Introduction to AI for Computer Vision Applications

June 12th 2024

Alan B McMillan, PhD
Dept. of Radiology
University of Wisconsin, Madison

Overview

1. The Paradigm Shift of Solving Problems with Machine and Deep Learning
2. The Steps to Build a (Supervised) Machine Learning Solution
3. How Do We Evaluate a DL Model

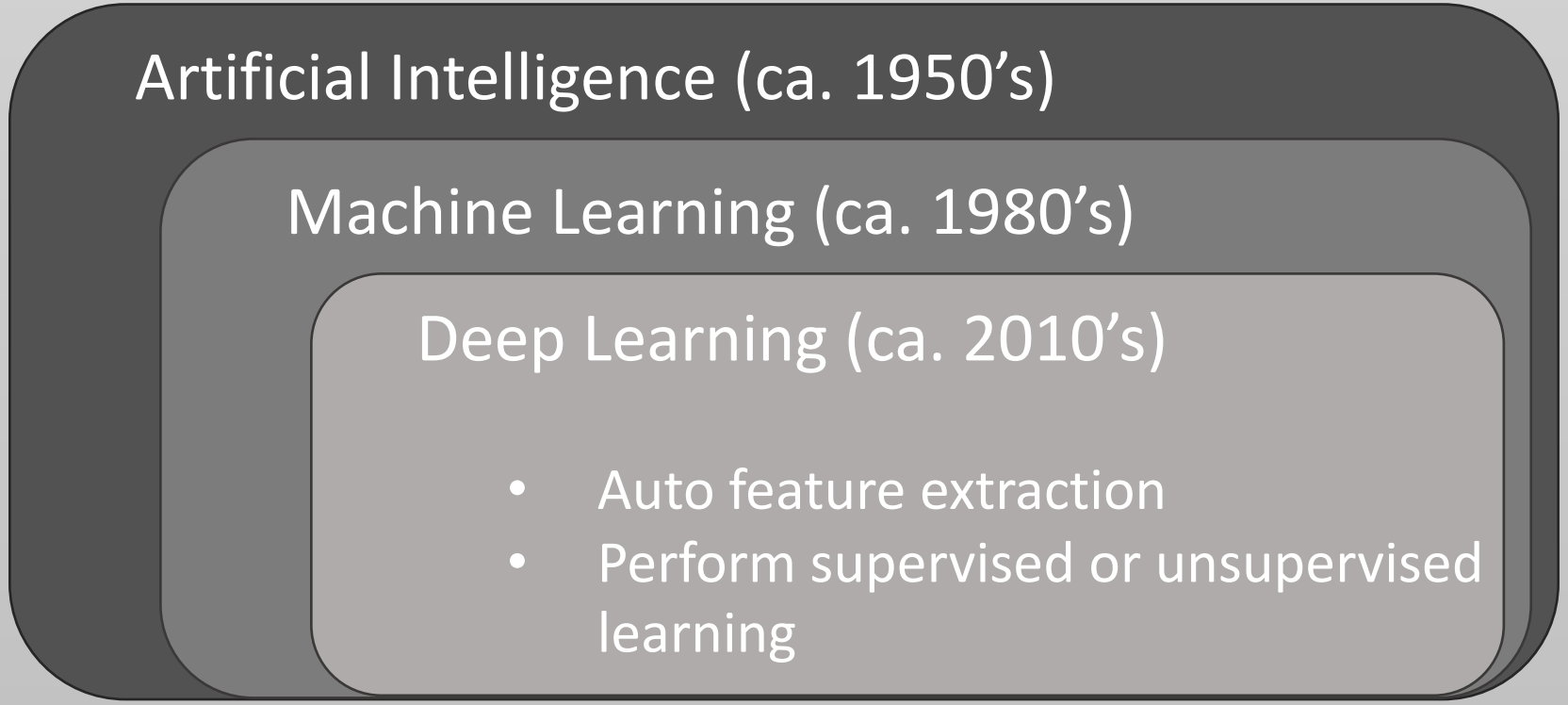
Limitations of this Talk

- Meant to provide a high-level overview of AI in computer vision tasks
- AI is moving very quickly
- Beyond our scope to talk deeply about approaches that combine text, image, and multi-modal data
  - Generative AI is rapidly developing: Firefly, Midjourney, Stable Diffusion, DALL-E, GPT-4, Gemini, Claude
  - The future is multi-modal.

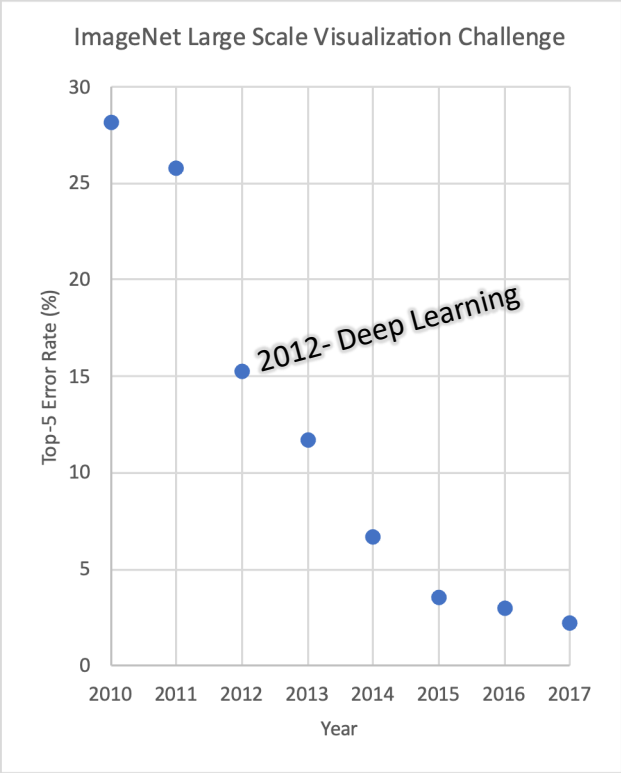
Overview

1. **The Paradigm Shift of Solving Problems with Machine and Deep Learning**
2. The Steps to Build a (Supervised) Machine Learning Solution
3. How Do We Evaluate a DL Model

AI vs. ML vs. DL


Artificial Intelligence (ca. 1950's)
Machine Learning (ca. 1980's)
Deep Learning (ca. 2010's)
• Auto feature extraction
• Perform supervised or unsupervised learning

