# Medical Image Analysis with CNNs

June 18, 2019

Machine Learning Summer School

Matthew Engelhard

Identifying Skin Cancer

# MEDICAL IMAGE CLASSIFICATION

nature
International journal of science

Letter | Published: 25 January 2017

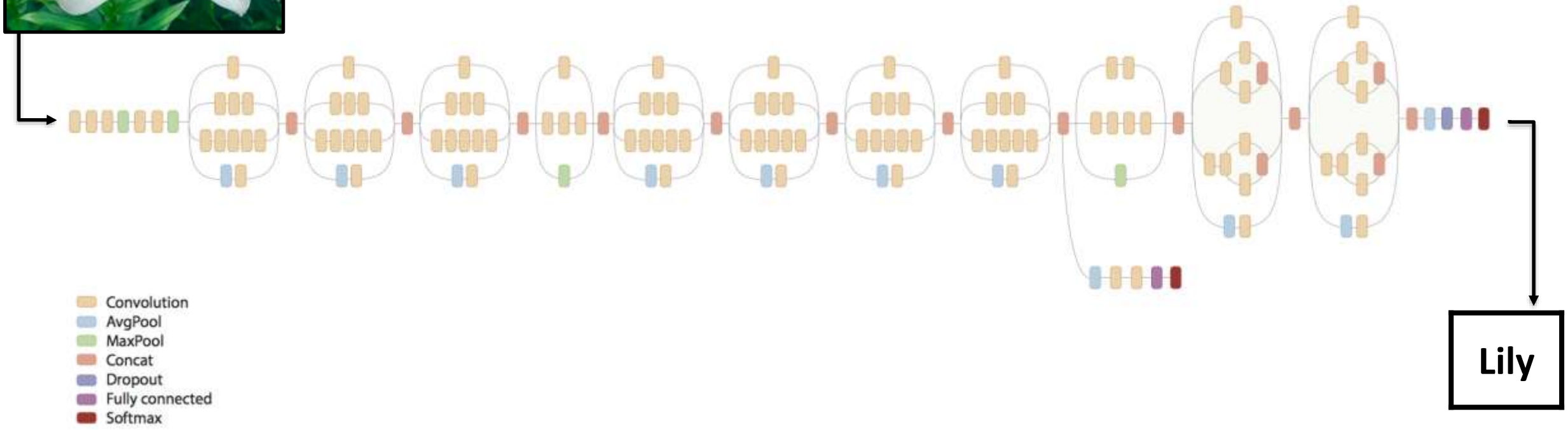
# Dermatologist-level classification of skin cancer with deep neural networks

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

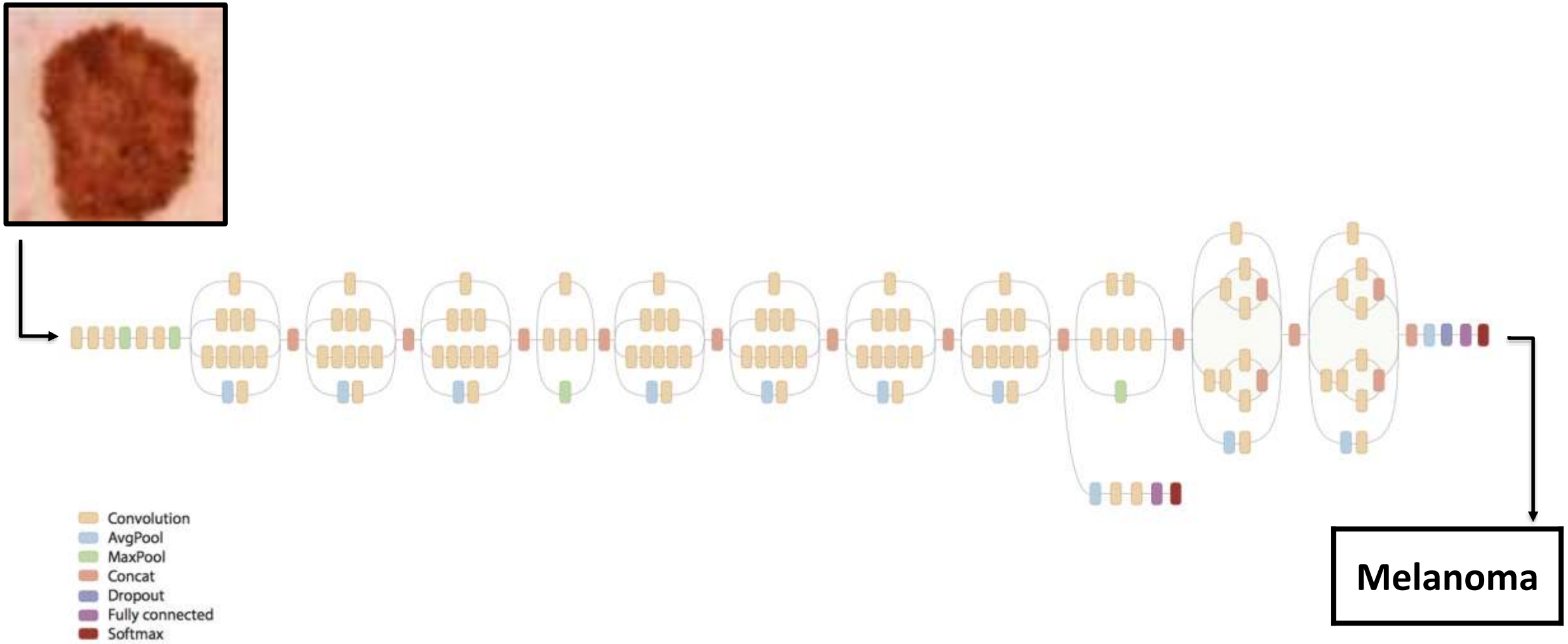
*Nature* **542**, 115–118 (02 February 2017) | Download Citation

# Classification:
# predict the label associated with each image

# Take a model trained on naturalistic images...



# …and repurpose it to evaluate medical images

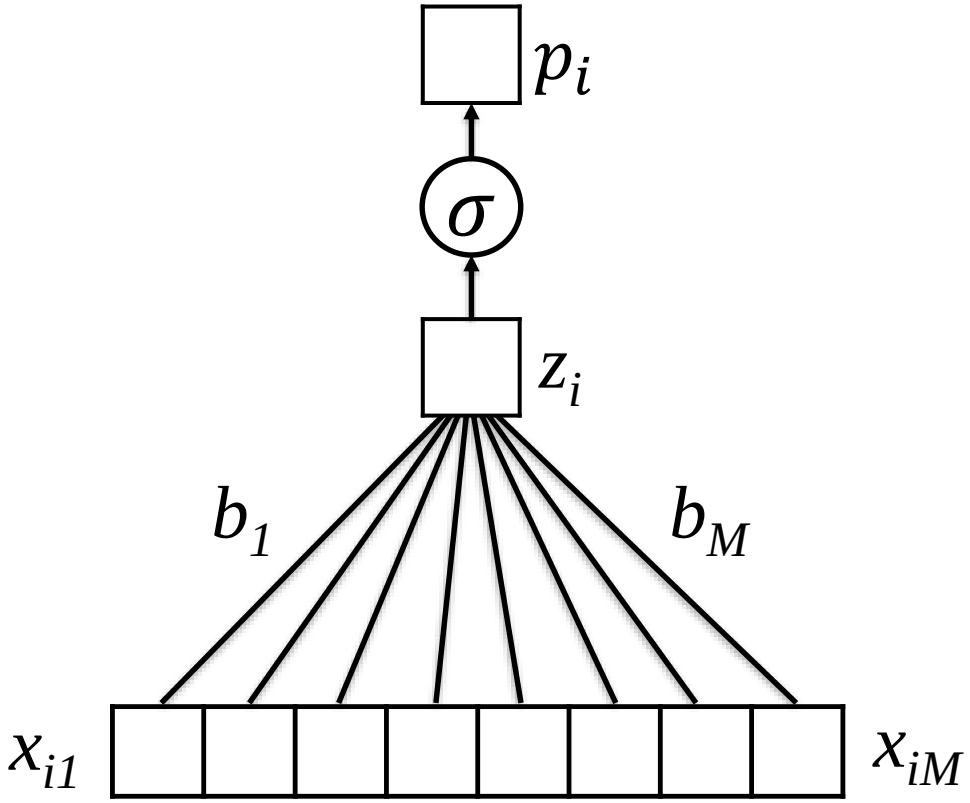


# Classifier Output: Two-Class (e.g. Yes/No)

**Diabetic retinopathy? (Yes/No)**

Gulshan et al. *JAMA* (2016)

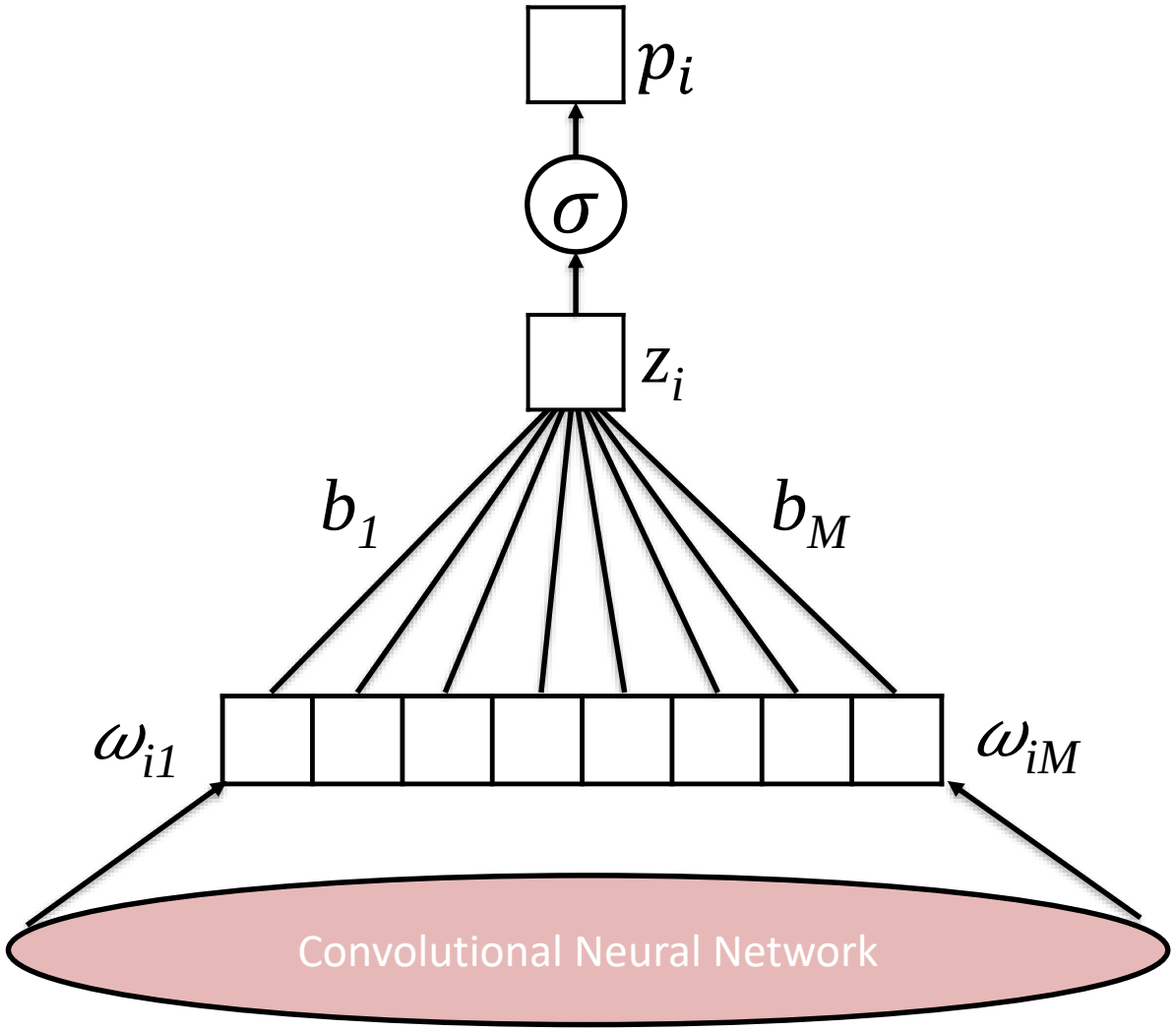
# Two-Class Predictions




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

In logistic regression, $x_i$ is a vector of predictor variables

# Two-Class Predictions



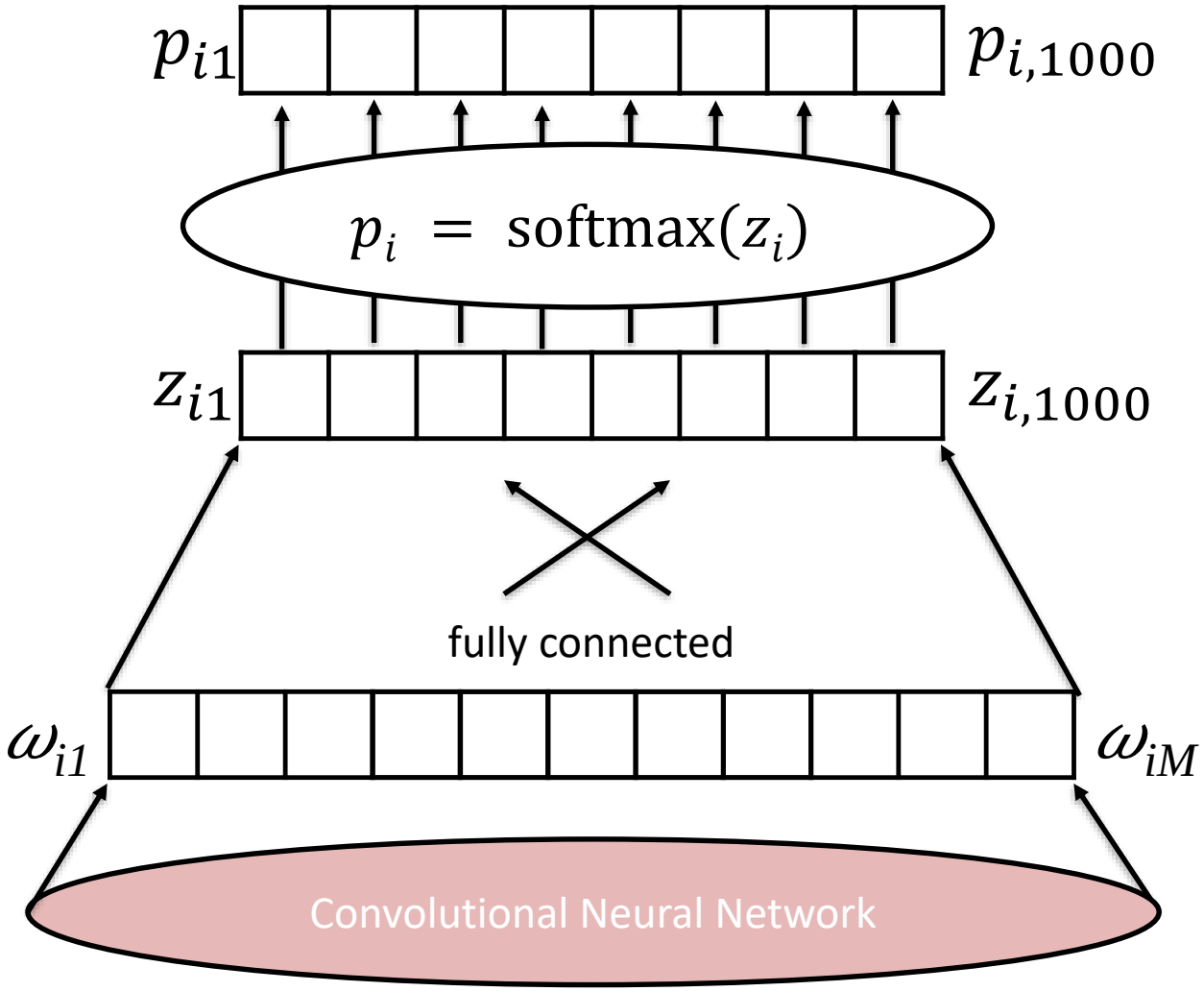

$$\sigma(z_i) = \frac{e^{z_i}}{1 + e^{z_i}}$$

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

# Classifier Output: Multi-Class (ImageNet)

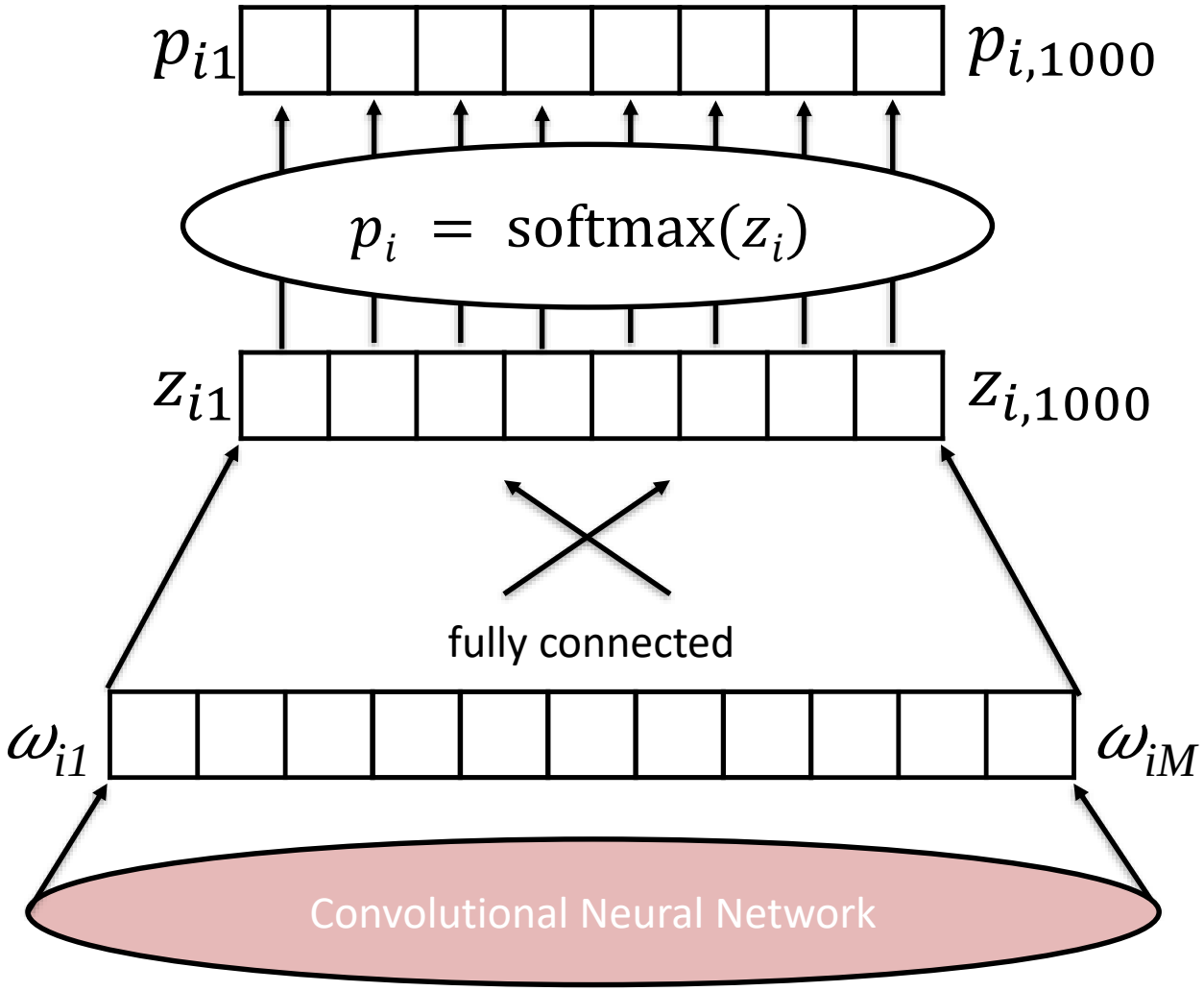

Esteva et al. *Nature* (2017)

