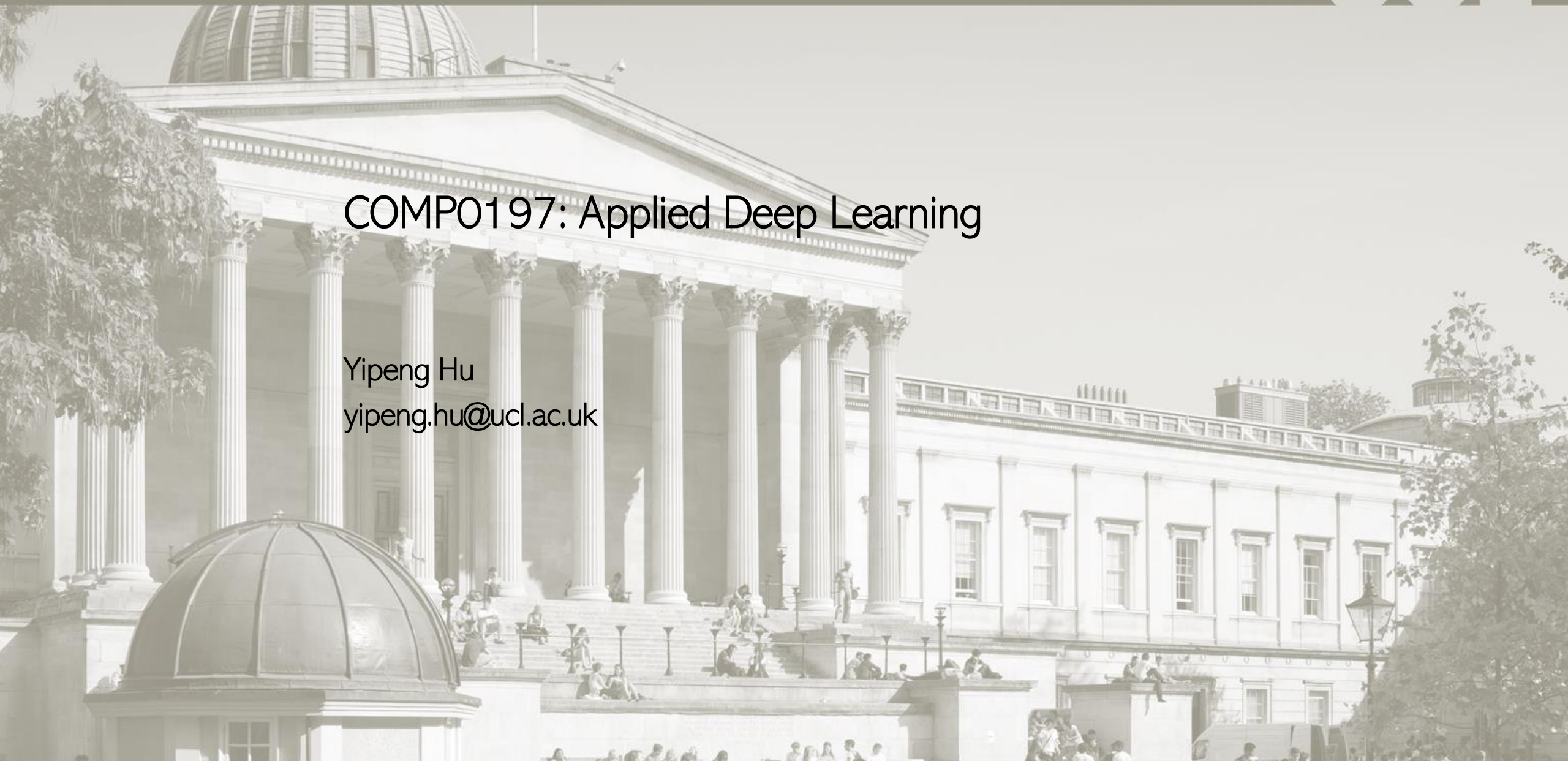


COMP0197: Applied Deep Learning

Yipeng Hu
yipeng.hu@ucl.ac.uk

Evaluation

A typical workflow to develop and evaluate a deep neural network

1. Formulate a machine learning problem

   e.g. classification/regression, input, output, model, training loss, evaluation metrics

2. Plan data

   Data collection, partitioning, collect more data

3. Developing models

   Training, tuning, regularising

4. Evaluation and validation

   Test, external validation

5. Deployment and monitoring

   Application-specific

Evaluation

- Goals

  Performance, overfitting and data leakage,

- Data

  Planning, cross-validation, collection, data cleaning, data bias, active learning

- Models

  Baseline models, hyperparameter tuning, autoML

- Applications

  Deployment considerations, computer vision, natural language processing, medical imaging, robotics

Evaluation | Goals

Performance

Code: maintainability, expandability, readability, tested, documented …

Development time and resource : developing cycles, model training time, GPU requirement, data requirement

Inference performance: speed, inference hardware requirement, *accuracy / generalisation*

Application: useful?