ImageNet: DL's Revolution in Capability

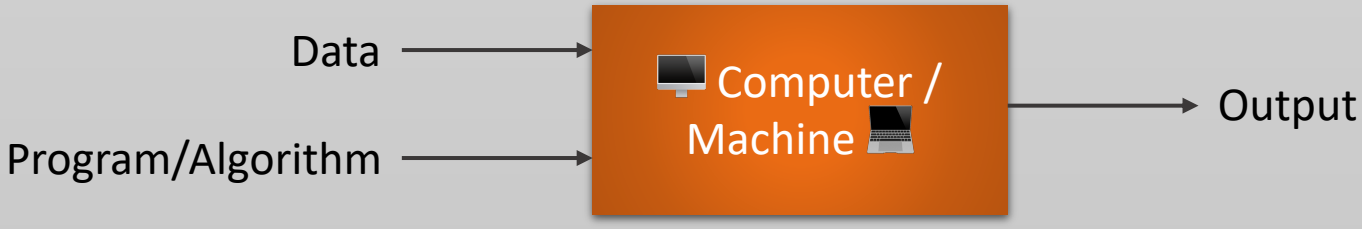


ImageNet Large Scale Visual Recognition Challenge
1000 object categories, localize them in images
DL introduced in 2012, enabled by GPU computing

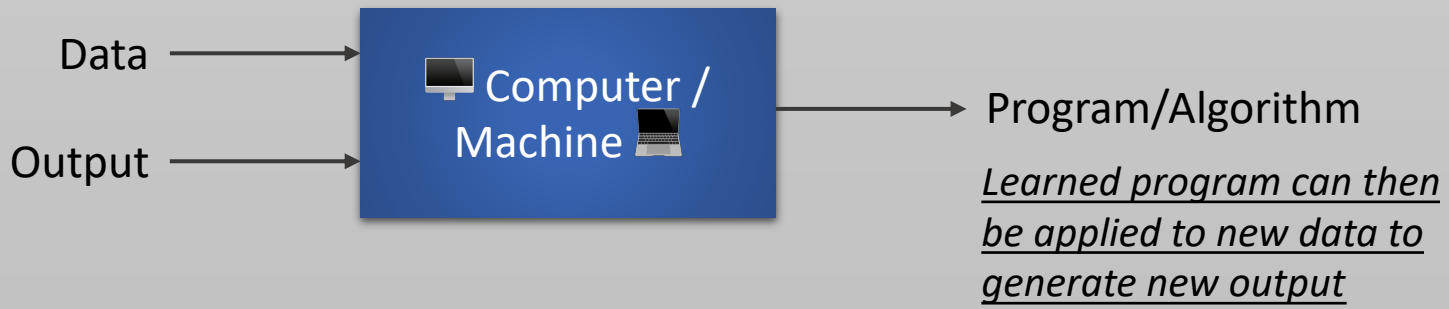
The Machine Learning Paradigm

The ingredients are similar. But the quality, quantity, and curation of the data becomes a primary focus in machine and deep learning applications.

Conventional Paradigm



Machine Learning Paradigm (Supervised)



Supervised vs. Unsupervised Learning

Machine Learning Paradigm (Supervised)

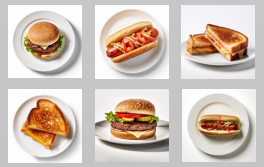
Data

Output

Computer / Machine

Program/Algorithm

Machine Learning Paradigm (Unsupervised)

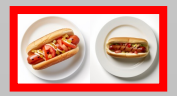


Data

Computer / Machine

Program/Algorithm

Output



Machine Learning vs Deep Learning

Machine Learning

- Boosting / Adaptive Boosting
- Decision tree
- K-means
- K-nearest neighbor (KNN)
- Logistic Regression
- Random forest (RF)
- Support vector machine (SVM)
- Etc…

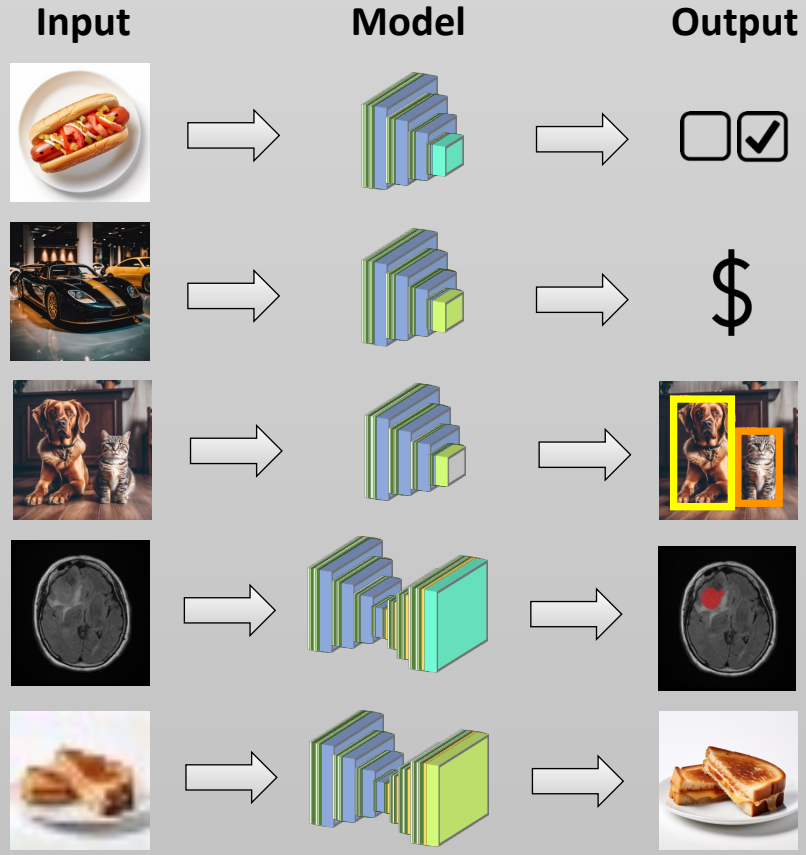
Deep Learning

- Convolutional neural network (CNN)
- Transformers / Vision Transfomer (ViT)
- Deep belief network (DBN)
- Generative adversarial network (GAN)
- Long / Short Term Memory (LTSM)
- Recurrent neural network (RNN)
- Diffusion
- Foundation Models
- Etc…

No time to explain them all! But training/evaluation approach is generally similar.

