SONOMA STATE UNIVERSITY

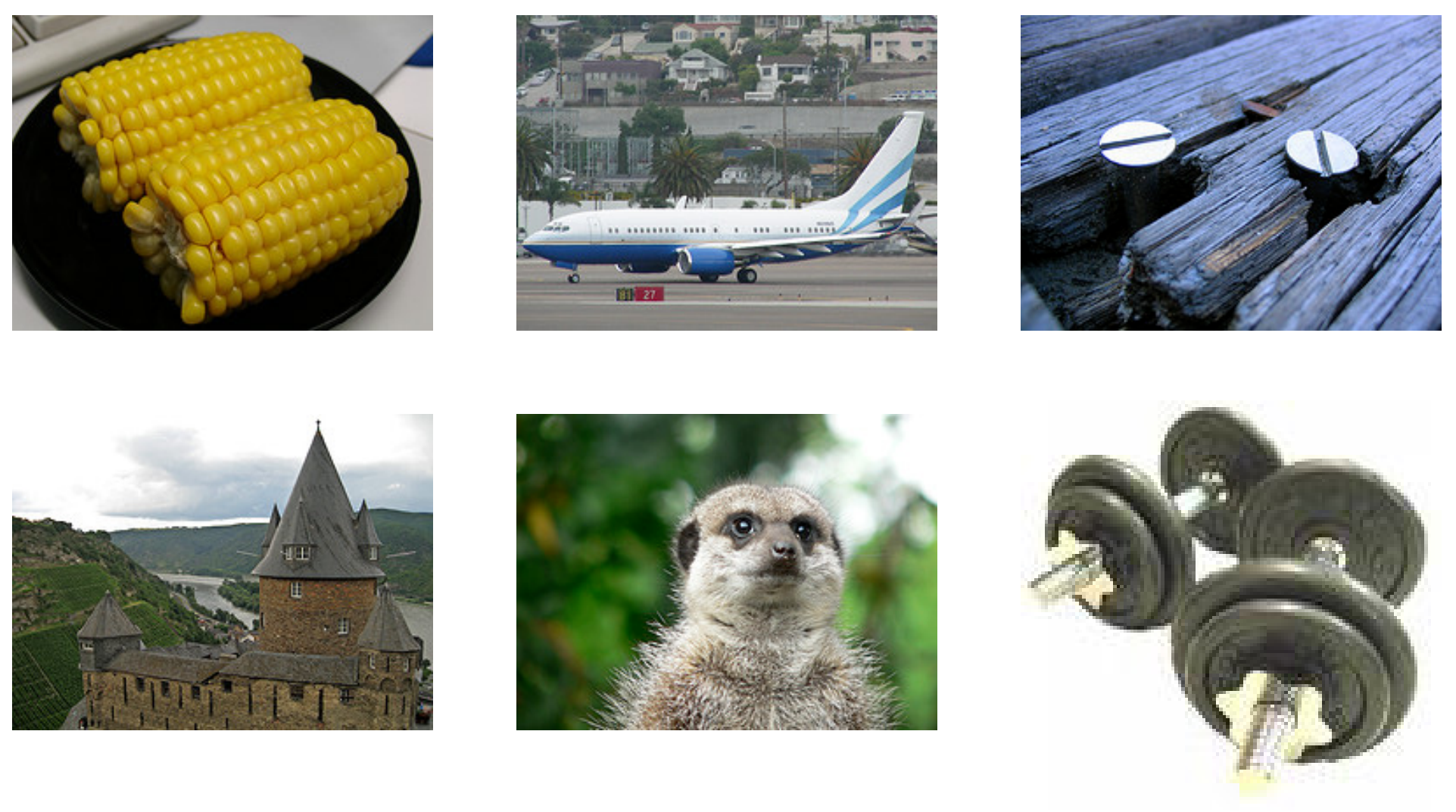
Using pre-trained convolutional neural networks to classify interstitial lung diseases in chest computed tomography scans

Joseph Granados (Mathematics and Statistics Department), Gurman Gill (Computer Science Department)

Chest computed tomography (CT) scans are widely used for automatic detection and classification of Interstitial lung disease (ILD) using computer-aided diagnostic (CAD) systems. The goal of these CAD systems is to reduce the time taken for and to optimize the diagnostic decisions made by radiologists. Convolutional neural networks (CNN) have been shown to be extremely effective at visual classification tasks. However, CNNs require large, labeled datasets in order to tune millions of parameters. Medical datasets large enough to train a CNN from scratch are not readily available. In this work, we explore the feasibility of using CNNs that have been fully trained with a non-medical dataset to classify ILDs in CT scan patches using "transfer learning": a technique where features identified by the pre-trained CNN are then used to train a new CAD system.

Data Sets

ImageNet Data Set (used to pre-train the CNN)
- 1.2 Million Images
- 1000 Categories
- Sample Images:



Talisman Test Suite Data Set
- 14594 Images from 85 Patients
- 5 Categories (disease types)
- Sample Images:

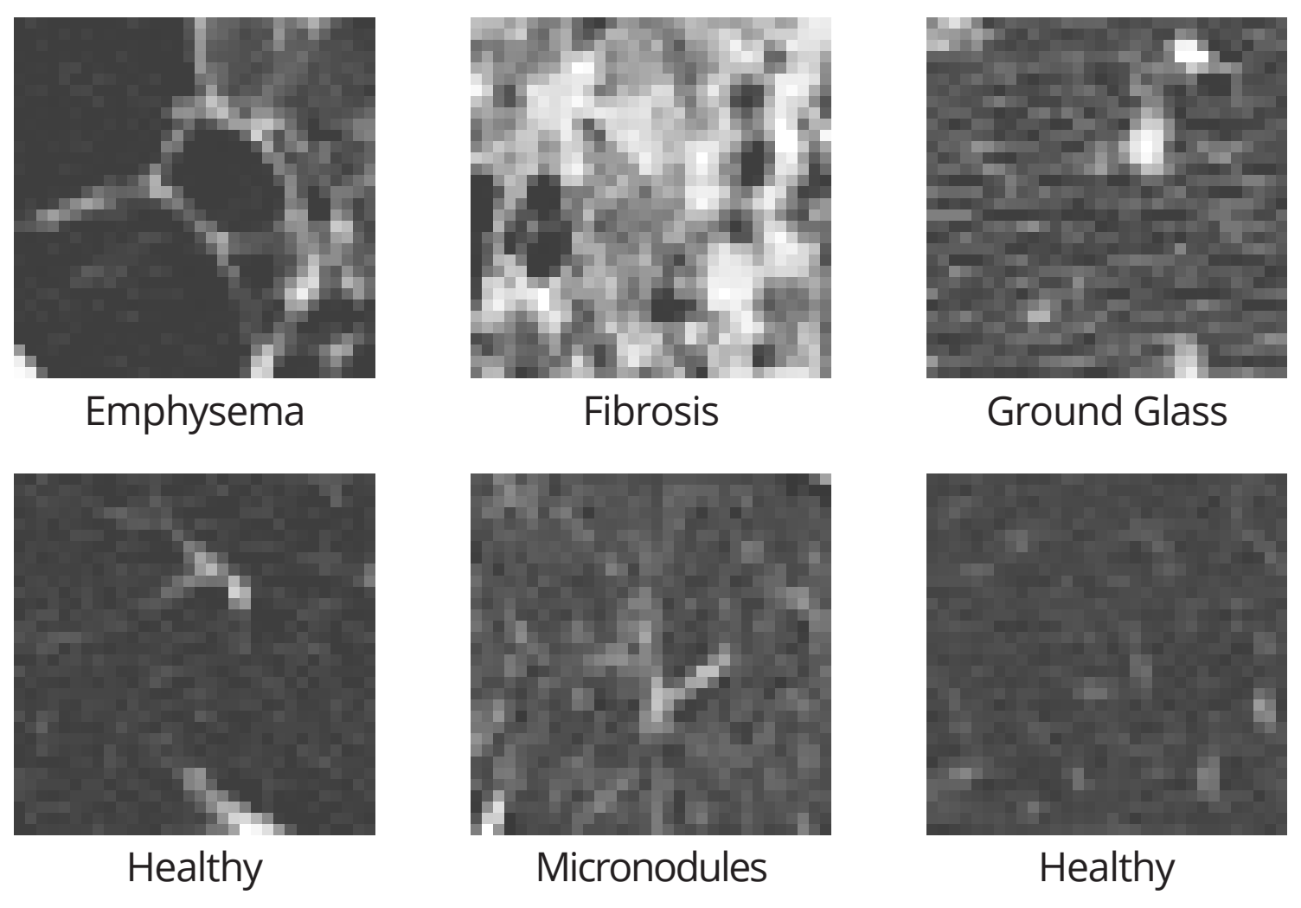


Image Count per Disease