Performance metrics

Loss on training data

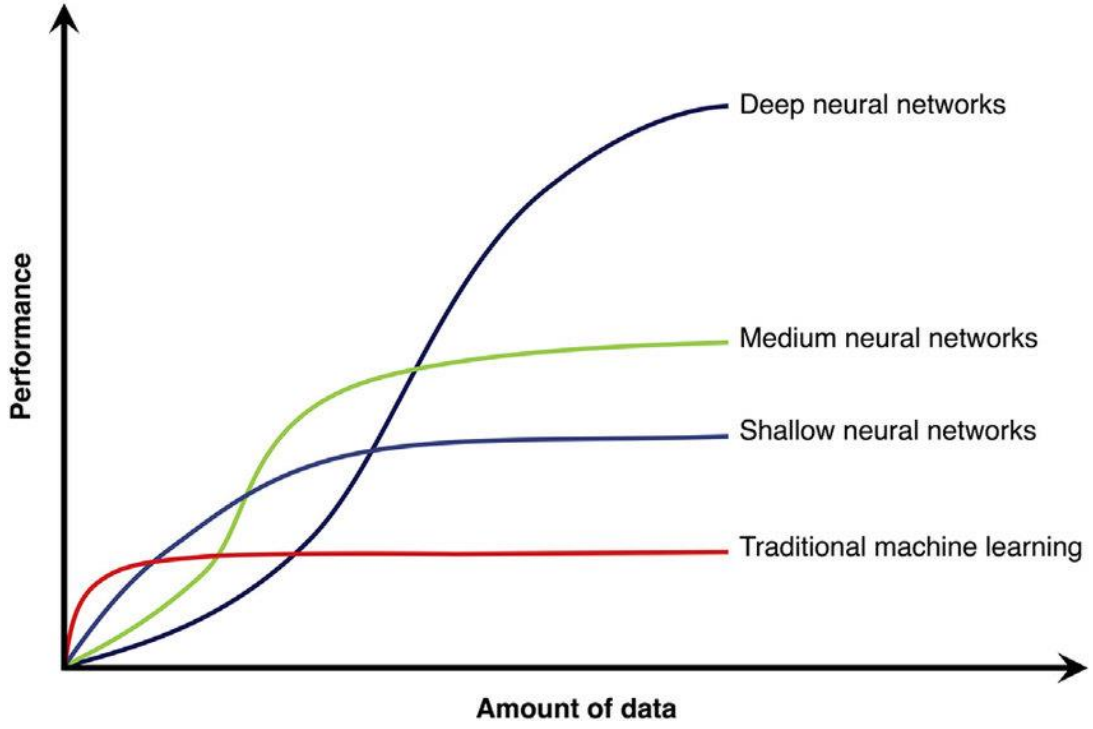
Loss on unseen data

Metrics on unseen data

Metrics on holdout data

Metrics on “out-of-distribution” data

Metrics on downstream tasks



Examples

Person recognition

Loss on training data: MSE, IoU, cross-entropy, IoU

Loss on unseen data: MSE, IoU, cross-entropy, IoU

Metrics on unseen data: recall, precision, average precision

Metrics on holdout data: accuracy, recall, precision

Metrics on “out-of-distribution” data: accuracy + financial consideration

Metrics on downstream tasks: financial consideration

Medical image diagnosis

Loss on training data: MSE, IoU, cross-entropy, Dice

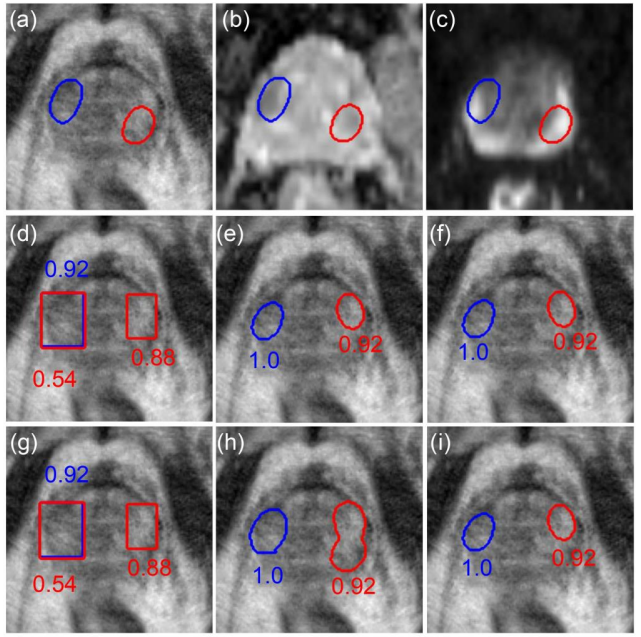
Loss on unseen data: MSE, IoU, cross-entropy, Dice

Metrics on unseen data: specificity-controlled sensitivity, Dice

Metrics on holdout data: specificity-controlled sensitivity, Dice

Metrics on “out-of-distribution” data: accuracy, specificity-controlled sensitivity, patient-cohort-defined