Data! Data! Data!

Model Structure is Specific to Application



Classification

- Determine category
- Binary or polynary

Regression

- Continuous measure

Detection

- Delineate objects within image

Segmentation

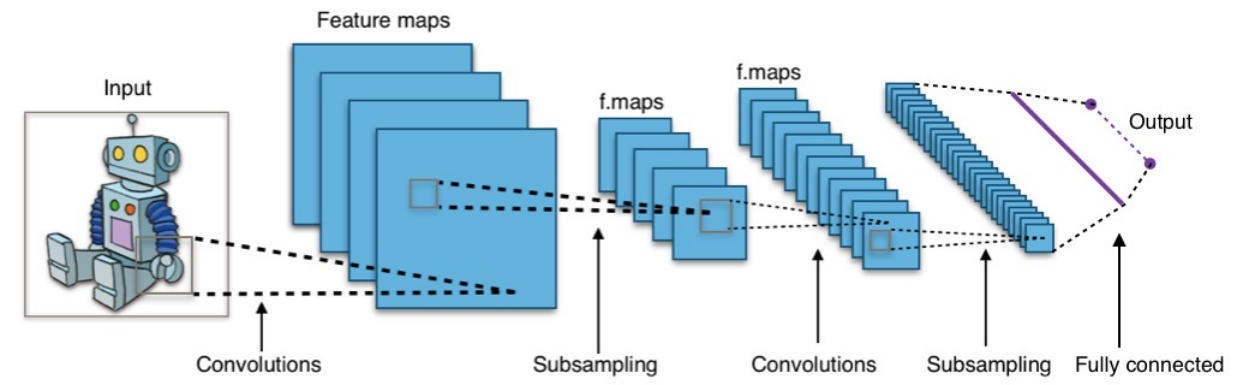
- Reduce manual labor
- Delineate structures/object boundaries in image

Synthesis

- Image denoising, super-resolution
- Image domain transformation (reconstruction)

Inputs can be images and/or other data: text, numerical data, etc.

Convolutional Neural Networks (CNNs)

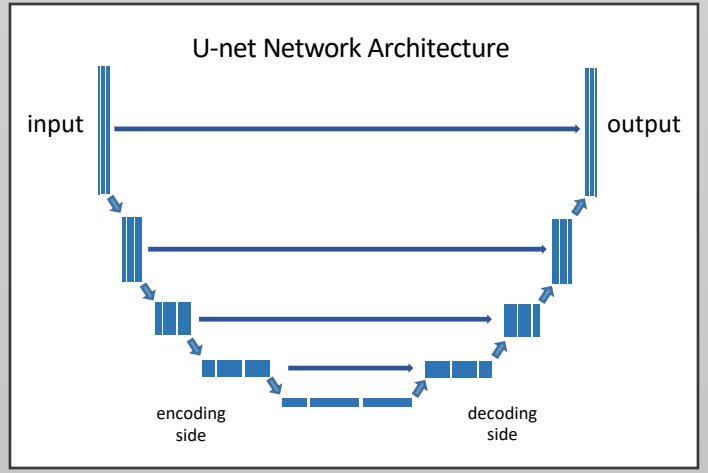


Aphex34, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=45679374

- CNNs utilize a series of convolutions, pooling layers, and activations to encode feature maps
- Strategy is used in many different network structures
- Responsible for the revolutionary performance in computer vision

Segmentation Architectures

- Incredible success of **U-net** CNNarchitecture
  - Contracting (encoder)
  - Expanding (decoder)



U-net: Convolutional networks for biomedical image segmentation

O Ronneberger, P Fischer, T Brox - International Conference on Medical …, 2015 - Springer

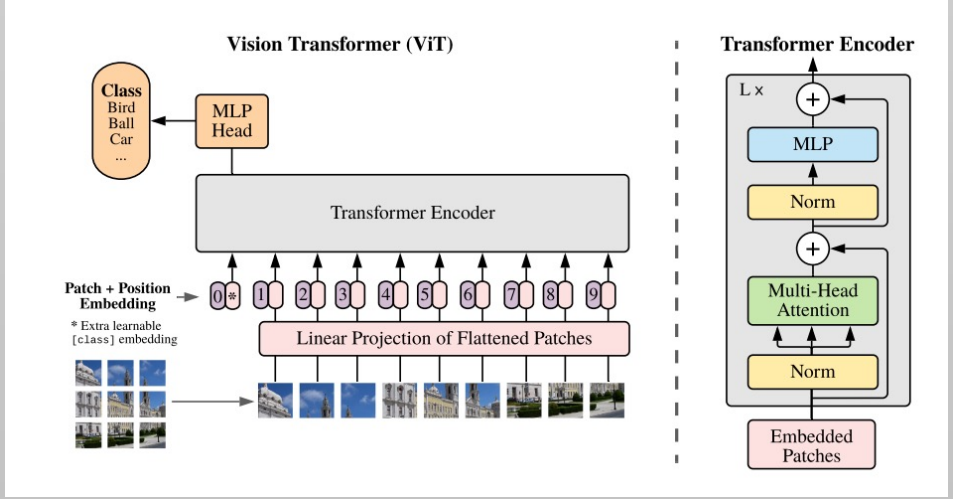
There is large consent that successful training of deep networks requires many thousand annotated training samples. In this paper, we present a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. We show that such a network can be trained end-to-end from very few images and outperforms the prior best method (a sliding …

Very widely used in image segmentation and image synthesis applications

Vision Transformers (ViTs)

- Transformer models have found great application in natural language processing (NLP) applications
  - Have also been adapted to computer vision applications (ViT)

- Decompose inputs into a sequence patches
- Potential advantages in receptive field
- Many state-of-the-art implementations are now using ViTs
  - Hybrid CNN and ViT models



An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (https://arxiv.org/abs/2010.11929)

GANs – Potential for Image Synthesis

https://www.thispersondoesnotexist.com

Training loss is based on the discriminator, so the output images are often sharper and more realistic.

