Medical Image Analysis with CNNs

Matthew Engelhard

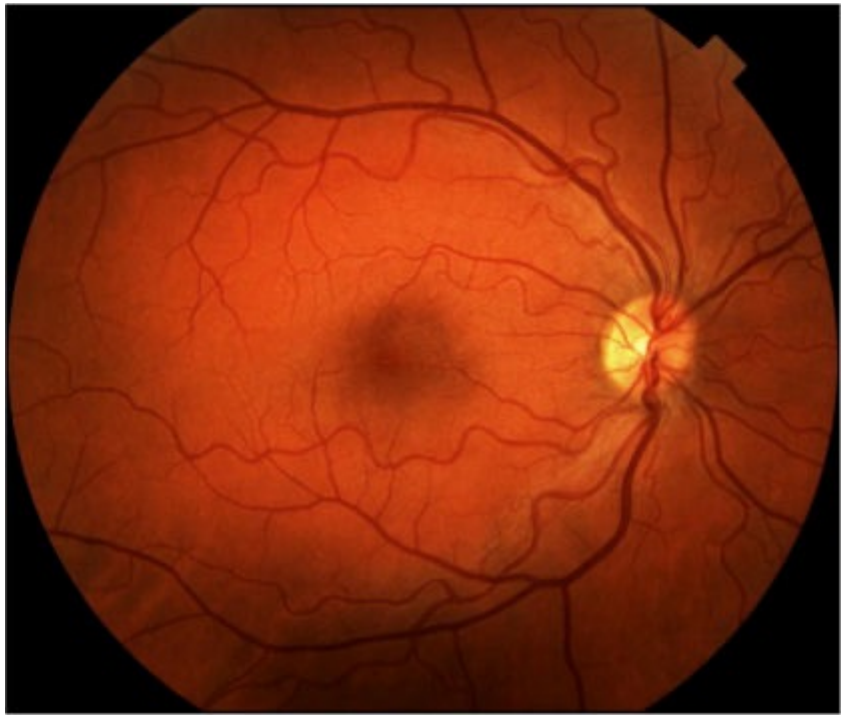
Overview

- Intuition on feature hierarchy

- Esteva et al as a jumping off point to discuss:
  - How do we adapt an existing model to classify medical images?
  - Why is it a good idea to do this?
  - How do we get labels for medical image classification tasks, and how “good” do they have to be?
  - How do we look inside the black box?
  - Preprocessing details for images

Deep Learning for Image Analysis

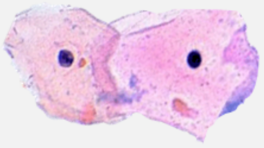
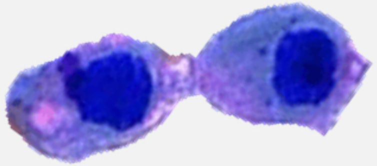
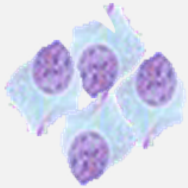
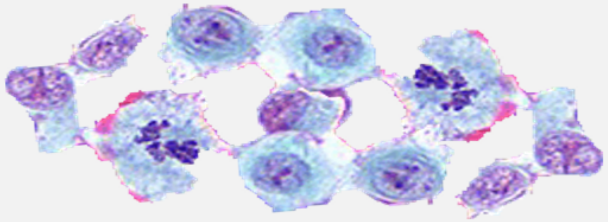
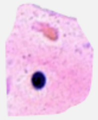
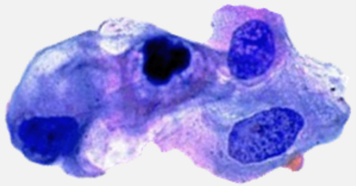
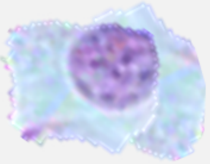
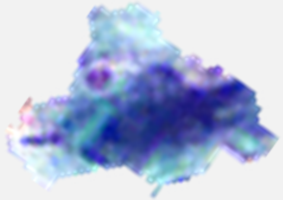
Diabetic Retinopathy Classification



Healthy Retina

Unhealthy Retina

To identify cancerous cells, first learn to identify cells and their components.

Normal	Cancer	
		Large, variably shaped nuclei
		Many dividing cells; Disorganized arrangement
		Variation in size and shape
		Loss of normal features

https://sphweb.bumc.bu.edu/otlt/mph-modules/ph/ph709_cancer/ph709_cancer7.html

nature
International journal of science

Letter | Published: 25 January 2017

Dermatologist-level classification of skin cancer with deep neural networks

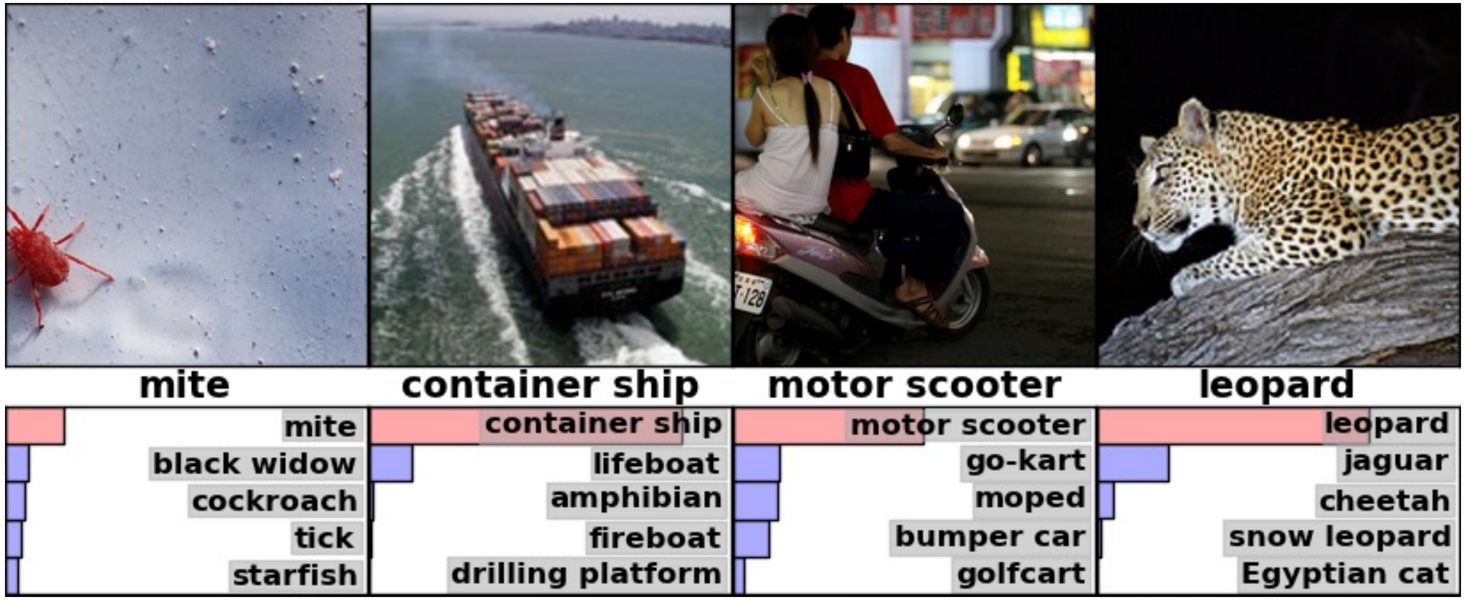
Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau & Sebastian Thrun

