate learning.
- Data normalisation.
  - Applies an algorithm to transform individual images before they are inputted to the model (often when training and when predicting).
  - Aims at making different images more similar.
  - Separate from standardising the images.
    - E.g. subtracting the channel-wise mean and dividing by the channel-wise standard deviation estimated over the training data.
- Data augmentation.
  - E.g. distort brightness, colour, contrast, rotation, sharpness, or size.
  - Shorten, C. & Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *J. Big Data* **6**, 60 (2019).
- More diverse training data.

Generalisation: Example for tumour detection

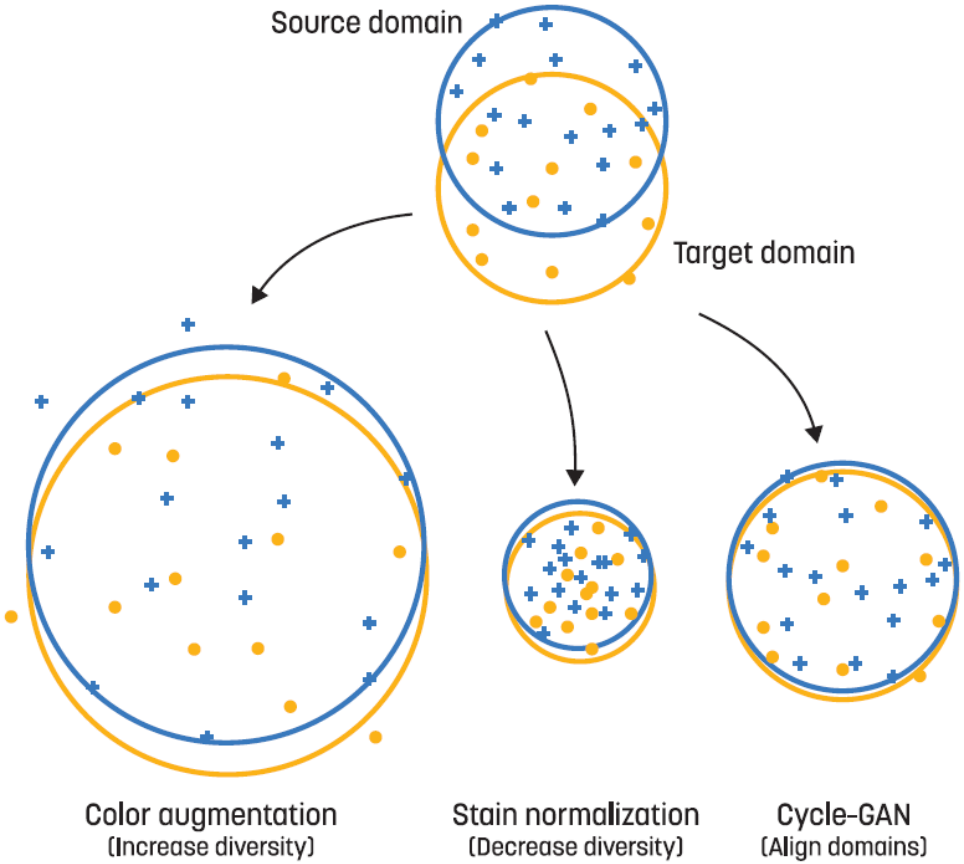
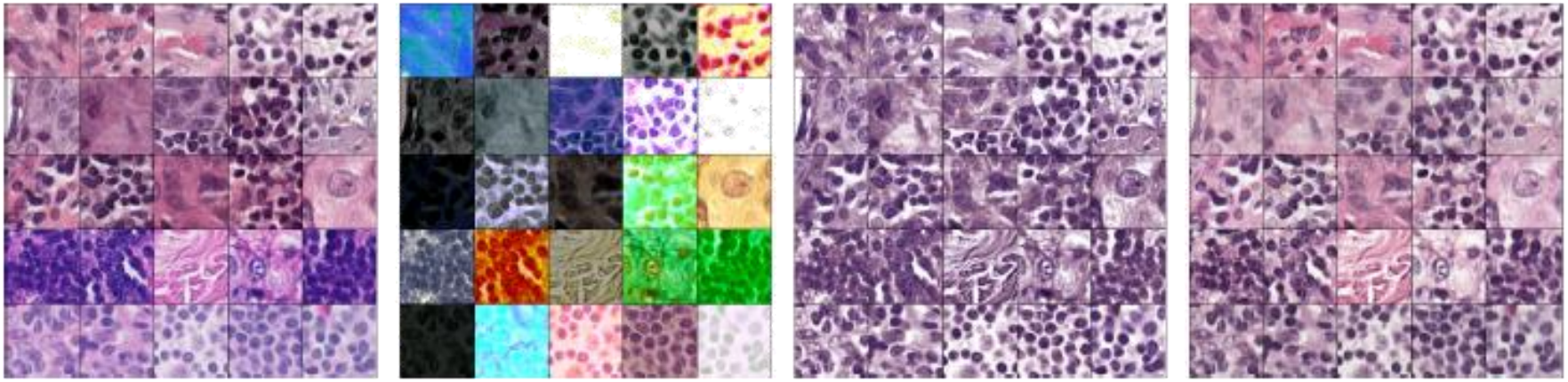


Fig. 6. Data preparation aim at increasing the intersection between domains. Different data transformation techniques do this with different strategies.

Stacke, K., Eilertsen, G., Unger, J. & Lundström, C. Measuring domain shift for deep learning in histopathology. *IEEE J. Biomed. Health Inform.* **25**, 325–336 (2021).

Generalisation: Example for tumour detection



(a) Original **(b)** Color augmentation **(c)** Stain normalization **(d)** Cycle-GAN

Fig. 5. Five example images per center from the CAMELYON17 dataset (total 5x5 images), shown with different transformations. The three top rows per image are scanned with Scanner 1, the forth row with Scanner 2 and the last row with Scanner 3.

Stacke, K., Eilertsen, G., Unger, J. & Lundström, C. Measuring domain shift for deep learning in histopathology. *IEEE J. Biomed. Health Inform.* **25**, 325–336 (2021).

