Lab 5: Transfer Learning

11210IPT 553000

Deep Learning in Biomedical Optical Imaging

2023/10/23

Outlines

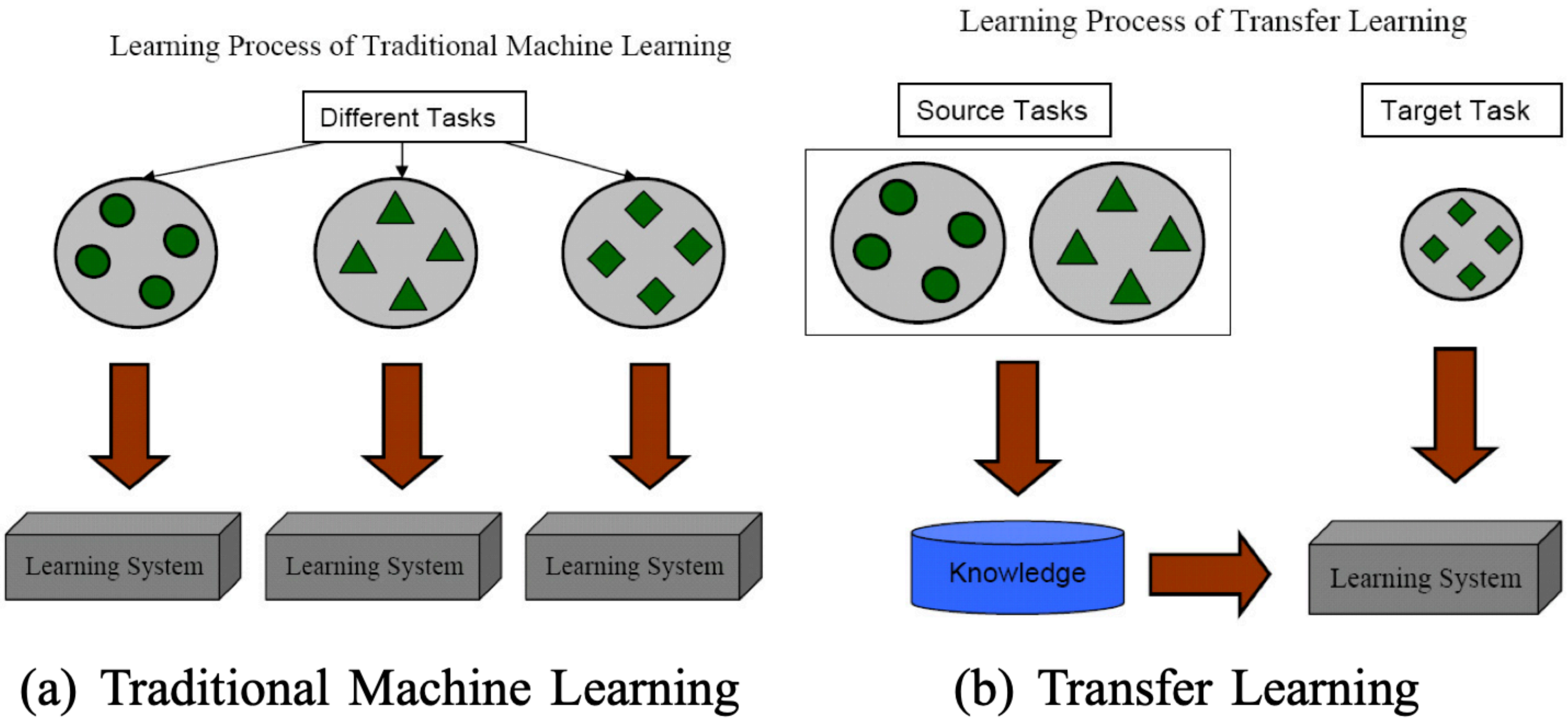
- Motivation & Benefits
- Methods
- Related Paper
- Homework 4

Motivation & Benefits

- There are many powerful trained models from precious works.
  - Could we reuse these trained knowledge in the models?
- If it's possible, we……..
  - Can save our time on training!
    - Buy less GPUs……
  - Can use smaller dataset!
  - Can have a good initial weights!