Nature **542**, 115–118 (02 February 2017) | Download Citation

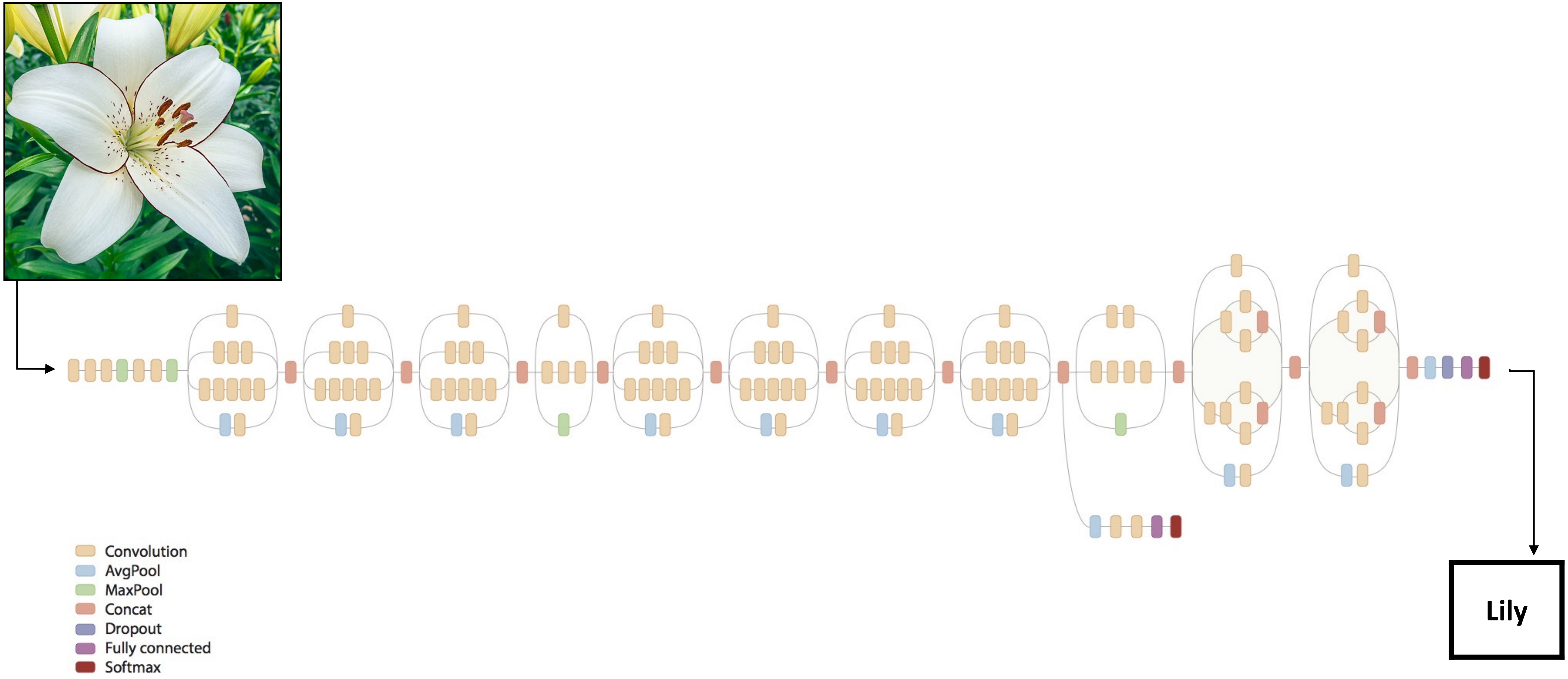
Medical Image Classification

Identifying Skin Cancer

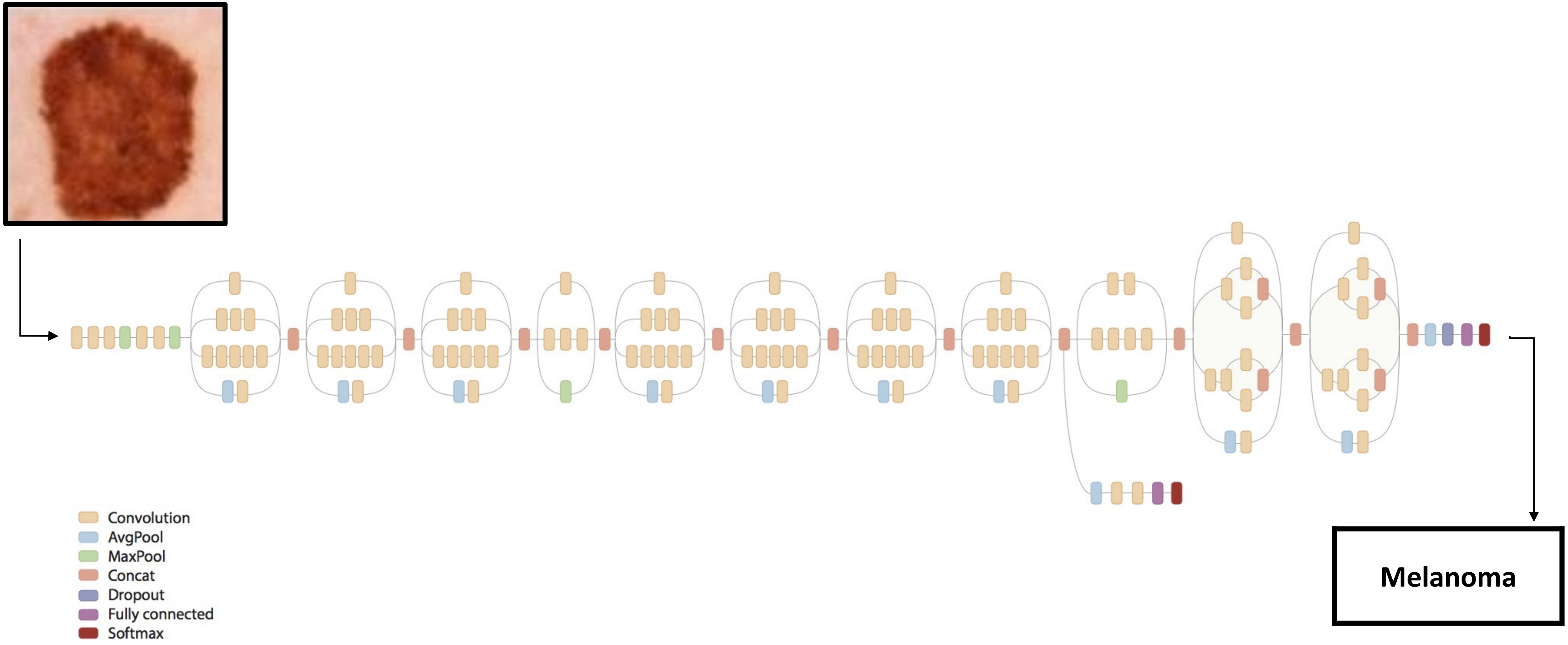
A Classification Task:
predict the label associated with each image



Take a model trained on naturalistic images...



…and repurpose it to evaluate medical images



Repurposing our model

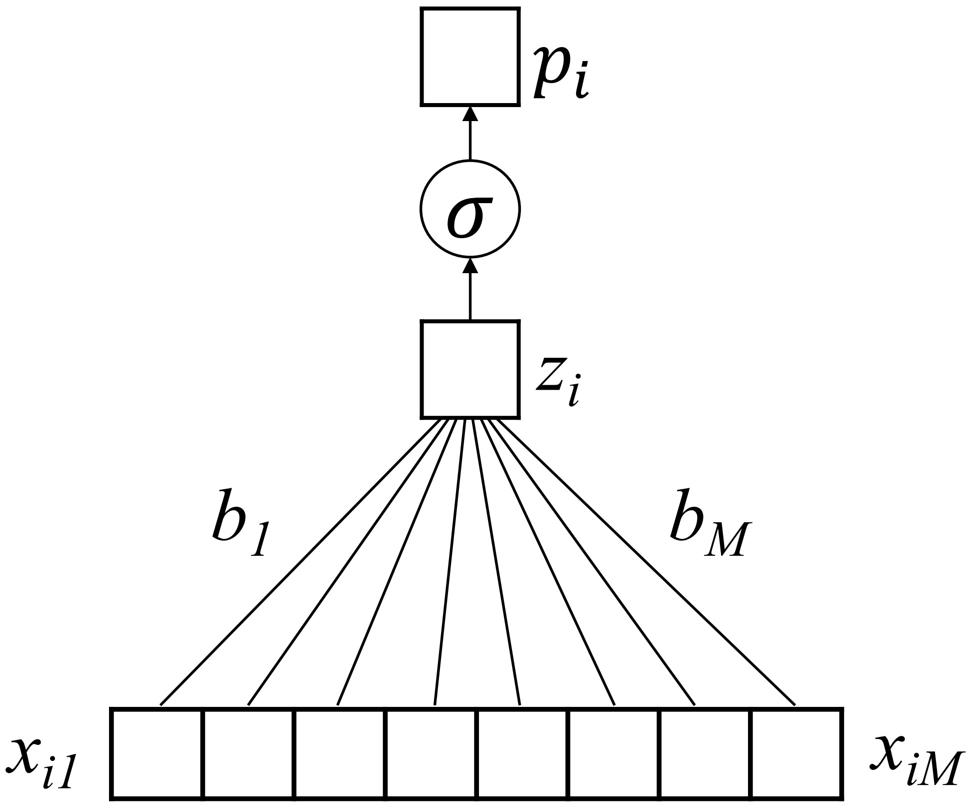
- Step 1: Modify the **architecture**
- Step 2: Fine-tune the **parameters**

Classifier Output: Two-Class (e.g. Yes/No)

Diabetic retinopathy? (Yes/No)

Gulshan et al. *JAMA* (2016)