Metrics on downstream tasks: outcome

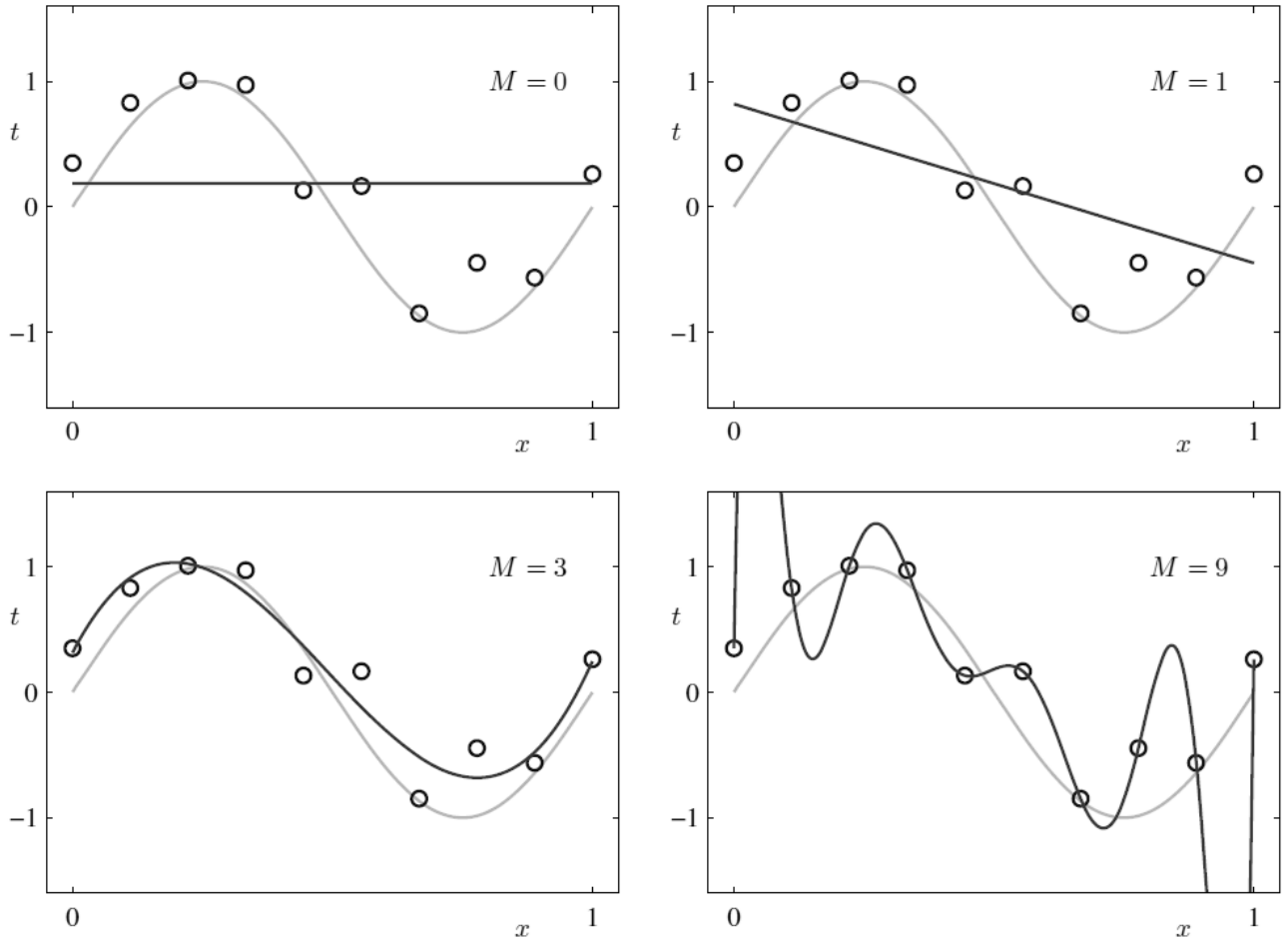


Overfitting and underfitting

- Generalisation to unseen data
- Underfitting

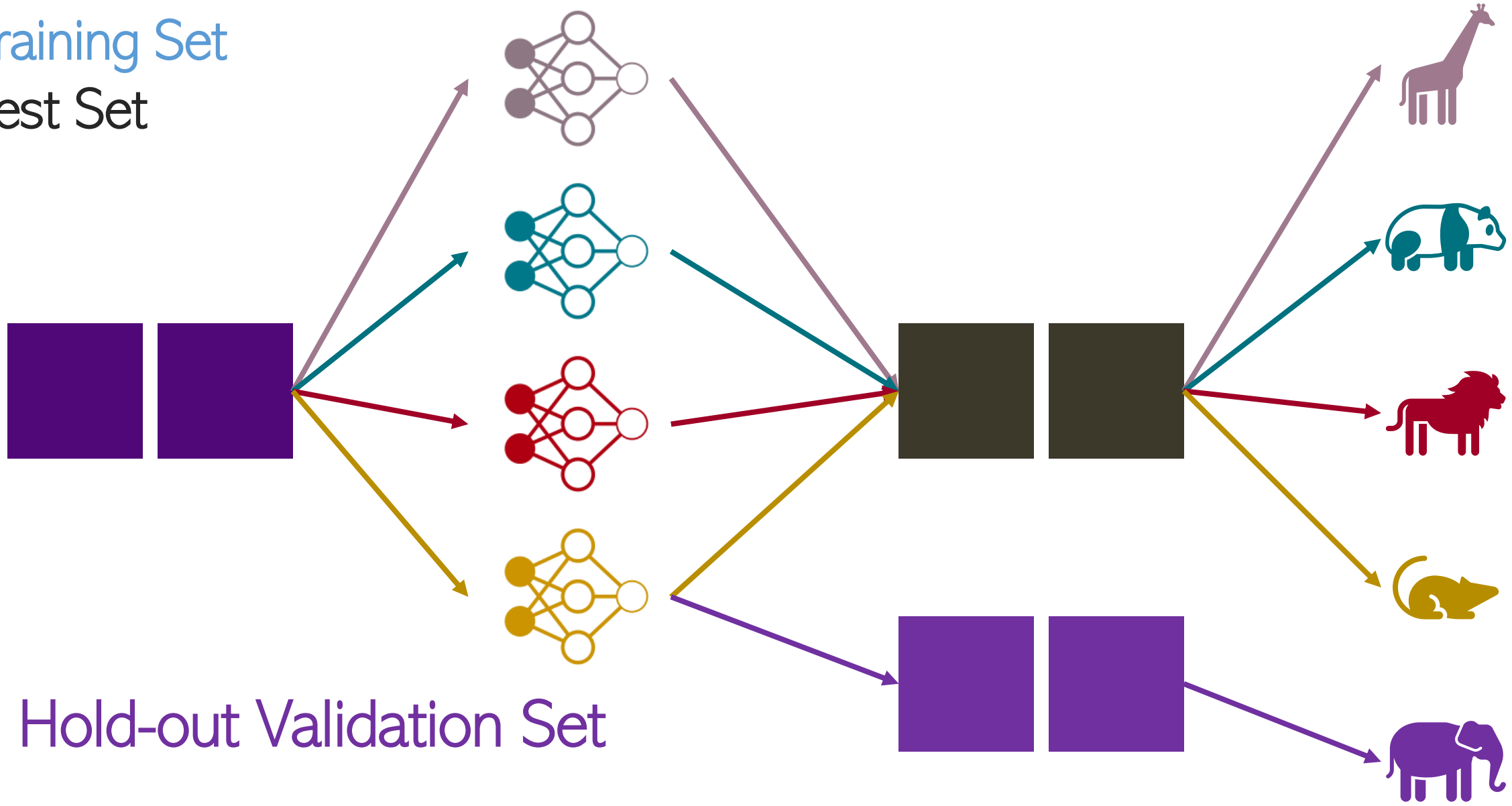
    e.g. better optimisation, larger model
- Overfitting