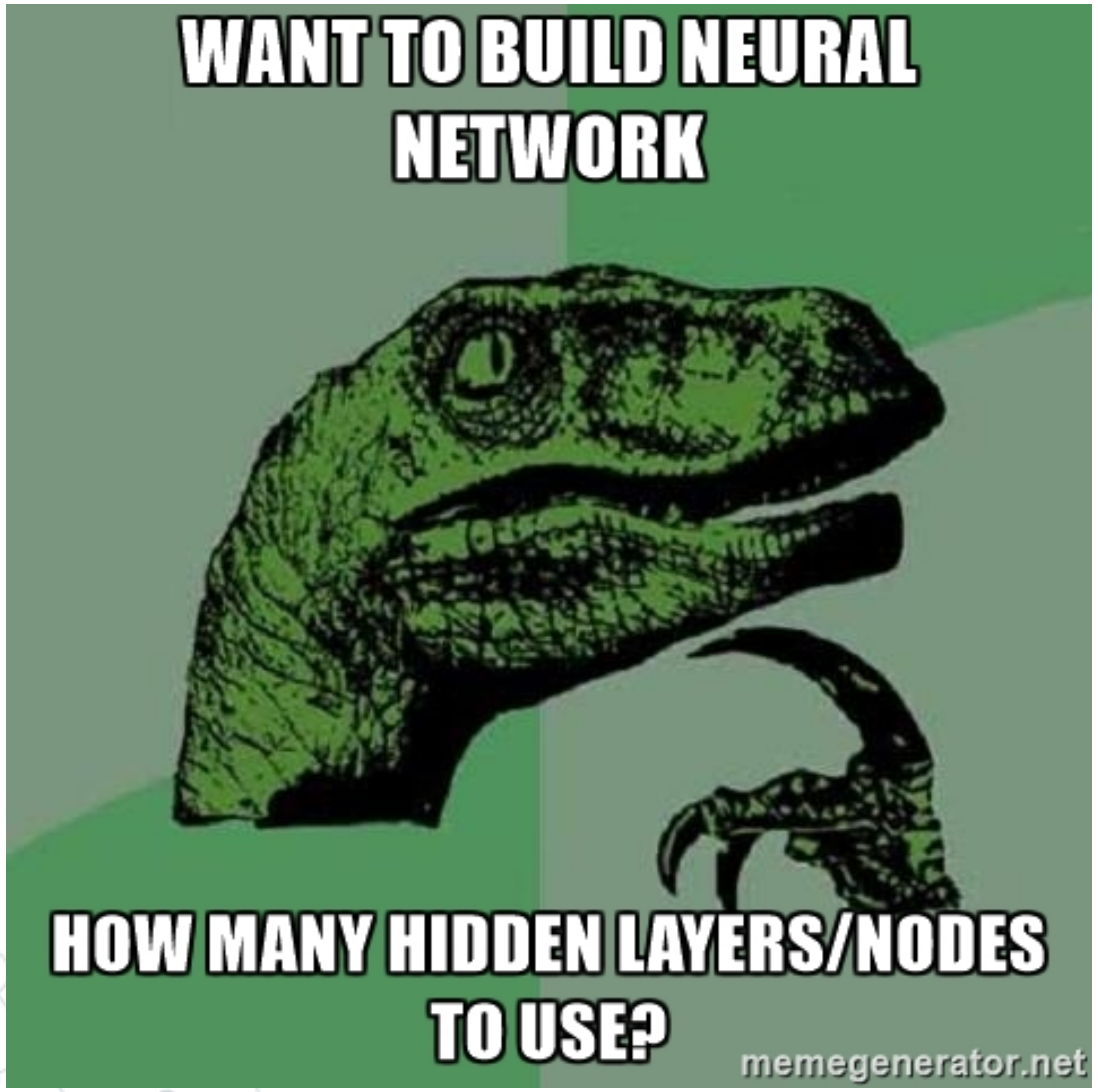
https://cs231n.github.io/transfer-learning/

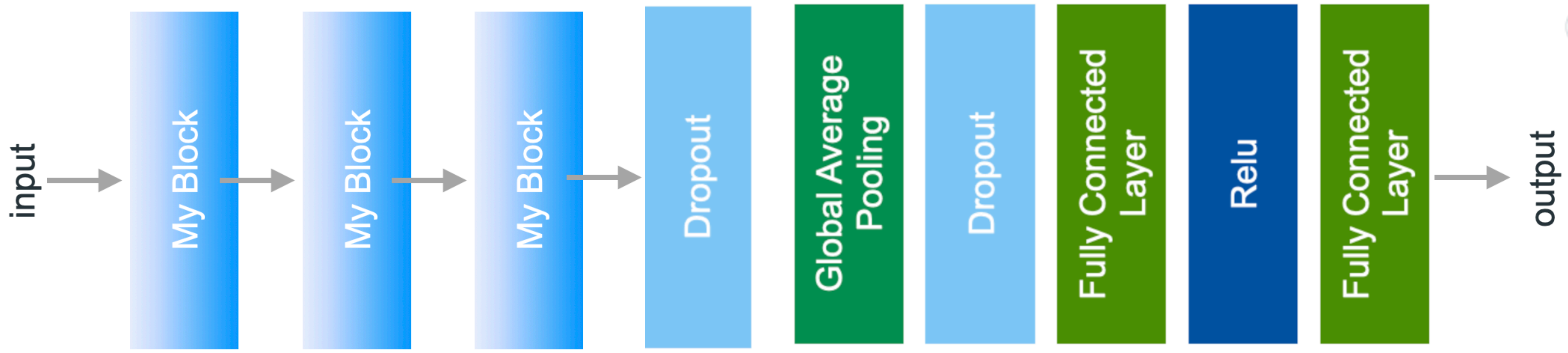
Motivation & Benefits



Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." *IEEE Transactions on knowledge and data engineering* 22.10 (2009): 1345-1359.