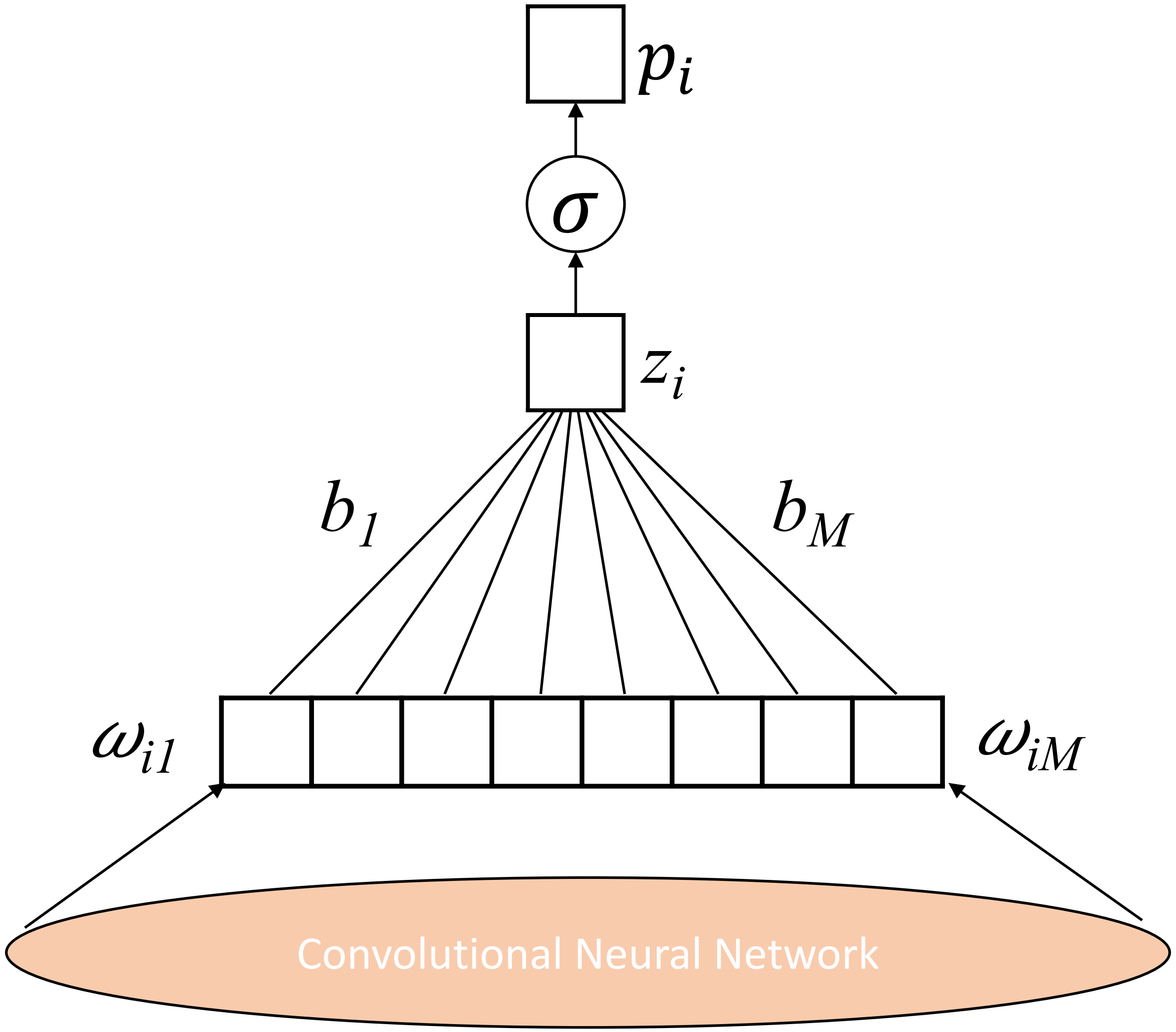
Two-Class Predictions



$$\sigma(z_i) = \frac{e^{z_i}}{1 + e^{z_i}}$$

In logistic regression, x_i is a vector of predictor variables

Two-Class Predictions



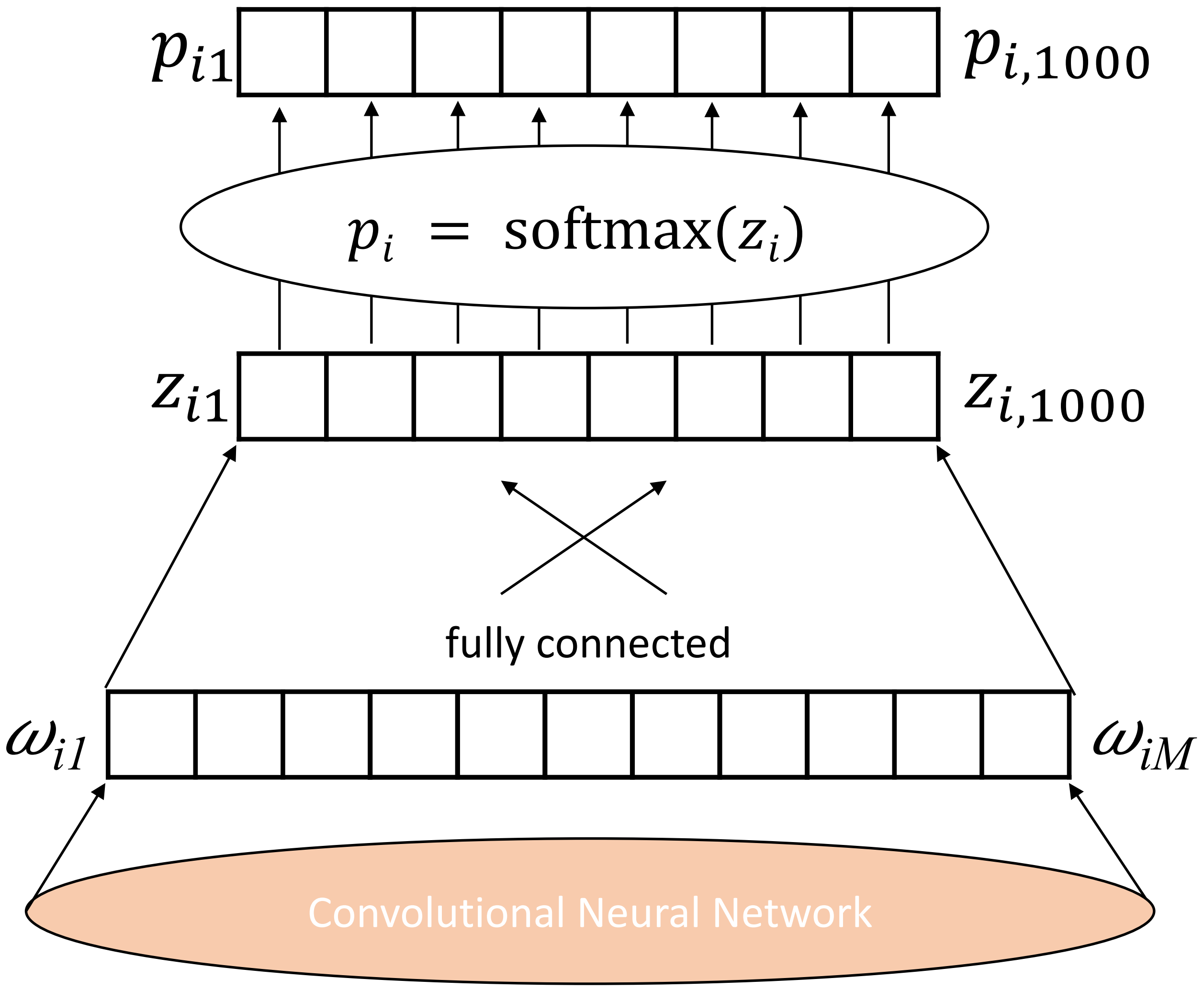
$$\sigma(z_i) = \frac{e^{z_i}}{1 + e^{z_i}}$$

When identifying diabetic retinopathy, consider ω_i , a vector of high-level features extracted by the CNN

Classifier Output:
Multi-Class (ImageNet)

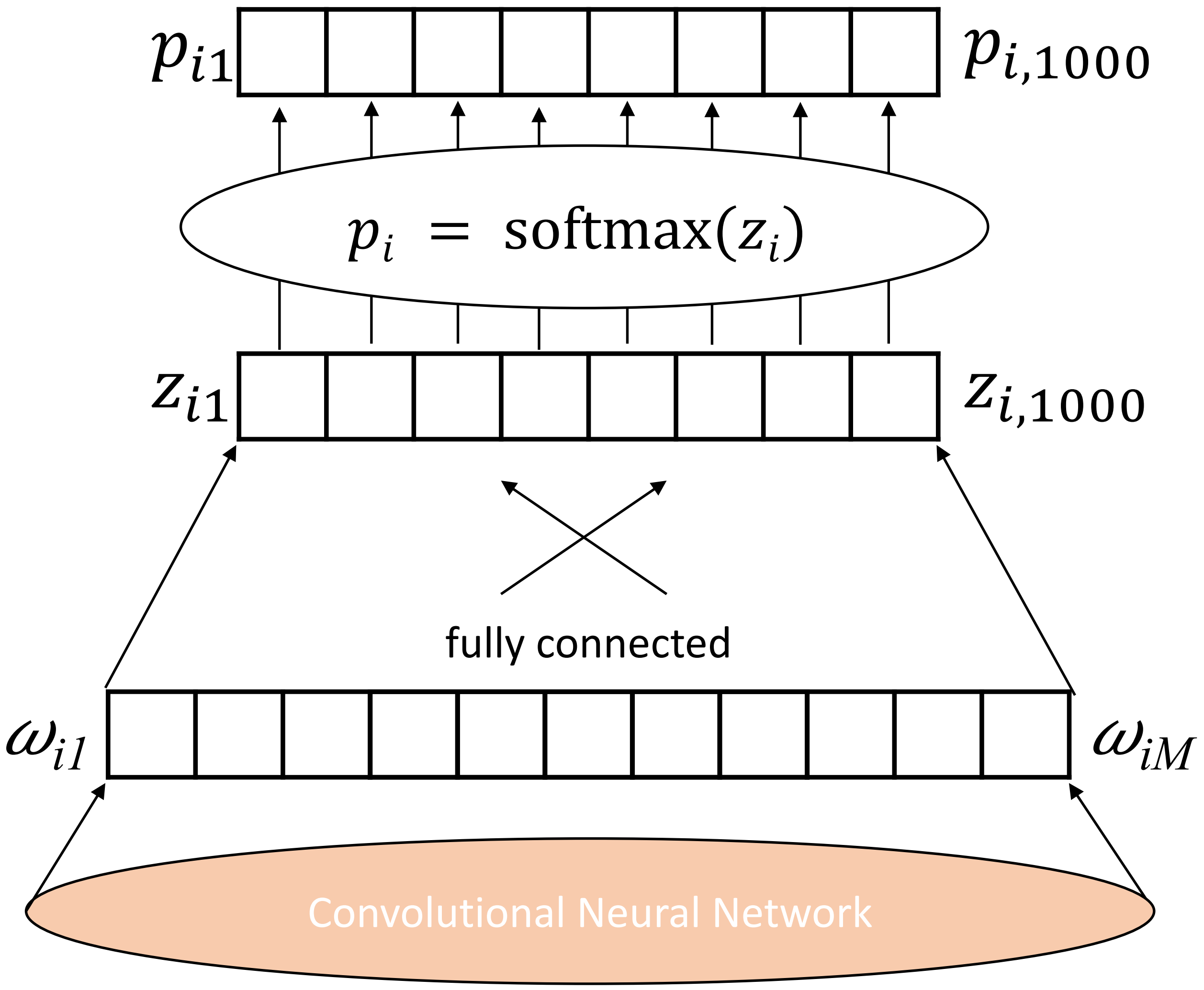
Image Label?
(1000 classes)