Generalisation: Example for tumour detection

TABLE I

CLASS BALANCED MEAN ACCURACIES (STANDARD DEVIATION) (%), TRAINING A TUMOR CLASSIFIER ON SCANNER 1 FOR CAMELYON17, AND CENTER 1 FOR AIDA-LNCO, TESTING IT ON SAME-SCANNER/CENTER DATA AND OTHER DATA, USING TWO DIFFERENT MODEL ARCHITECTURES, EVALUATED ACROSS 5 INDEPENDENT TRAINING RUNS

CAMELYON17

	Simple CNN				*Mini-GoogLeNet*			
Train \ Test	Scanner 1	Scanner 2	Scanner 3	Mean drop (p.p.)	Scanner 1	Scanner 2	Scanner 3	Mean drop (p.p.)
Orig. data	90.0 (5.0)	48.9 (1.2)	72.3 (12.4)	29.5 (9.7)	95.4 (2.5)	49.7 (3.4)	73.8 (12.8)	33.6 (7.4)
Color aug.	94.5 (1.3)	91.4 (0.8)	89.3 (3.9)	4.1 (1.7)	96.4 (0.7)	90.3 (0.2)	93.5 (1.1)	4.5 (0.5)
Stain norm.	94.1 (2.1)	93.5 (1.3)	92.7 (2.0)	0.9 (1.5)	97.8 (1.0)	93.5 (2.5)	96.5 (0.7)	2.8 (1.3)
Cycle-GAN	95.0 (1.3)	91.0 (1.7)	91.8 (0.6)	3.6 (0.8)	97.0 (0.7)	90.1 (2.5)	94.4 (0.4)	4.7 (1.6)

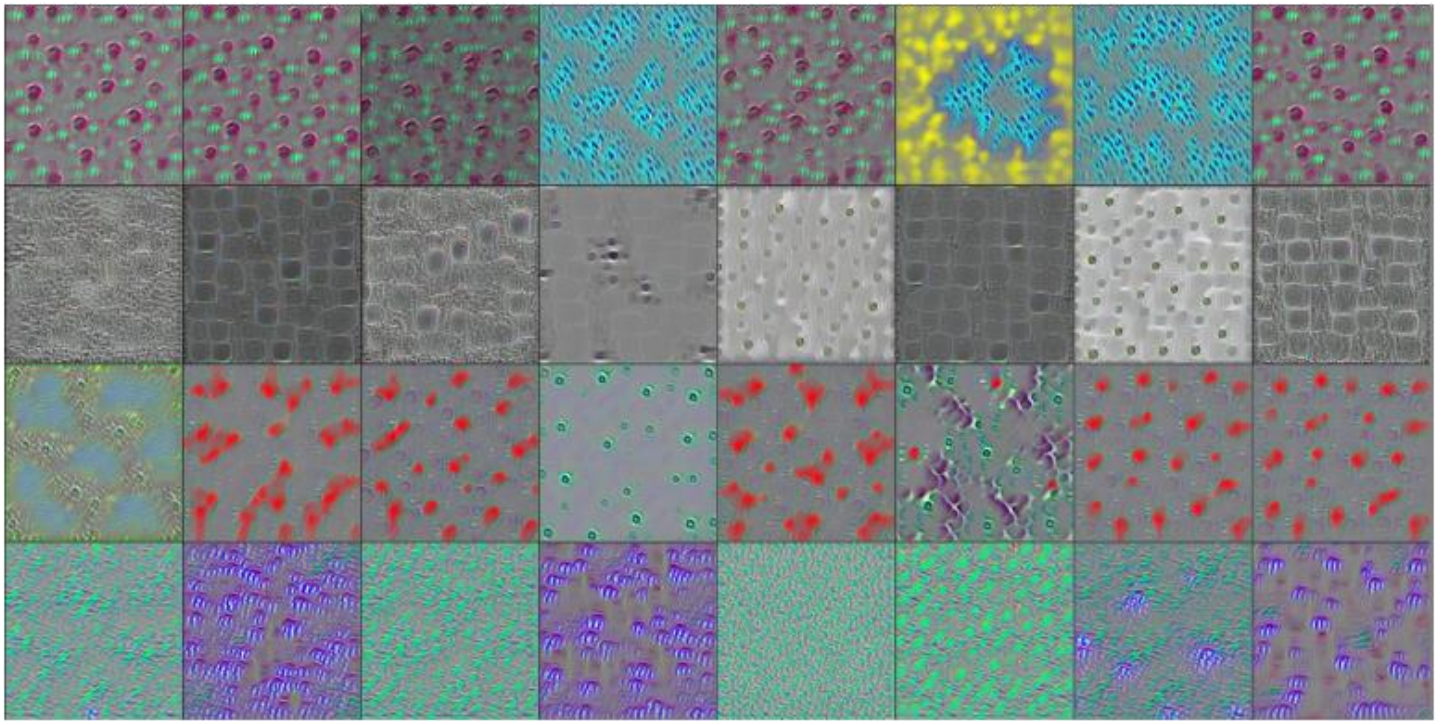
AIDA-LNCO

	Simple CNN			*Mini-GoogLeNet*		
Train \ Test	Center 1	Center 2	Mean drop (p.p.)	Center 1	Center 2	Mean drop (p.p.)
Orig. data	98.3 (0.8)	73.3 (6.6)	25.0 (5.8)	97.3 (1.3)	72.8 (8.1)	24.5 (7.5)
Color aug.	96.0 (3.3)	95.8 (1.0)	0.3 (2.5)	95.3 (4.6)	95.5 (2.0)	−0.2 (2.7)
Stain norm.	96.3 (4.0)	95.5 (3.1)	0.8 (0.9)	95.4 (3.7)	95.7 (2.6)	−0.3 (2.0)
Cycle-GAN	98.3 (1.2)	77.7 (3.4)	20.6 (3.6)	97.2 (1.3)	81.7 (10.7)	15.5 (10.5)

Stacke, K., Eilertsen, G., Unger, J. & Lundström, C. Measuring domain shift for deep learning in histopathology. *IEEE J. Biomed. Health Inform.* **25**, 325–336 (2021).

- For the generalisation to external data, it could be that colour augmentation helps primarily because it increases the variability of the input, not because it artificially increases the amount of training data.

Generalisation: Example for tumour detection



(b) Mini-GoogLeNet, the convolutional layer in the first auxiliary branch

Fig. 2. Each row shows example images that maximally activates different filters in models Simple CNN (a) and Mini-GoogLeNet (b), trained with (from top to bottom): original data, color augmentation, stain normalization, and Cycle-GAN. Best viewed digitally.

Stacke, K., Eilertsen, G., Unger, J. & Lundström, C. Measuring domain shift for deep learning in histopathology. *IEEE J. Biomed. Health Inform.* **25**, 325–336 (2021).

Generalisation: Augmentation+normalisation?