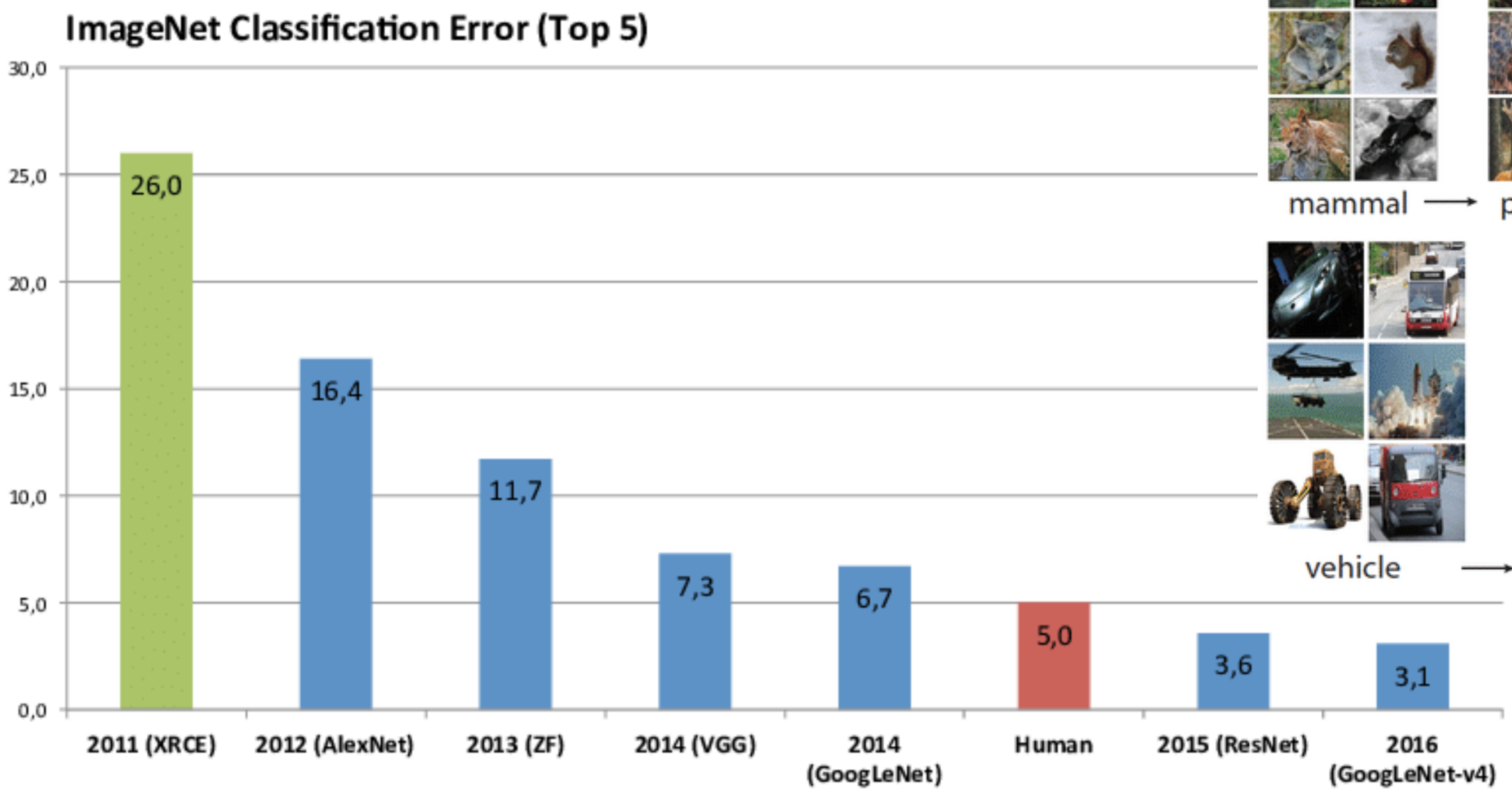
Motivation & Benefits



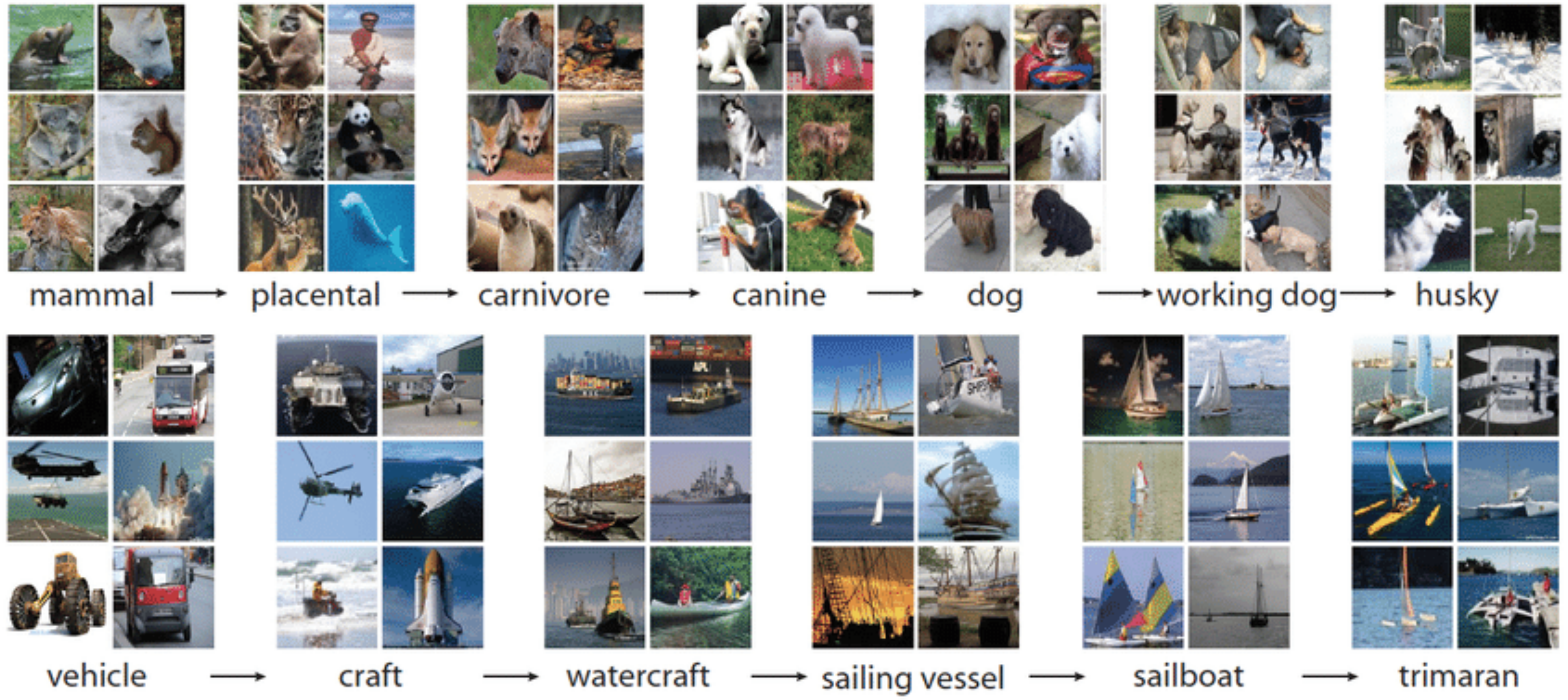


Methods

- ImageNet
  - Total: 150 GB (1281167 images in 1000 categories)