Multi-Class Predictions



ω_i is a vector of high-level features extracted by the CNN

Multi-Class Predictions



$$p_{ij} = \frac{e^{z_{ij}}}{\sum_{c=1}^{1000} e^{z_{ic}}}$$

$$\sigma(z_i) = \frac{e^{z_i}}{1 + e^{z_i}}$$

z_i are log-odds scores for each class

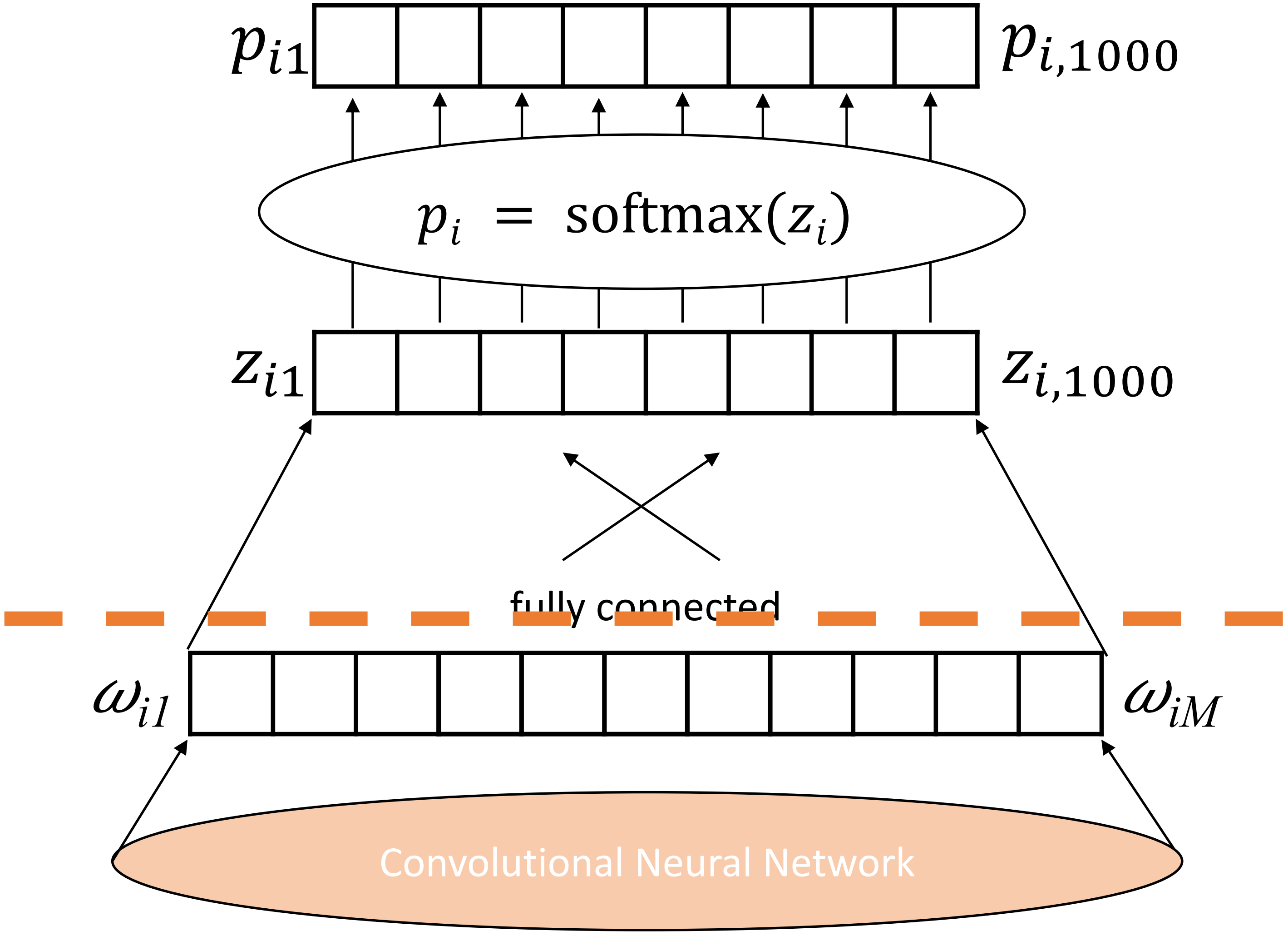
ω_i is a vector of high-level features extracted by the CNN

Classifier Output:
Multi-Class (Lesion Type)

Type of Skin Lesion?
(757 classes)

Esteva et al. *Nature* (2017)

Step 1: Modify the Architecture



1000 training classes

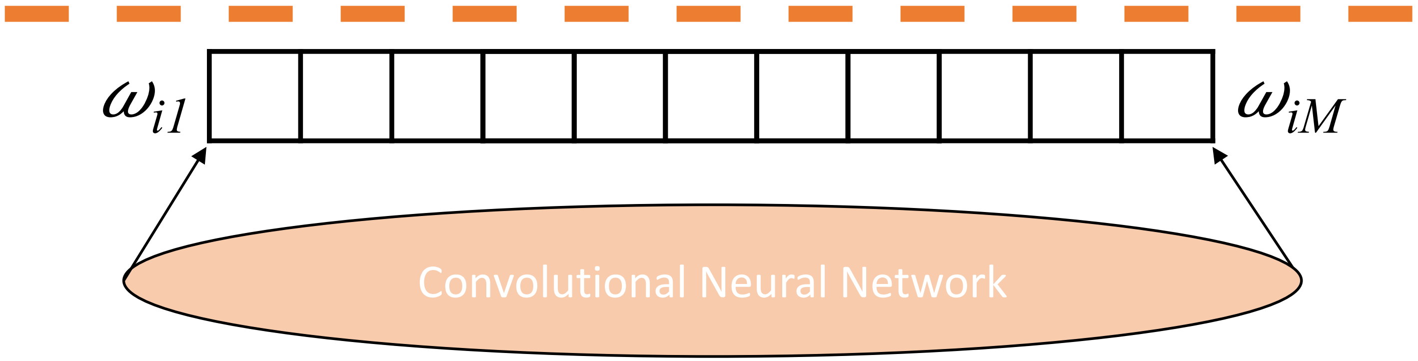
757 training classes

ω_i is a vector of high-level features extracted by the CNN

Step 1: Modify the Architecture

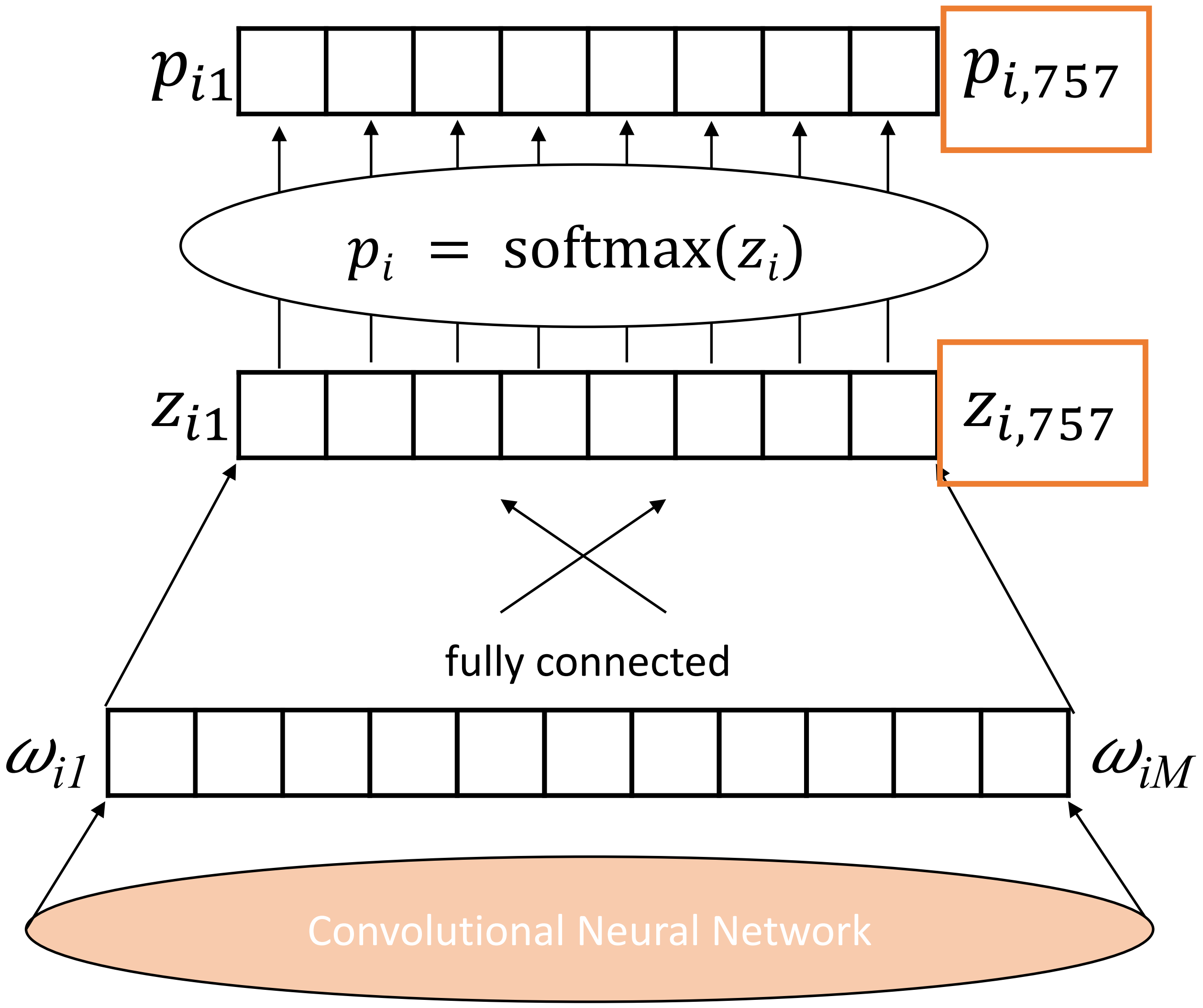
1000 training classes

757 training classes



ω_i is a vector of high-level features extracted by the CNN

Step 1: Modify the Architecture



New layers with randomly initialized weights

ω_i is a vector of high-level features extracted by the CNN

Step 2: *fine-tune* the parameters

- This just means we train the model.
- The only difference is that normally, when we train the model, we start with randomly selected parameters
- This time, we start with parameters from the original model
- Everything else is the same: we follow the gradient to reduce the cross-entropy loss as much as we can

Why is pre-training a good idea?

https://docs.gimp.org/en/plug-in-convmatrix.html

A filter that detects edges may be useful for many classification tasks.