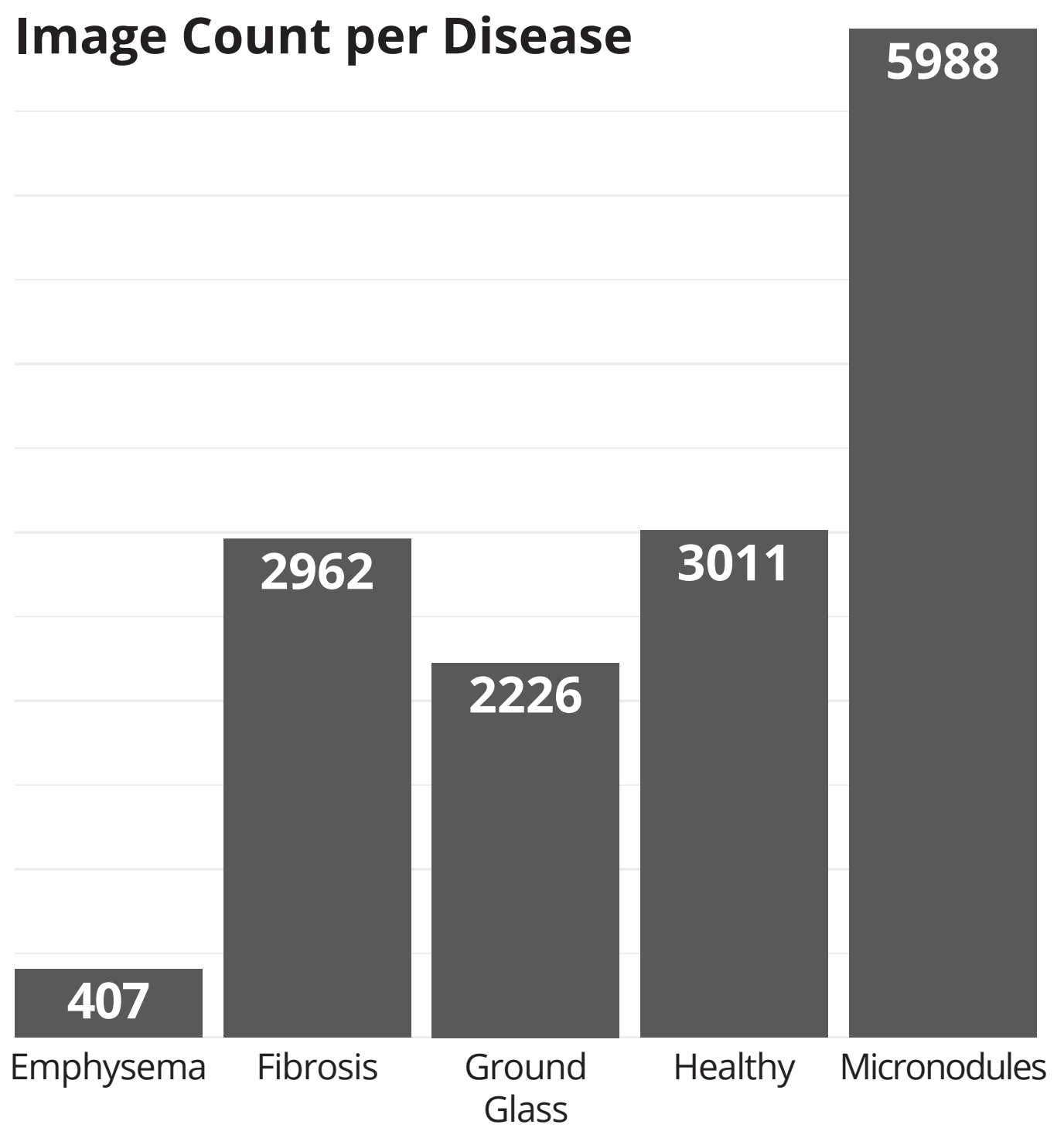
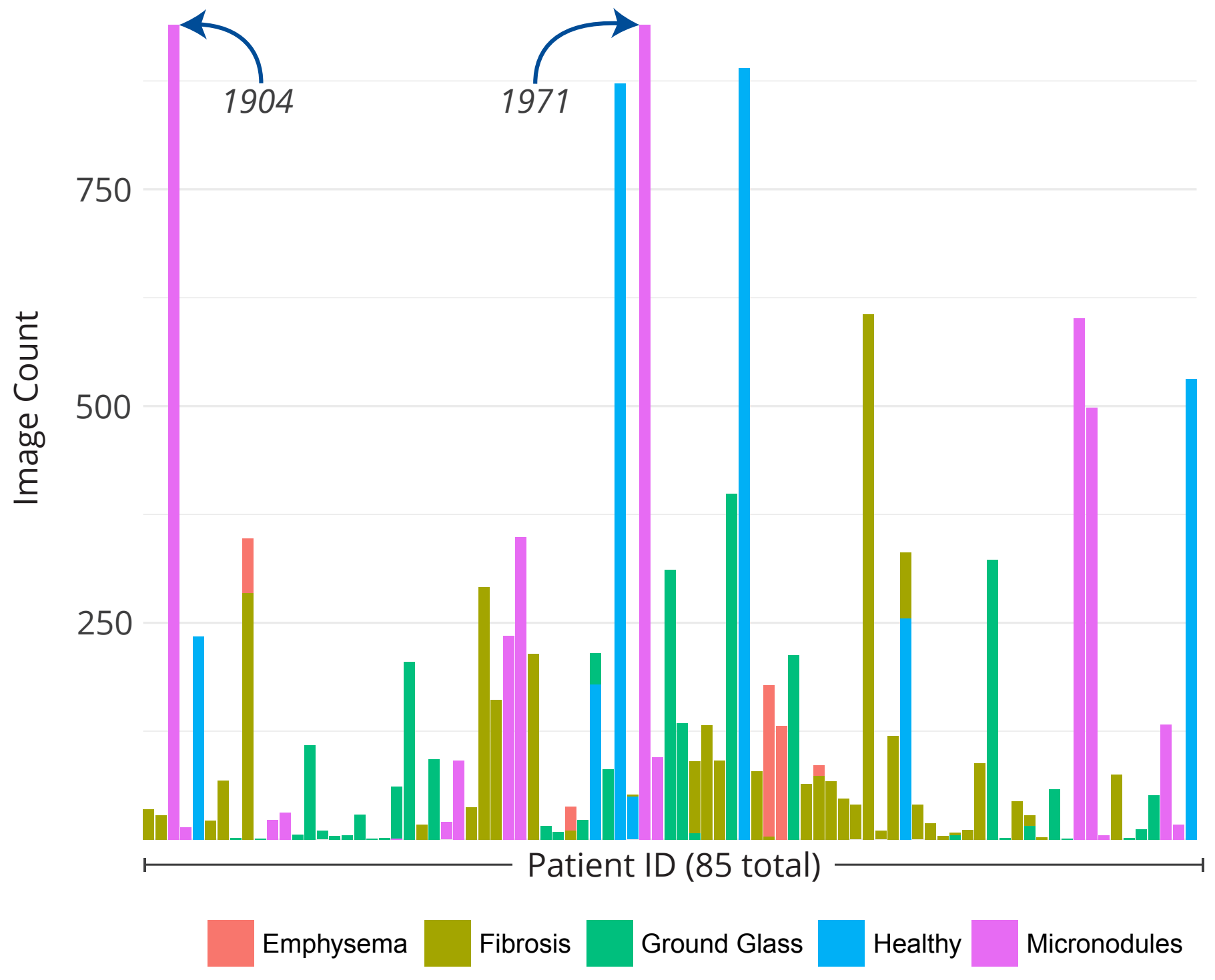