Input (e.g., Random Noise)

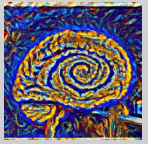
Generator Network

Fake Images

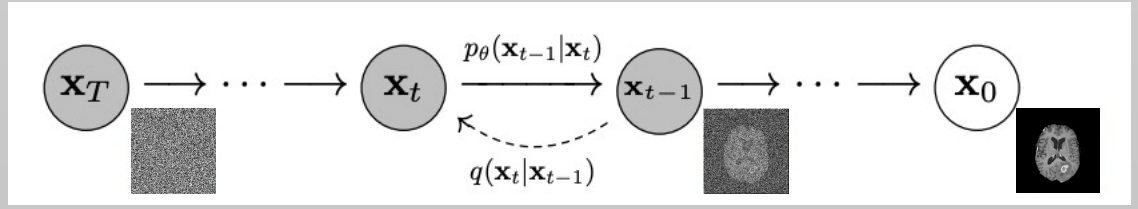
Real Images

Discriminator Network

Diffusion Models



- Recently demonstrated for synthesis applications
  - denoising, super-resolution, inpainting, compressed sensing, art
- Utilize progressive perturbation (noise) in a Markov chain to eliminate the desired image, and learn to recover it



Modified from Denoising Diffusion Probabilistic Models (arXiv:2006.11239)

- Currently provide state-of-the-art performance in many applications

Multi-Modal Models

- Integration of Text and Other Data (Images, Video, Sounds, etc.)
  - Combine natural language processing and computer vision techniques
  - Enable understanding and generation of content that involves both modalities
- Applications
  - Image captioning
  - Visual question answering
  - Text-based image retrieval
  - Art and content generation
- Advantages
  - Enhance contextual understanding by leveraging multiple data types
  - Improve accuracy and performance in tasks requiring both text and visual comprehension

CLIP (Contrastive Language–Image Pre-training)

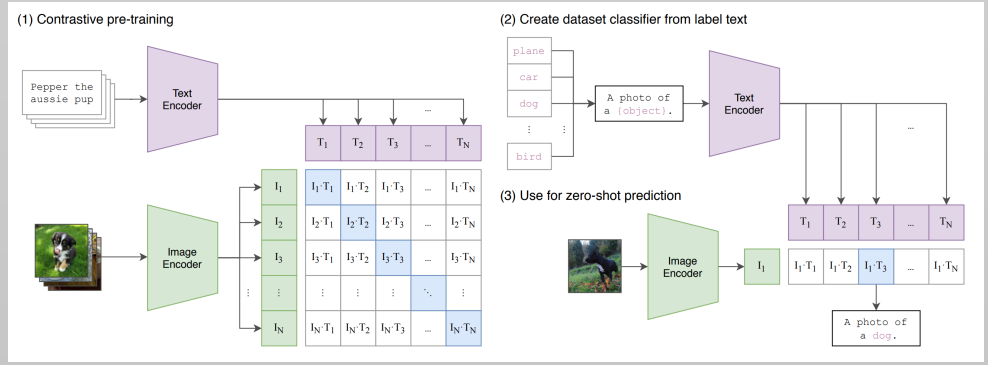
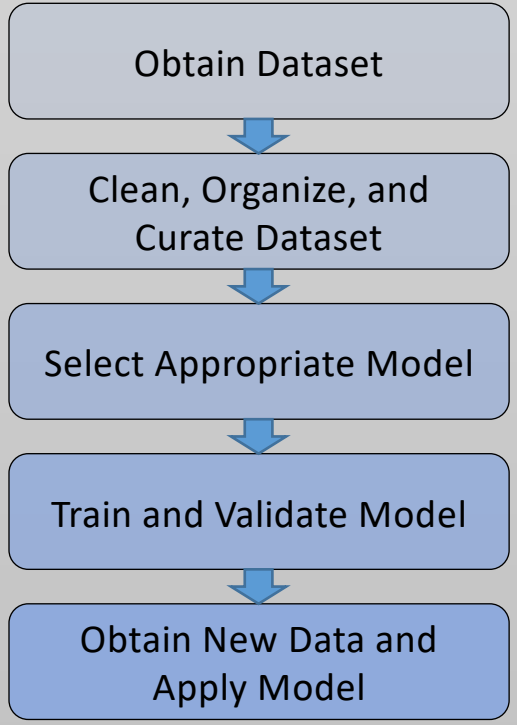


Fig. 1 from Learning Transferable Visual Models From Natural Language Supervision (arXiv:2103.00020)

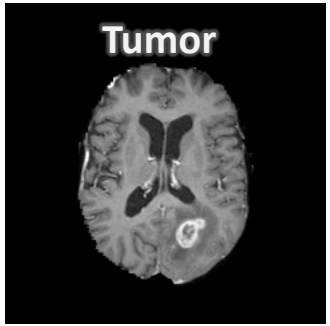
Overview

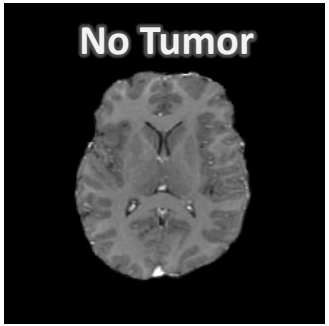
1. The Paradigm Shift of Solving Problems with Machine and Deep Learning
2. **The Steps to Build a (Supervised) Machine Learning Solution**
3. How Do We Evaluate a DL Model

Steps needed to implement ML & DL

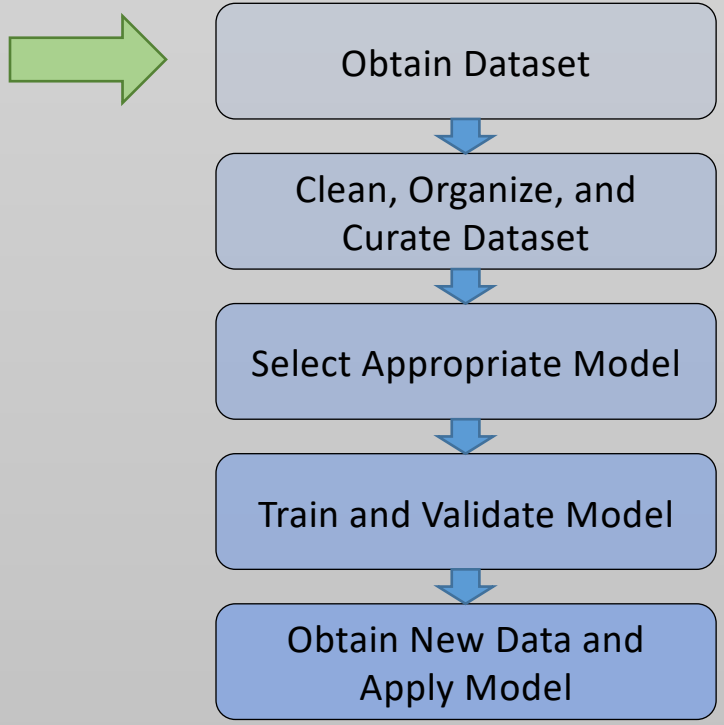


Let's make a system to detect tumors on brain images:





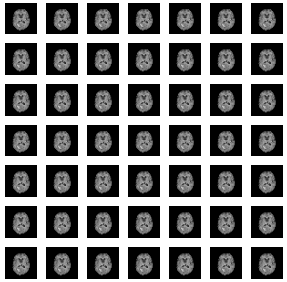
Steps needed to implement ML & DL (1)



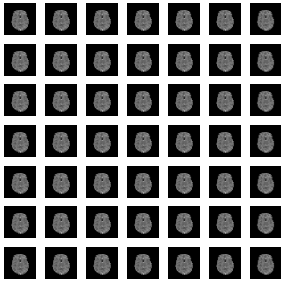
Examples of input and desired output.
Sufficiently large to represent the diversity in your population

For our brain tumor detection system, we need many examples!

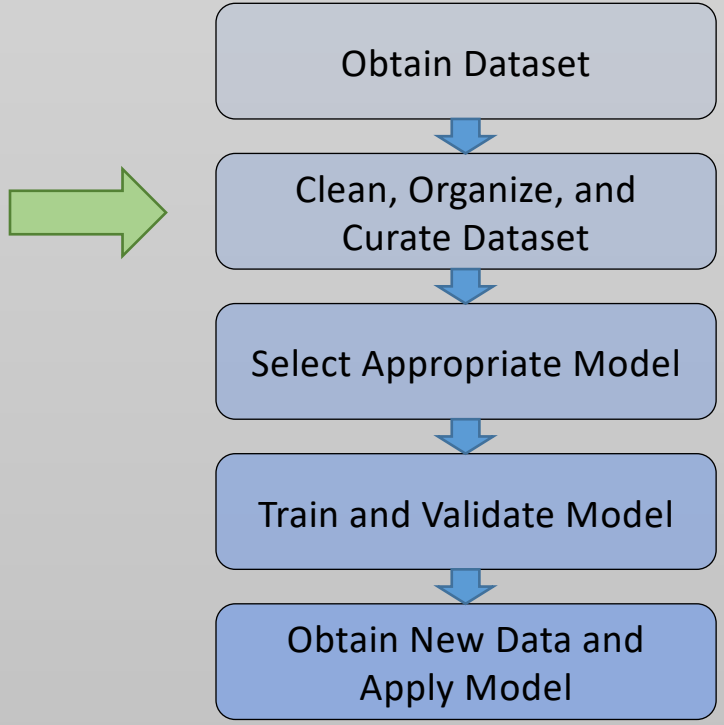
Tumor:



No Tumor:



Steps needed to implement ML & DL (2)

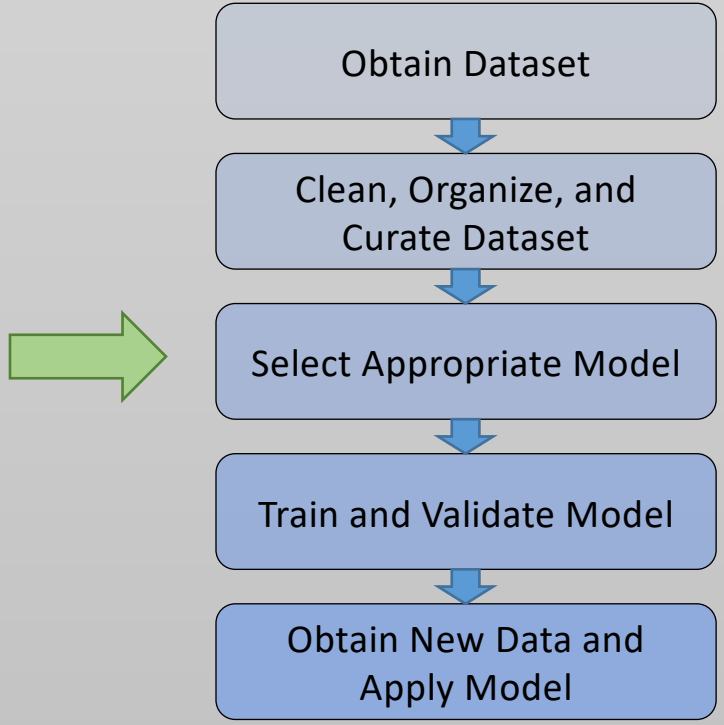


Often the most time-consuming step. Garbage in = Garbage out

We want our brain tumor detection system to learn from the correct answer only (true negatives and true positives)

- **Hematoma**
- **Infection**
- **Radiation necrosis**

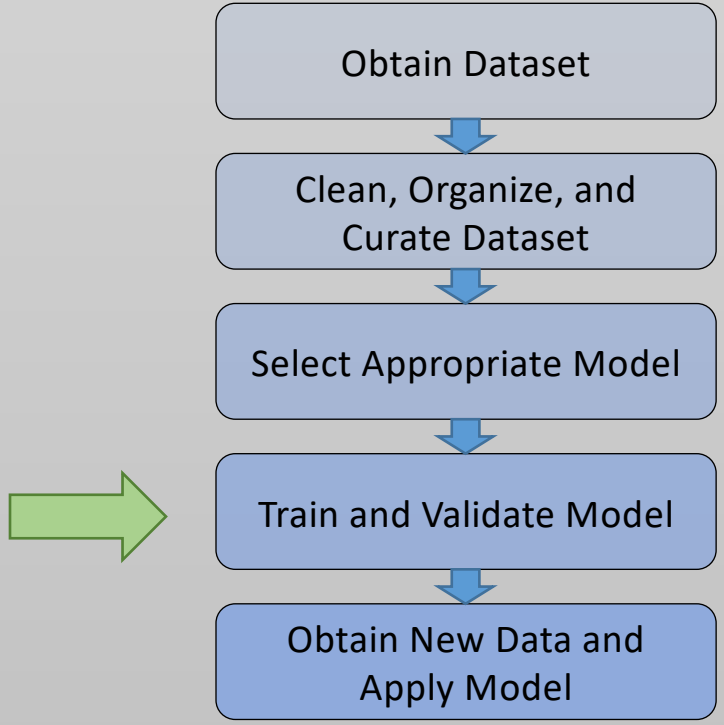
Steps needed to implement ML & DL (3)



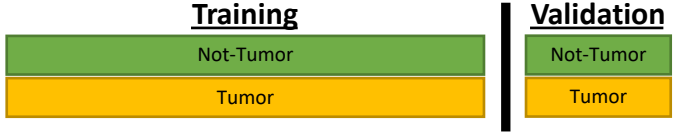
Sometimes an iterative process after training/validation.