# Multi-Class Predictions




$\omega_i$ is a vector of high-level features extracted by the CNN

# Multi-Class Predictions

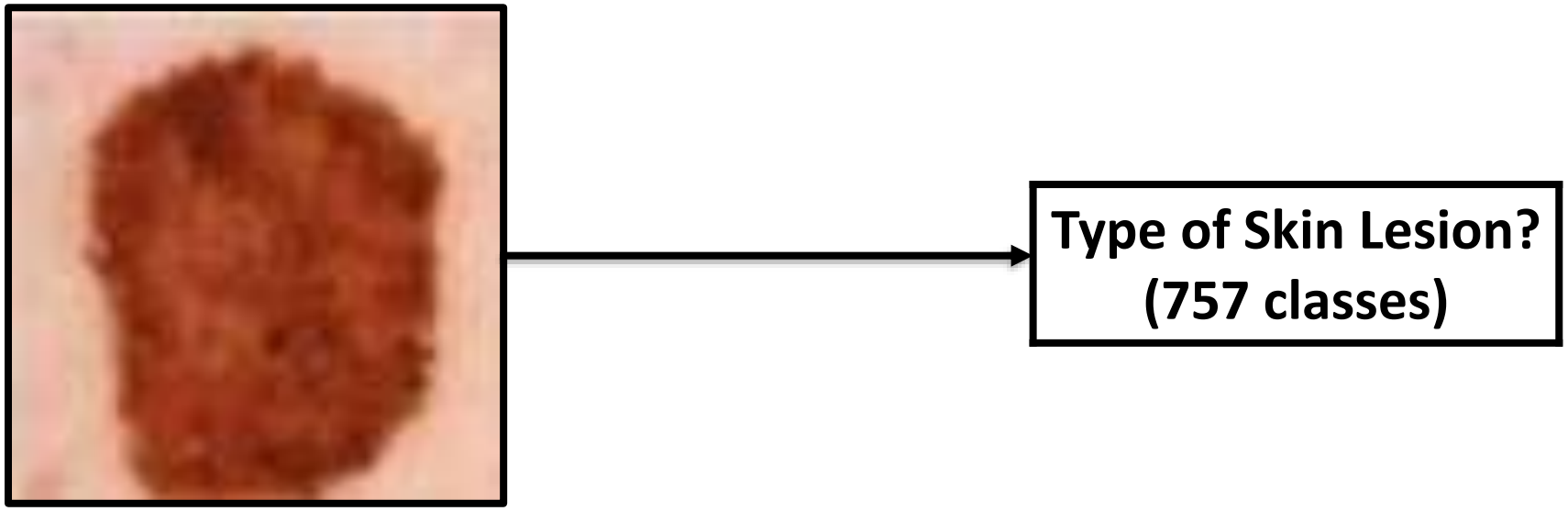



$$p_{ij} = \frac{e^{z_{ij}}}{\sum_{c=1}^{1000} e^{z_{ic}}}$$

$z_i$ are log-odds scores for each class

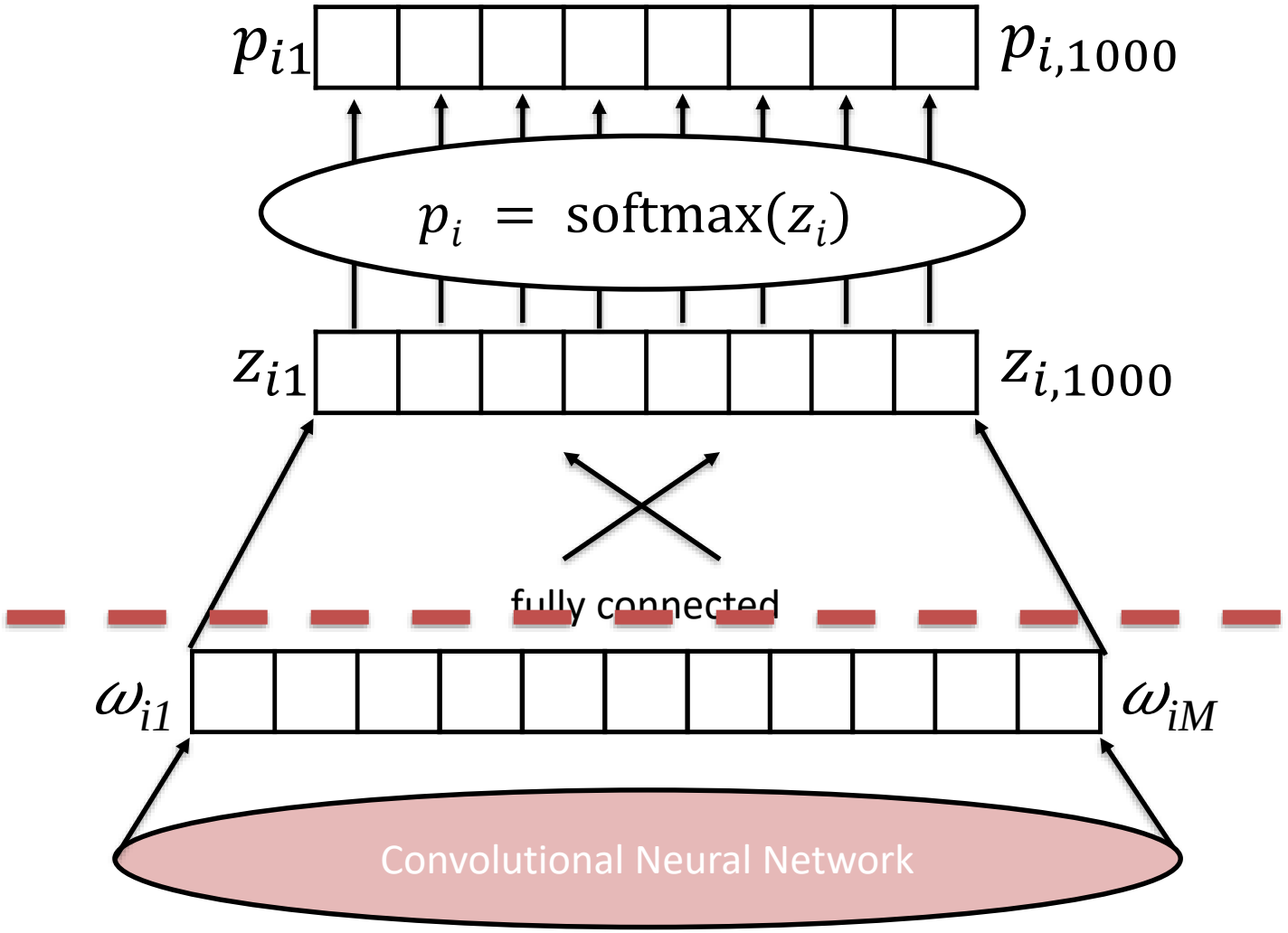
$\omega_i$ is a vector of high-level features extracted by the CNN

# Classifier Output: Multi-Class (Lesion Type)




Esteva et al. *Nature* (2017)