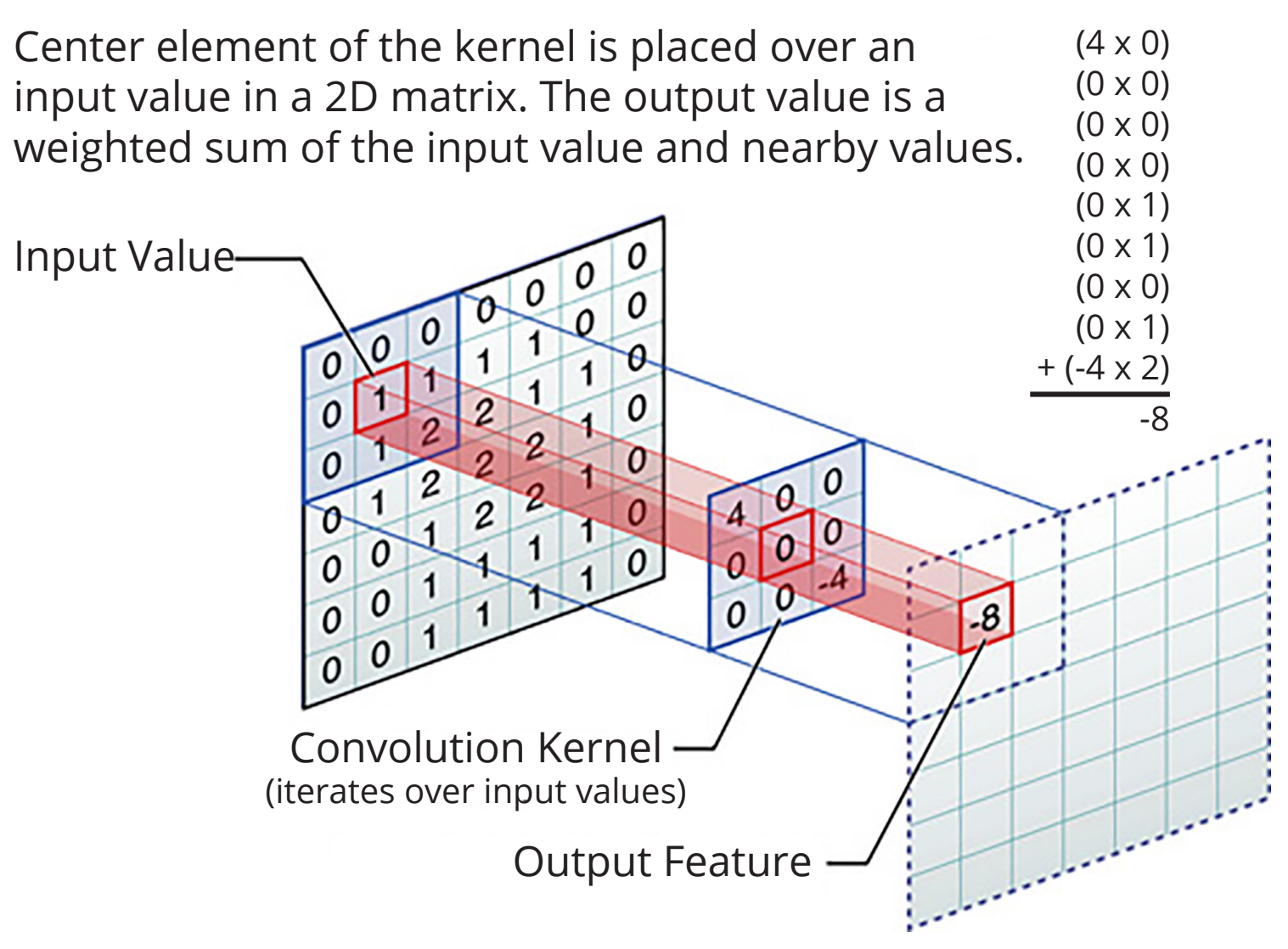
Image Count per Patient



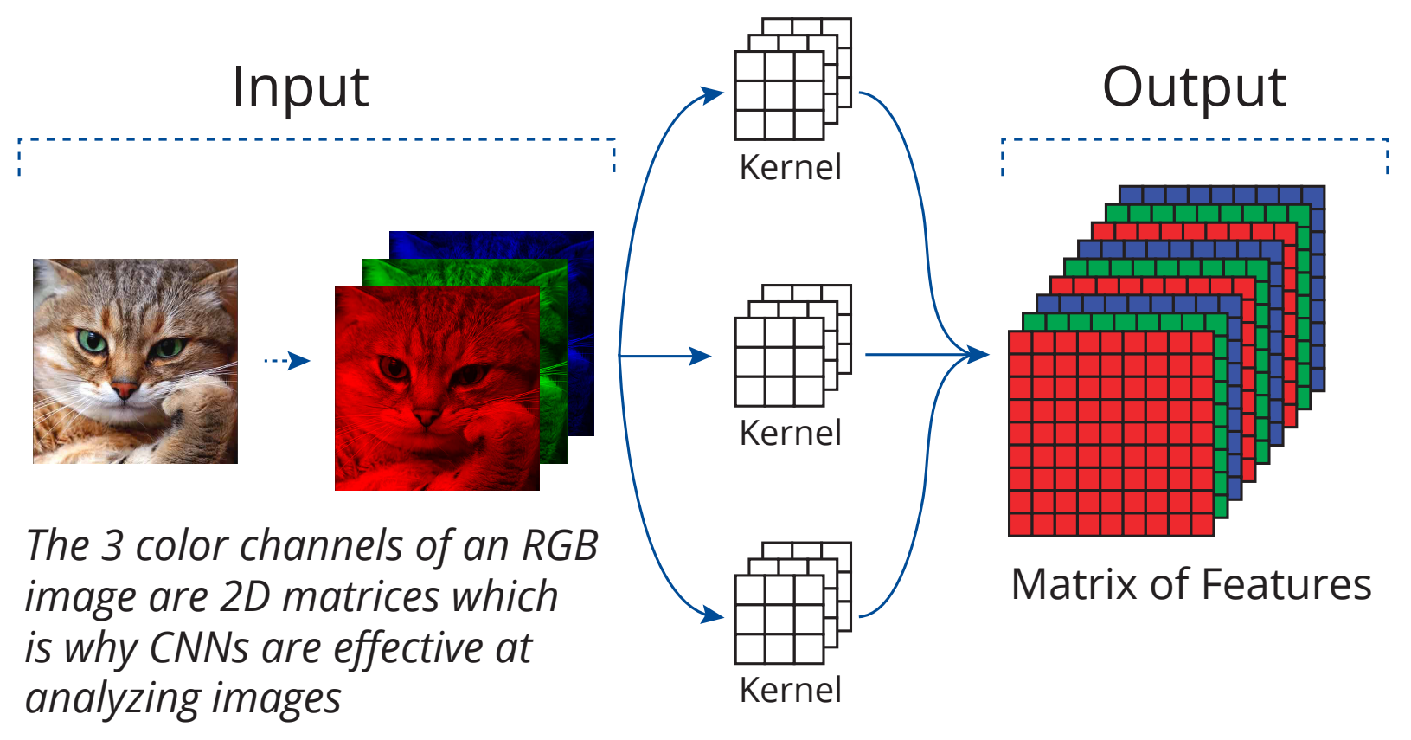
Brief Introduction to CNNs

Convolution Kernel

Center element of the kernel is placed over an input value in a 2D matrix. The output value is a weighted sum of the input value and nearby values.