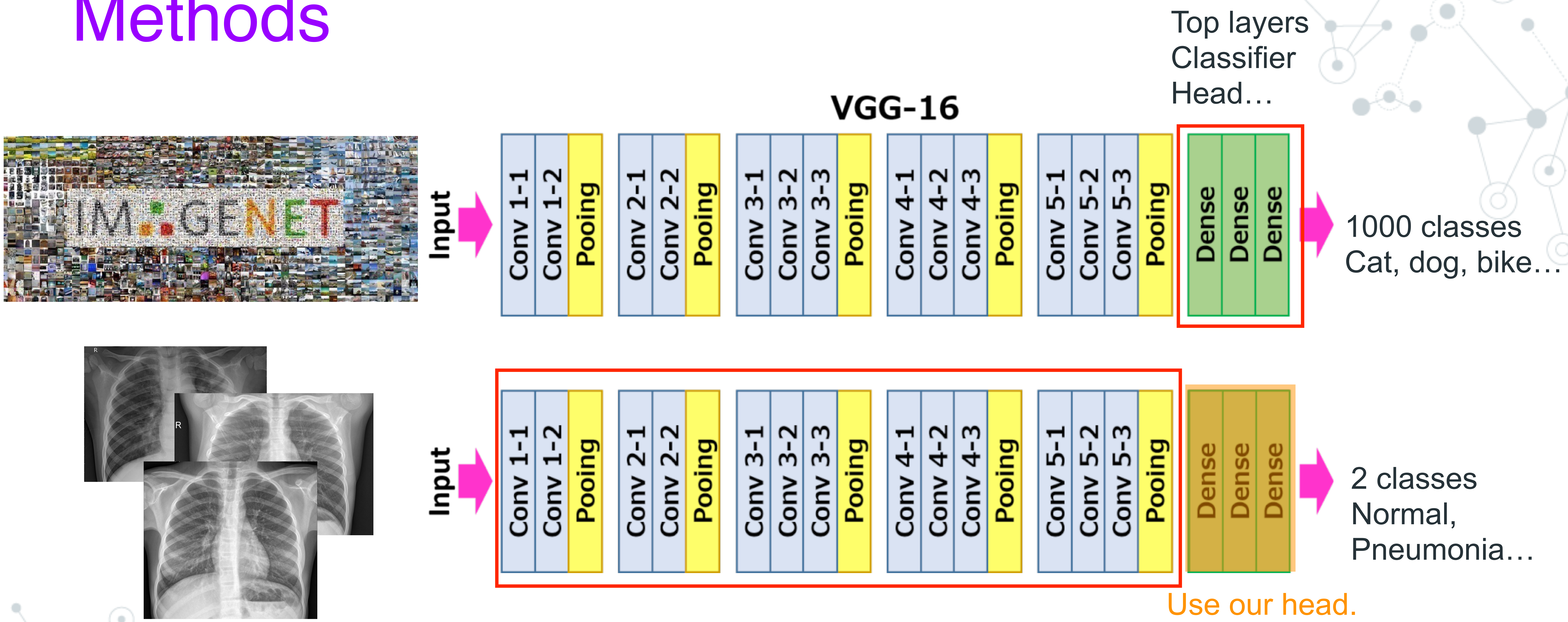


AlexNet triggers a wave of better solutions to the ImageNet classification problem. Source: von Zitzewitz 2017, fig. 11.*