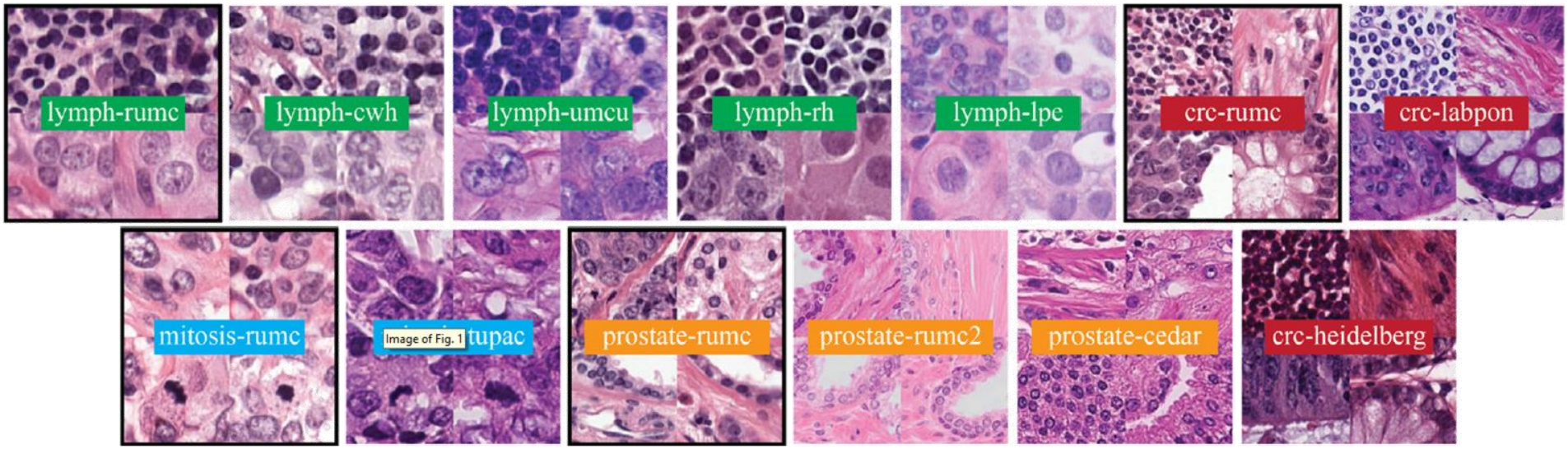


Fig. 1. Example images from training and test datasets. Applications are indicated by colors and keywords: tumor detection in lymph nodes (*lymph*), colorectal cancer tissue classification (*crc*), mitosis detection (*mitosis*) and prostate epithelium detection (*prostate*). Training set images are indicated by the keyword *rumc* and black outline. The rest belong to test sets from other centers. Stain variation can be observed between training and test images.

Tellez, D. et al. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Med. Image Anal.* **58**, 101544 (2019).

Generalisation: Augmentation+normalisation?

Table 1
Experimental results ranking *stain color augmentation* and *stain color normalization* methods. Values correspond to AUC scores, except for the last column, averaged across 5 repetitions with standard deviation shown between parenthesis. Each column represents a different external test dataset, with the last column *Ranking* indicating the position of each method within the global benchmark, computed as described in Section 4.1. Normalization methods: *Network* is our proposal; *Style* is from Bug et al. (2017); *LUT* is from Bejnordi et al. (2016); and *Deconvolution* is from Macenko et al. (2009).