# Step 1: Modify the Architecture




1000 training classes

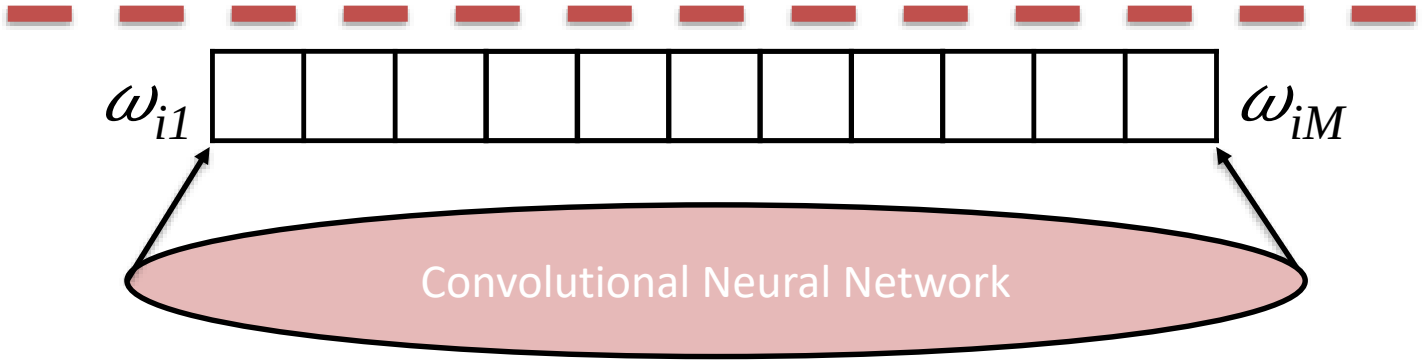
757 training classes

$\omega_i$ is a vector of high-level features extracted by the CNN

# Step 1: Modify the Architecture

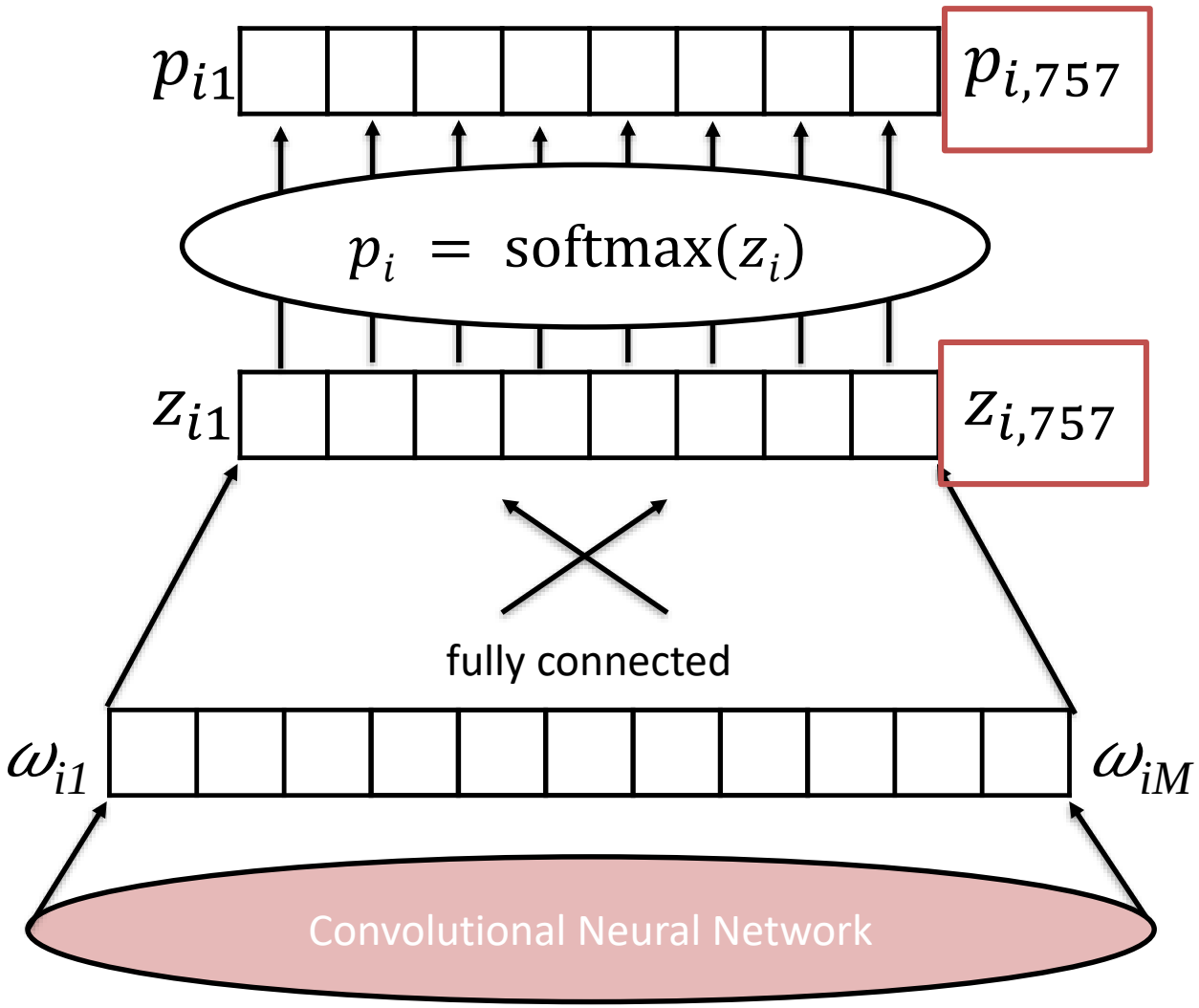
1000 training classes

757 training classes




$\omega_i$ is a vector of high-level features extracted by the CNN

# Step 1: Modify the Architecture

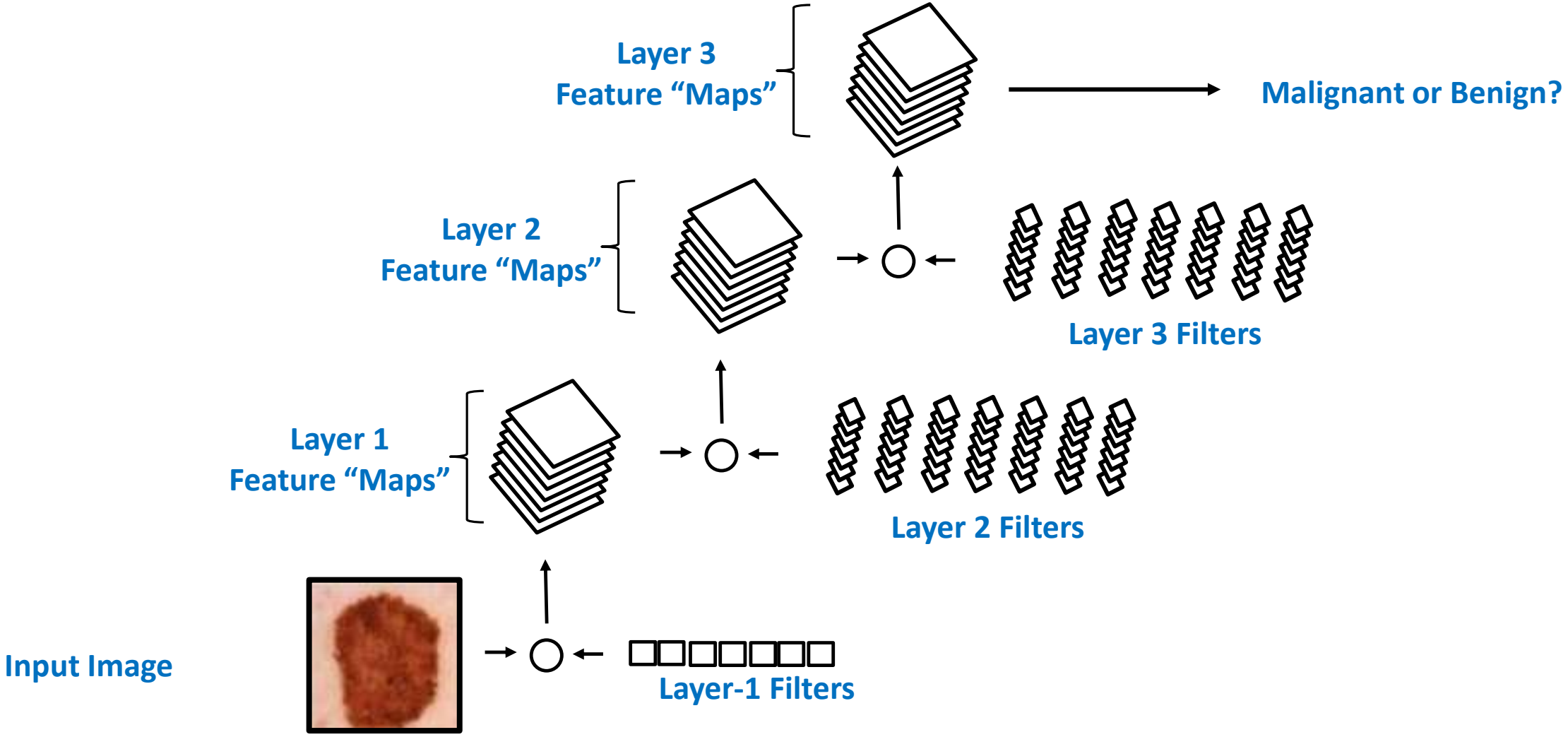



New layers with randomly initialized weights

$\omega_i$ is a vector of high-level features extracted by the CNN

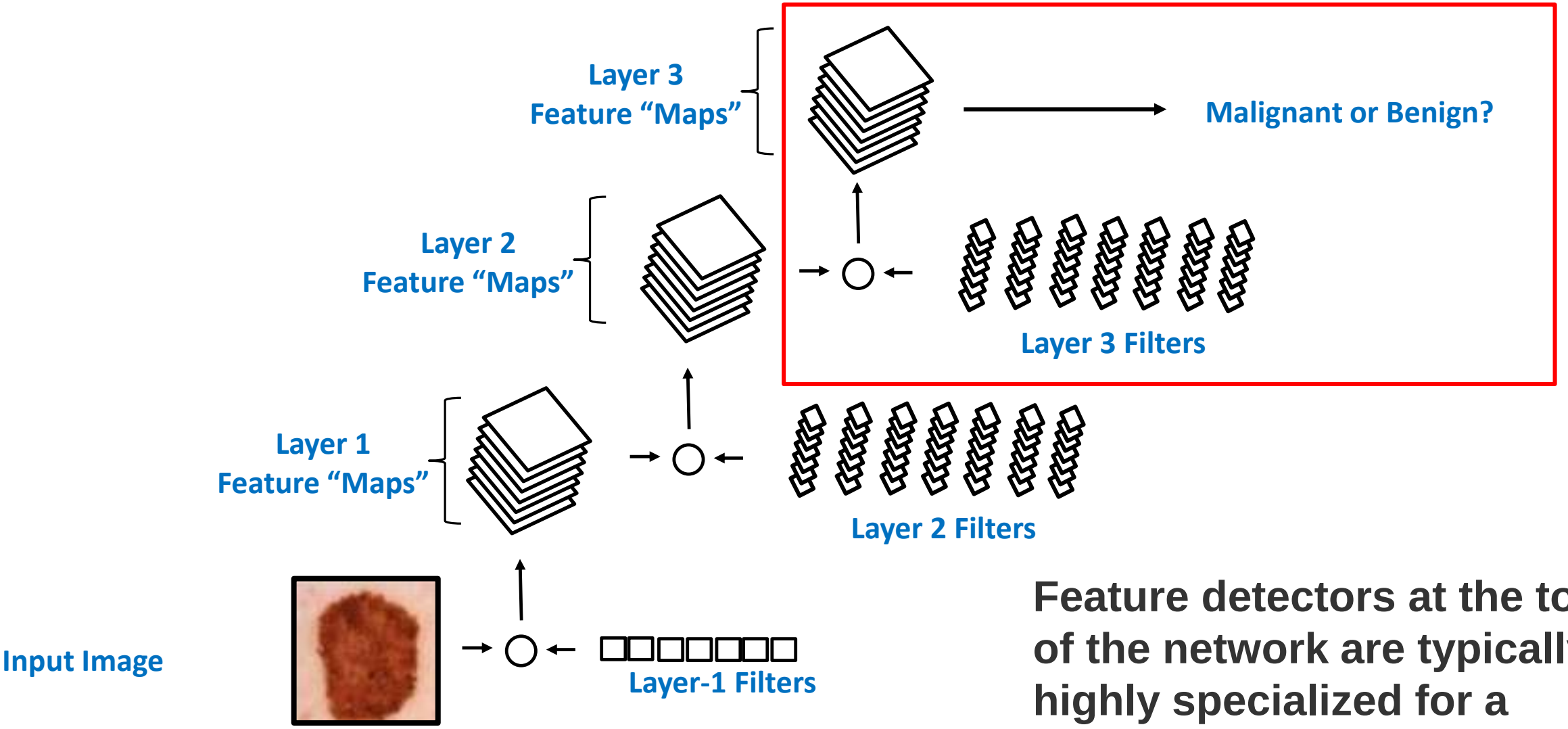
# Step 2: Fine-tune the Parameters

## "pre-training", or "transfer learning"



# Step 2: Fine-tune the Parameters

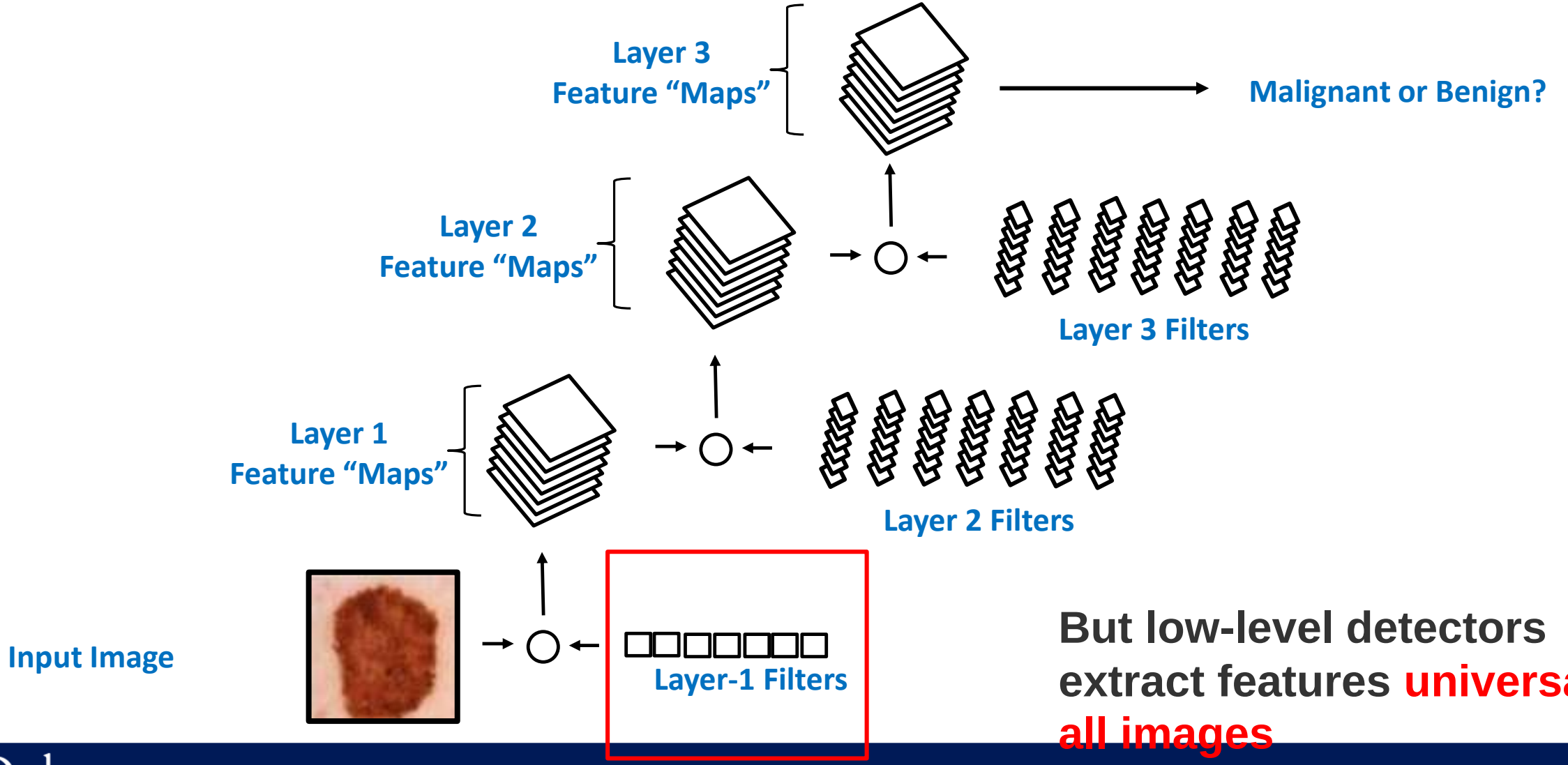
"pre-training", or "transfer learning"




**Feature detectors at the top of the network are typically highly specialized for a particular task**

# Step 2: Fine-tune the Parameters

"pre-training", or "transfer learning"




**But low-level detectors extract features universal to all images**

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

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

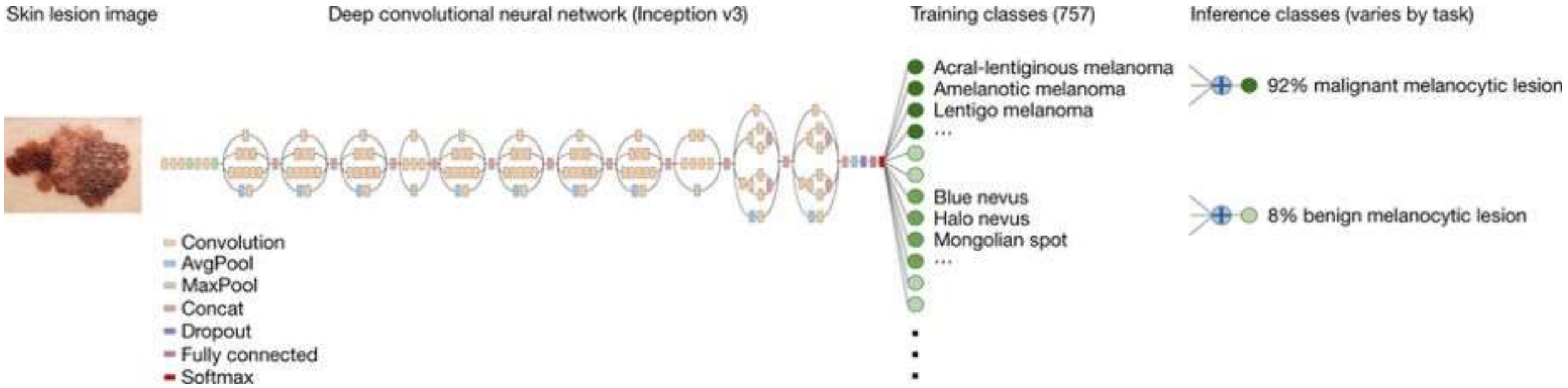
# Pre-training, in brief

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 architecture to match the new number of training classes
- Fine-tune parameters 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

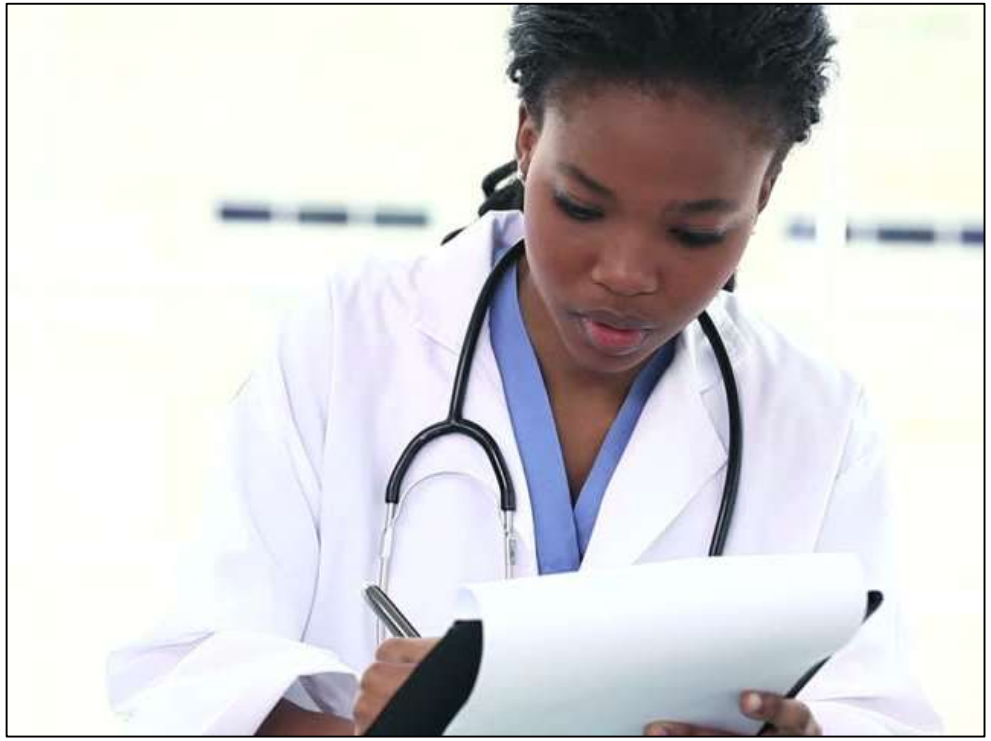
https://github.com/tensorflow/models/tree/master/research/slim#Pretrained

What are the labels?

# “GROUND TRUTH” IN MEDICINE

# Esteva et al: Two Types of Labels

All images: dermatologists' annotations



Some images: biopsy results

# Two Rounds of Evaluation

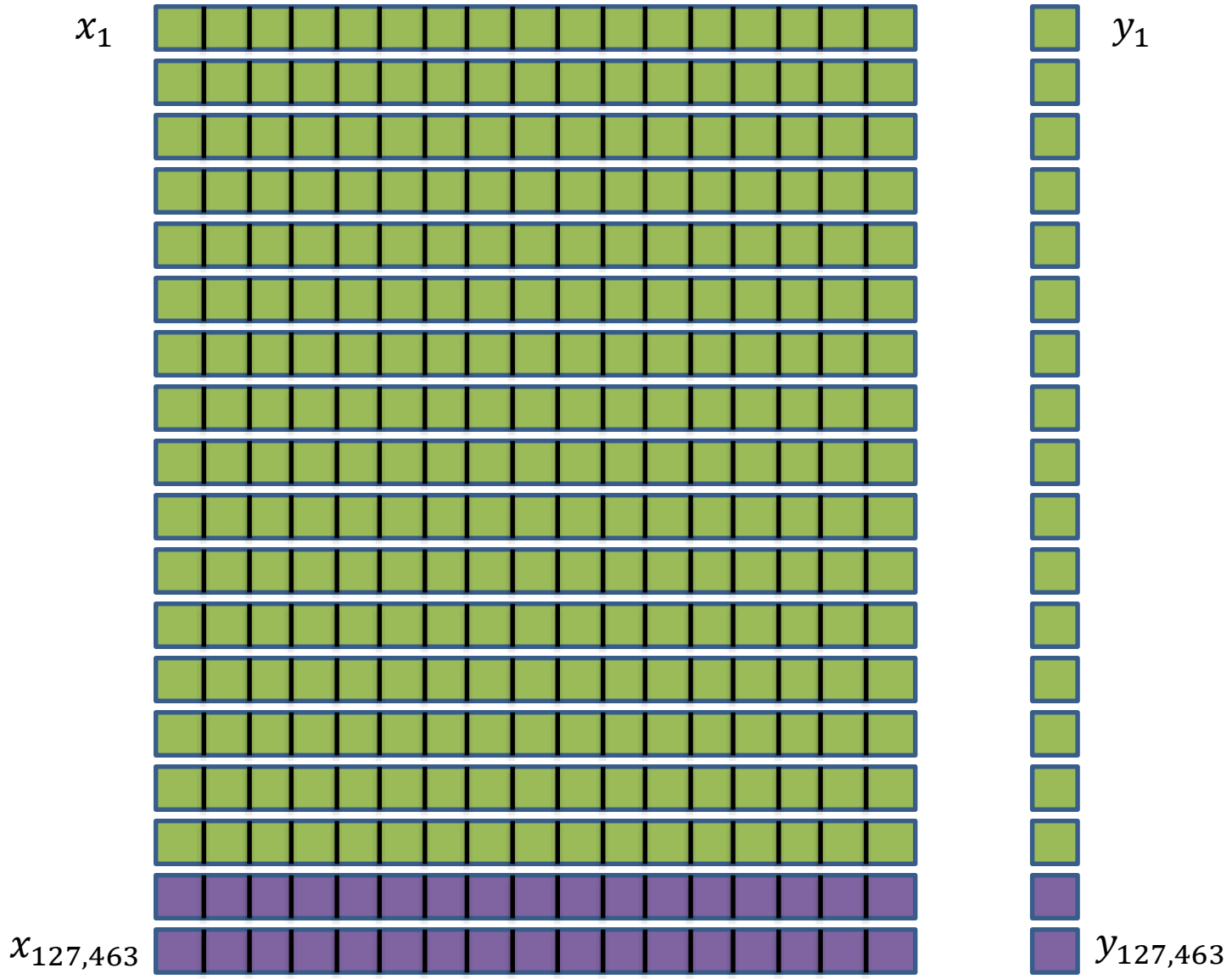
1. Model development: predict dermatologists' annotations:
   - Three-class disease partition
   - Nine-class disease partition