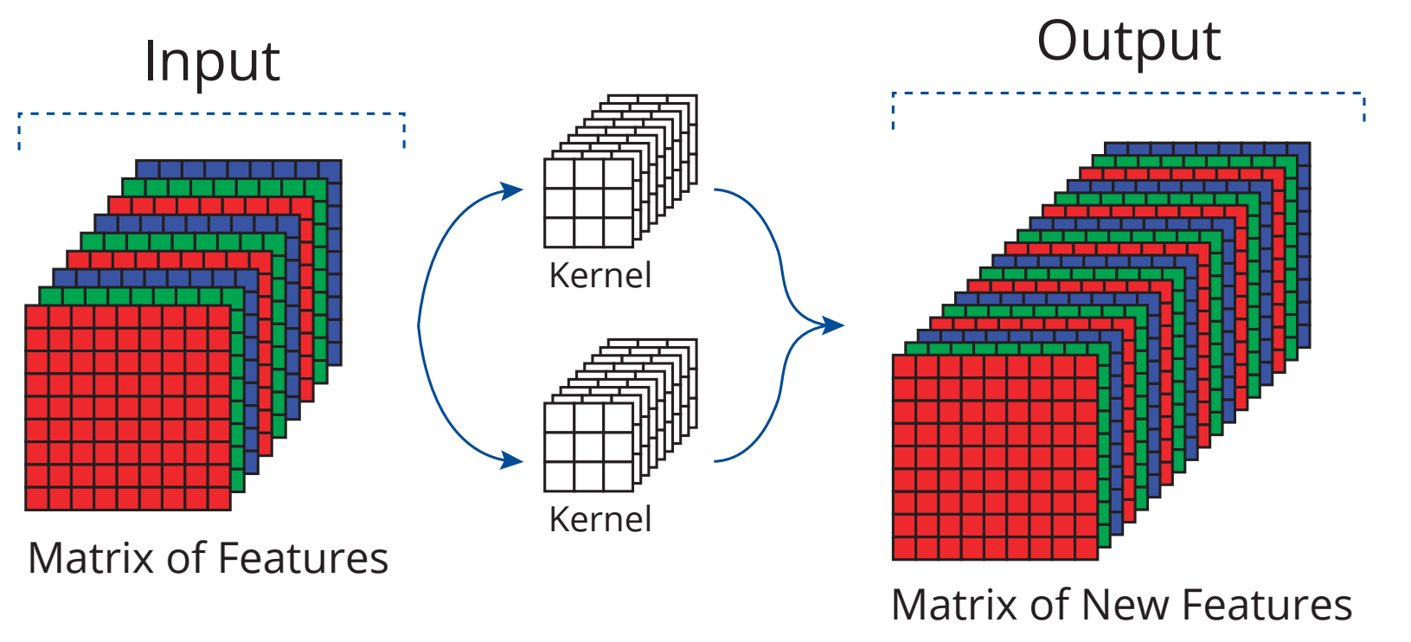


Convolutional Layer is Composed of Multiple Kernels



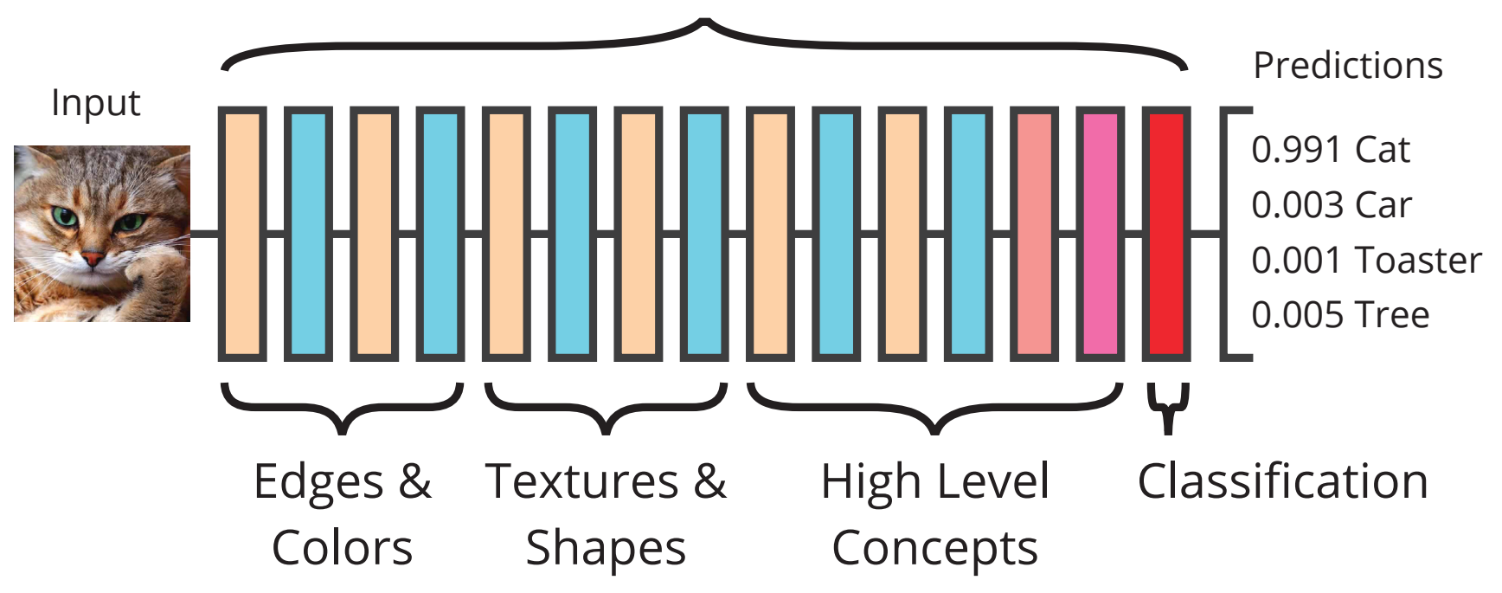
The 3 color channels of an RGB image are 2D matrices which is why CNNs are effective at analyzing images

Note that a convolutional layer can evaluate any stack of 2D matrices



- Other layers commonly used in Neural Networks: pooling, fully connected, softmax, dropout, concatenation
- Sequences of layers make up a CNN architecture
- Training involves feeding images into a CNN and optimizing parameters, such as kernel values, so that they minimize a loss function

Simple CNN composed of multiple Layers



It is generally the case that in a trained CNN initial layers respond to simple concepts such as edges or colors and later layers respond to high level concepts such as faces.