Normalization	Augmentation	lymph-cwh	lymph-lpe	lymph-rh	lymph-umcu	mitosis-tupac	prostate-rumc2	prostate-cedar	crc-labpon	crc-heidelberg	Ranking
Identity	HED-light	0.952(0.004)	0.976(0.001)	0.946(0.009)	0.968(0.004)	0.996(0.001)	0.957(0.001)	0.879(0.011)	0.973(0.002)	0.895(0.002)	1.2(0.4)
Style	HED-light	0.961(0.002)	0.953(0.004)	0.952(0.001)	0.972(0.004)	0.991(0.003)	0.925(0.003)	0.879(0.006)	0.975(0.001)	0.917(0.003)	2.8(0.7)
Network	HSV-light	0.946(0.006)	0.962(0.001)	0.941(0.002)	0.965(0.004)	0.992(0.001)	0.957(0.000)	0.872(0.013)	0.980(0.001)	0.900(0.003)	3.9(1.9)
Network	HED-light	0.949(0.005)	0.968(0.001)	0.942(0.002)	0.963(0.004)	0.989(0.003)	0.958(0.001)	0.862(0.011)	0.980(0.001)	0.906(0.003)	4.1(1.6)
Identity	HSV-strong	0.955(0.003)	0.965(0.004)	0.929(0.002)	0.973(0.003)	0.988(0.003)	0.945(0.009)	0.886(0.005)	0.977(0.001)	0.902(0.003)	4.7(1.7)
Network	HSV-strong	0.953(0.002)	0.964(0.003)	0.946(0.002)	0.964(0.005)	0.991(0.003)	0.951(0.002)	0.852(0.006)	0.975(0.002)	0.894(0.005)	6.6(0.9)
Network	HED-strong	0.956(0.003)	0.959(0.002)	0.940(0.003)	0.965(0.004)	0.985(0.005)	0.943(0.003)	0.861(0.009)	0.974(0.002)	0.916(0.003)	7.9(1.9)
Identity	HED-strong	0.950(0.005)	0.959(0.005)	0.936(0.007)	0.957(0.007)	0.992(0.002)	0.945(0.003)	0.872(0.005)	0.967(0.003)	0.920(0.005)	8.1(2.8)
Style	HSV-strong	0.953(0.004)	0.956(0.004)	0.940(0.003)	0.959(0.007)	0.986(0.004)	0.932(0.005)	0.878(0.003)	0.976(0.001)	0.917(0.004)	9.0(2.6)
Style	HSV-light	0.940(0.011)	0.960(0.004)	0.944(0.007)	0.926(0.012)	0.992(0.001)	0.958(0.002)	0.852(0.008)	0.974(0.001)	0.921(0.003)	9.4(3.6)
Style	HED-strong	0.955(0.002)	0.949(0.004)	0.936(0.003)	0.954(0.005)	0.982(0.005)	0.942(0.002)	0.884(0.004)	0.975(0.000)	0.925(0.003)	9.9(1.2)
Grayscale	BC	0.956(0.003)	0.962(0.003)	0.935(0.005)	0.961(0.002)	0.989(0.002)	0.939(0.004)	0.851(0.002)	0.972(0.000)	0.884(0.003)	12.2(1.2)
Deconvolution	HSV-strong	0.955(0.003)	0.936(0.008)	0.941(0.004)	0.943(0.009)	0.991(0.001)	0.865(0.010)	0.867(0.004)	0.961(0.002)	0.928(0.001)	13.9(1.9)
LUT	HED-strong	0.934(0.006)	0.941(0.006)	0.925(0.006)	0.963(0.005)	0.989(0.002)	0.945(0.002)	0.871(0.005)	0.956(0.002)	0.945(0.001)	14.0(2.2)
Deconvolution	HED-strong	0.942(0.003)	0.962(0.003)	0.897(0.006)	0.967(0.003)	0.993(0.002)	0.827(0.018)	0.853(0.006)	0.969(0.001)	0.927(0.002)	14.4(1.2)
LUT	HSV-strong	0.923(0.009)	0.939(0.003)	0.928(0.005)	0.947(0.008)	0.987(0.002)	0.949(0.003)	0.862(0.007)	0.962(0.002)	0.940(0.002)	17.0(2.0)
Network	BC	0.944(0.003)	0.950(0.003)	0.903(0.003)	0.934(0.006)	0.983(0.005)	0.953(0.003)	0.869(0.009)	0.981(0.001)	0.881(0.005)	17.4(1.5)
Identity	HSV-light	0.888(0.013)	0.951(0.009)	0.942(0.004)	0.930(0.023)	0.962(0.015)	0.949(0.001)	0.905(0.005)	0.976(0.000)	0.894(0.003)	17.4(2.9)
LUT	HED-light	0.914(0.011)	0.926(0.011)	0.923(0.006)	0.932(0.019)	0.993(0.001)	0.948(0.003)	0.852(0.021)	0.966(0.003)	0.940(0.002)	17.9(2.2)
LUT	HSV-light	0.894(0.006)	0.936(0.006)	0.921(0.003)	0.942(0.007)	0.987(0.002)	0.951(0.002)	0.860(0.010)	0.971(0.002)	0.945(0.002)	19.2(1.2)
LUT	BC	0.925(0.025)	0.948(0.027)	0.853(0.016)	0.790(0.061)	0.985(0.004)	0.951(0.004)	0.848(0.018)	0.973(0.003)	0.924(0.005)	21.3(3.3)
Style	BC	0.949(0.005)	0.858(0.031)	0.938(0.001)	0.411(0.065)	0.987(0.004)	0.949(0.006)	0.764(0.047)	0.946(0.002)	0.903(0.005)	23.3(2.2)
Deconvolution	HSV-light	0.942(0.004)	0.930(0.009)	0.913(0.023)	0.961(0.002)	0.982(0.005)	0.850(0.019)	0.840(0.009)	0.958(0.006)	0.917(0.002)	23.5(1.0)
Network	Basic	0.944(0.003)	0.954(0.007)	0.887(0.010)	0.959(0.004)	0.969(0.005)	0.905(0.006)	0.815(0.019)	0.977(0.002)	0.855(0.006)	23.6(1.4)
Network	Morphology	0.939(0.010)	0.949(0.006)	0.890(0.012)	0.950(0.009)	0.980(0.006)	0.913(0.011)	0.823(0.022)	0.977(0.001)	0.868(0.002)	23.9(1.1)
Deconvolution	HED-light	0.930(0.005)	0.912(0.015)	0.916(0.005)	0.948(0.006)	0.982(0.002)	0.816(0.011)	0.834(0.004)	0.970(0.003)	0.927(0.005)	25.4(2.3)
Deconvolution	Morphology	0.951(0.003)	0.938(0.006)	0.849(0.021)	0.951(0.008)	0.993(0.002)	0.754(0.027)	0.749(0.037)	0.903(0.008)	0.865(0.015)	27.7(0.6)
Grayscale	Morphology	0.943(0.010)	0.820(0.021)	0.922(0.005)	0.941(0.011)	0.991(0.006)	0.910(0.009)	0.816(0.005)	0.929(0.006)	0.813(0.009)	27.7(1.2)
Style	Morphology	0.935(0.011)	0.725(0.082)	0.934(0.002)	0.361(0.113)	0.992(0.004)	0.918(0.006)	0.754(0.006)	0.873(0.010)	0.890(0.006)	28.6(2.4)
Grayscale	Basic	0.940(0.007)	0.692(0.064)	0.926(0.010)	0.938(0.019)	0.992(0.001)	0.882(0.008)	0.661(0.039)	0.934(0.002)	0.798(0.006)	30.0(0.6)
Deconvolution	BC	0.942(0.004)	0.896(0.008)	0.682(0.044)	0.949(0.005)	0.989(0.006)	0.794(0.021)	0.792(0.028)	0.930(0.007)	0.872(0.004)	30.4(1.2)
LUT	Morphology	0.898(0.007)	0.920(0.007)	0.801(0.021)	0.874(0.025)	0.969(0.008)	0.895(0.013)	0.803(0.007)	0.939(0.006)	0.906(0.006)	32.6(1.4)
Deconvolution	Basic	0.919(0.015)	0.896(0.038)	0.810(0.081)	0.902(0.026)	0.993(0.001)	0.753(0.006)	0.791(0.009)	0.903(0.003)	0.836(0.008)	32.8(0.7)
Style	Basic	0.918(0.002)	0.334(0.133)	0.926(0.004)	0.124(0.041)	0.991(0.003)	0.865(0.025)	0.723(0.024)	0.863(0.020)	0.857(0.010)	33.6(1.0)
LUT	Basic	0.908(0.010)	0.894(0.030)	0.809(0.022)	0.772(0.072)	0.951(0.009)	0.906(0.011)	0.741(0.018)	0.930(0.014)	0.890(0.013)	34.6(0.8)
Identity	BC	0.899(0.006)	0.634(0.100)	0.741(0.016)	0.177(0.047)	0.906(0.034)	0.936(0.006)	0.704(0.060)	0.684(0.009)	0.761(0.012)	36.2(0.4)
Identity	Morphology	0.811(0.026)	0.671(0.099)	0.673(0.027)	0.214(0.174)	0.986(0.006)	0.374(0.191)	0.602(0.023)	0.569(0.028)	0.720(0.009)	37.2(0.7)
Identity	Basic	0.811(0.009)	0.563(0.309)	0.790(0.047)	0.406(0.375)	0.965(0.009)	0.631(0.178)	0.624(0.053)	0.556(0.057)	0.701(0.028)	37.6(0.5)

Tellez, D. et al. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Med. Image Anal.* **58**, 101544 (2019).

Generalisation: Diverse training data

- The natural variability in the training data matters.
- Data distortion might not be able to fully compensate for reduction in natural data variation.
- Artificial and natural data variation complements each other.