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

# Model Development:

# Predict dermatologists' annotations

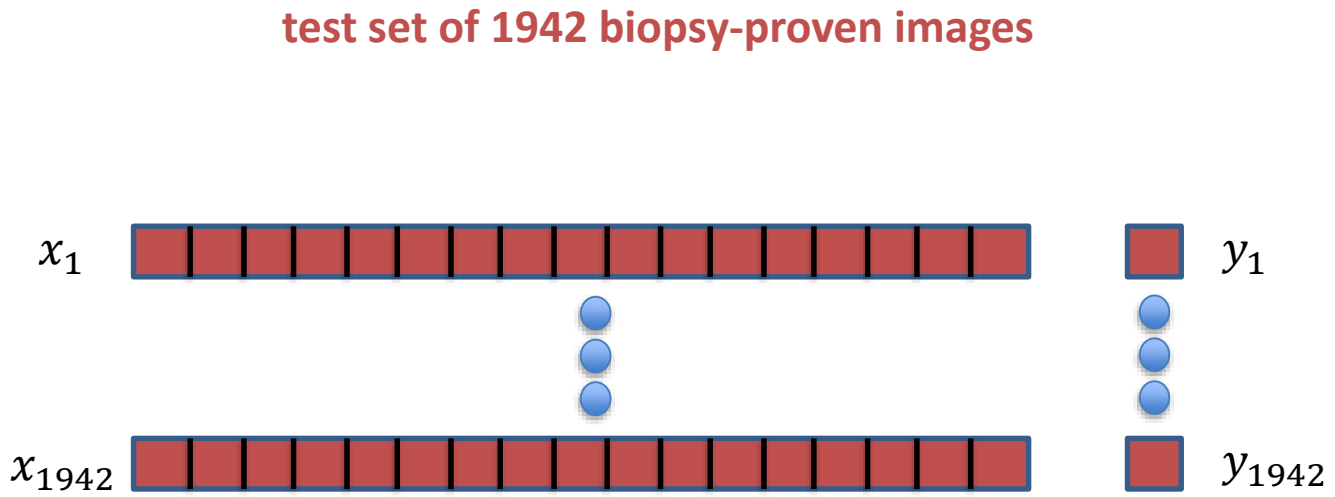
- 9-fold cross-validation
  - 757 training classes derived from dermatologists' annotations
  - 3 and 9-class validation partitions
  - two dermatologists



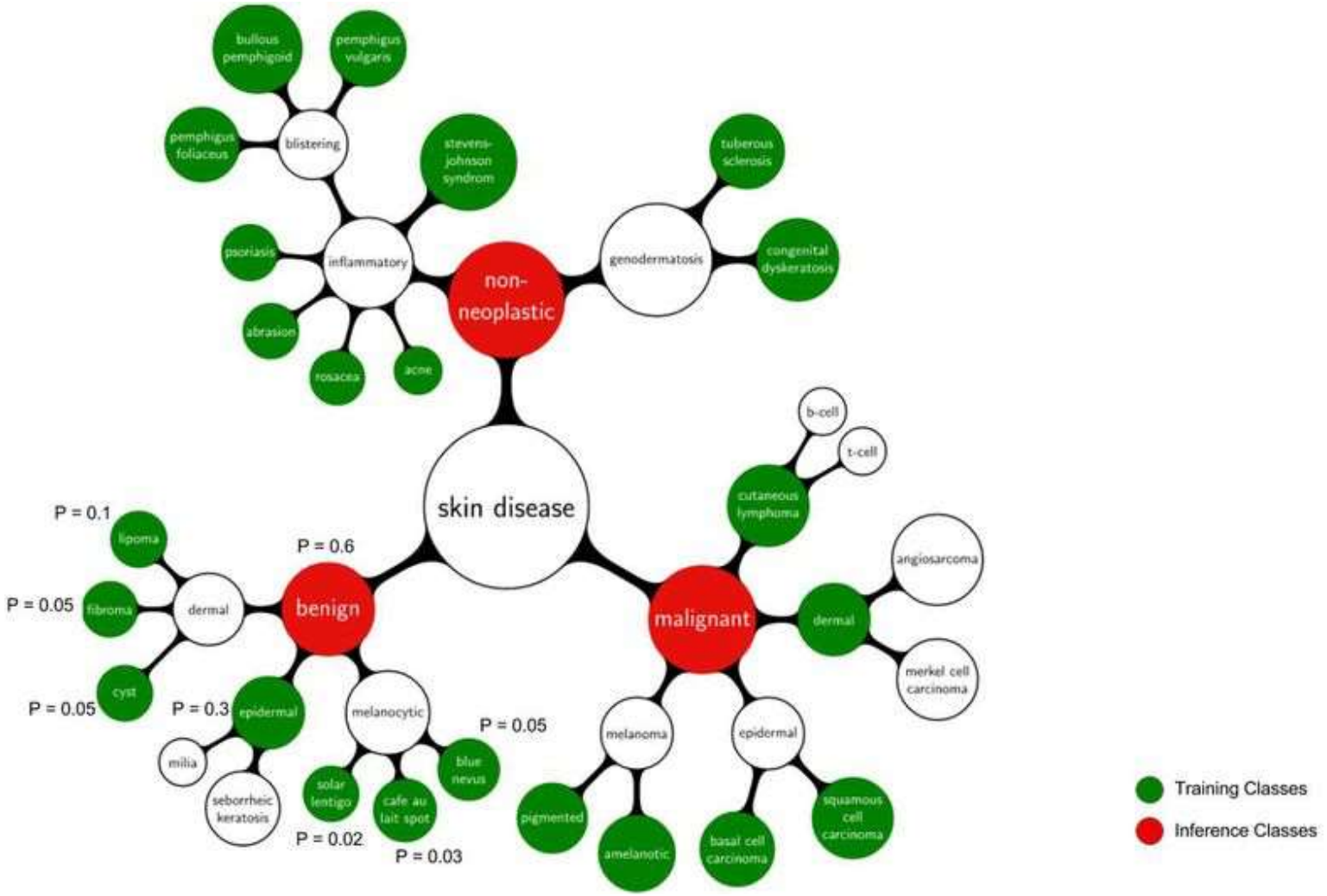

**training set** **validation set**

# Model Evaluation: Predict Biopsy Result

**Performance of the trained model is compared to 21 dermatologists on a test set of biopsy-proven images**

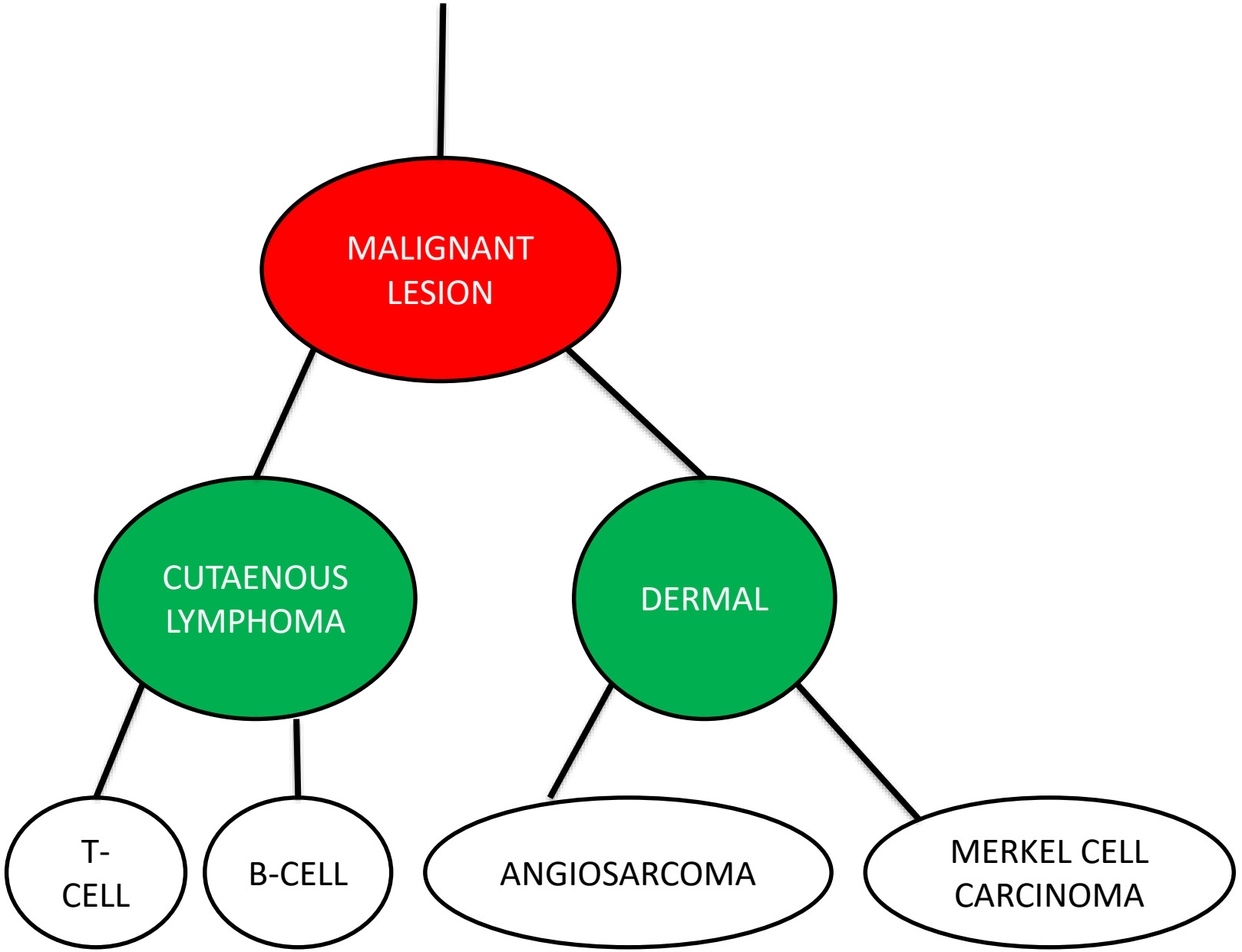


# Specifying training classes based on taxonomy of lesions

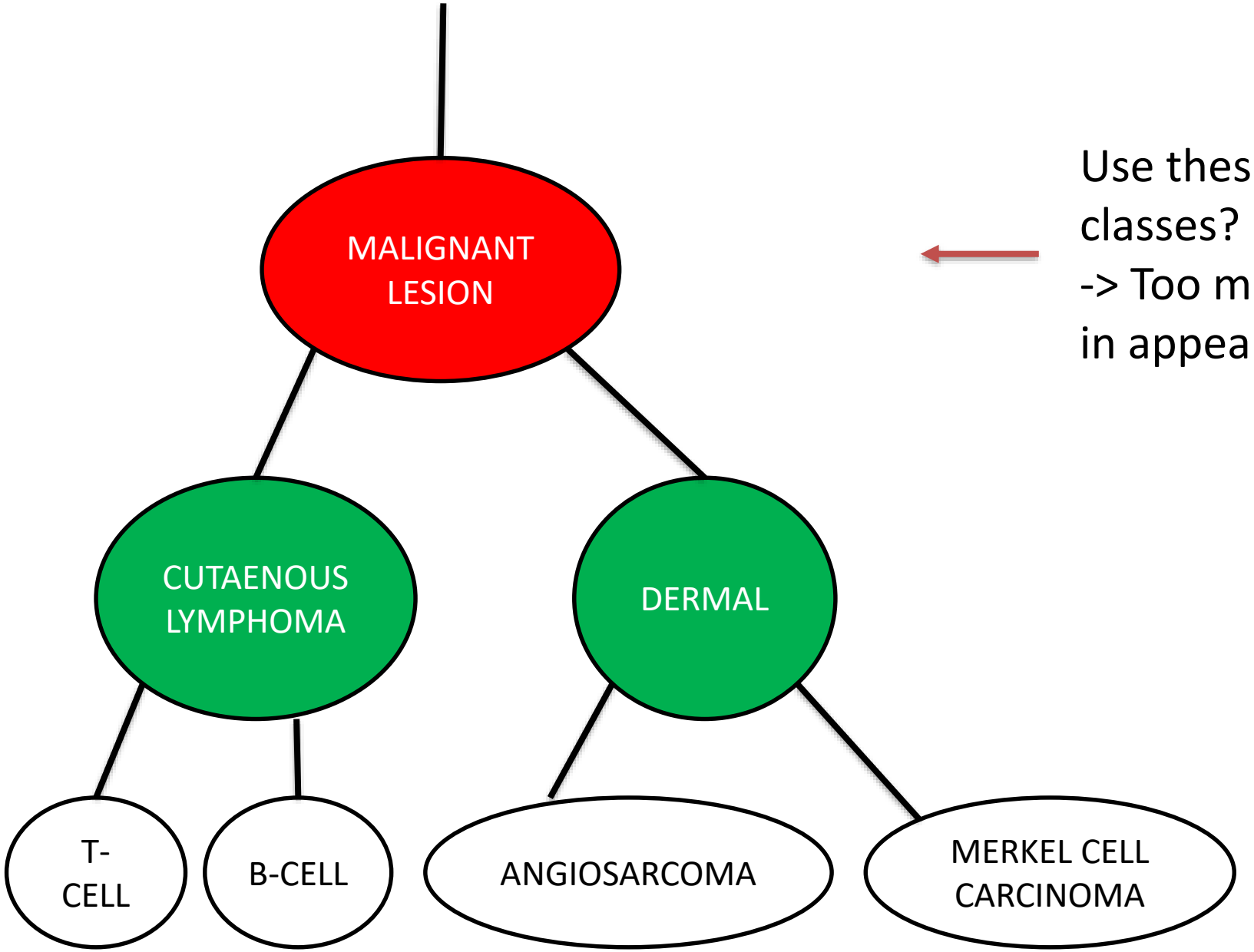



**Disease Partitioning Algorithm:**

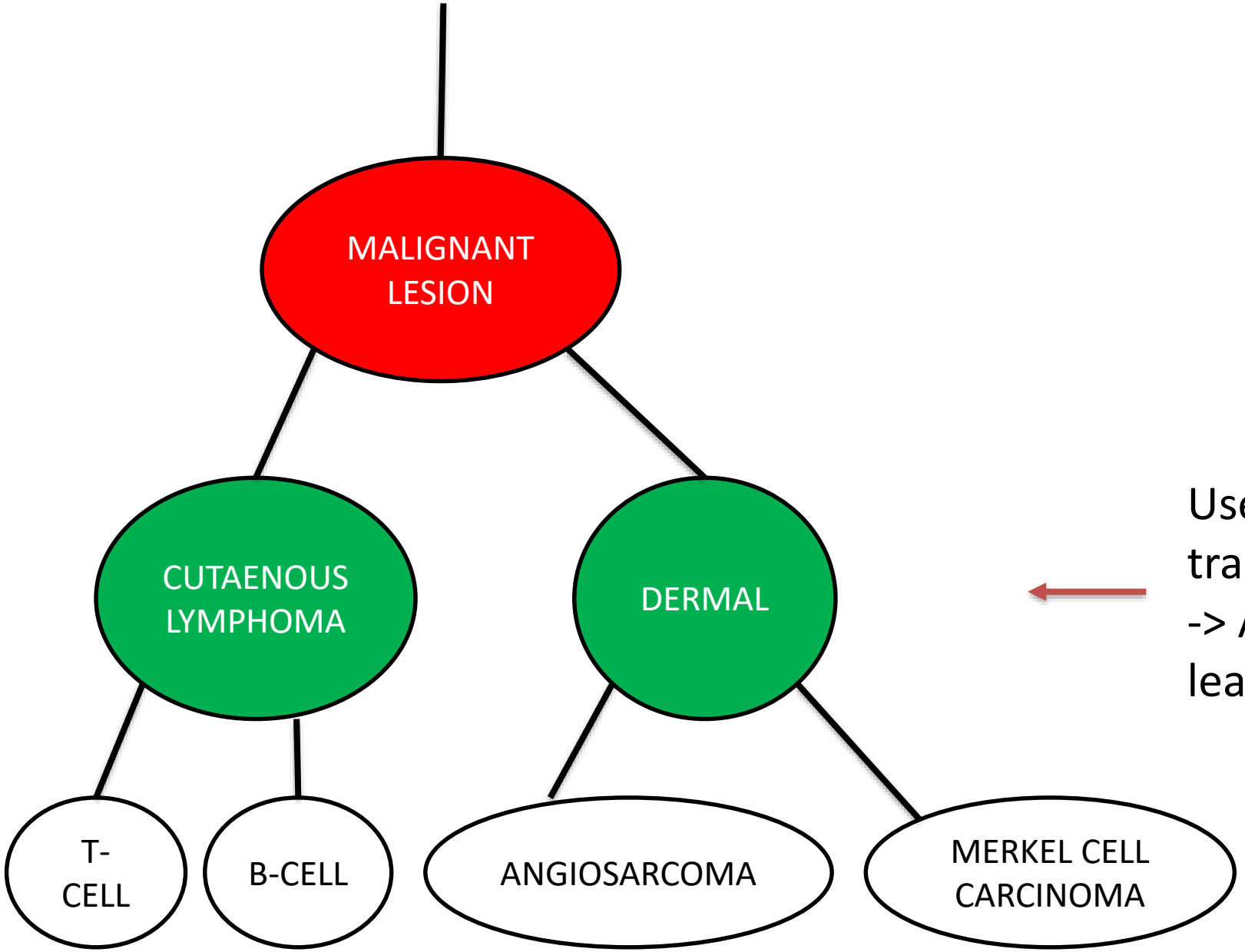
- Ascend the tree until the current node contains <1000 images across all child nodes. Add these images as a distinct training class.
- This resulted in 757 training classes.
- However, performance was assessed based on higher-level nodes.