Source: https://devopedia.org/imagenet

Methods

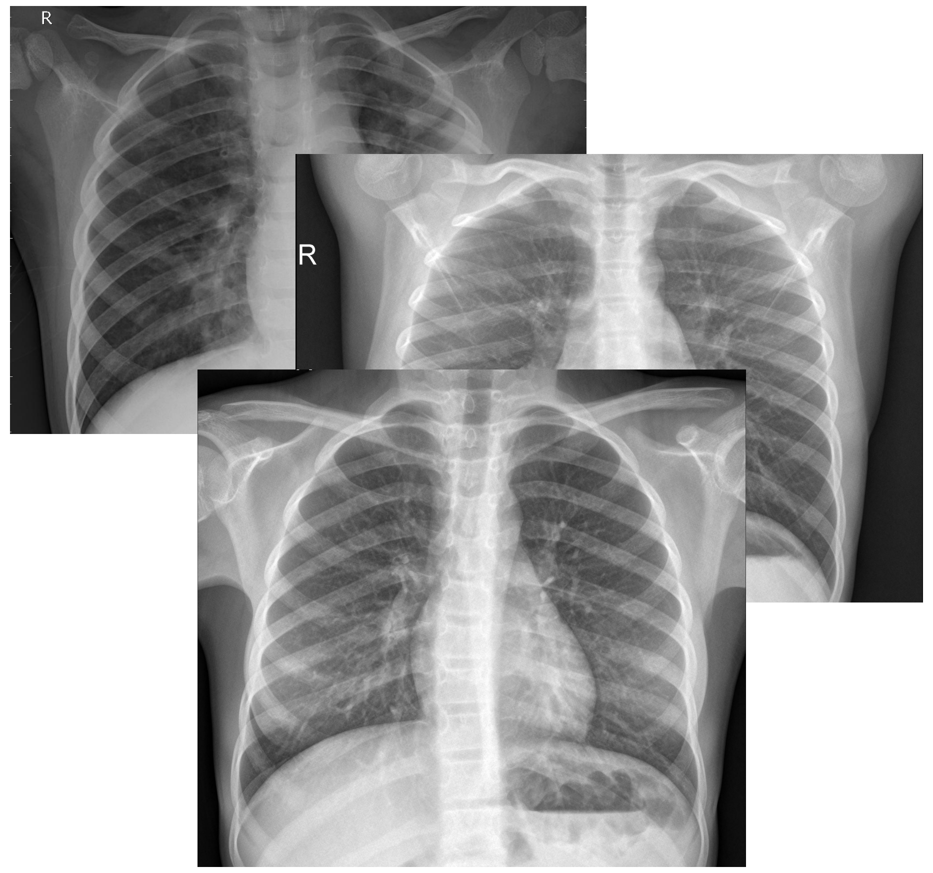


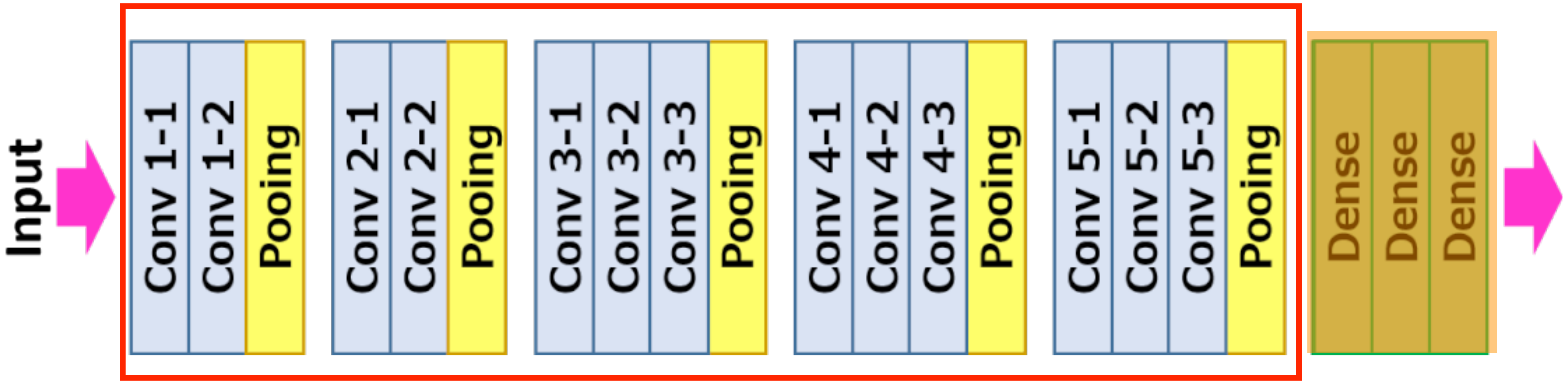
Use pre-trained model as a **feature extractor**.

Methods

Step 1: **Freeze** pre-trained model, only train our head!