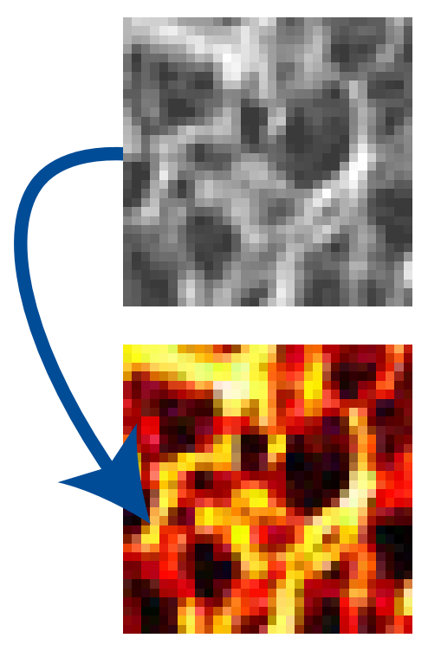
A Neural Network assigns a score for each class it knows about. In the example above, the CNN gives the image a score of 0.991 for the class Cat, indicating that the CNN is highly confident the image is of a cat.

Methodology

HU to RGB

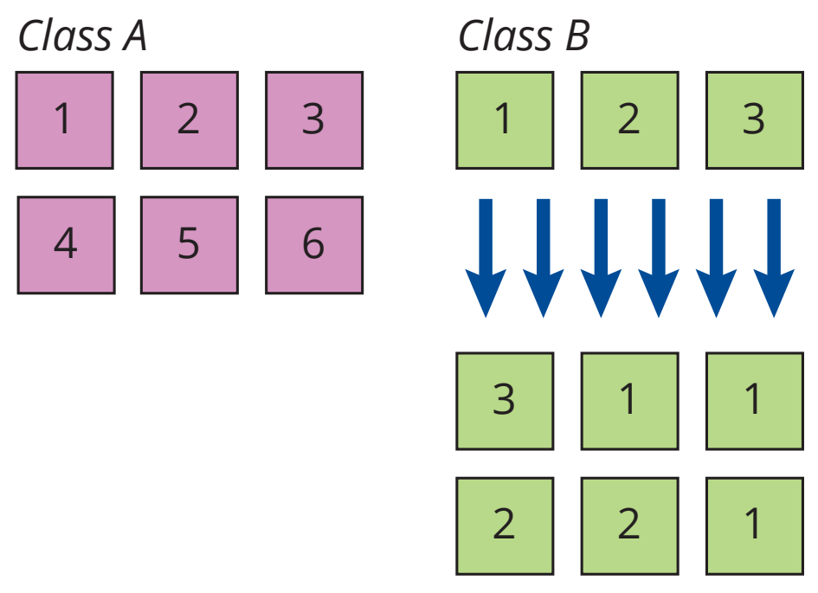
CT-scans are saved in Hounsfield units (HU) but the CNN we're using expects RGB images. RGB images consist of 3 color channels that have values between 0 and 255 for each pixel. We linearly mapped
- HU range (-1000, -600) to the Red Channel,
- HU range (-601, -200) to the Green Channel,
- HU range (-201, 200) to the Blue Channel,

and values outside of the HU ranges of each channel were mapped to either 0 or 255.



Oversampling

When class sizes are imbalanced, take random samples with replacement to make training class sizes equal. This way all classes are represented equally during training.