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




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

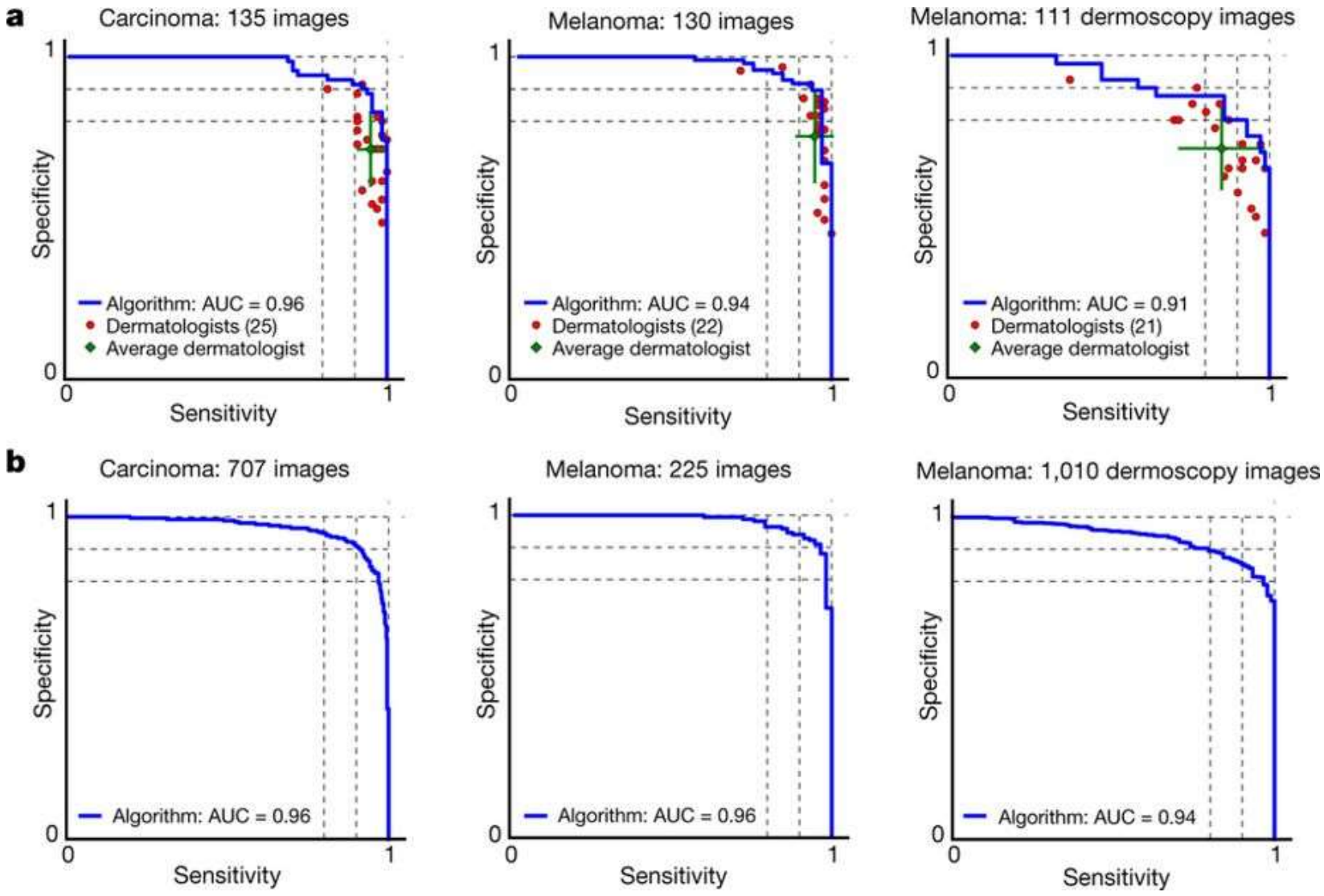



Use these as training classes.
-> Allows effective learning

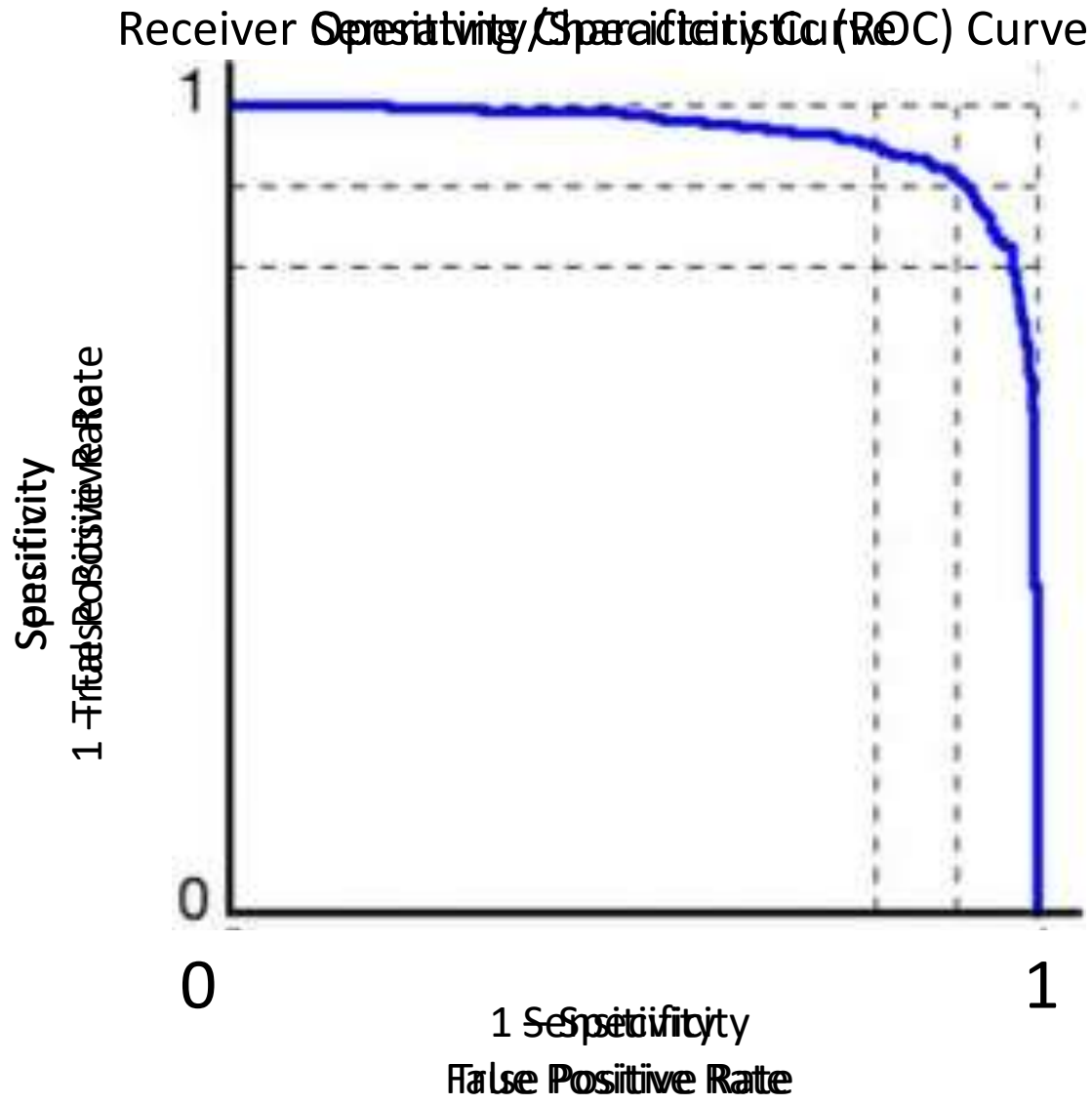
Interpreting the ROC Curve

# CLASSIFICATION RESULTS

# Results: CNN Performance vs Dermatologists



# Evaluation Measures: Classification




Sensitivity, or True Positive Rate:

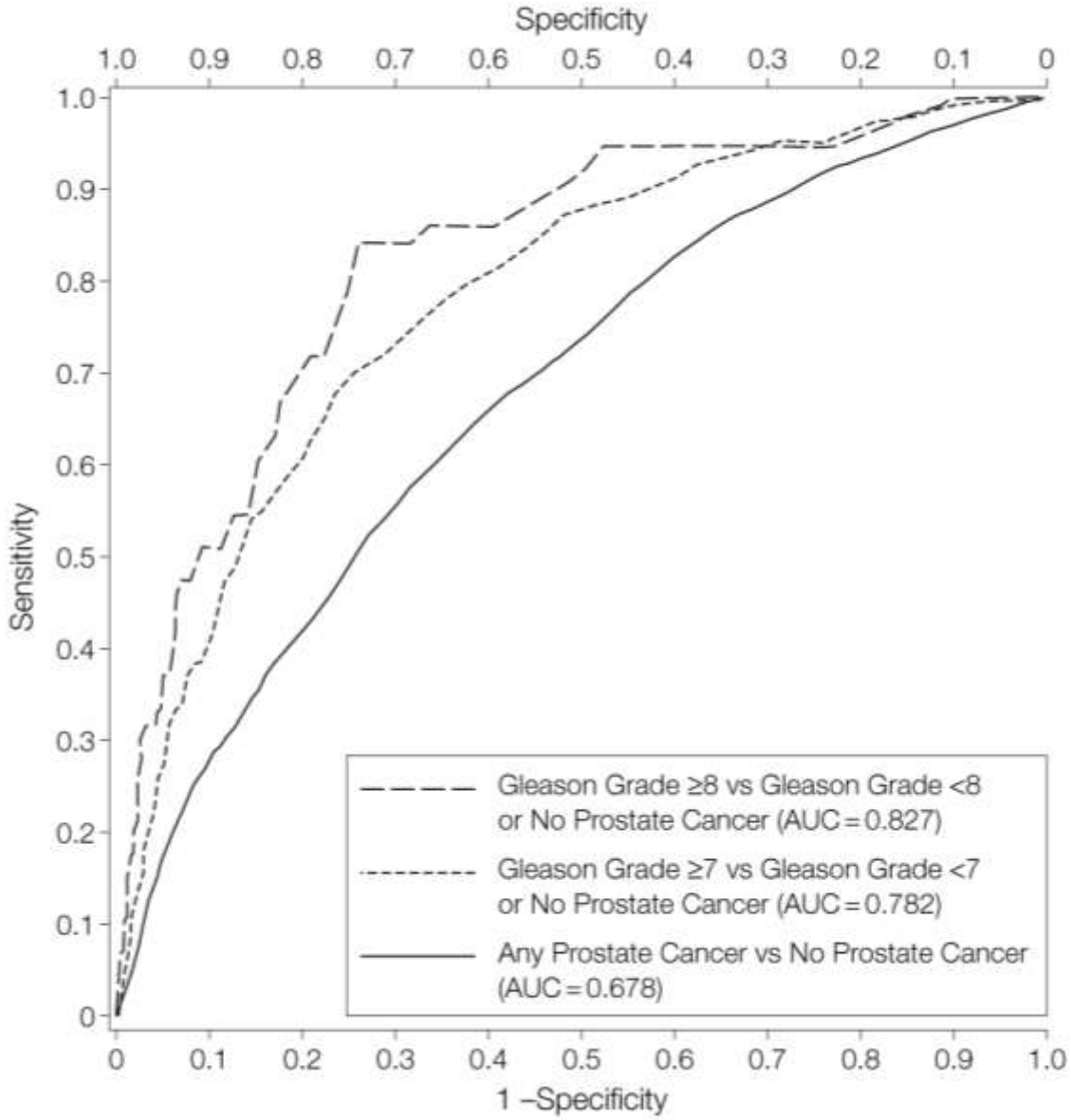
$$\frac{\text{true positives}}{\text{all condition positives}}$$

Specificity, or (1 – False Positive Rate):

$$\frac{\text{true negatives}}{\text{all condition negatives}}$$

Accuracy:

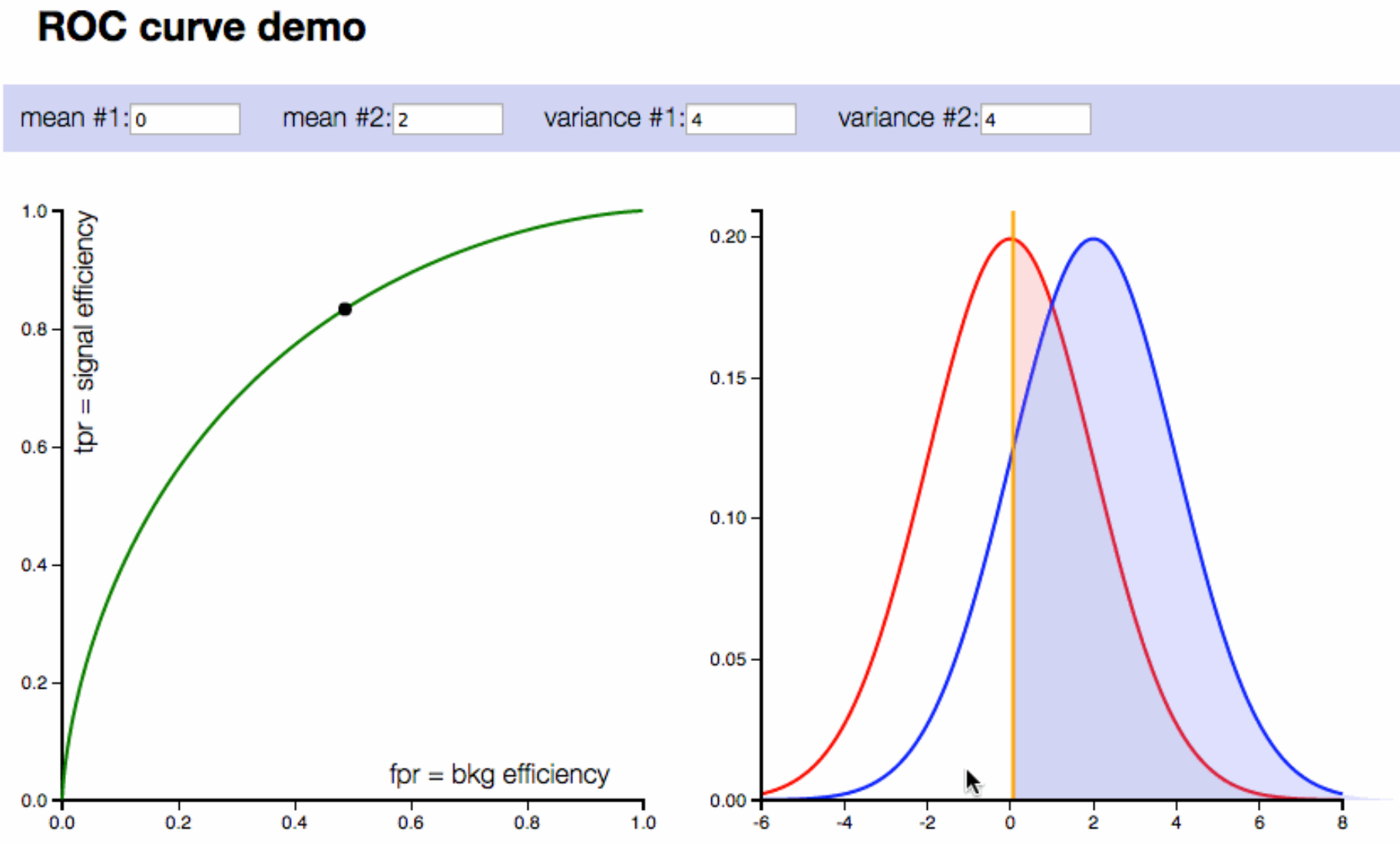
$$\frac{\text{true positives} + \text{true negatives}}{\text{total cases}}$$




# Receiver Operating Characteristic Curve for Prostate-Specific Antigen (PSA)

Thompson IM, Ankerst DP, Chi C, et al. Operating Characteristics of Prostate-Specific Antigen in Men With an Initial PSA Level of 3.0 ng/mL or Lower. *JAMA.* 2005;294(1):66–70. doi:10.1001/jama.294.1.66

# Set a “classification threshold” to distinguish between groups




http://arogozhnikov.github.io/2015/10/05/roc-curve.html

# Once a threshold is set, we get a “confusion matrix”

| | **Condition Positive** | **Condition Negative** |
|---|---|---|
| **Prediction Positive** | True Positive | False Positive |
| **Prediction Negative** | False Negative | True Negative |

Sensitivity, or True Positive Rate:

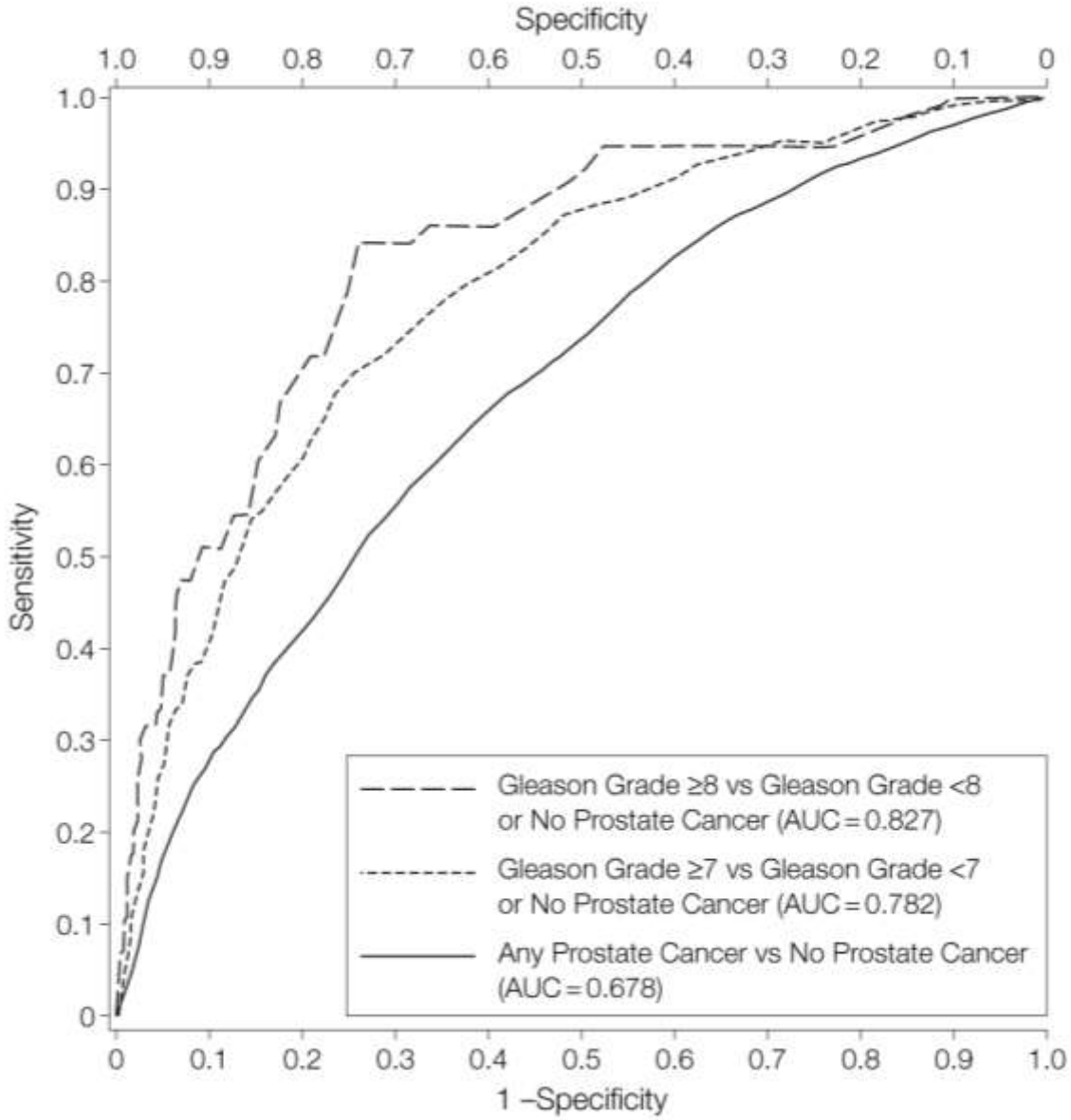
$$\frac{\text{true positives}}{\text{all condition positives}}$$