Every model needs an edge detector… why waste time re-learning it?

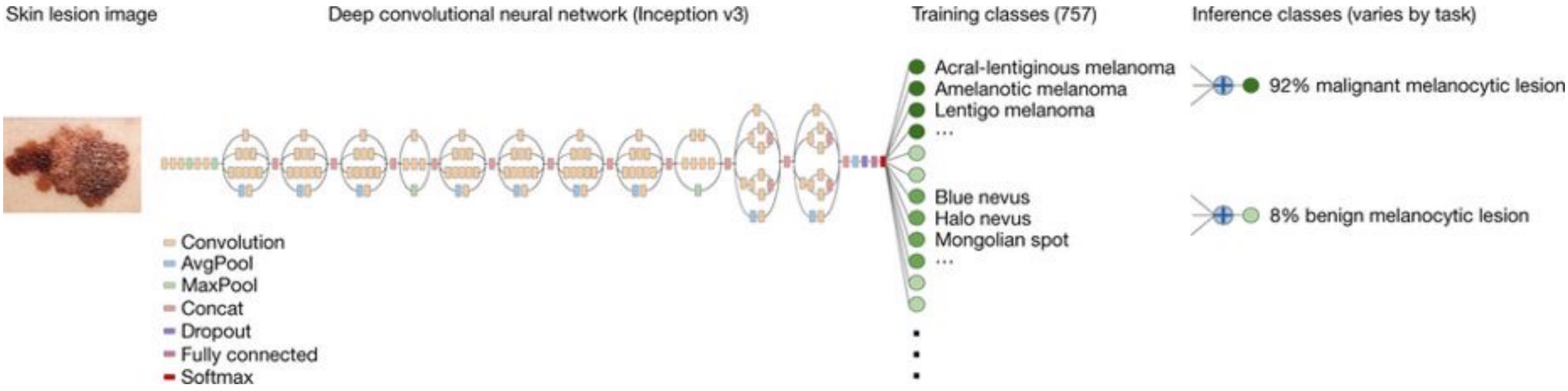
Pre-training, in brief

1) fine-tuning a pre-trained model tends to be **at least as good as learning from scratch**
   (empirical result)

2) freeze early layers and fine-tune later layers
   more data → fine-tune more layers

3) best tuning depth depends on the application, and should be explored

Repurposing the Inception v3 CNN



- Begin with a model trained on ImageNet (to classify everyday images)
- Modify the <u>architecture</u> to match the new number of training classes
- Fine-tune <u>parameters</u> using images of skin lesions

Inception v3 and many other models are freely available

Pre-trained Models

Neural nets work best when they have many parameters, making them powerful function approximators. However, this means they must be trained on very large datasets. Because training models from scratch can be a very computationally intensive process requiring days or even weeks, we provide various pre-trained models, as listed below. These CNNs have been trained on the ILSVRC-2012-CLS image classification dataset.

In the table below, we list each model, the corresponding TensorFlow model file, the link to the model checkpoint, and the top 1 and top 5 accuracy (on the imagenet test set). Note that the VGG and ResNet V1 parameters have been converted from their original caffe formats (here and here), whereas the Inception and ResNet V2 parameters have been trained internally at Google. Also be aware that these accuracies were computed by evaluating using a single image crop. Some academic papers report higher accuracy by using multiple crops at multiple scales.

Model	TF-Slim File	Checkpoint	Top-1 Accuracy	Top-5 Accuracy
Inception V1	Code	inception_v1_2016_08_28.tar.gz	69.8	89.6
Inception V2	Code	inception_v2_2016_08_28.tar.gz	73.9	91.8
Inception V3	Code	inception_v3_2016_08_28.tar.gz	78.0	93.9
Inception V4	Code	inception_v4_2016_09_09.tar.gz	80.2	95.2

TF-Slim Code:
Defines the model architecture

Checkpoint File:
Trained model parameters

https://github.com/tensorflow/models/tree/master/research/slim#Pretrained

“Ground Truth” in Medicine

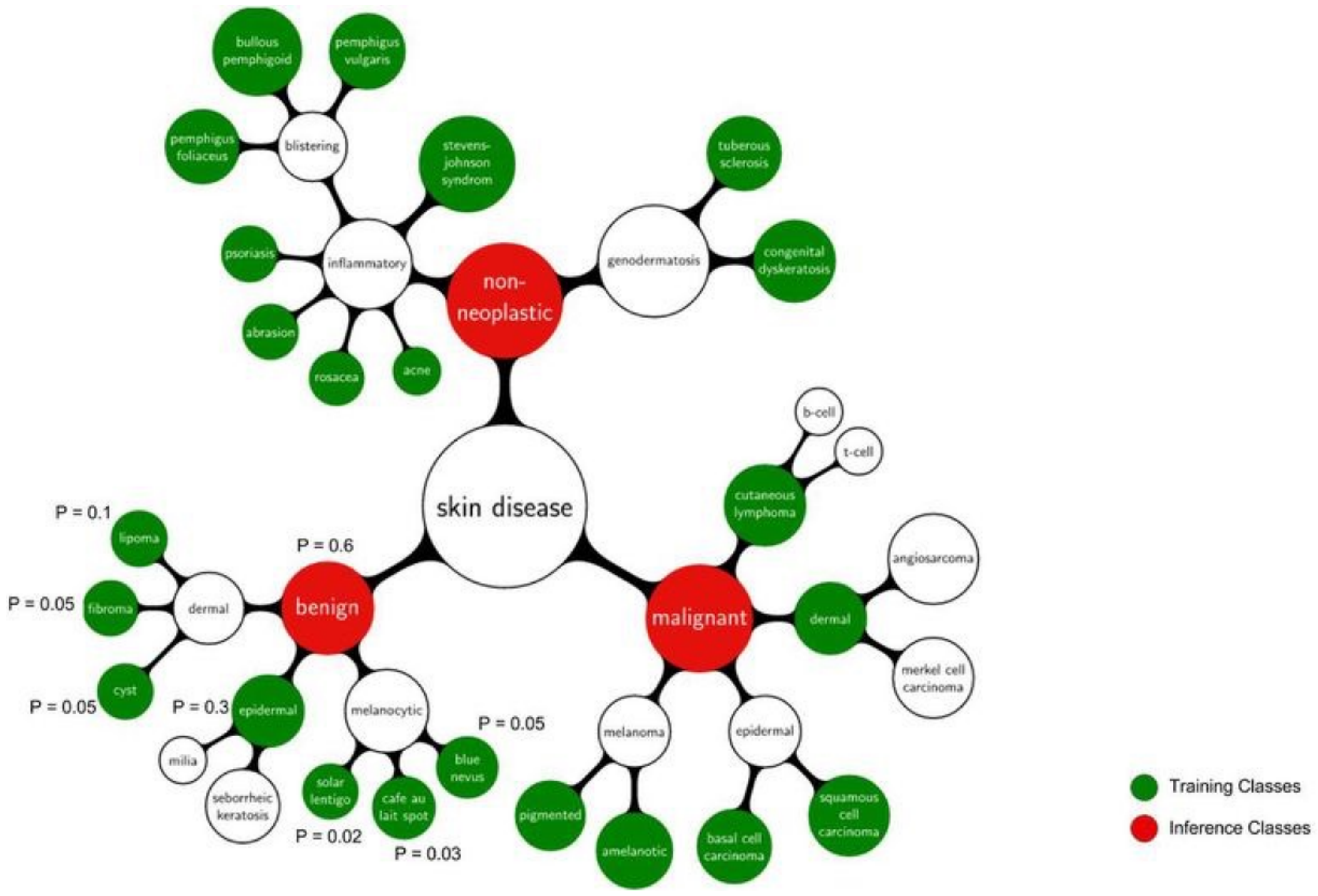
What should the labels be, and how do we get a lot of them?