Our model should be structured to deliver the answer to the question we are asking it: Tumor or Not?

Steps needed to implement ML & DL (4)



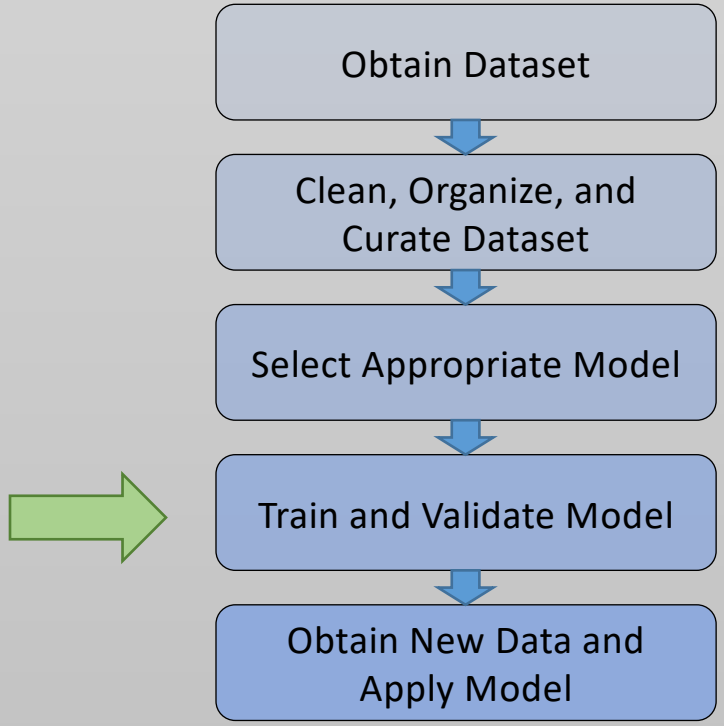
The learning algorithm requires hours (or days) to process. We reserve a portion of our data set to validate the training process (are we learning?)

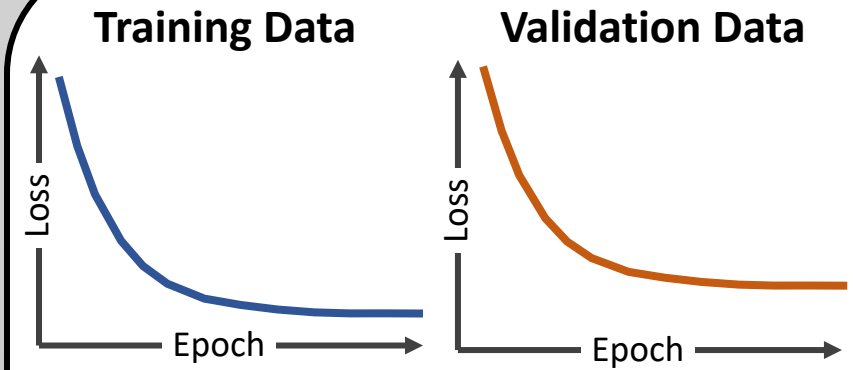


Need big compute to train (GPUs). Naïve validation data important

Often we want to shuffle Training and Validation sets and retrain from scratch to test robustness => Cross Validation

Steps needed to implement ML & DL (4)