Specificity, or (1 – False Positive Rate):

$$\frac{\text{true negatives}}{\text{all condition negatives}}$$

Accuracy:

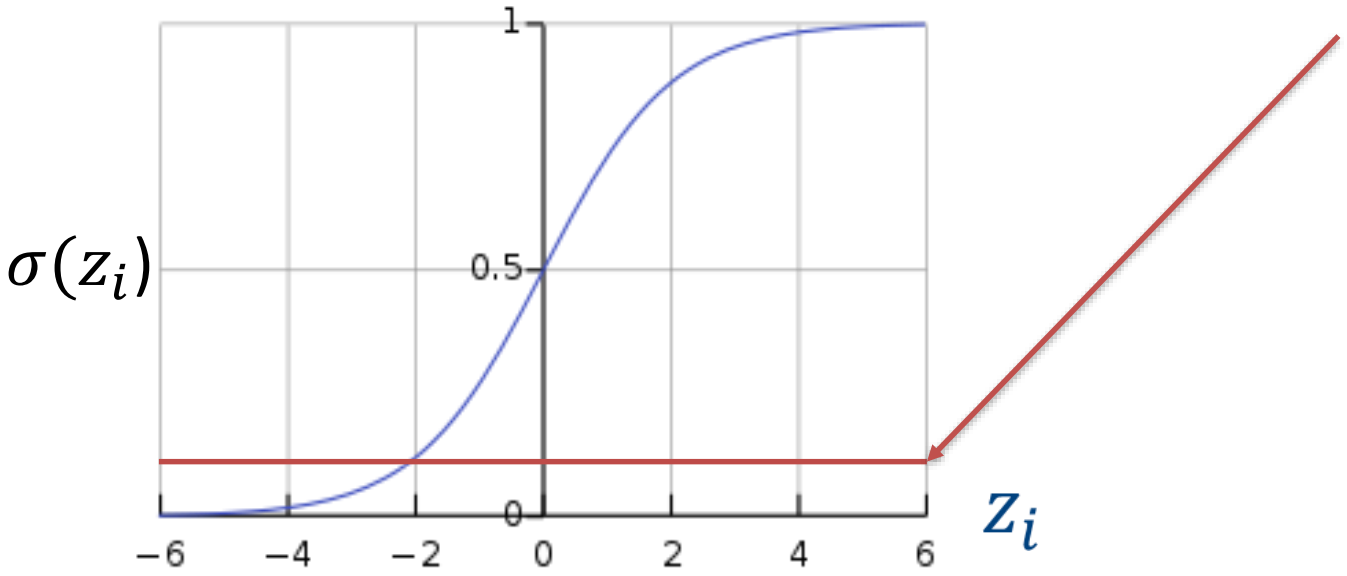
$$\frac{\text{true positives} + \text{true negatives}}{\text{total cases}}$$




1) Set a threshold on PSA
2) Make predictions:
   - Above threshold: cancer-positive
   - Below threshold: cancer-negative
3) Count true positives, true negatives, false positives, and false negatives
4) Calculate sensitivity and specificity
5) Plot point and repeat

# Set a threshold on classifier predictions

$$p(y_i = 1|x_i) = \sigma(z_i)$$

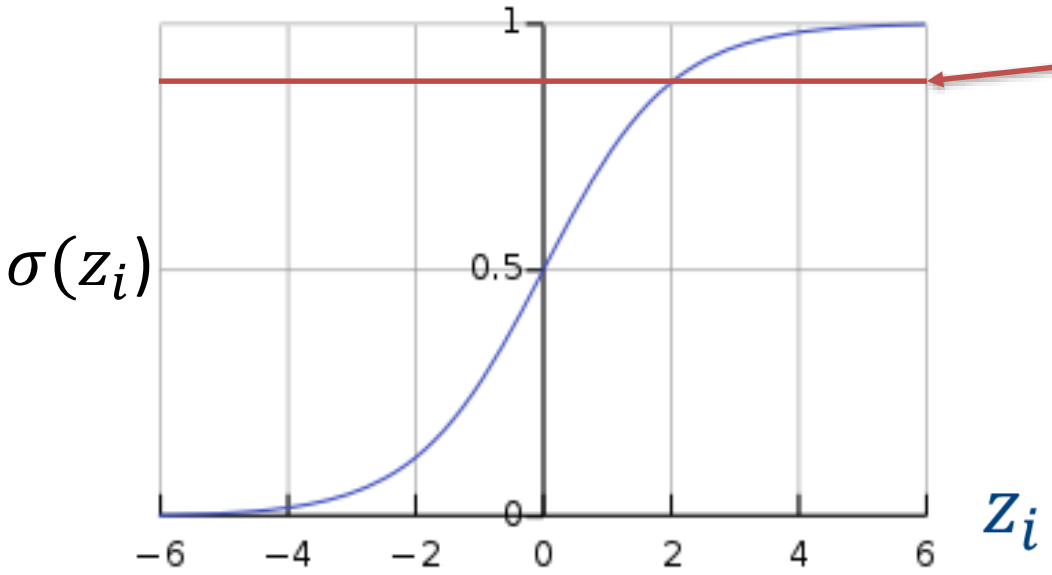



classification threshold

A low threshold favors sensitivity, because more points are predicted to be ones

# Set a threshold on classifier predictions

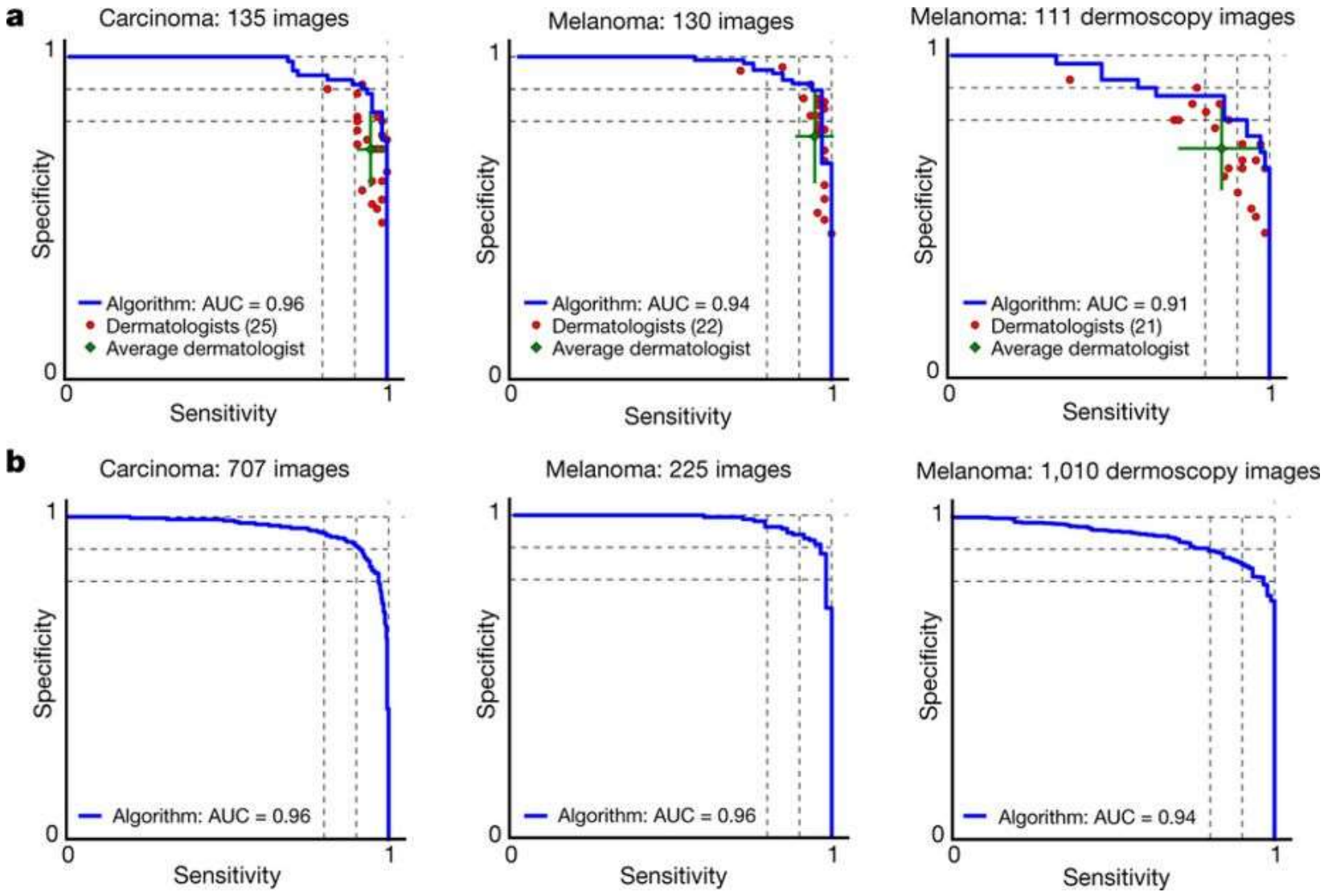
$$p(y_i = 1|x_i) = \sigma(z_i)$$




classification threshold

A high threshold favors specificity, because more points are predicted to be zeros

# Results: CNN Performance vs Dermatologists



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

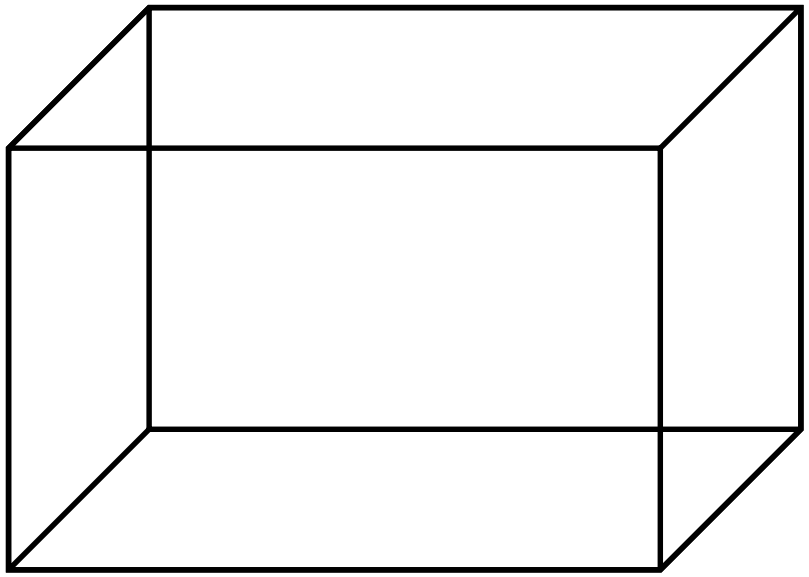
# MODEL INTERPRETATION

# Machine Learning: A Black Box?

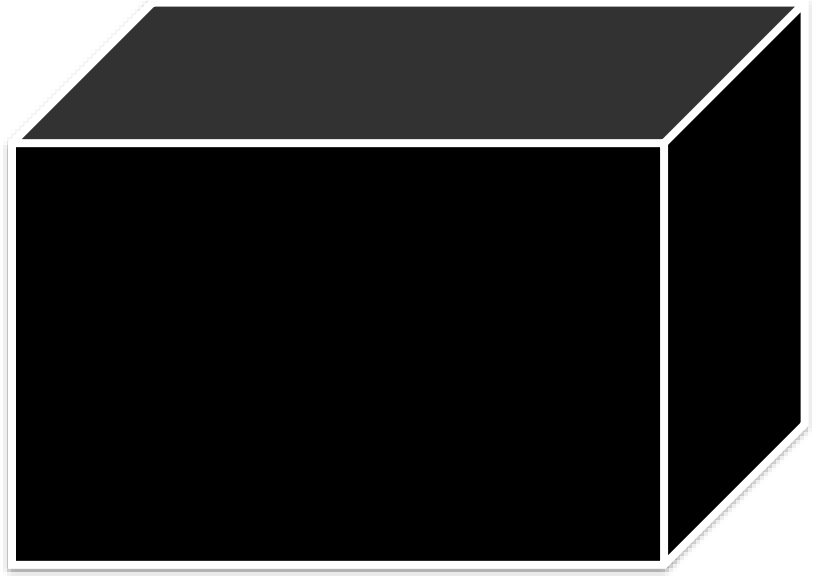
# Prostate-specific antigen measurement: A Black Box?

# Two competing perspectives

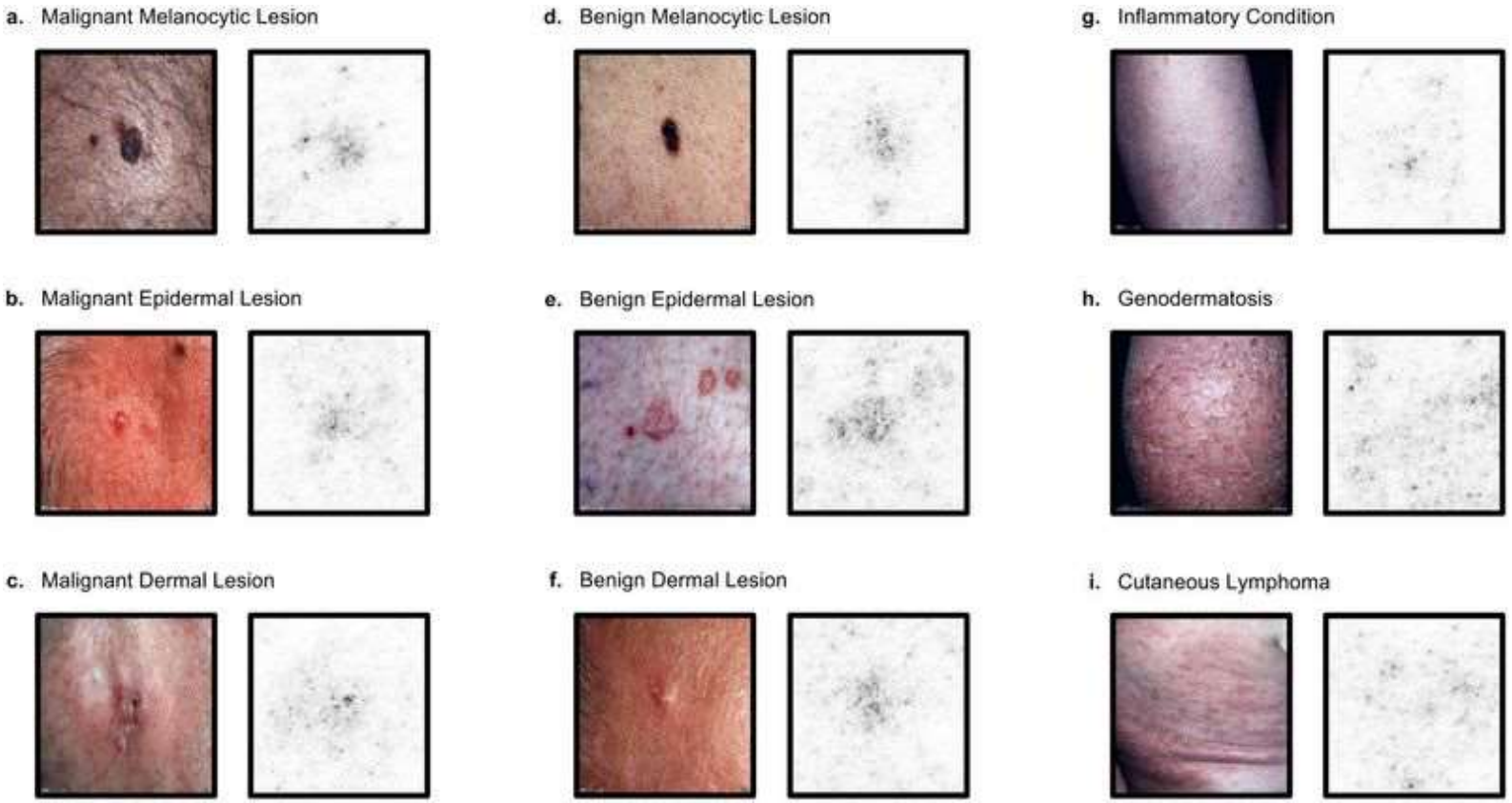
Clinicians must fully understand how their diagnostic tools work



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



# Saliency maps for example images



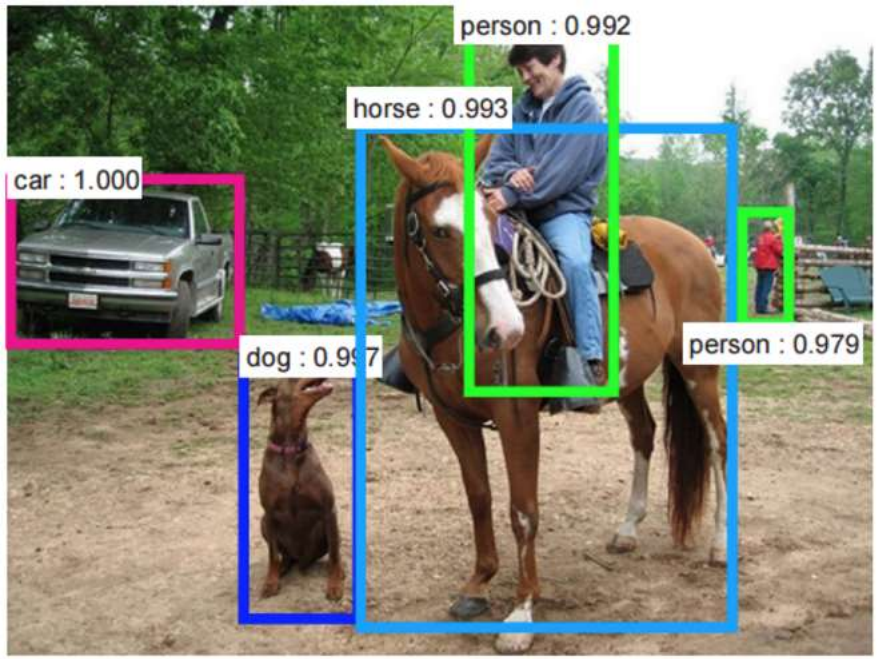

Saliency maps show gradients for each pixel with respect to the CNN's loss function. Darker pixels represent those with more influence.