Two Types of Labels, Two Rounds of Evaluation

1. Model development: predict dermatologists' annotations:
   - Three-class disease partition
   - Nine-class disease partition

2. Model evaluation: predict biopsy result (benign vs malignant)
   - Keratinocyte carcinoma vs benign seborrheic keratosis
   - Malignant melanoma vs benign nevus
     - Standard images
     - Dermoscopy

Specifying training classes based on taxonomy of lesions

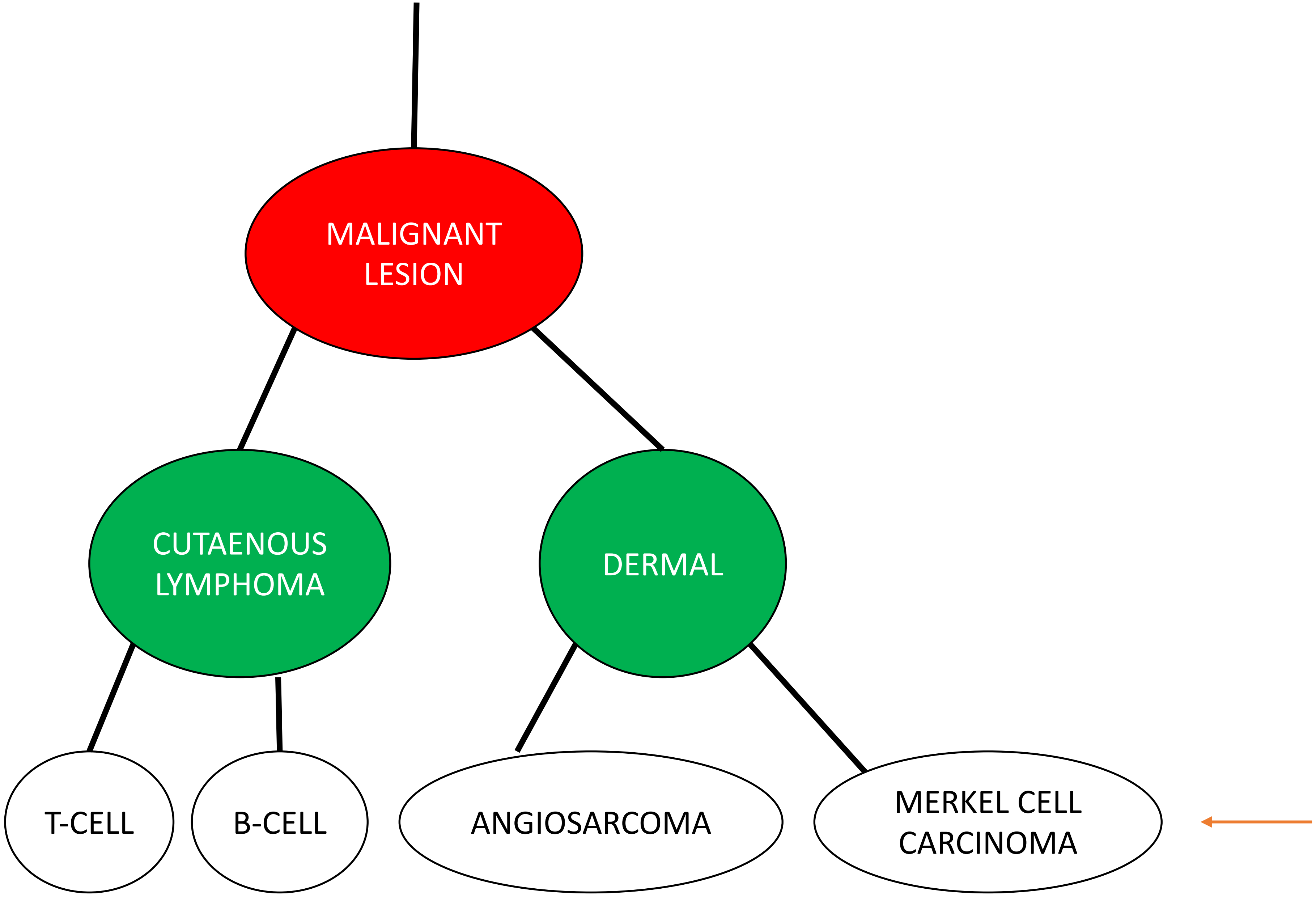


Disease Partitioning Algorithm:

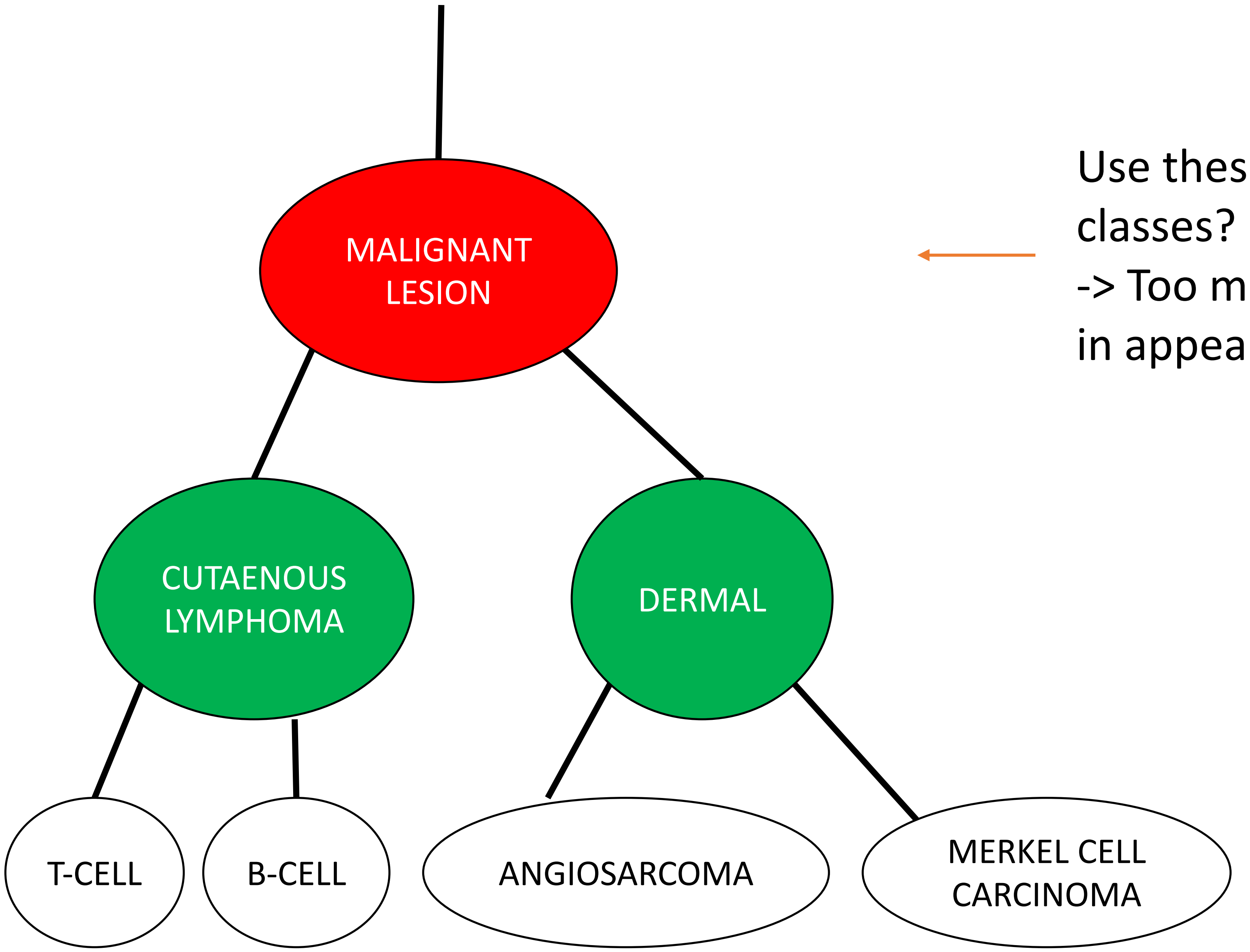
- Ascend the tree until the current node contains <1000 images across all child nodes. Add these images as a distinct training class.

- This resulted in 757 training classes.

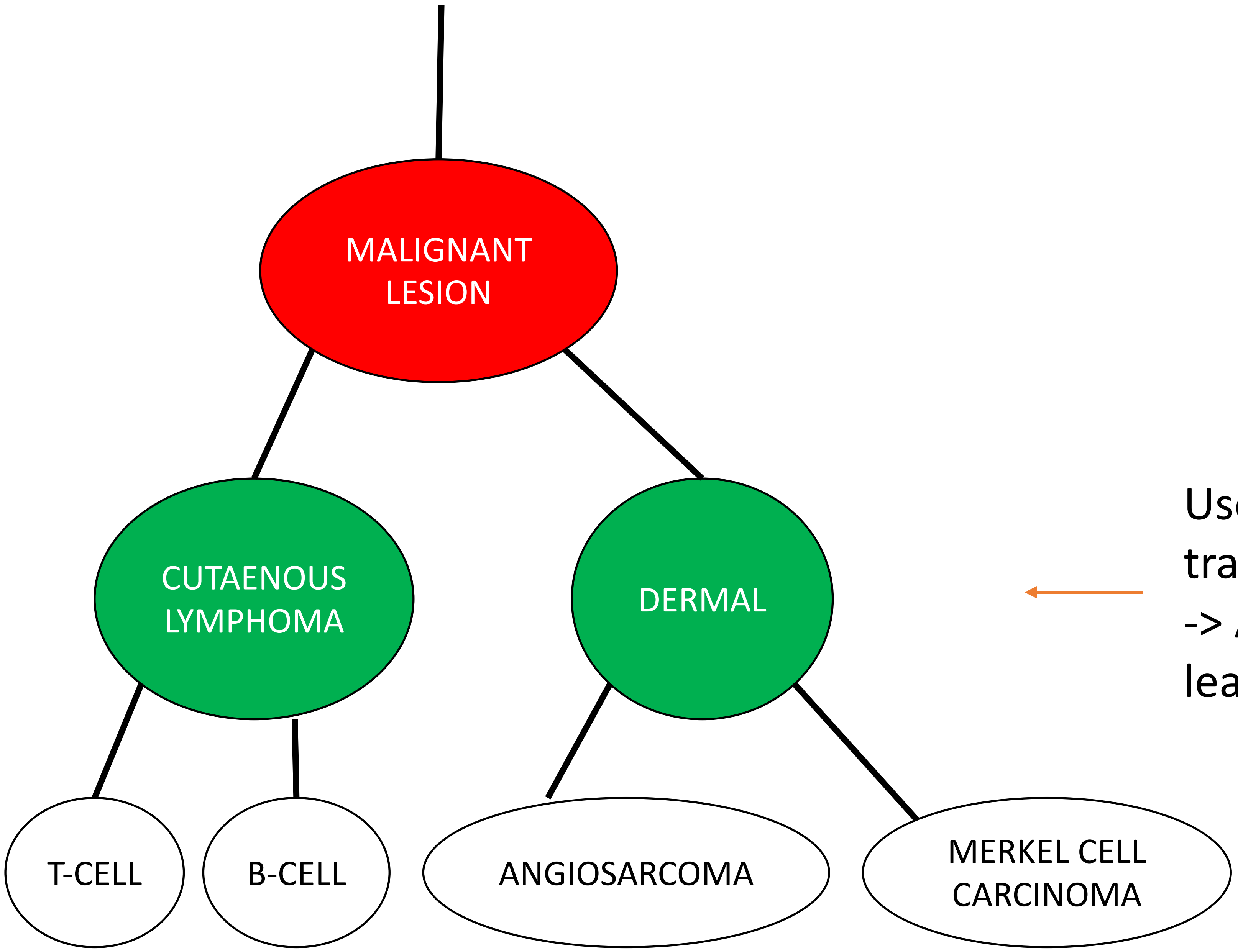
- However, performance was assessed based on higher-level nodes.