Only update these layers.



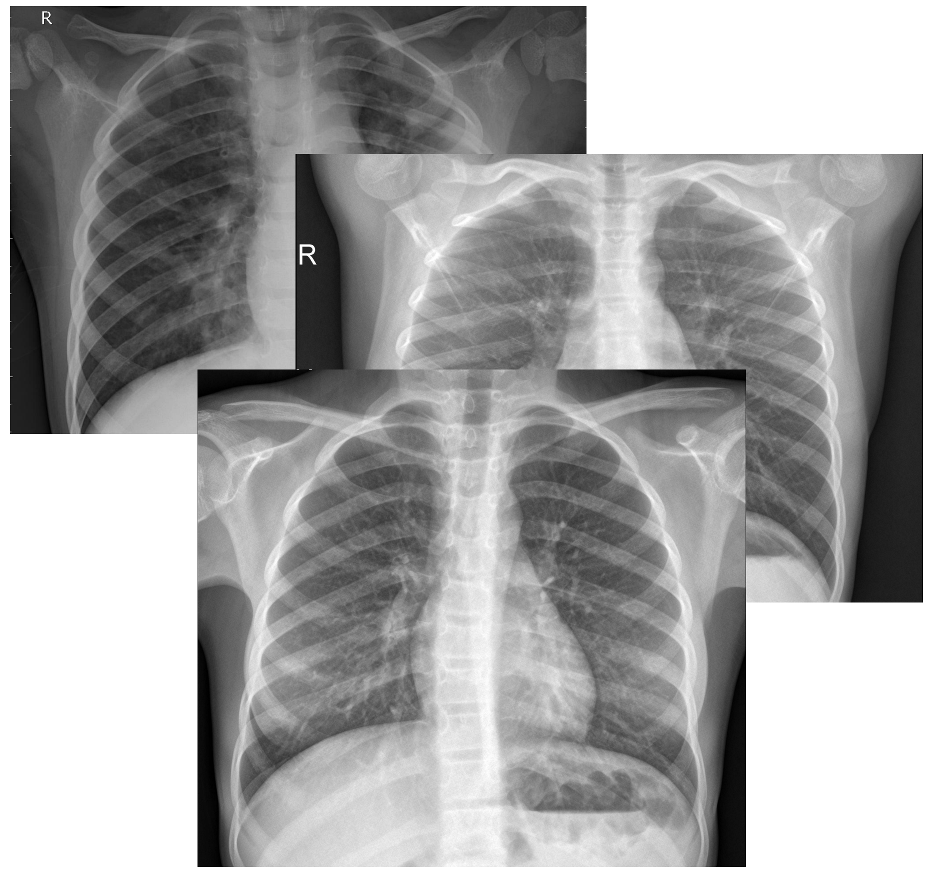


2 classes
Normal,
Pneumonia…

Freeze here!

Methods

Step 2: Fine-tune the weights in all layers.



2 classes
Normal,
Pneumonia…

Docs > Models and pre-trained weights

MODELS AND PRE-TRAINED WEIGHTS

The `torchvision.models` subpackage contains definitions of models for addressing different tasks, including: image classification, pixelwise semantic segmentation, object detection, instance segmentation, person keypoint detection, video classification, and optical flow.

General information on pre-trained weights

TorchVision offers pre-trained weights for every provided architecture, using the PyTorch `torch.hub`. Instancing a pre-trained model will download its weights to a cache directory. This directory can be set using the *TORCH_HOME* environment variable. See `torch.hub.load_state_dict_from_url()` for details.

NOTE

The pre-trained models provided in this library may have their own licenses or terms and conditions derived from the dataset used for training. It is your responsibility to determine whether you have permission to use the models for your use case.

NOTE

Backward compatibility is guaranteed for loading a serialized `state_dict` to the model created using old PyTorch version. On the contrary, loading entire saved models or serialized `ScriptModules` (serialized using older versions of PyTorch) may not preserve the historic behaviour. Refer to the following documentation

https://pytorch.org/vision/stable/models.html


WHAT ARE MY STRENGTHS?
NONE. LET'S TALK ABOUT YOUR WEAKNESSES!
Researcher
Transfer Learning
STFUHERO.COM

Transfusion: Understanding Transfer Learning for Medical Imaging

- A performance evaluation on two large scale medical imaging tasks shows that surprisingly, **transfer offers little benefit** to performance, and simple, lightweight models can perform comparably to ImageNet architectures.
- Some of the differences from transfer learning are due to the **over-parametrization** of standard models rather than complex feature reuse.
  - The large, standard ImageNet models do not change significantly through the fine-tuning process.

Raghu, Maithra, et al. "Transfusion: Understanding transfer learning for medical imaging." *NeruIPS* (2019).