1) Class B has 3 observations

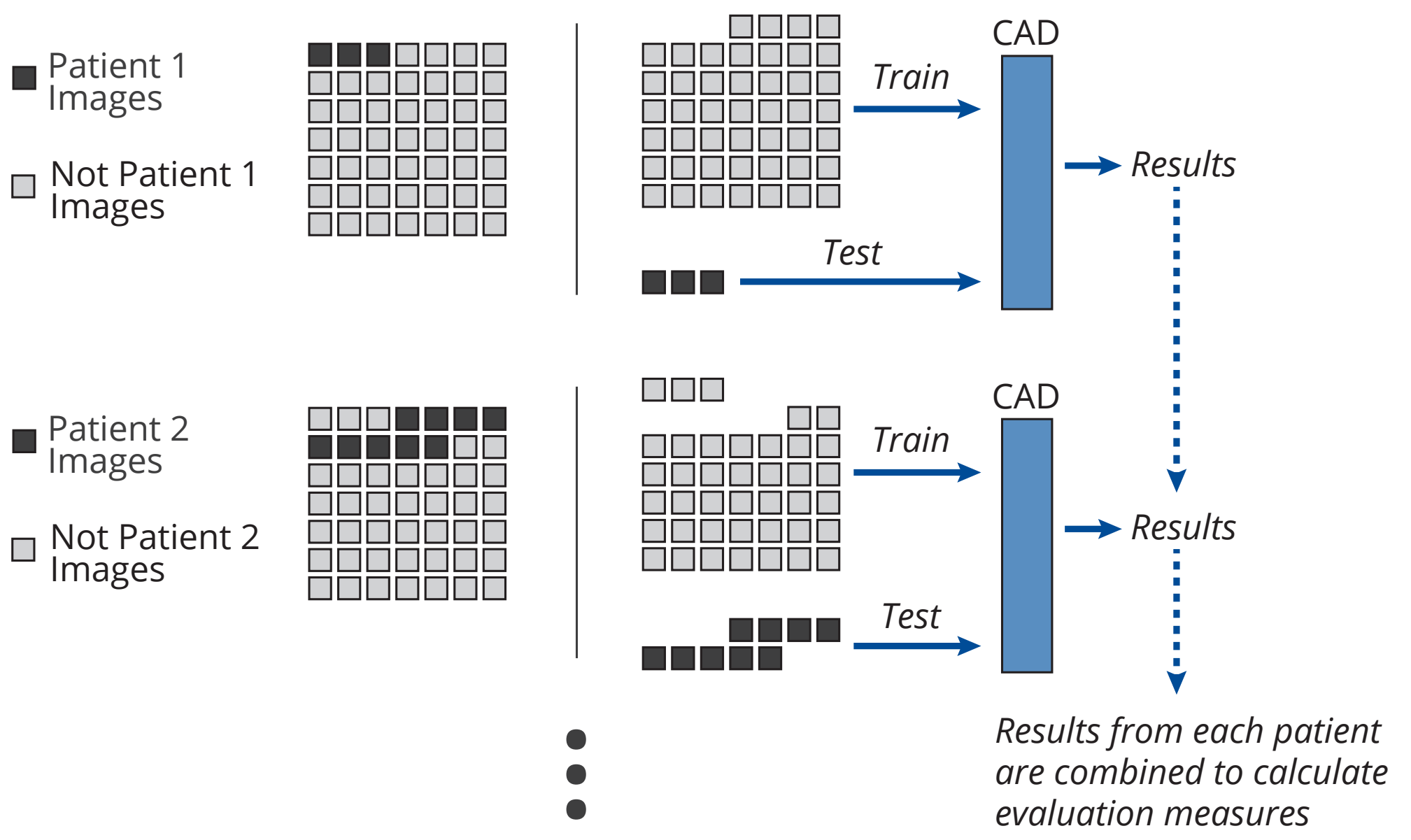
2) Random Sample with Replacement

3) Class B now has 6 observations, some of which are duplicates

Leave One Patient Out Cross Validation

To get the best estimate of model performance we want to:
- ensure that only information specific to the diseases is used for training
- use as much training data as possible

By removing all images associated with a single patient we can be sure that information specific to that patient doesn't influence the model and we can use all the remaining images from other patients for training. Then we evaluate the removed images with the newly trained CAD. Doing this for each patient allows us to make an estimate of model performance for any patient.

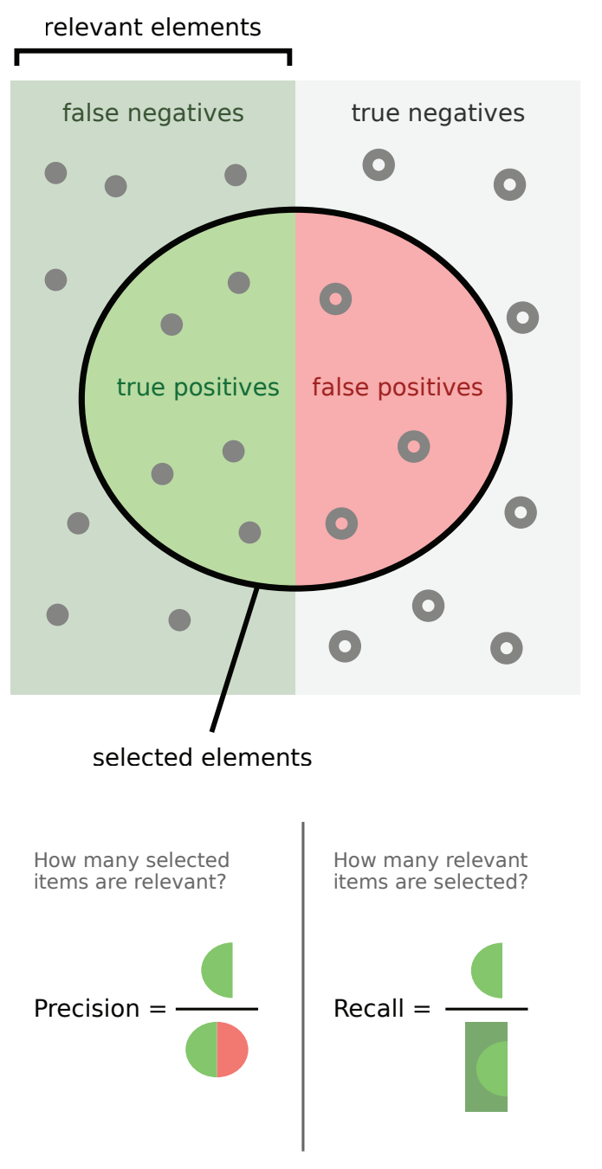


Results from each patient are combined to calculate evaluation measures

Evaluation Measures

Confusion Matrix: contingency table showing actual class versus predicted class for all images.