Use these as training classes?
-> Too few examples to learn effectively



Use these as training classes?
-> Too much variability in appearance

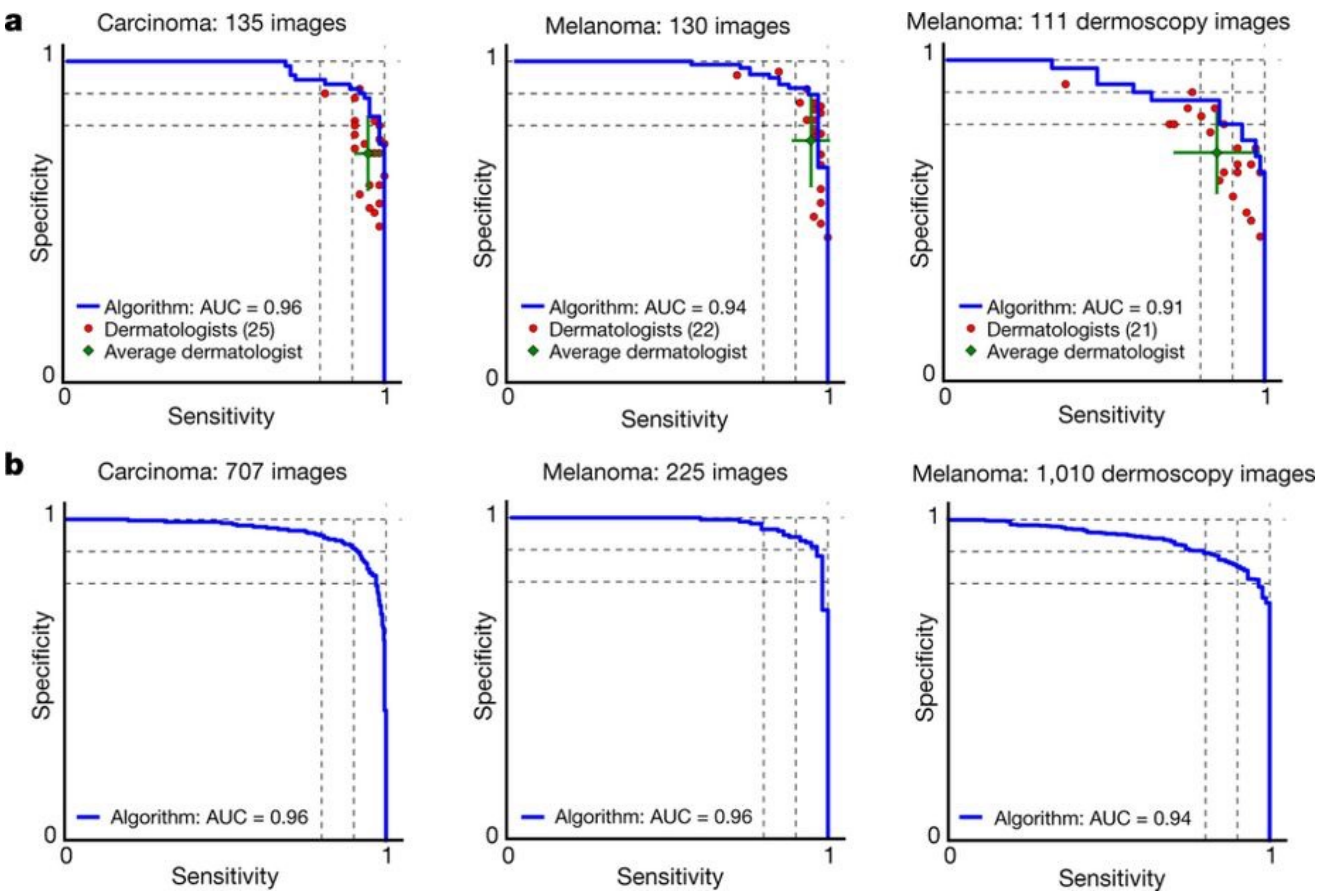


Use these as training classes.
-> Allows effective learning

Classification Results

Interpreting the ROC Curve

Results: CNN Performance vs Dermatologists

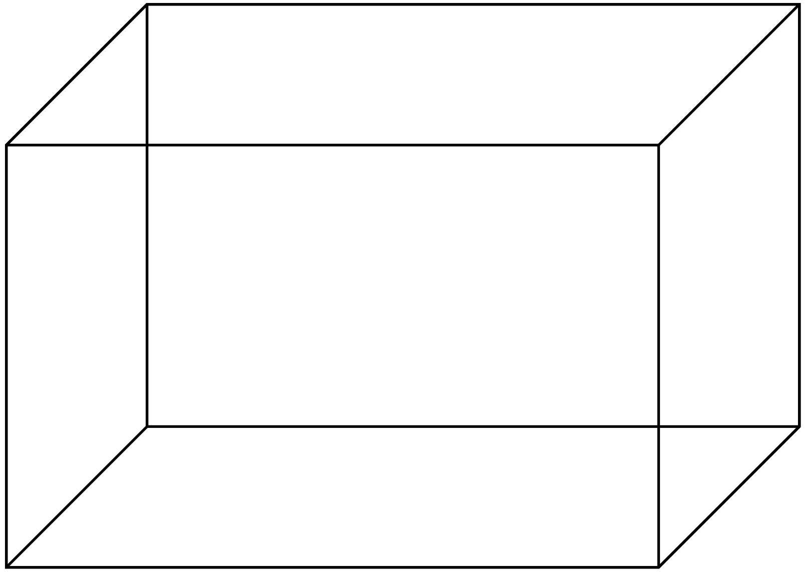


Model Interpretation

How do the authors attempt to look inside the “black box”?

Two competing perspectives

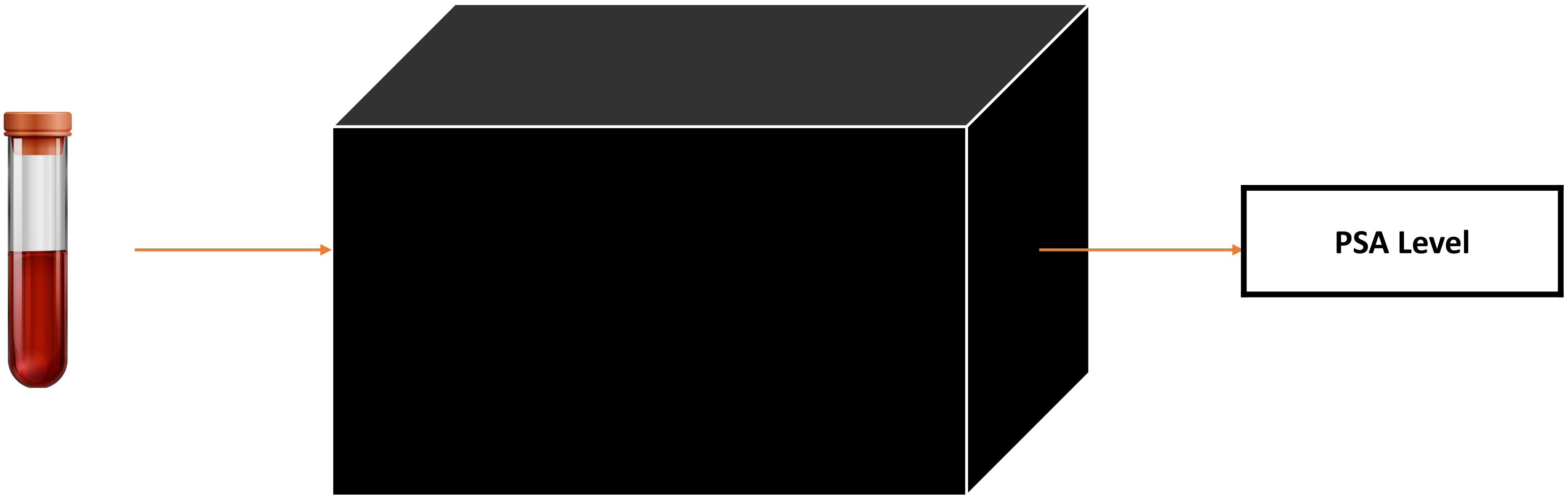
Clinicians must fully understand how their diagnostic tools work