    e.g. regularisation, model data



Training Set

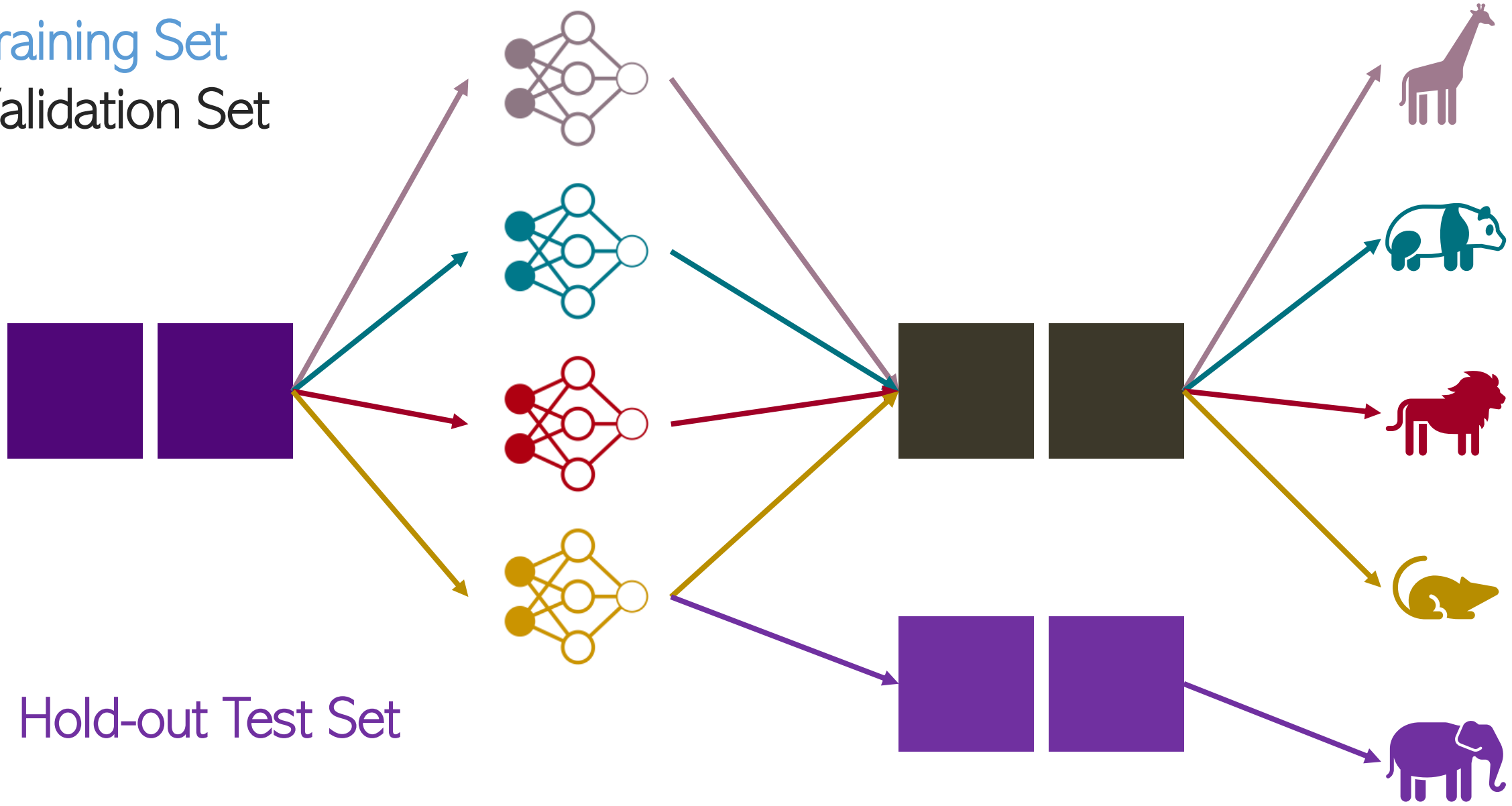
Test Set



Hold-out Validation Set

Training Set

Validation Set



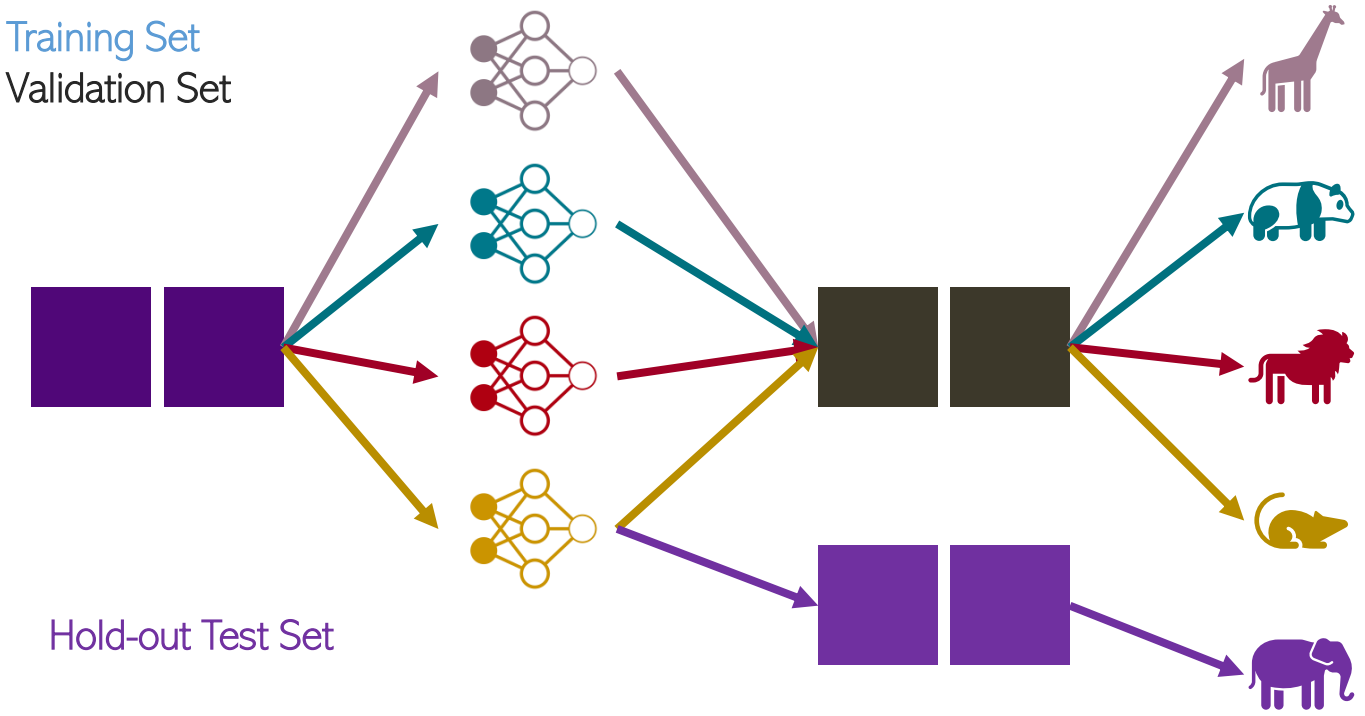
Hold-out Test Set

Data leakage, aka information leak

- Different data sets
  - Development set = training + validation set
  - Holdout set
  - "External" validation/test set
  - "Real-world" performance

    ...

- Evaluation is an estimation problem
  - Goal is to estimate the model performance
  - Avoid "meta-overfitting" as much as possible
  - Test set distribution and target distribution



Evaluation | Data

Planning

The more the better!

Planning

Development set

Training data -> for obtain adequate models

ImageNet: 150 GB, ImageNet holds 1 million (bounding-box) labelled images for training and 50k images for validation, organised in 1k categories.

Unet: The data set is provided by the EM segmentation with a set of 30 images (512x512 pixels) …

Validation set -> for estimate performance on training-unseen data

This is how refine/optimise models

e.g. hyperparameter tuning, architecture selecting

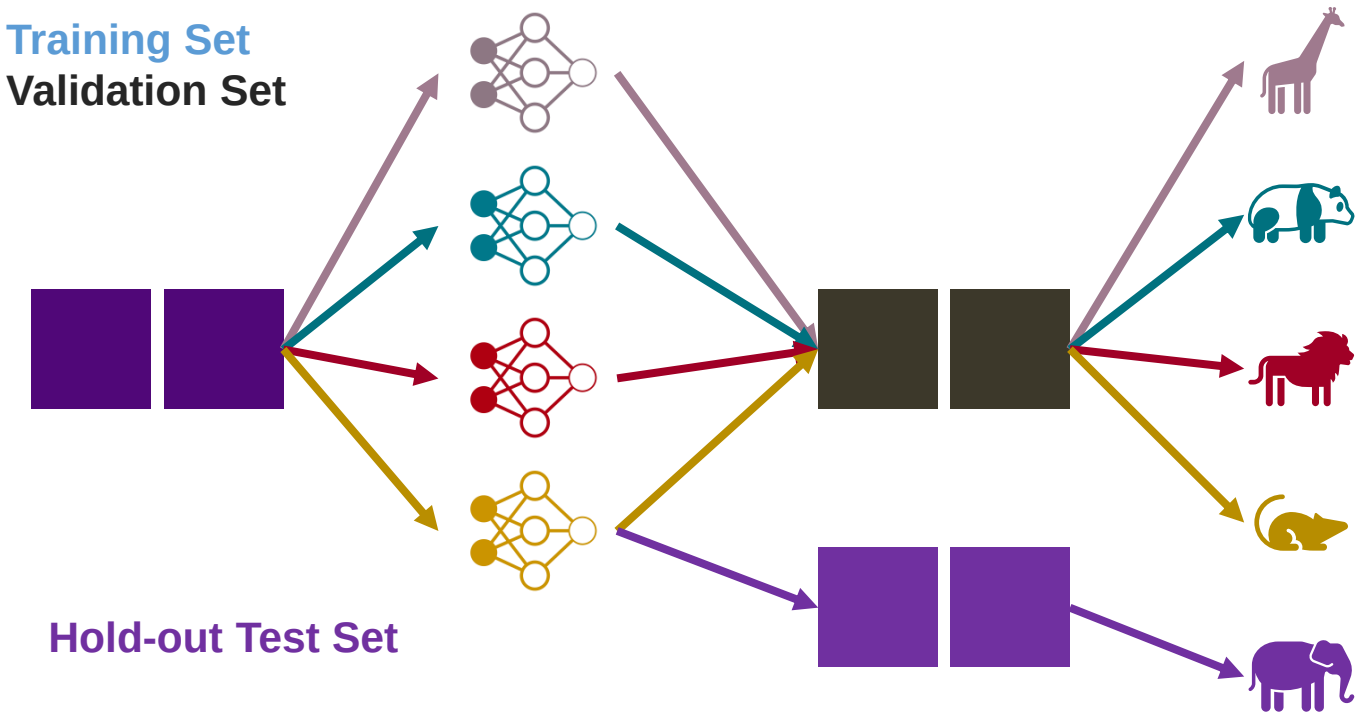
Bootstrapping?