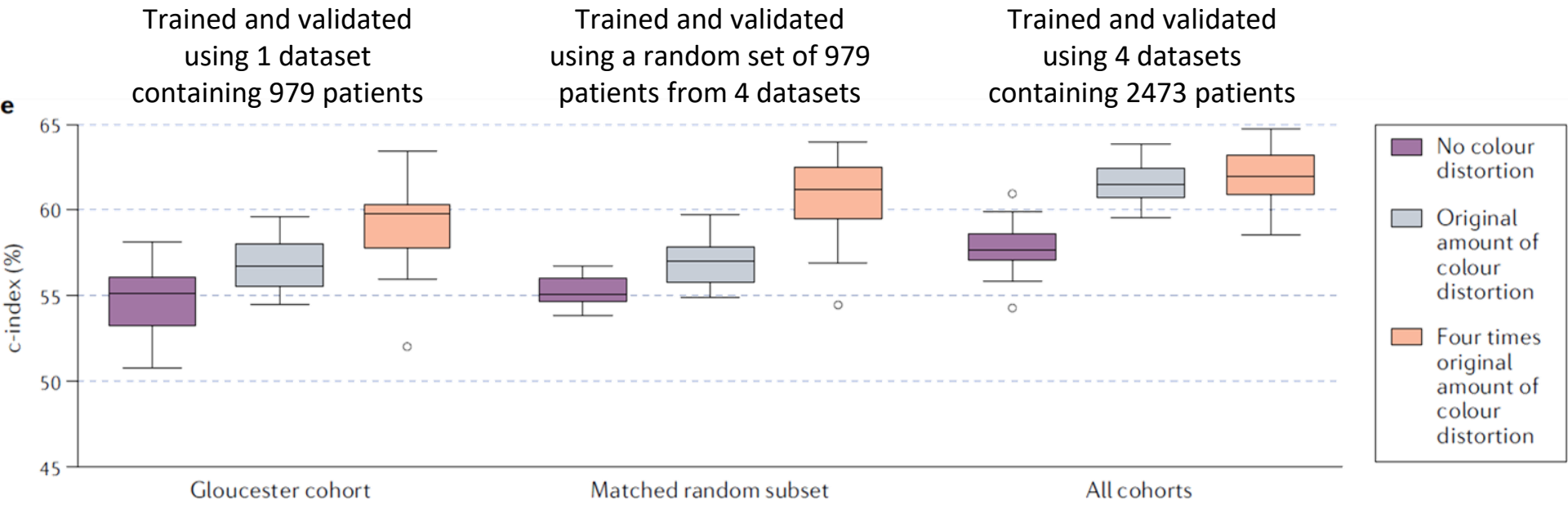


Figure 2e in Kleppe, A. et al. Designing deep learning studies in cancer diagnostics. *Nat. Rev. Cancer* **21**, 199–211 (2021).

Generalisation: Too much distortion

- There is a trade-off between facilitating the learning of relations that generalise well beyond the development dataset and not occluding relevant information for the prediction task.

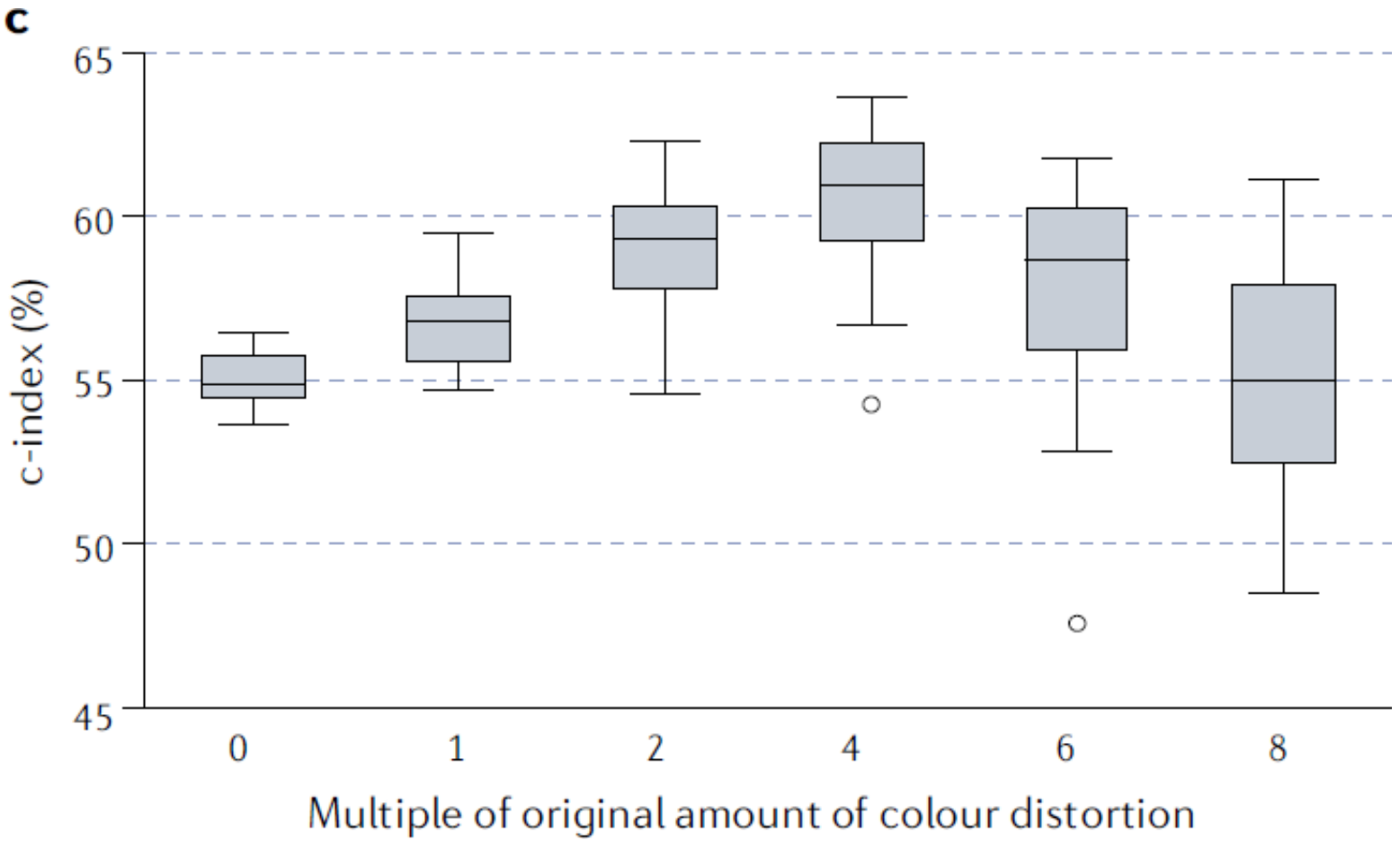


Figure 2c in Kleppe, A. et al. Designing deep learning studies in cancer diagnostics. *Nat. Rev. Cancer* **21**, 199–211 (2021).

Facilitating generalisation

- Varied training data or data normalisation might be particularly important to facilitate generalisation to external datasets.
  - Controlling the network's capacity and facilitating learning might also help.
- Amount of data distortion (i.e. which distortion algorithms and the selected values of its hyperparameters) or the particular normalisation method should be adapted to the input type and the prediction task.
  - Not easy to identify the best option because subsets of the development dataset and external datasets might indicate different options to be best.
  - If external data is not available to perform the adaption:
    - For data augmentation, a heuristic is to use as much distortion as possible without decreasing the validation performance substantially.
    - For data normalisation, the train (and any validation) subset could be inspected visually and by looking at relevant quantitative statistics.