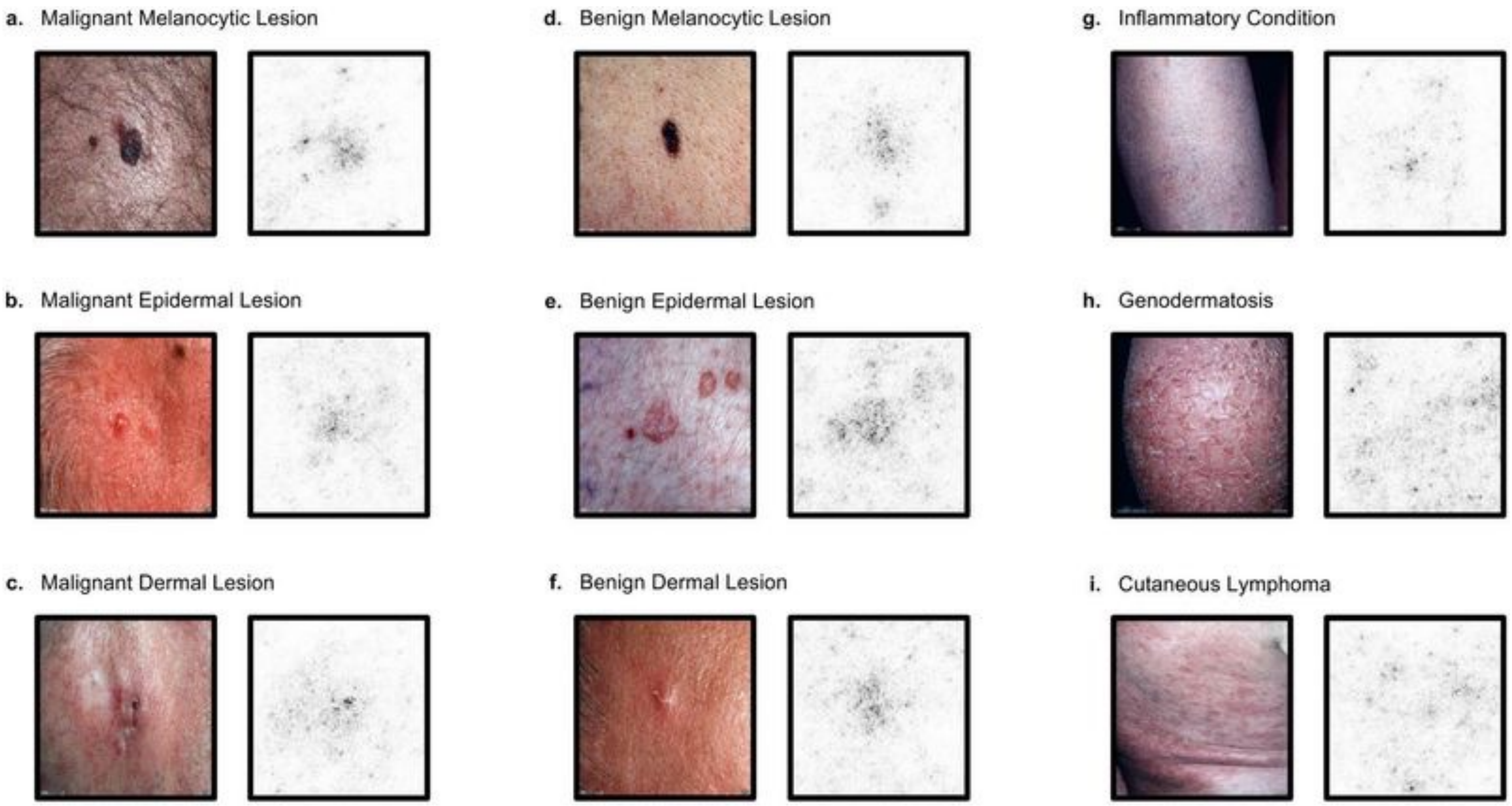
Clinicians must be sure these tools are *valid* and *reliable*

Machine Learning: A Black Box?

Prostate-specific antigen measurement: A Black Box?



Saliency maps for example images



Saliency maps show gradients for each pixel with respect to the CNN's loss function. Darker pixels represent those with more influence.

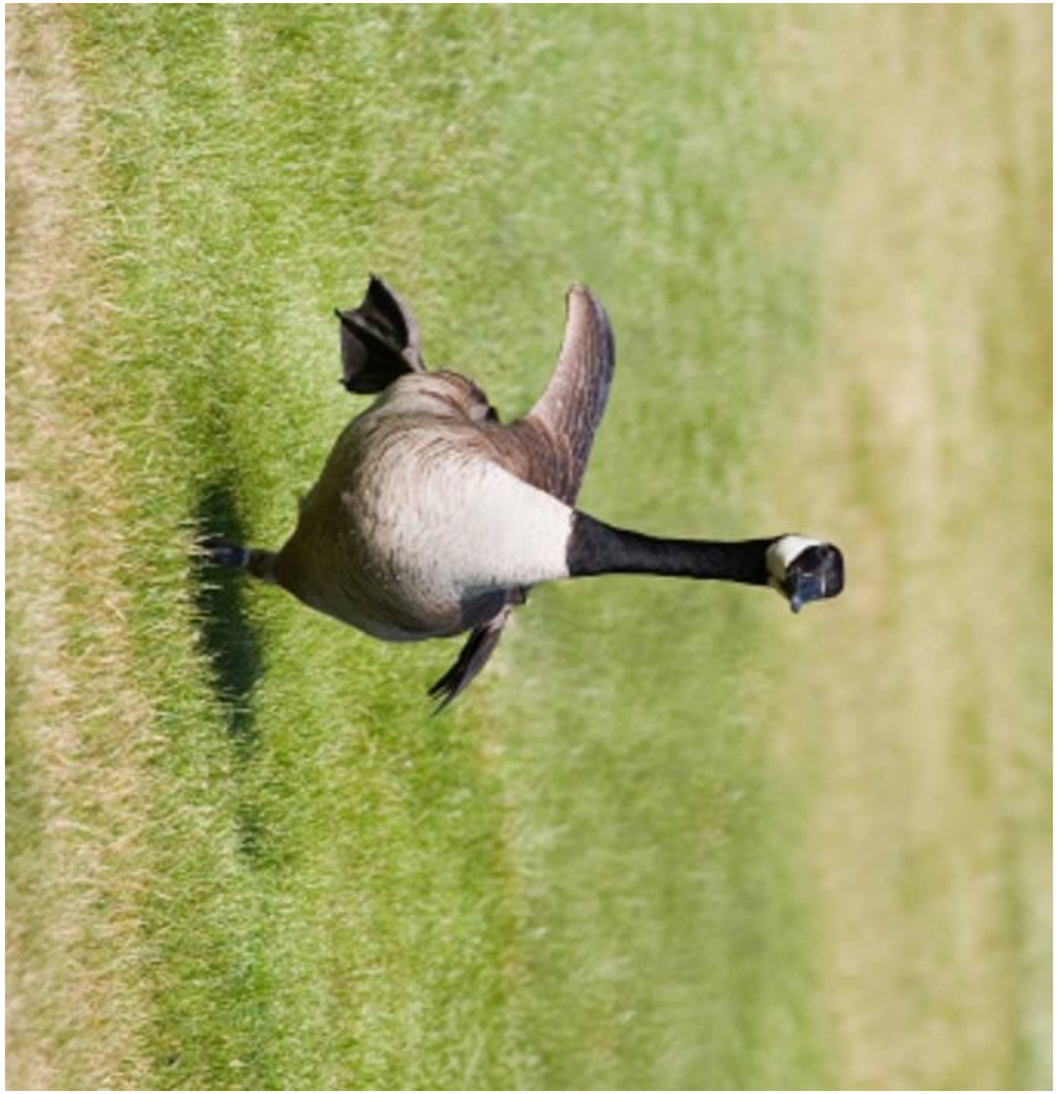
Q: How much does this visualization help us understand the model?

Data Preprocessing / Augmentation

Sometimes the details matter…

I know this one, it’s a goose!

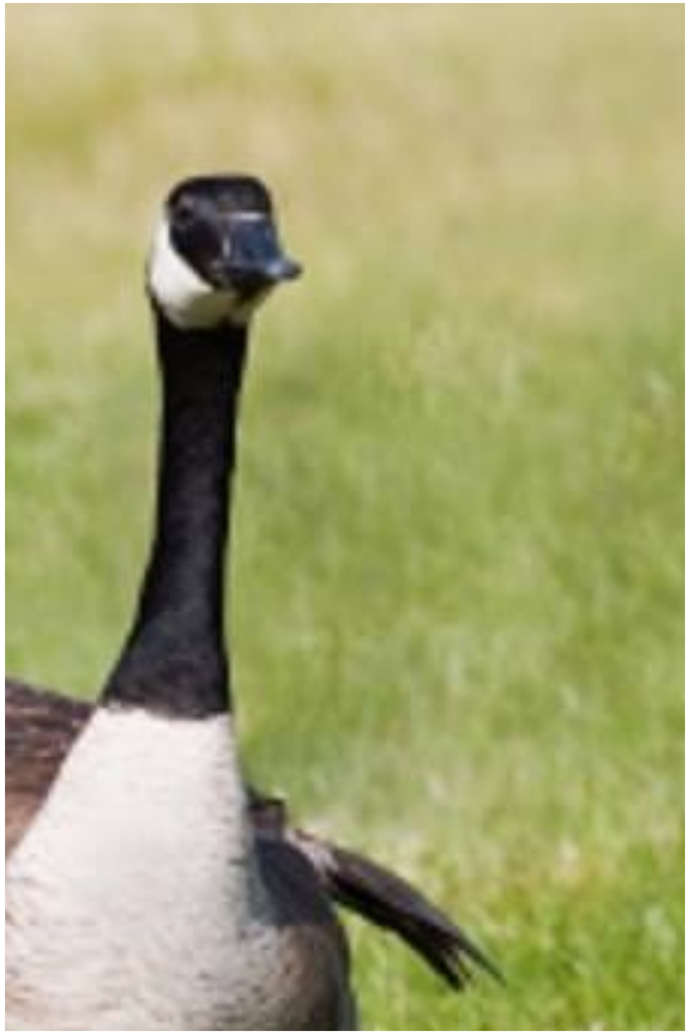
I have no idea what this is.



I’ve never seen anything like it in my life.

I don’t see anything.

I’m certain this is a space shuttle.



Solution: Augment your dataset

- The details depend on the application.