Cross-validation

- Underestimate variance

Random splits

- Better estimate vs. computation



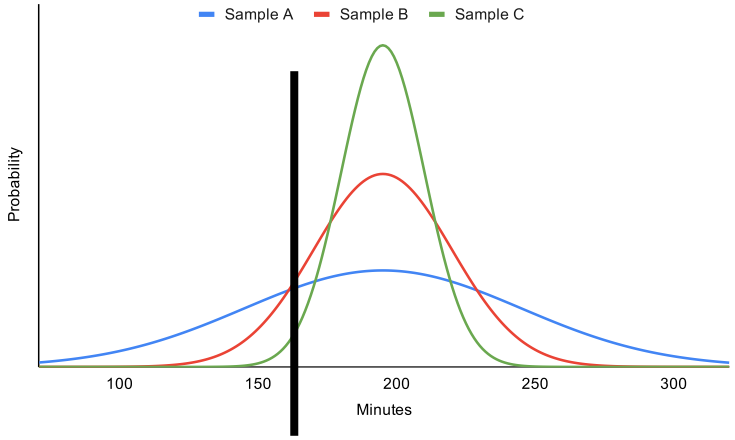
Planning

Test set

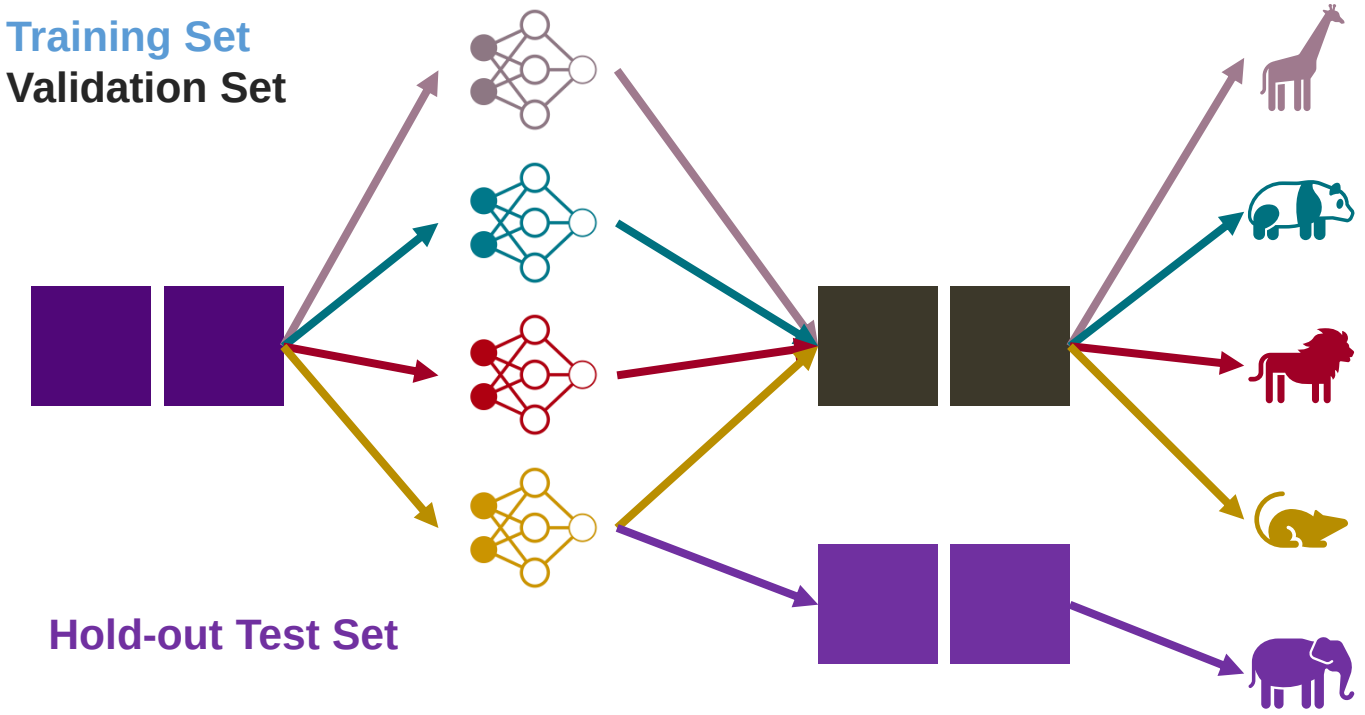
Statistical significance, p-values,

Variance vs. required data size

Power analysis



Why deep learning papers do not report variance or p-values?



Collection

The initial data size

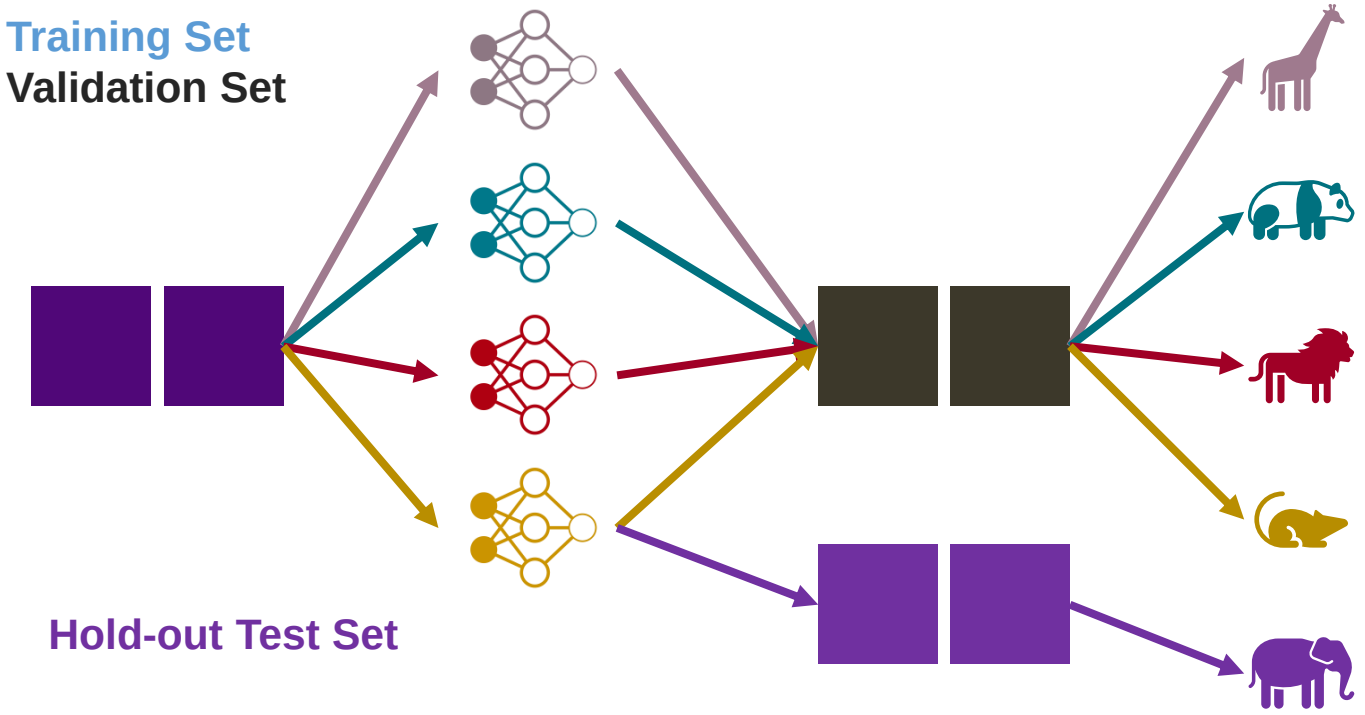
When to collect more data

During training: high training loss (always can overfit?)

During development: lack of performance on validation set

During evaluation: lack of performance on test set

- Cost
- Collect noisy/raw data
- Data cleaning
- Data labelling
- Unlabelled data, e.g. semi-supervised learning
- Methodology development, e.g. better regulariser



Cleaning

Examples

Format

Data quality, noise, error, missing data

Labels

Example:

ImageNet: starting 2006, crowdsourced, first release 2009, still errors frequently reported today

~110 pairs of MR and ultrasound images from a trial, three senior imaging researchers several months

Bias

Sample bias (randomness)

Exclusion bias (design)

Measurement bias (tools, observer)

Social bias (e.g. gender, racial and their associated)

Data governess

Rights and time and geographic limitation, legitimacy, privacy, sensitivity

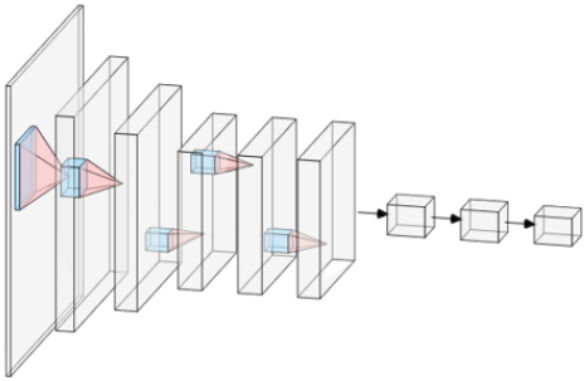
Store, access and destruction

Trustworthiness

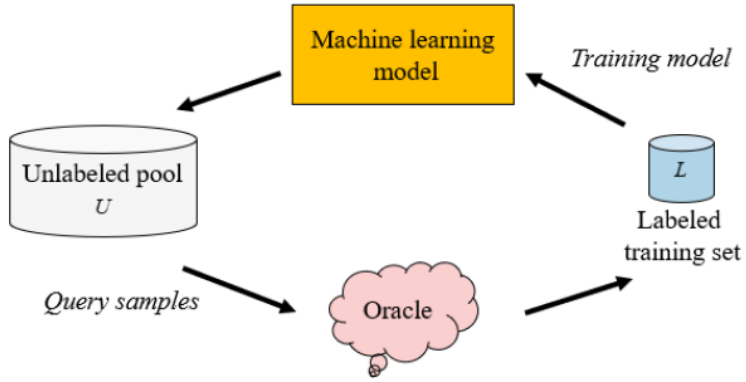
Performance: reliability, safety and usability