Performance metric

- Select a performance metric that reflects the intended usage of your model.
  - If the end user of your model will simply use the classifications, then the performance metric should also only use the classifications (not the prediction scores).
    - Note that metrics calculated from classifications are often more conservative than metrics calculated from prediction scores.
  - If chosen inappropriately, it could become difficult to interpret how well the model will perform in the intended application, both in itself and relative to other models.
- Some performance metrics ignore relevant aspects of an evaluation and should be complemented by other statistics.

Recap: Precision and recall

Assuming a ***single class***: “animal”

FP (false positive)

FN (false negative)

TP (true positives)

Note: Requires an ***IoU threshold***, e.g. 0.5 or 0.75.

Recap: Precision and recall

Remember: Precision/recall is calculated **per class**!

$$\text{Precision} = \frac{TP}{TP + FP}$$

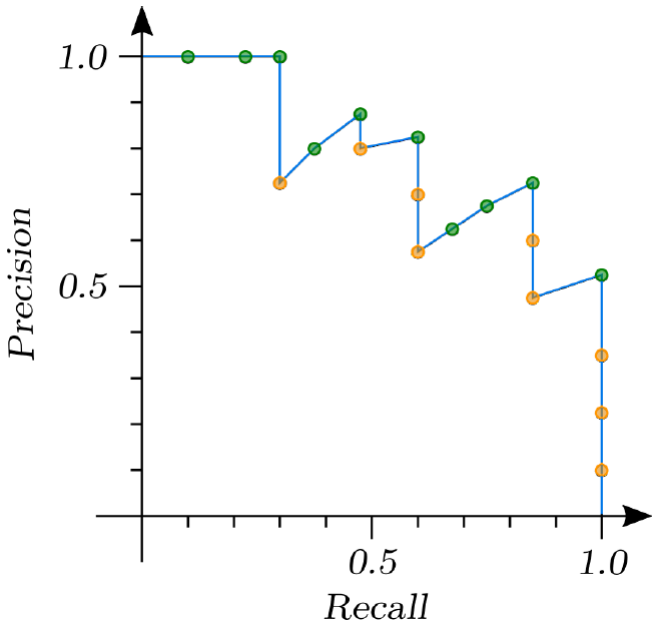
(How many of our predictions are actually true/objects)

$$\text{Recall} = \frac{TP}{TP + FN}$$

(How many of the objects of interest are found)

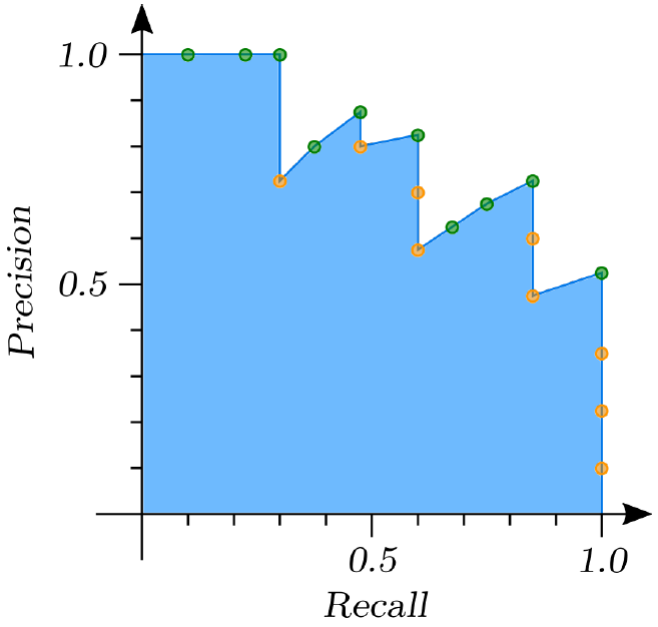
Recap: Precision-recall curves

1. Sort the prediction scores for all detected objects.
2. If requiring a prediction score larger than all observed values, we get no predictions (so recall is 0) and define precision to be 1 (the expression is 0/0).
3. Repeat until the threshold is below all prediction scores:
    1. Lower the threshold to include one more of the observed prediction scores.
    2. Calculate the recall and precision when using the new threshold.
    3. Plot the (recall, precision) pair and draw a straight line from the previous point.
4. Draw a straight line from the previous point to (1, 0).



Recap: Average precision (AP)

- The area under the precision-recall curve is the *average precision* (for that class).
  - It is common to compute the area below an interpolated precision, which is calculated at each observed recall by taking the maximum precision measured for that recall. This removes the “wiggles” in the precision-recall curve.

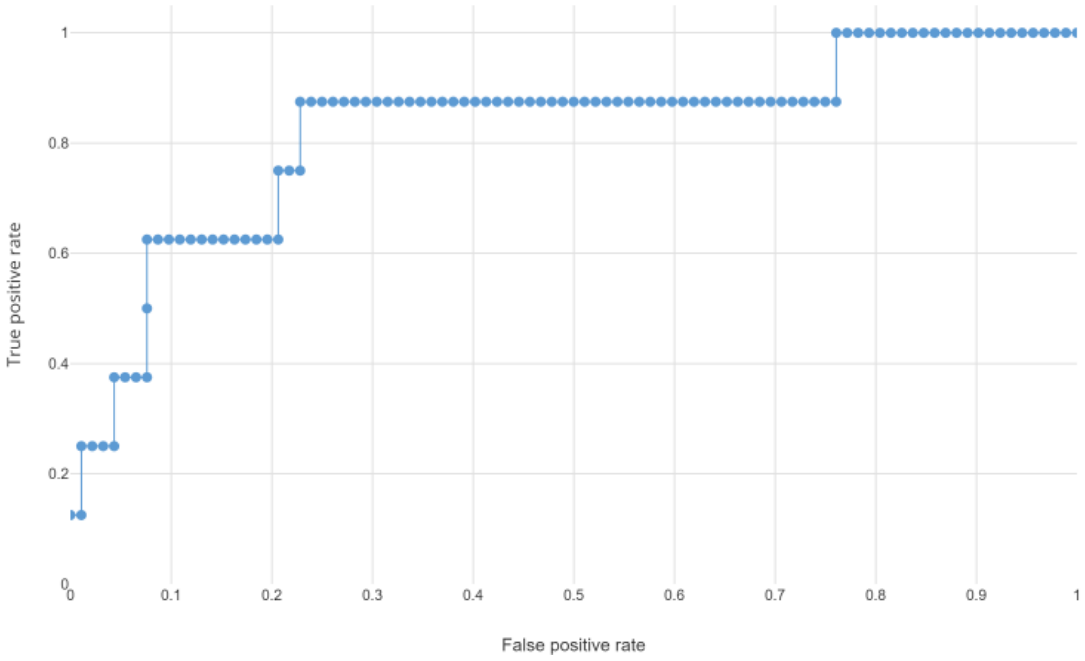


Sensitivity and specificity

- For classification tasks, recall is often called *sensitivity*.
  - Is the proportion of images with a specific class that are correctly classified by the model as belonging to that class.
- For binary classification tasks:
  - There are only two sensitivities.
  - The sensitivity of the “negative” class (e.g. those without a condition) is often called the specificity or true negative rate.
  - The sensitivity of the “positive” class (e.g. those with a condition) is often referred to as the sensitivity or true positive rate.