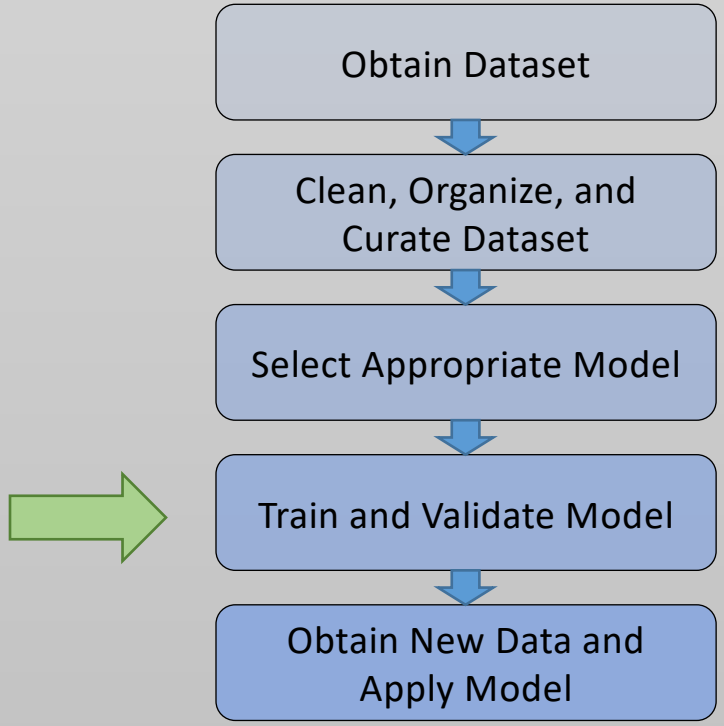


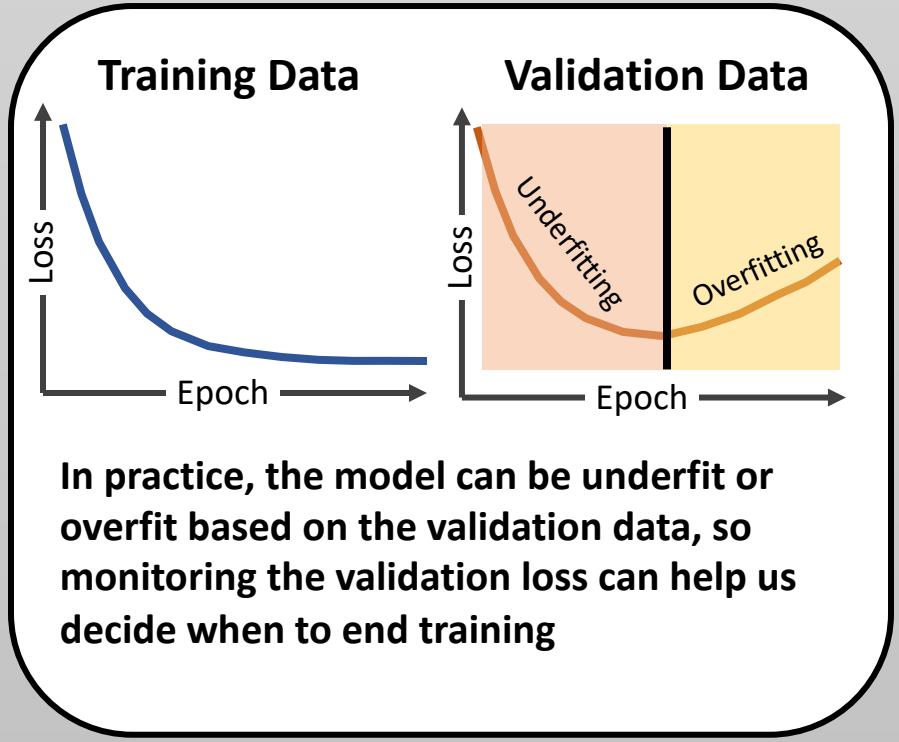


During training, the loss function should decrease at each epoch. Validation data allows us to guess how well the model might work at this point if we stop training

The loss function needs to match the problem at hand.

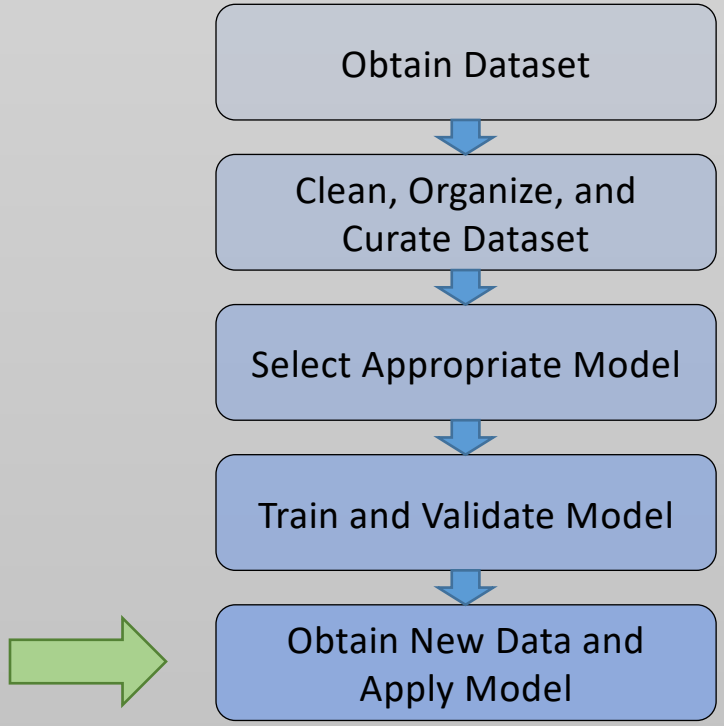
Steps needed to implement ML & DL (4)





In practice, the model can be underfit or overfit based on the validation data, so monitoring the validation loss can help us decide when to end training

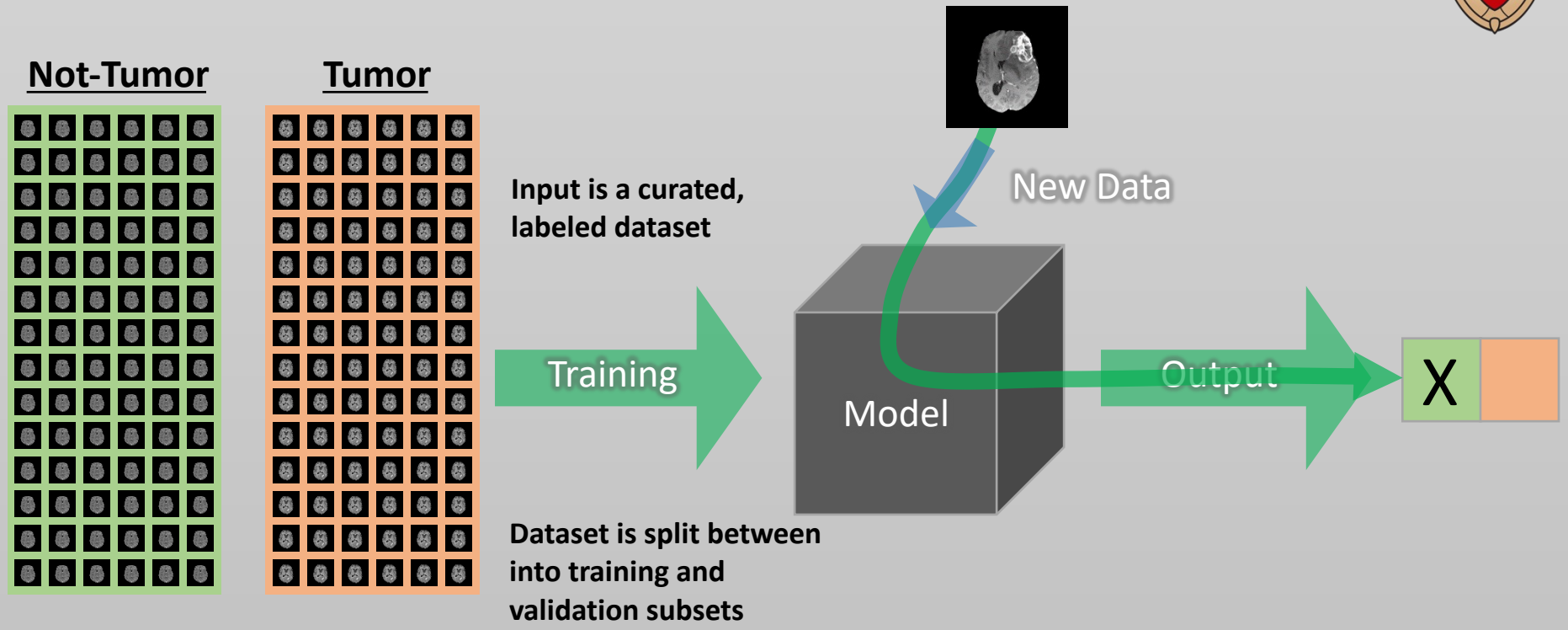
Steps needed to implement ML & DL (5)



If successful, our model can make accurate inferences on new data

New input data can be rapidly assessed.