Design: fairness, security (including privacy-preservation) and transparency (including explainability and interpretability)

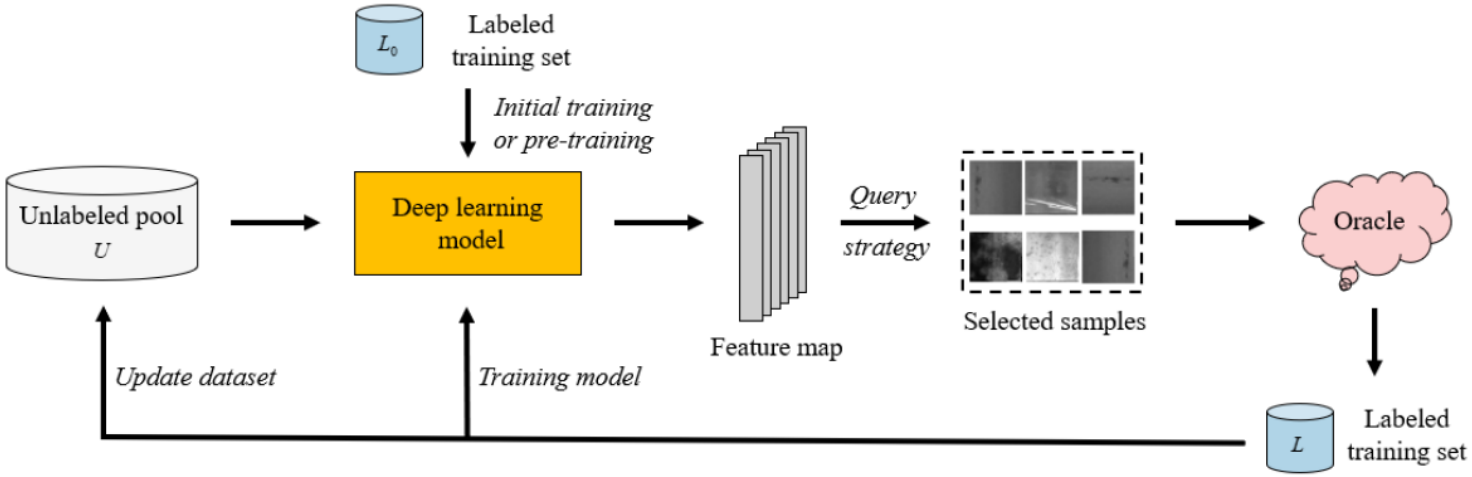
Active learning



(a) Structure diagram of convolutional neural network.



(b) The pool-based active learning cycle.



Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Chen, X. and Wang, X., 2020. A survey of deep active learning. *arXiv preprint arXiv:2009.00236*.

Evaluation | Models

Baseline models

Image classification: VGG, resnet, EfficientNet…

Sequence data: RNN with LSTM and attention, transformer

More specialised:

Object detection: Mask-RCNN

Segmentation: Unet

Generative models: GANs, diffusion

Training: SGD with momentum, Adam, minibatch depending on the available hardware, MSE/CE losses

“No harm tricks”: some form of data augmentation, normalisation, batchnorm, small weight decay

Cross-validation for learning rate, model size, and other hyperparameters*

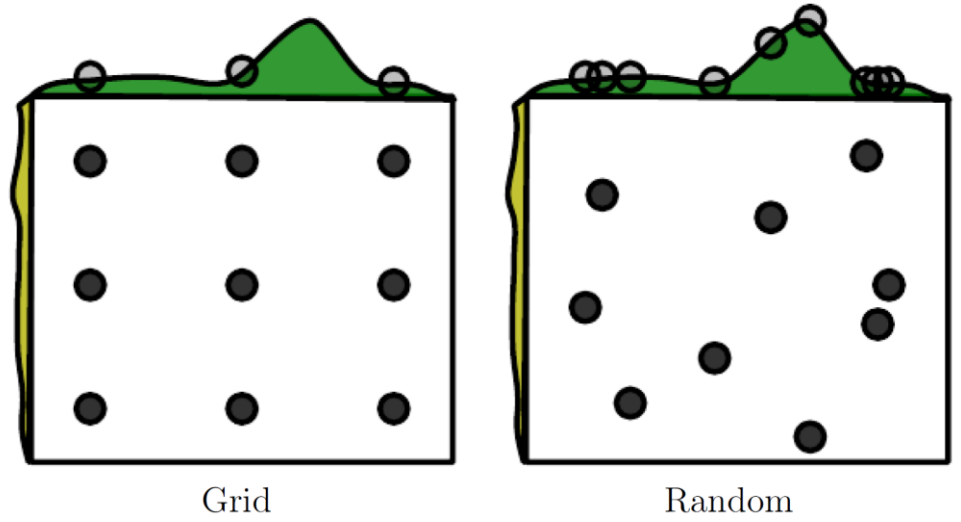
Data: labelled, cleaned and partitioned

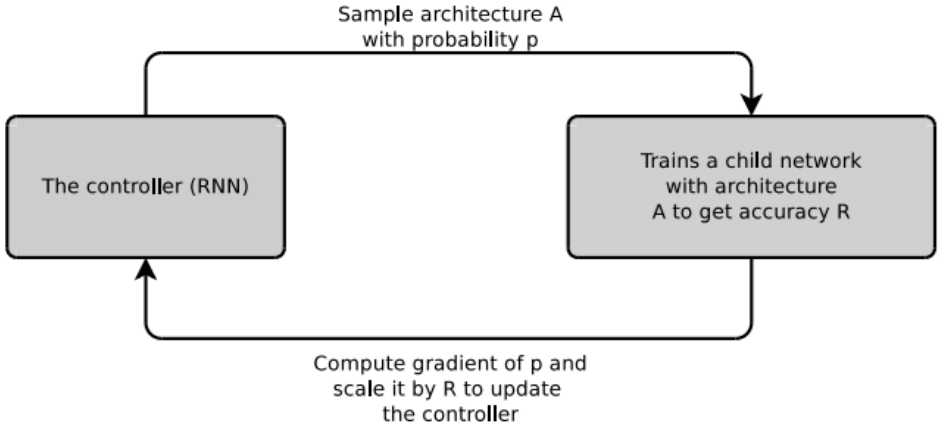
Unsupervised learning?