Receiver Operator Characteristic (ROC) curves

- ROC curves can be created using the approach described for precision-recall curves, except that:
  - The false positive rate (1-specificity) and true positive rate (sensitivity) is calculated and plotted instead of recall and precision.
  - ROC curves start at (0,0) and end at (1,1), while precision-recall curves start at (0,1) and end at (1,0).



Area under the curve (AUC)

- AUC is the area under the ROC curve.
- It can be shown that AUC can be calculated as the proportion of pairs where the prediction scores and target output class are concordant.
  - “Concordant” means that the prediction score for the positive class is higher for the data point with positive label than for the data point with negative label.
    - Requires one positive and one negative in each pair, thus these are the pairs that are considered when calculating the AUC in this way.
  - This means that the AUC can be viewed as the probability that the predicted score for the positive class is larger than the predicted score for the negative class.

Not exam relevant: c-index (Harrell's concordance index)

- A generalisation of AUC to time-to-event data.
  - In time-to-event data, each subject is associated with a time in addition to the binary output.
  - For subjects in the positive class, the time specifies when an event occurred.
  - For subjects in the negative class, the time specifies how long the subject was observed to not experience the event (it is unknown whether or not the subject subsequently experience the event).
- The pairs considered when calculating the c-index are those where one subject experienced the event and the other subject was followed at least as long without experiencing the event.
  - So the other subject had not experienced the event at the time when the first subject in the pair experienced the event.
  - This implies that the other subject's time must be equal or larger to the time of the first subject, but note that both subjects could have experience the event (though at different points in time).
  - If all times are equal, then the pairs are as for the AUC, which makes the c-index and the AUC equal in this case.

AP, AUC, and c-index

- Summarises the discriminatory value of the prediction scores.
- Only uses the ranking of the prediction scores, not their absolute values.
  - => Invariant to any strictly monotonic transform of the prediction scores.
    - That is, the metric value of the transformed scores will be the same as the metric value of the original scores.
- Thus, these metrics ignore possibly relevant aspects of an evaluation.
  - Assume we want to evaluate a model using two external datasets where each of the two target output class are equally common.
  - If the model classifies all samples in one of the datasets as the positive class (the prediction score for the positive class is always very high) and all samples in the other dataset as the negative class (the prediction score for the positive class is always very low), the model clearly has generalised poorly.
  - However, if the prediction scores rank the subjects in a fairly correct order in each of the datasets, then the AP, AUC, and c-index would all be rather high.

AP, AUC, and c-index

- Looking at the distribution of the prediction scores for samples with a particular output class in a given dataset could suggest that the prediction scores inadvertently differ between datasets.
  - Not trivial because differences between datasets might also affect these distributions.
- If the prediction scores are expected to relate to the probability of some observable event, then this relation could be plotted in a calibration plot.

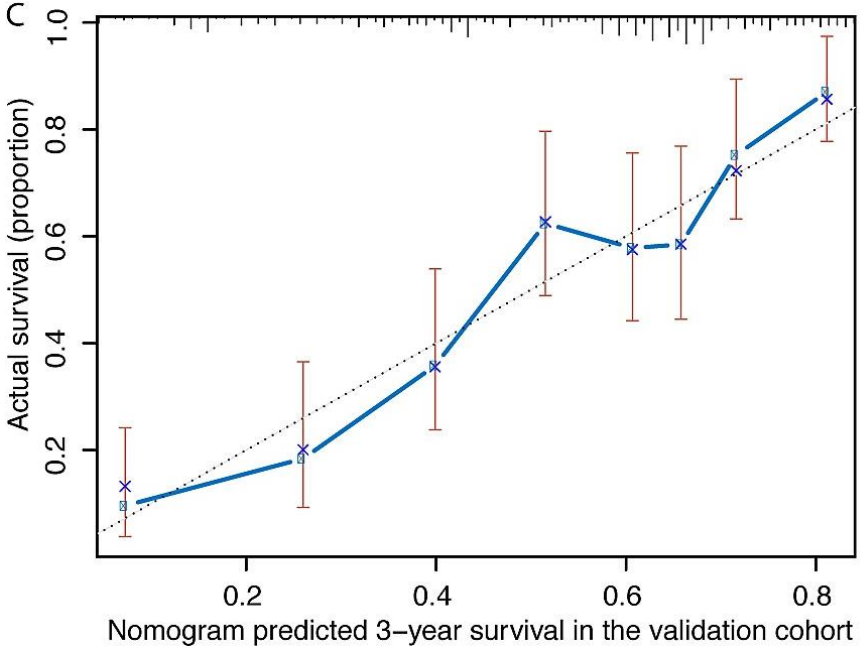


Figure 2c by Su, D. et al. Prognostic Nomogram for Thoracic Esophageal Squamous Cell Carcinoma after Radical Esophagectomy. *PLOS ONE* **10**, e0124437 (2015).

Performance metric

- For many classification tasks, the ultimate goal would simply be to predict the correct class.
- If so, use a classification metric such as balanced accuracy.
  - “Balanced accuracy” is the average of the sensitivity of each class.
- How to obtain classifications from the prediction scores should be predefined.
  - Could be e.g. selecting the class with largest prediction score or defined by thresholds found to be suitable using cross-validation.
  - If instead using the test set to determine how to convert prediction scores to classifications, e.g. by finding thresholds using the test set, then inadvertent transformations of the prediction scores could be occluded even if using a classification metric.