Example: Brain tumor detection system

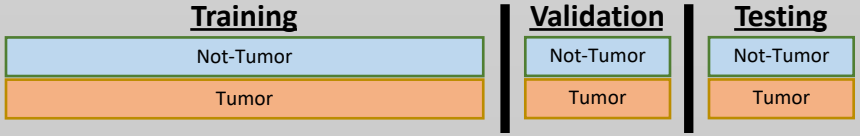


Overview

1. The Paradigm Shift of Solving Problems with Machine and Deep Learning
2. The Steps to Build a (Supervised) Machine Learning Solution
3. **How Do We Evaluate an AI Model**

Evaluating Performance

- Once trained, the model should be evaluated on a testing dataset that also includes ground-truth:



  - Cross-validation is a good strategy for small datasets
- Several performance metrics should be evaluated, depending on problem, for example:

Classification
Accuracy
Sensitivity
Specificity
AUC
TP, FP, FN, TN, etc.

Regression
Mean Absolute Error
Mean Square Error
%-Difference

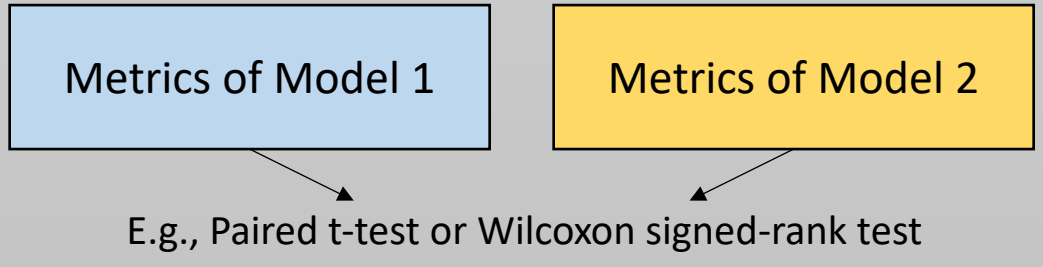
Segmentation
Dice coefficient
Intersection over union
TP, FP, FN, TN, etc.

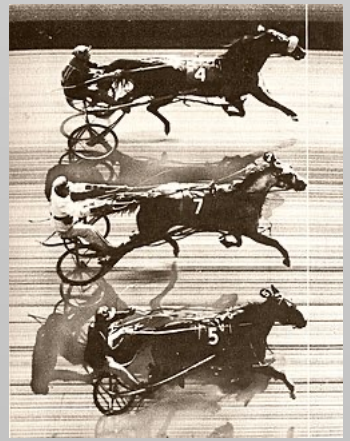
Synthesis
Mean Absolute Error
Mean Square Error
%-Difference
Peak SNR
Structural Similarity
Subjective interpretation

Evaluating Performance

- Model performance should be reported as a confidence interval or mean ± standard deviation.

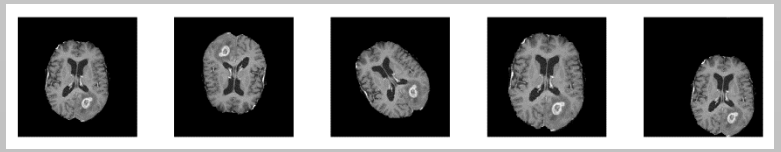
- Statistical tests should be used to demonstrate meaningful differences between developed approaches, e.g.:





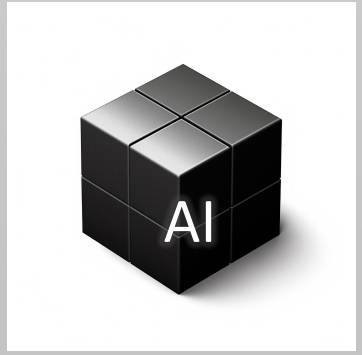
Generalizability and Bias

- ML and DL will always perform best on the data it was trained on
- Is it Generalizable? Is the training data biased?
  - Does model work on data outside of the training/validation/testing cohort?
  - Technical dependence? Specific scanner, PSD, vendor, artifact-free, etc.
  - Biased training data? Gender, ethnic, disease, anatomical abnormalities, etc.
- What can we do about these issues?
  - Enlarge dataset
    - Naturally, by getting more data, e.g., multi-institutional federated learning
    - Artificially, by synthesizing more data, e.g., doing data augmentation
  - Be mindful about potential biases



Interpretability

- Deep learning is often considered a black box
  - However, there is a great deal of recent focus on interpretable AI

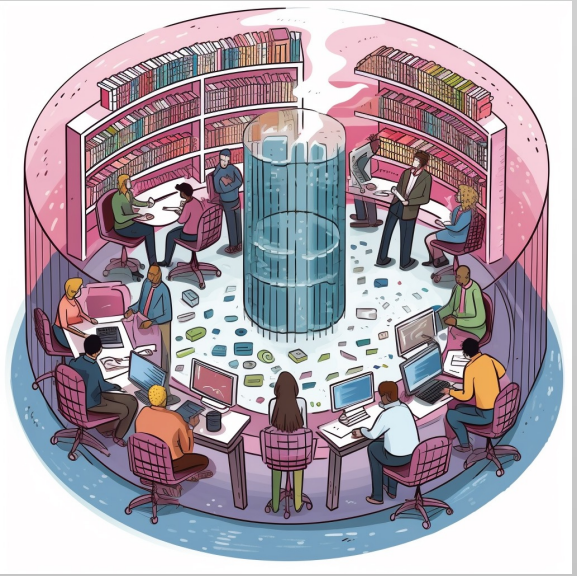


<u>Potential approaches towards interpretable AI:</u>

- Output Probabilities
  - An output probability of the classification
- Visual Saliency
  - A salience map to depict which image regions most influence the output
- By Example
  - Show cases with the same classification
- Semantic Explanation
  - Text description of the output

Reproducibility through sharing data and code

- Most ML and DL frameworks are open source
- Many authors have shared source code to implementations
- The availability of large databases and example code will help translate technology and foster reproducible research.
  - Challenging in the intelligence community



Summary

1. The Paradigm Shift of Solving Problems with Machine and Deep Learning
2. The Steps to Build a Machine Learning Solution
3. How Do We Evaluate a DL Model

Thanks!

linkedin.com/in/alan-b-mcmillan/

abmcmillan@wisc.edu

go.wisc.edu/mimrtl