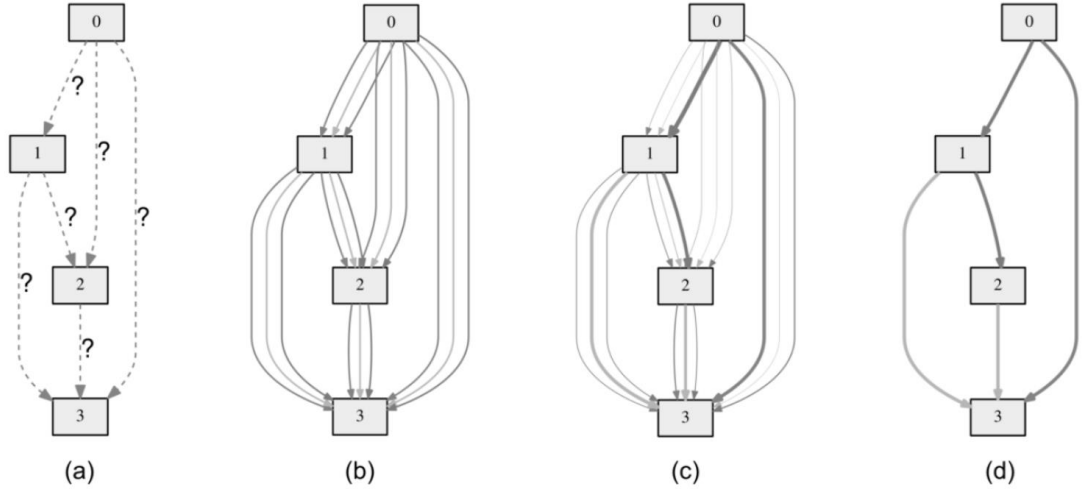
Deep reinforcement learning?

Hyperparameter tuning

- Definition of hyperparameters
- Search: grid search vs. random search
- Non-gradient optimisers: GA, RL, meta-learning
- Differentiable hyperparameters



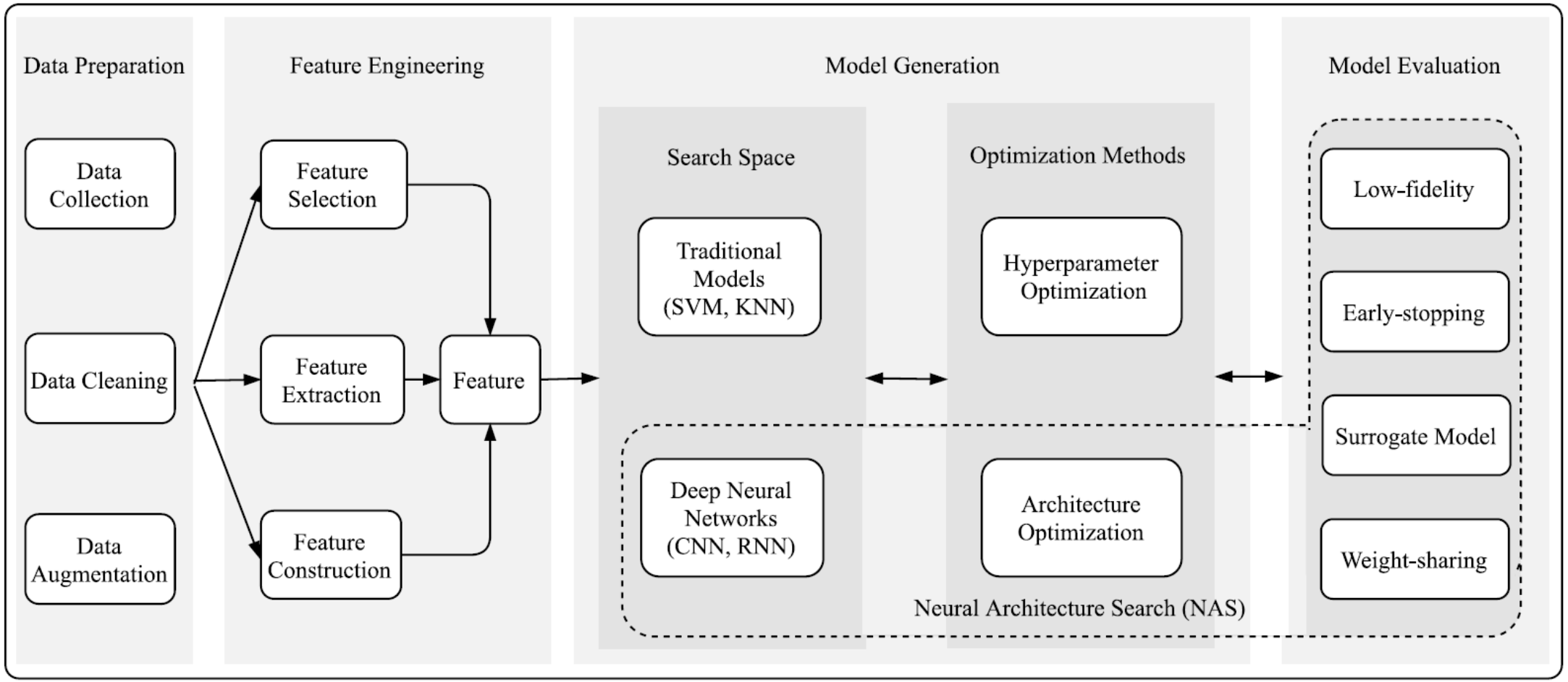




AutoML

- Hyperparameter tuning
- Model selection

.....



He, X., Zhao, K. and Chu, X., 2021. AutoML: A Survey of the State-of-the-Art. *Knowledge-Based Systems*, *212*, p.106622.

Evaluation | Applications

Deployment considerations

- Machine learning engineering
- System testing
- Continuous learning
- Local validation
- Hardware