Transfusion: Understanding Transfer Learning for Medical Imaging

Model Architecture	Atelectasis	Cardiomegaly	Consolidation	Edema	Pleural Effusion
Resnet-50	79.52±0.31	75.23±0.35	85.49±1.32	88.34±1.17	88.70±0.13
Resnet-50 (trans)	79.76±0.47	74.93±1.41	84.42±0.65	88.89±1.66	88.07±1.23
CBR-LargeT	81.52±0.25	74.83±1.66	88.12±0.25	87.97±1.40	88.37±0.01
CBR-LargeT (trans)	80.89±1.68	76.84±0.87	86.15±0.71	89.03±0.74	88.44±0.84
CBR-LargeW	79.79±0.79	74.63±0.69	86.71±1.45	84.80±0.77	86.53±0.54
CBR-LargeW (trans)	80.70±0.31	77.23±0.84	86.87±0.33	89.57±0.34	87.29±0.69
CBR-Small	80.43±0.72	74.36±1.06	88.07±0.60	86.20±1.35	86.14±1.78
CBR-Small (trans)	80.18±0.85	75.24±1.43	86.48±1.13	89.09±1.04	87.88±1.01
CBR-Tiny	80.81±0.55	75.17±0.73	85.31±0.82	84.87±1.13	85.56±0.89
CBR-Tiny (trans)	80.02±1.06	75.74±0.71	84.28±0.82	89.81±1.08	87.69±0.75

Table 2: **Transfer learning provides mixed performance gains on chest x-rays.** Performances (AUC%) of diagnosing different pathologies on the CHEXPERT dataset. Again we see that transfer learning does not help significantly, and much smaller models performing comparably.

Raghu, Maithra, et al. "Transfusion: Understanding transfer learning for medical imaging." *NeruIPS* (2019).

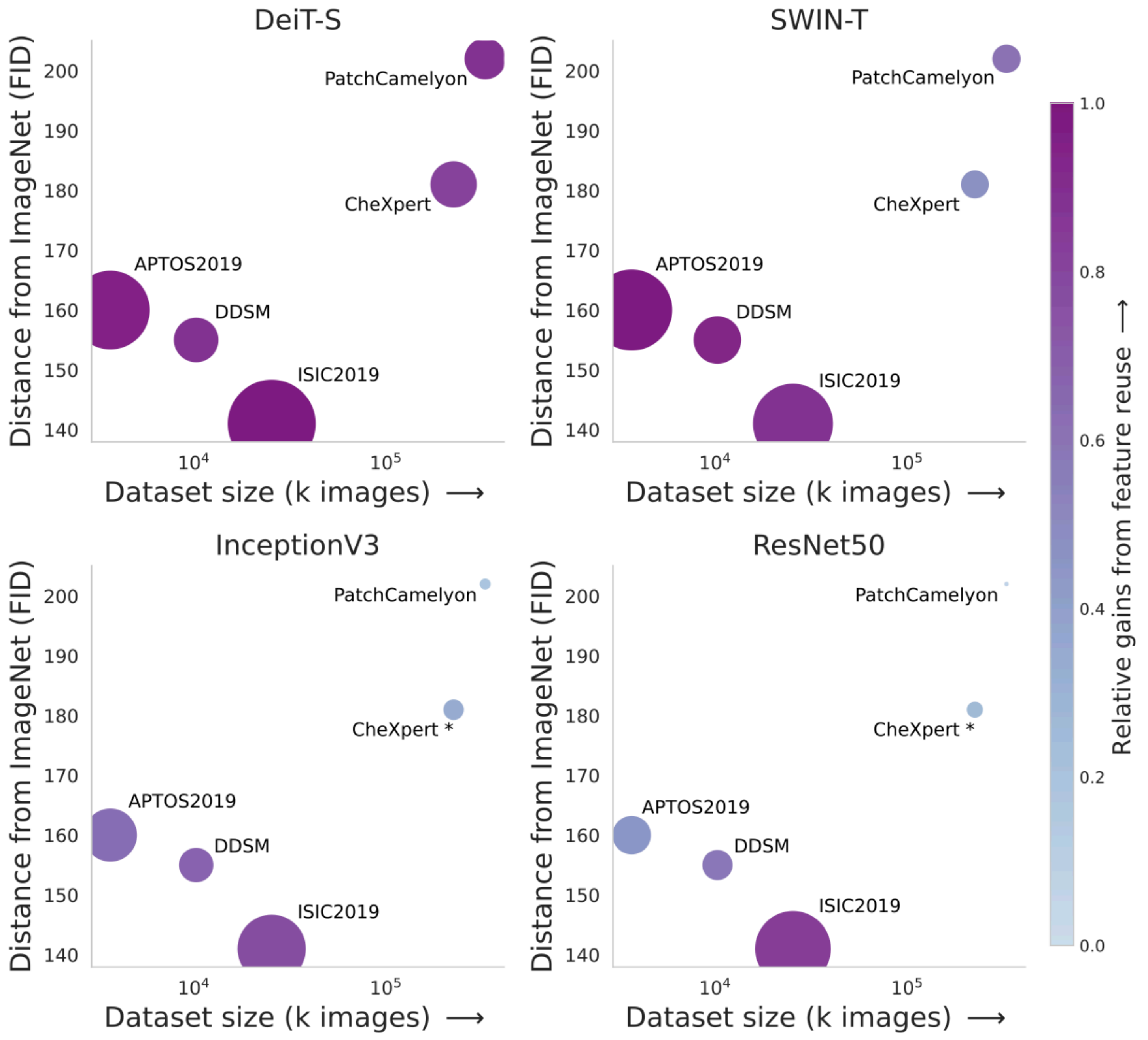

WHAT ARE MY WEAKNESSES?
NONE. LET'S TALK ABOUT YOUR STRENGTHS!
Researcher
Transfer Learning
STFUHERO.COM