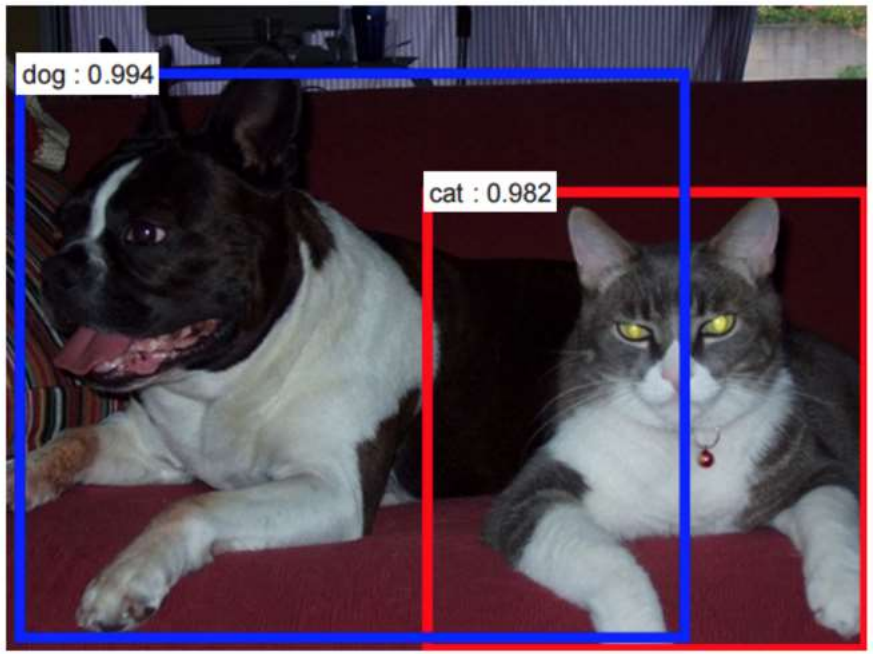
**Q: How much does this visualization help us understand the model?**

Detection and Segmentation for Medical Images

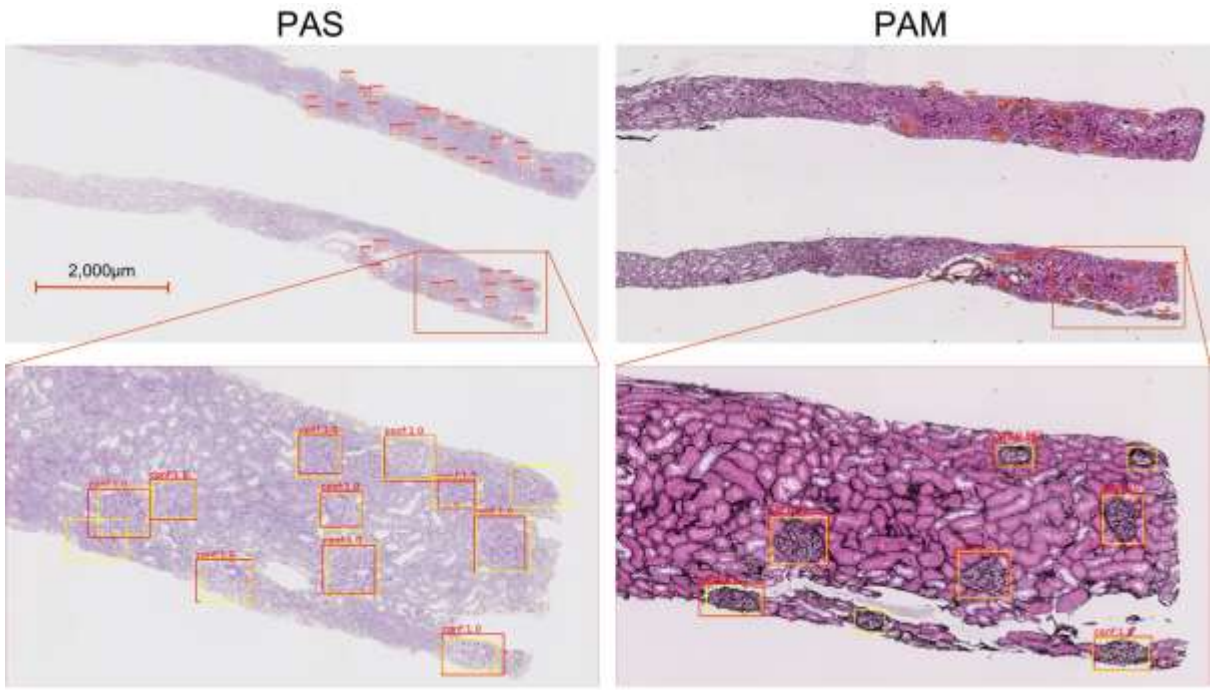
# BEYOND CLASSIFICATION

# Detection: propose regions and predict their labels



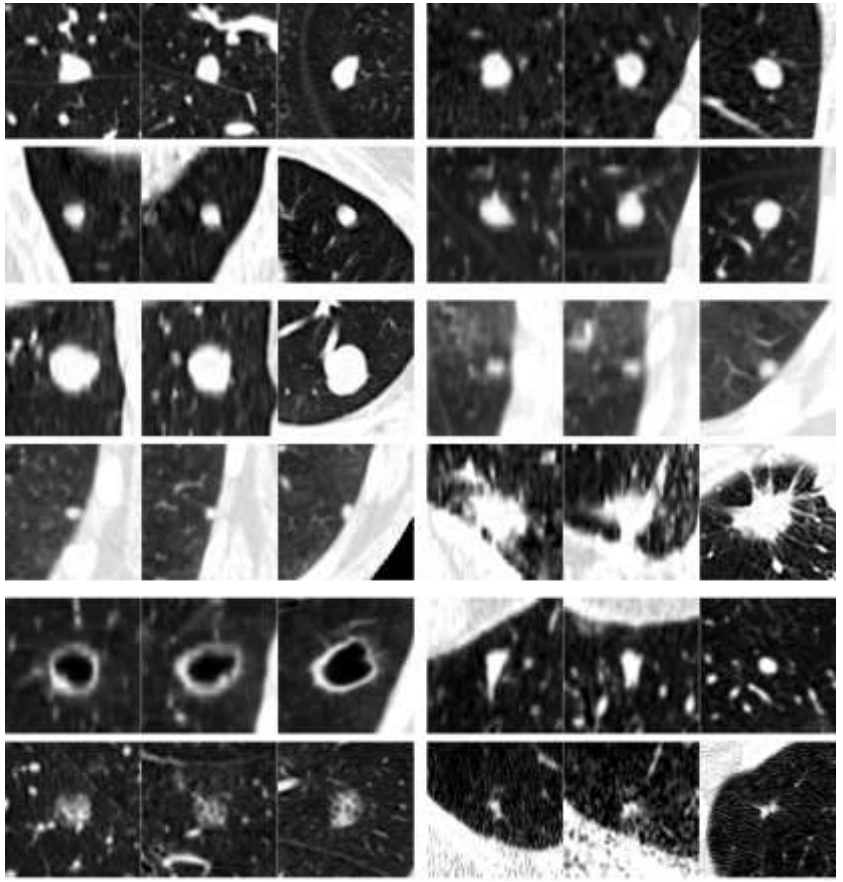



# Detection in medicine




Glomerular Detection with Faster-RCNN

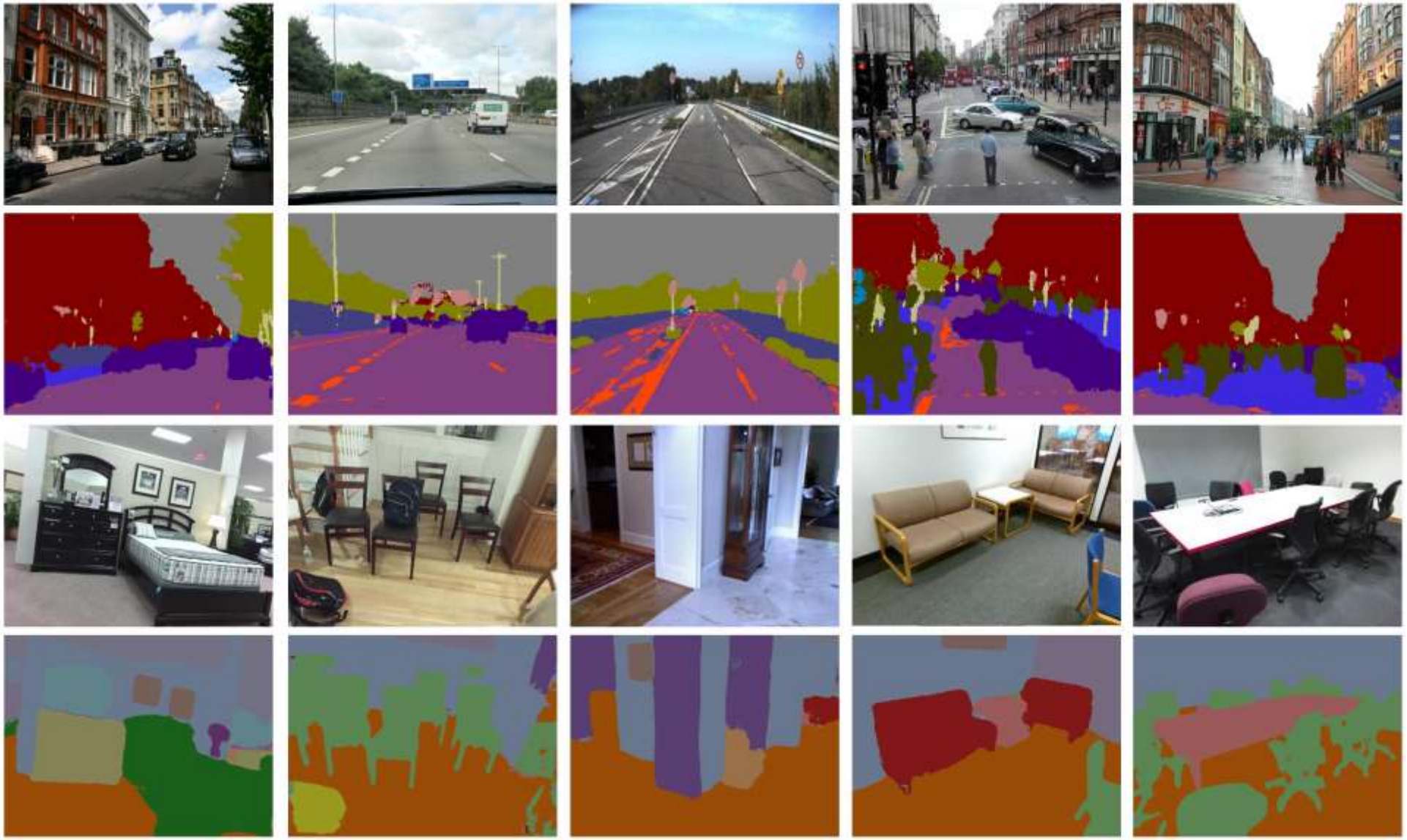
Kawazoe et al., *J. Imaging, 2018*



Pulmonary Nodule detection in CT

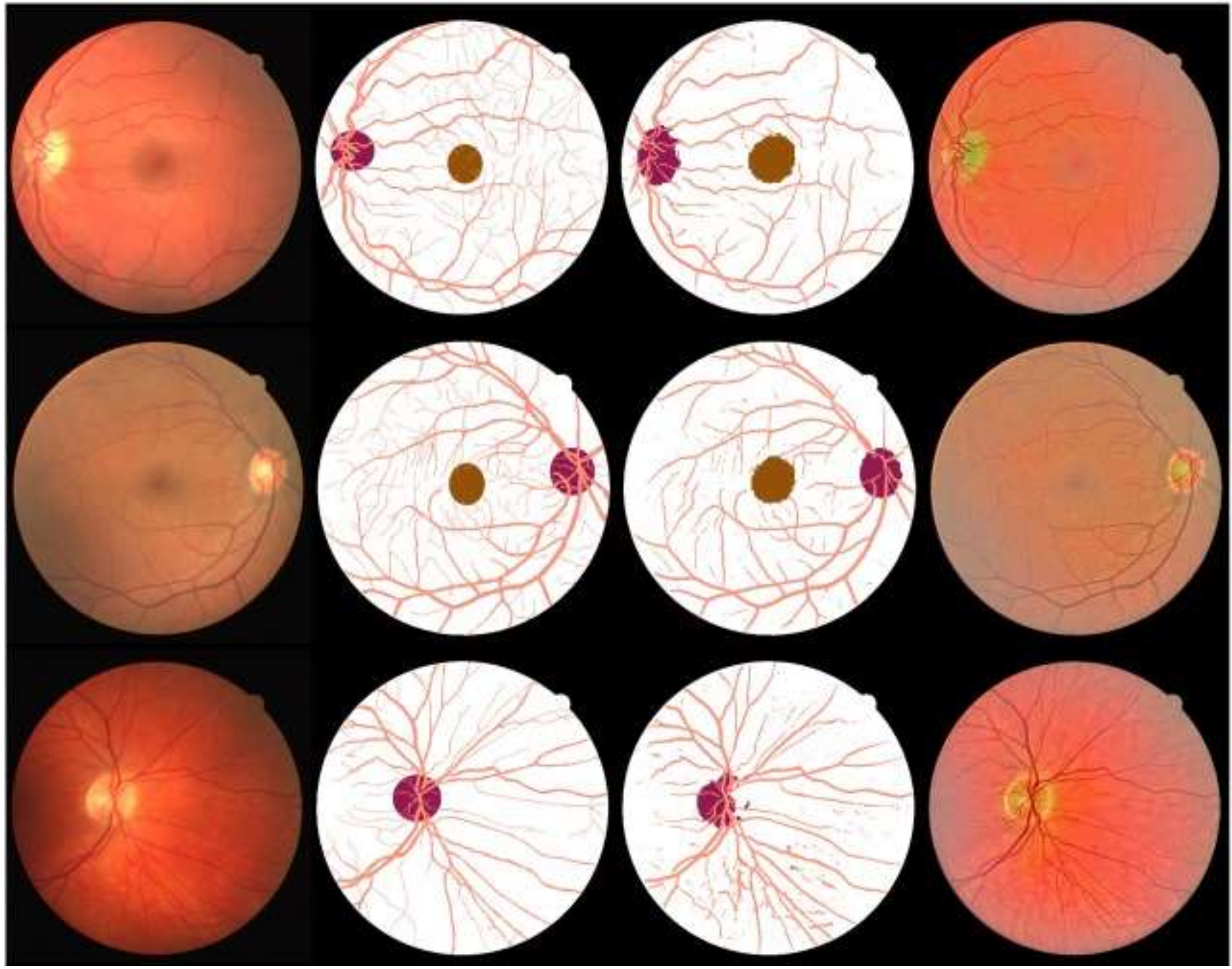
van Ginneken et al., *Biomedical Imaging,* 2015