Computer vision

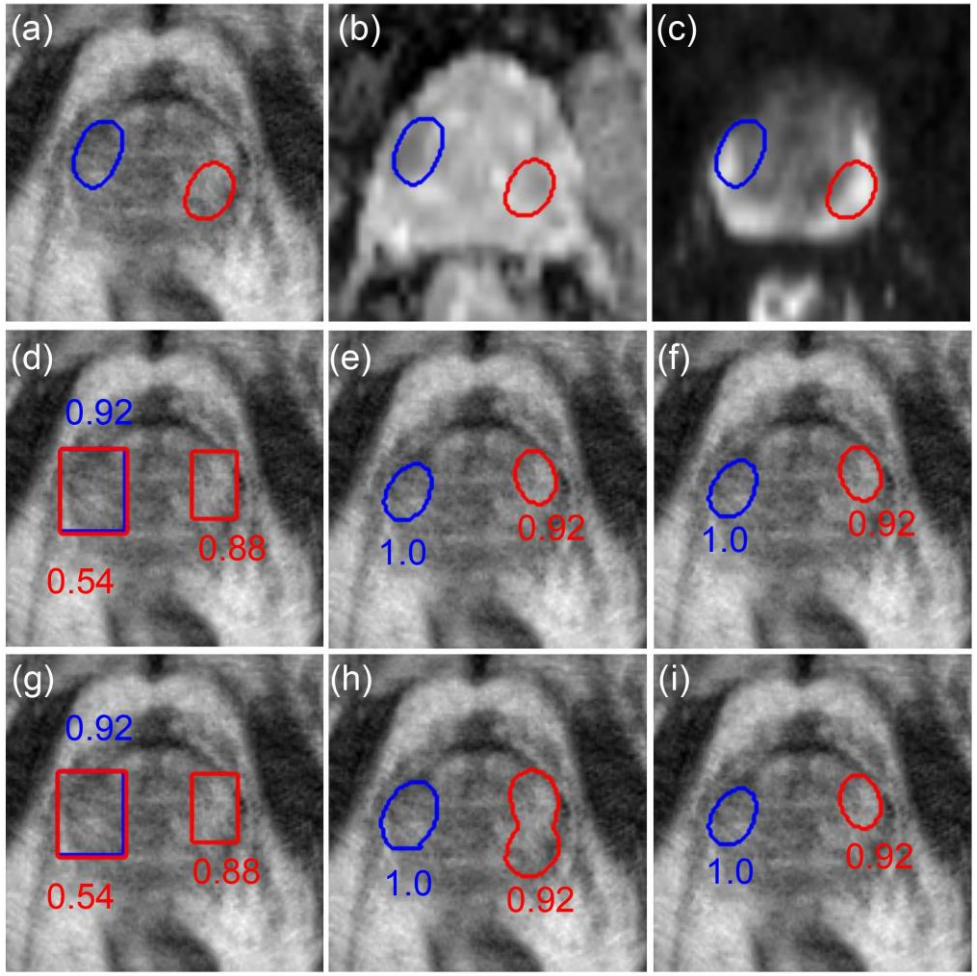
Body temperature measuring

Natural language processing

Voice labelling for sports statistics

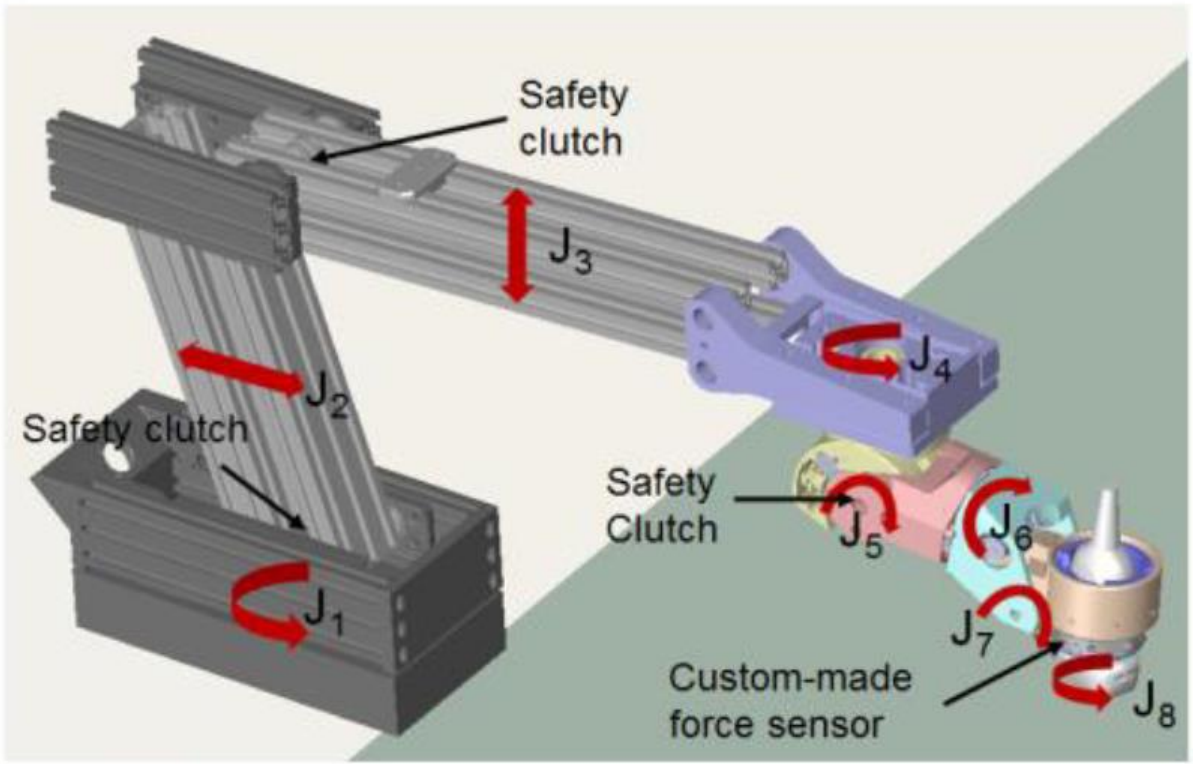
Medical imaging

CAD for prostate cancer on mpMR imaging

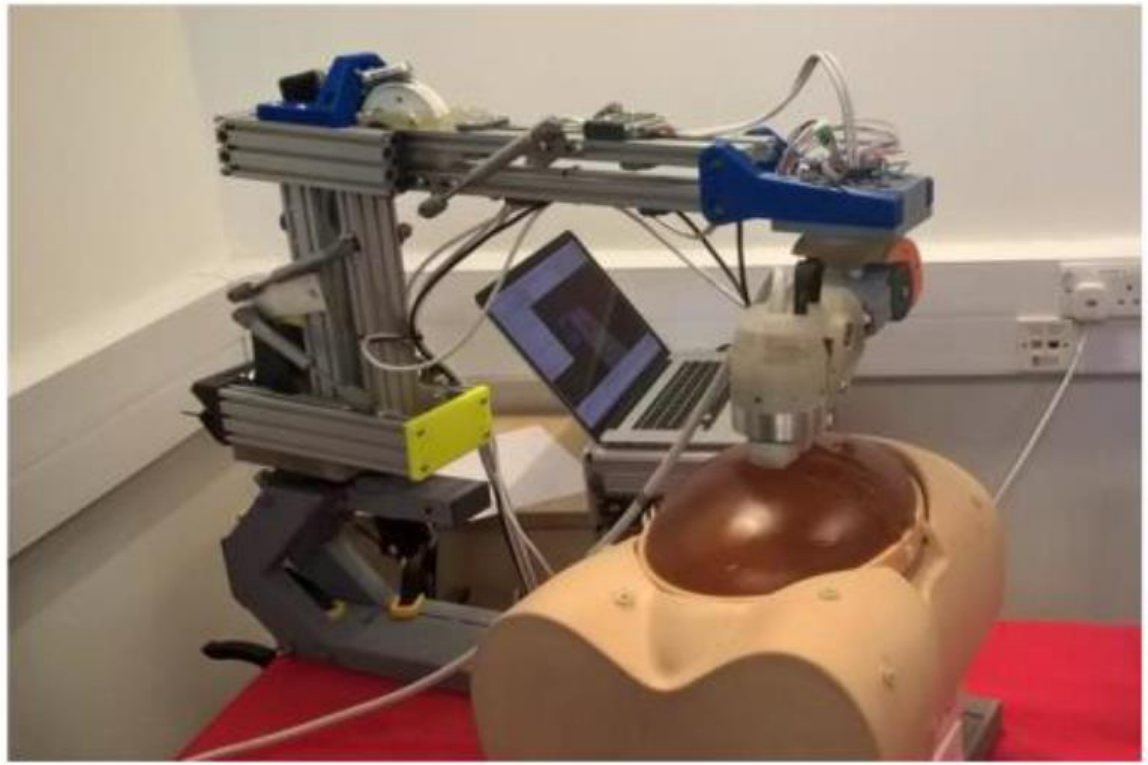


Robotics

Assisted ultrasound for fetal imaging



(a)



(b)

Wang, S., Housden, J., Noh, Y., Singh, D., Singh, A., Skelton, E., Matthew, J., Tan, C., Back, J., Lindenroth, L. and Gomez, A., 2019, July. Robotic-assisted ultrasound for fetal imaging: evolution from single-arm to dual-arm system. In *Annual Conference Towards Autonomous Robotic Systems* (pp. 27-38). Springer, Cham.

Evaluation

- Goals

  Performance, overfitting and data leakage,

- Data

  Planning, cross-validation, collection, data cleaning, data bias, active learning

- Models

  Baseline models, hyperparameter tuning, autoML

- Applications

  Deployment considerations, computer vision, natural language processing, medical imaging, robotics

"Classical" machine learning methods

- Linear regression, logistic regression
- Support vector machine
- Random forest

Tools

- Deep neural networks, TensorFlow, PyTorch
- Computer vision and natural language processing
- Supervised learning with quality labeled (data)

Research

- Advanced regularisation
- Generative modelling
- Unsupervised learning
- Monte Carlo and approximate inference
- Model interpretation
- Meta-learning
- Reinforcement learning

.....