Uncertainty of performance estimate

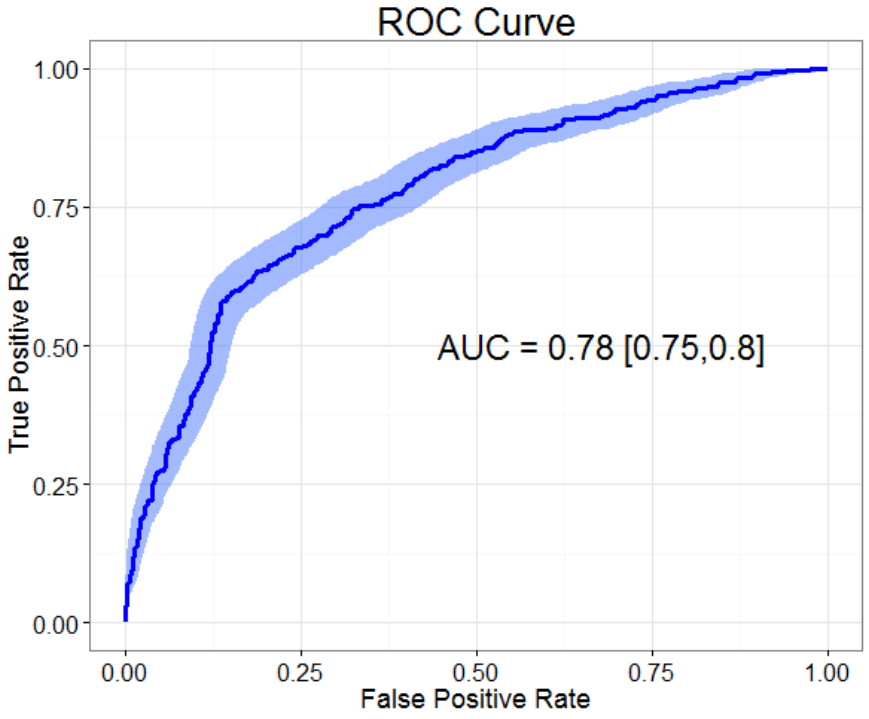
- When estimating the performance using a finite set of data, the estimate is always associated with an uncertainty.
  - Technically, the estimator (i.e. the rule for calculating the estimate) has a distribution around the true performance.
- This uncertainty should be quantified and reported with the estimate of the performance metric.
- Common to estimate a 95% confidence interval.
  - A procedure for creating 95% confidence intervals for a parameter (e.g. a performance metric) should be such that on average 95% of created confidence intervals contain the true value of the parameter.
  - Often misinterpreted as “95% probability of containing the true value”, but given a concrete confidence interval, the interval either contains the true value or not (it is no longer a matter of probability).

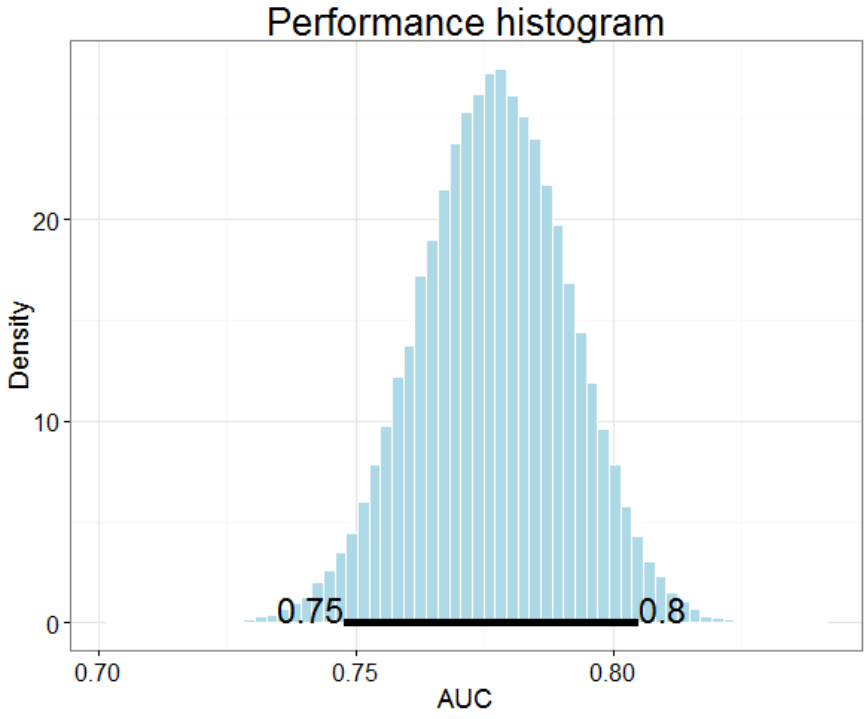
Uncertainty of performance estimate

- For many performance metrics, there exists specific procedures for estimating confidence intervals.
  - E.g. for sensitivities, one of the formulas for binomial proportion confidence interval could be applied, e.g. Clopper-Pearson interval or normal approximation interval.
- Bootstrap confidence intervals are common and can be calculated for (almost?) all metrics.
  - Although bootstrap confidence intervals might be a bit narrow because of the similarity between the bootstrapped samples.
  - Isaksson, A., Wallman, M., Göransson, H. & Gustafsson, M. G. Cross-validation and bootstrapping are unreliable in small sample classification. *Pattern Recognit. Lett.* **29**, 1960–1965 (2008).

Bootstrap confidence interval

- Create a distribution of the parameter by repeating e.g. 1000 times:
  1. Sample with replacement from the original set to obtain a resampled set with the same size as the original set.
  2. Compute the parameter of interest using the resampled set.





https://i2.wp.com/www.epeter-stats.de/wp-content/uploads/2015/05/example-1.png

Bootstrap confidence interval

- The standard bootstrap confidence interval is computed by estimating the standard error of the bootstrapped distribution of the parameter, *s*, and constructing a confidence interval when assuming normal distribution:
  - (<point estimate of parameter> - $z_{\alpha/2}s$, <point estimate of parameter> + $z_{\alpha/2}s$)
  - $z_{\alpha/2}$ is the 100(1-α/2)th percentile of the standard normal distribution, e.g. about 1.96 if constructing a 95% confidence interval (α=0.05).
- There exists other bootstrap confidence interval that are based on the percentiles of the bootstrapped distribution of the parameter.
  - These are better in particular if the performance estimate (i.e. point estimate of the parameter) is close to the theoretical maximum, e.g. 100%.
  - Efron, B. Better Bootstrap Confidence Intervals. *J. Am. Stat. Assoc.* **82**, 171-185 (1987).

Uncertainty of performance estimate

- The confidence interval should be calculated using a test set and will relate to the point estimate of the parameter (the performance estimate) calculated on the same set.
  - This test set is ideally an external dataset.
- If the model performs overoptimistically on the test set, the confidence interval will inherent this overoptimism.
  - This may also happen for external datasets that are used for model selection through testing multiple options and selecting the best one.

Key learning goals

- Be able to critically reason about the reliability of evaluations of the performance of a trained neural network model.
- Be able to suggest approaches for facilitating generalisation, in particular generalisation to external data.
- Understand AUC and limitations of performance metrics that only applies the ranking of prediction scores.
- Interpret a performance estimate as a value with associated uncertainty.