# Segmentation: predict the label for each pixel

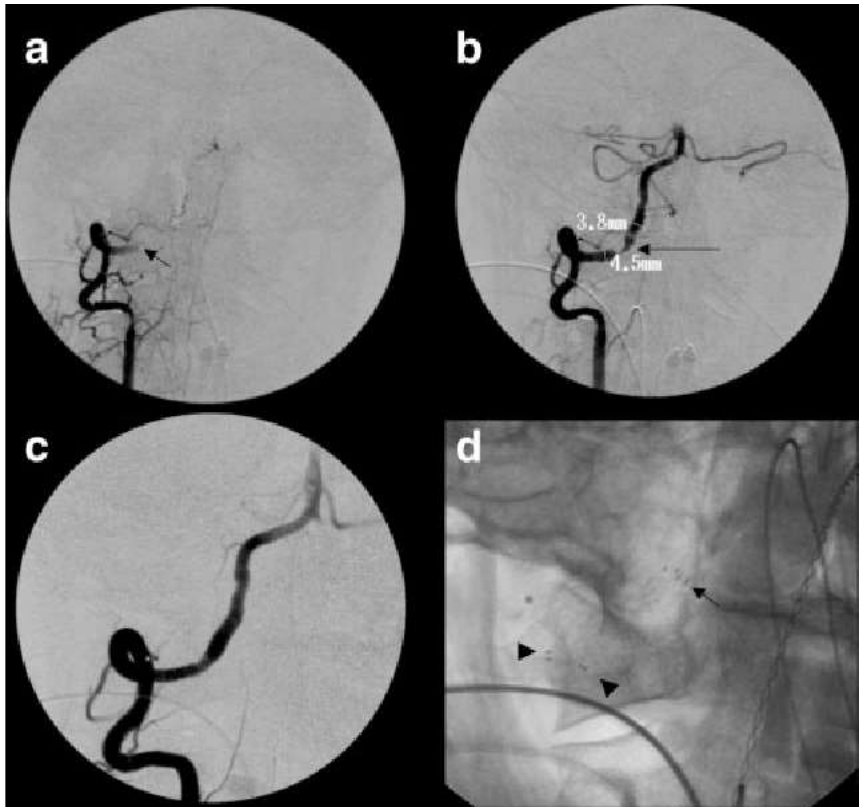


# Segmentation of optic disc, fovea and retinal vasculature

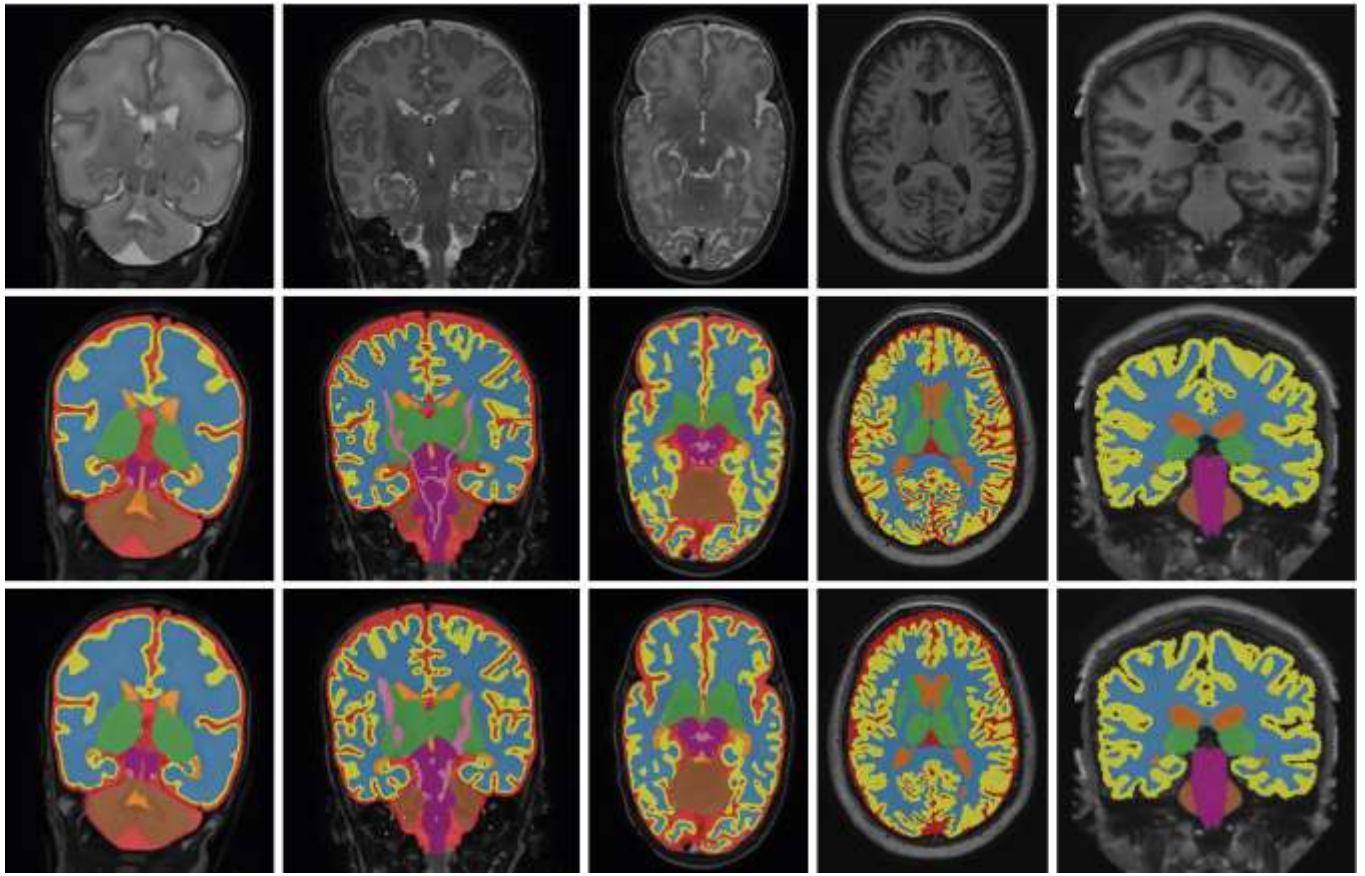
*Journal of Computational Science*, *20*, 70-79 (2017).



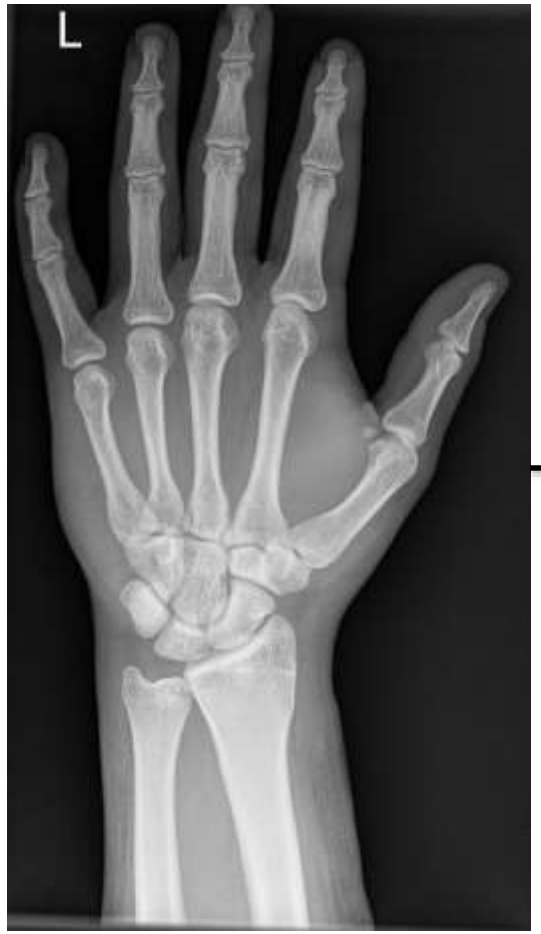
# Precisely Identify Boundaries



# Determine Areas or Volumes



# Segmentation-based features when end-to-end classification is not feasible




Segment Bones → Calculate lengths and areas → Simple Classifier for Genetic or Growth Disorder

nature biomedical engineering

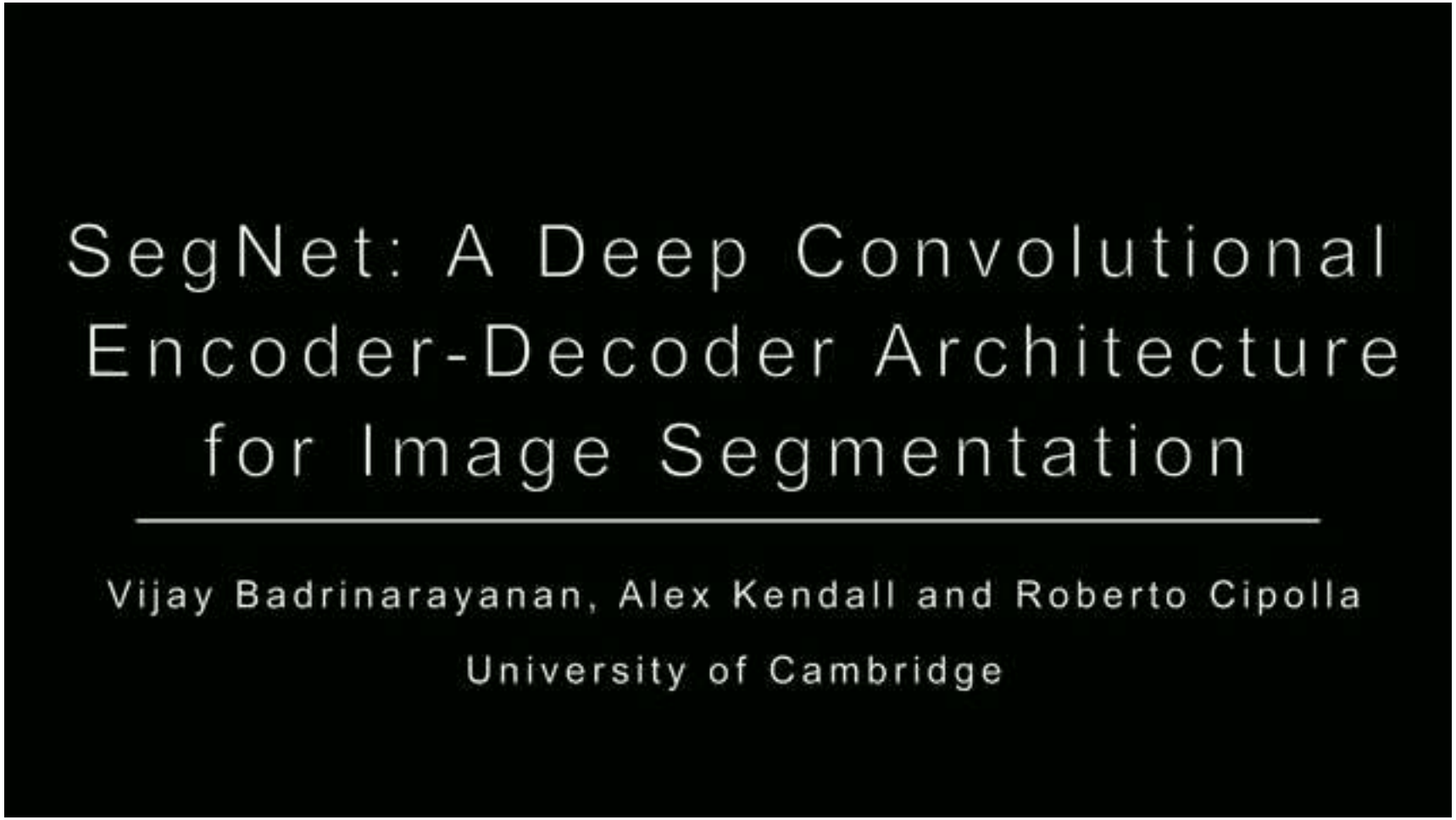
Article | Published: 10 October 2018

# Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy

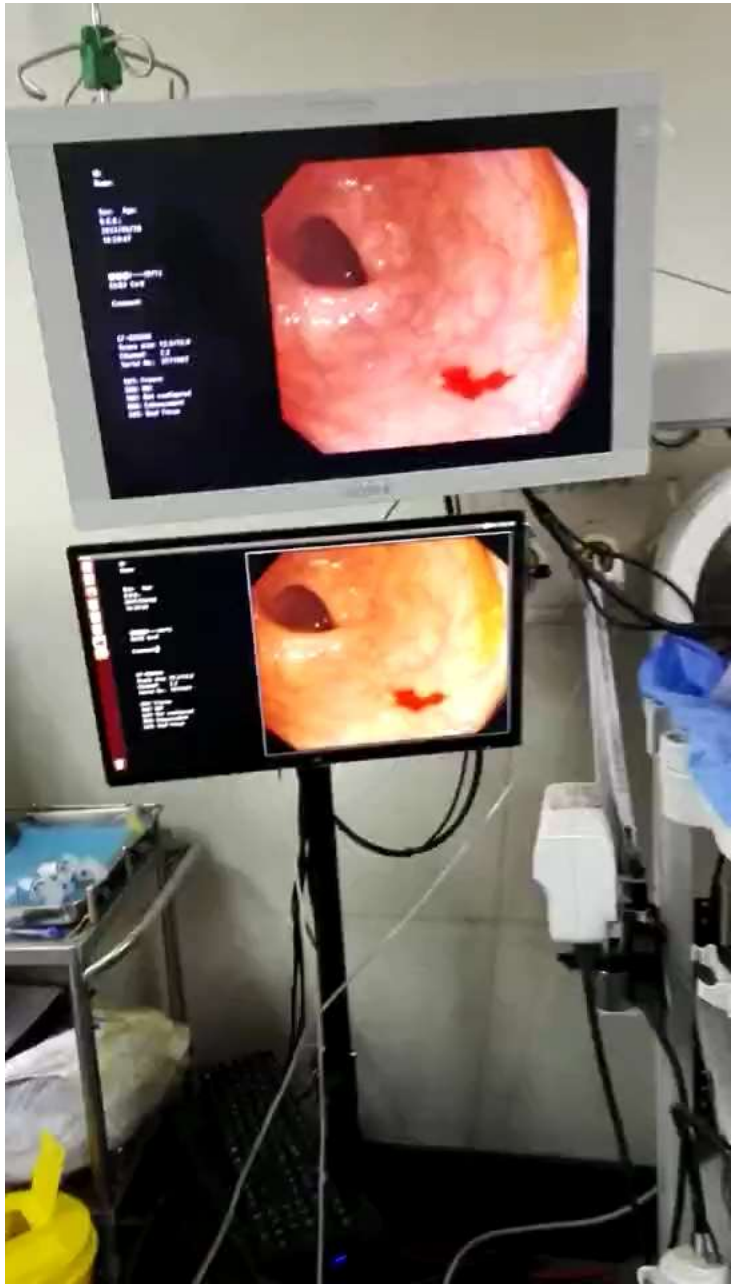
Pu Wang, Xiao Xiao, Jeremy R. Glissen Brown, Tyler M. Berzin, Mengtian Tu, Fei Xiong, Xiao Hu, Peixi Liu, Yan Song, Di Zhang, Xue Yang, Liangping Li, Jiong He, Xin Yi, Jingjia Liu & Xiaogang Liu ✉

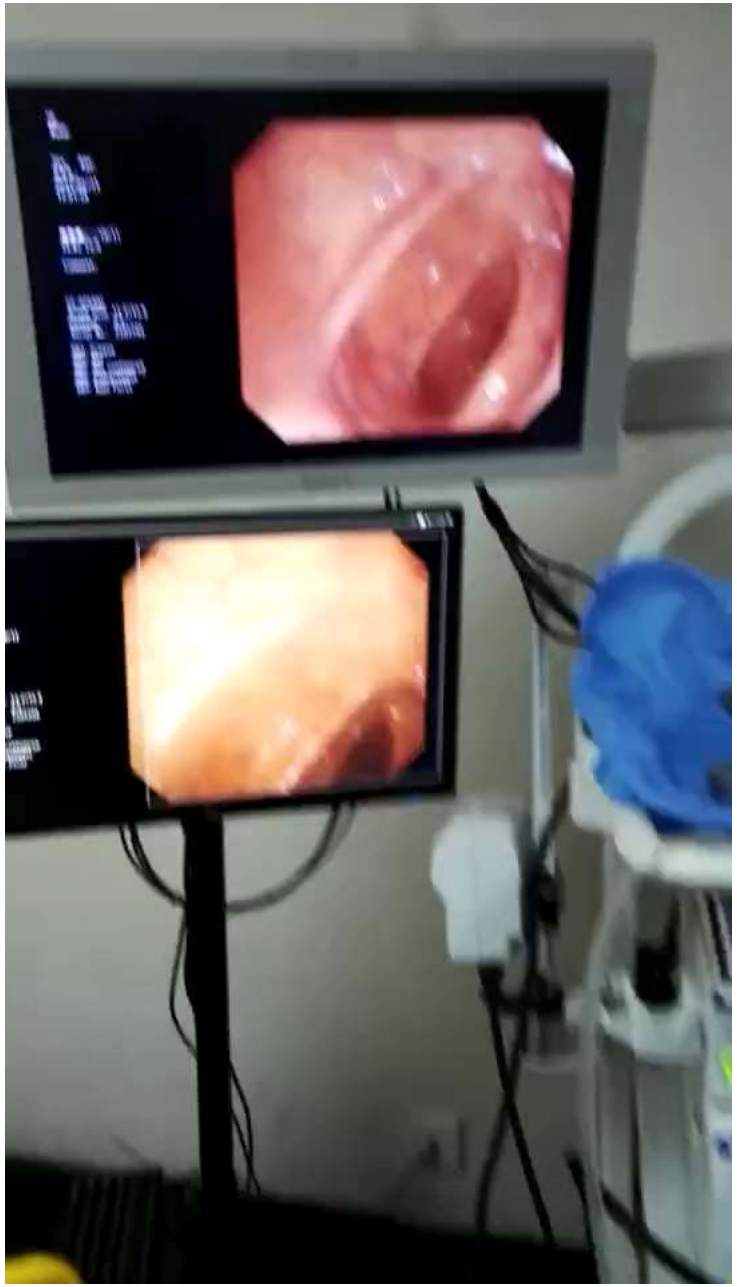
*Nature Biomedical Engineering* **2**, 741–748 (2018) | Download Citation ↓

# Approach: Start with SegNet (2015)



# Retrain to segment polyps in real time





# THANK YOU!

Questions or ideas? Please contact me at m.engelhard@duke.edu