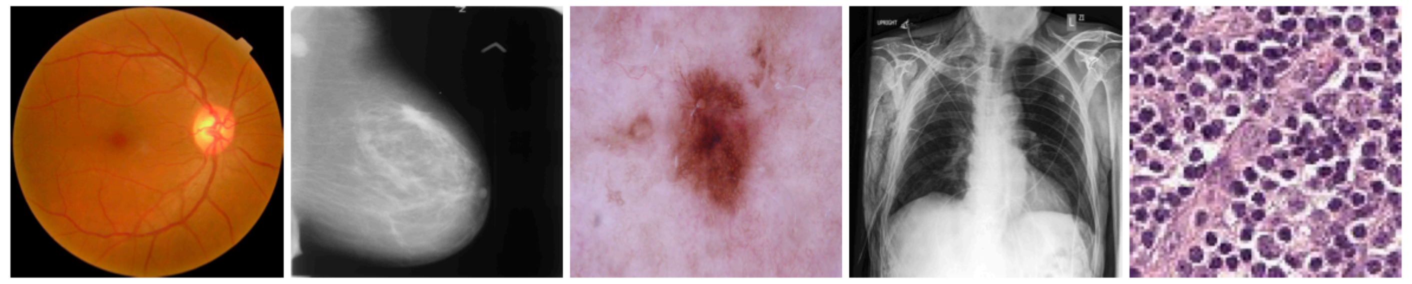
What Makes Transfer Learning Work For Medical Images?

- We study the effectiveness of transfer learning as a function of dataset size, the distance from the source domain, the model's capacity, and the model's inductive bias.
- Find a strong correlation between the observed benefits from transfer learning and evidence for ***feature reuse***.
- The benefits from transfer learning increase with
  - (1) reduced data size
  - (2) smaller distances between the source and target domain
  - (3) less inductive bias.

Matsoukas, Christos, et al. "What Makes Transfer Learning Work For Medical Images: Feature Reuse & Other Factors." *CVPR.* 2022.

What Makes Transfer Learning Work For Medical Images?





APTOS 2019 | CBIS-DDSM | ISIC 2019 | CheXpert | PatchCamelyon

FID: Fre’chet Inception Distance

Model (# parameters)	**Init**	**APTOS2019**, κ ↑ n = 3,662 FID = 160	**DDSM**, AUC ↑ n = 10,239 FID = 155	**ISIC2019**, Rec. ↑ n = 25,333 FID = 141	**CheXpert**, AUC ↑ n = 224,316 FID = 181	**Camelyon**, AUC ↑ n = 327,680 FID = 202
DeiT-S (22M)	RI	0.684 ± 0.017	0.907 ± 0.005	0.576 ± 0.013	0.740 ± 0.006	0.921 ± 0.002
	ST	0.721 ± 0.016	0.895 ± 0.005	0.607 ± 0.017	0.734 ± 0.002	0.916 ± 0.005
	WT	0.894 ± 0.017	0.949 ± 0.011	0.824 ± 0.008	0.792 ± 0.001	0.962 ± 0.003
SWIN-T (29M)	RI	0.689 ± 0.022	0.898 ± 0.005	0.597 ± 0.080	0.780 ± 0.001	0.936 ± 0.002
	ST	0.722 ± 0.017	0.900 ± 0.004	0.654 ± 0.008	0.785 ± 0.000	0.948 ± 0.013
	WT	0.906 ± 0.005	0.961 ± 0.007	0.833 ± 0.008	0.805 ± 0.000	0.968 ± 0.006
InceptionV3 (24M)	RI	0.835 ± 0.012	0.923 ± 0.003	0.668 ± 0.008	0.794 ± 0.001	0.956 ± 0.006
	ST	0.796 ± 0.014	0.907 ± 0.014	0.629 ± 0.013	0.787 ± 0.001	0.956 ± 0.003
	WT	0.873 ± 0.007	0.939 ± 0.010	0.758 ± 0.011	0.797 ± 0.000	0.958 ± 0.004
ResNet50 (25M)	RI	0.845 ± 0.022	0.919 ± 0.005	0.664 ± 0.016	0.796 ± 0.000	0.948 ± 0.008
	ST	0.848 ± 0.006	0.933 ± 0.006	0.635 ± 0.012	0.794 ± 0.001	0.959 ± 0.003
	WT	0.888 ± 0.004	0.957 ± 0.003	0.795 ± 0.011	0.800 ± 0.001	0.960 ± 0.006

Matsoukas, Christos, et al. "What Makes Transfer Learning Work For Medical Images: Feature Reuse & Other Factors." *CVPR.* 2022.

Homework 4

- **Deadline**: 23:59, 6th Nov. (GMT+8)
- We need to write a report to answer questions. Details are in hw4_description.pdf
- **Important**: Make sure your commit is timestamped before the deadline. Late submissions might not be graded or could incur a penalty. Only the GitHub link is required on NTHU EEclass.

CODING TIME!