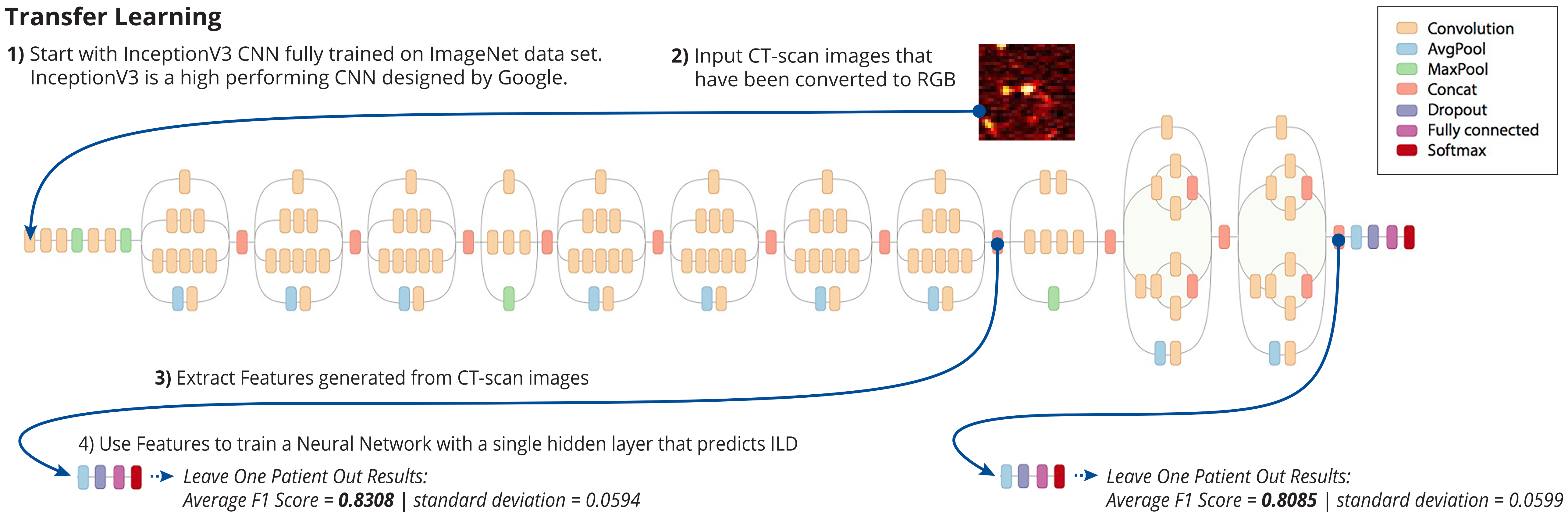
F1 Score: average of precision and recall. A number between 0 and 1 where 1 is the best possible performance.



Transfer Learning

1) Start with InceptionV3 CNN fully trained on ImageNet data set. InceptionV3 is a high performing CNN designed by Google.

2) Input CT-scan images that have been converted to RGB



3) Extract Features generated from CT-scan images

4) Use Features to train a Neural Network with a single hidden layer that predicts ILD

Leave One Patient Out Results:
*Average F1 Score = **0.8308** | standard deviation = 0.0594*

Leave One Patient Out Results:
*Average F1 Score = **0.8085** | standard deviation = 0.0599*

Best Leave One Patient Out Results

Using features extracted from an intermediate layer of the ImageNet trained InceptionV3 CNN.

Average F1 Score = **0.8308**

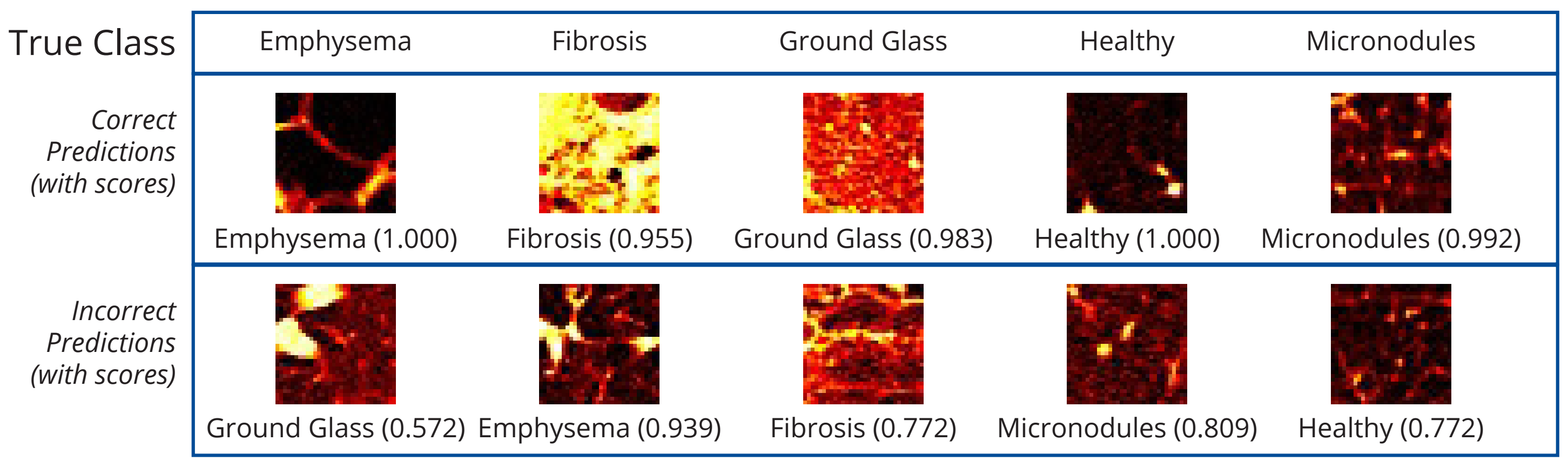
Standard deviation = 0.0594

Normalized Confusion Matrix

(predicted)

(actual)	EM	FI	GG	HE	MI
EM	**0.59**	0.13	0.10	0.07	0.11
FI	0.01	**0.90**	0.07	0.01	0.02
GG	0.01	0.09	**0.78**	0.04	0.08
HE	0.01	0.00	0.04	**0.82**	0.12
MI	0.01	0.02	0.04	0.06	**0.87**

CAD Prediction Examples:



True Class	Emphysema	Fibrosis	Ground Glass	Healthy	Micronodules
Correct Predictions (with scores)	Emphysema (1.000)	Fibrosis (0.955)	Ground Glass (0.983)	Healthy (1.000)	Micronodules (0.992)
Incorrect Predictions (with scores)	Ground Glass (0.572)	Emphysema (0.939)	Fibrosis (0.772)	Micronodules (0.809)	Healthy (0.772)

Conclusions

- Transfer learning is feasible for classifying ILD patches
- Extracting features from an intermediate layer of the InceptionV3 model gave the best results, indicating that information learned by later layers is not applicable to ILD classification
- Emphysema has an average recall of 0.59 but it is underrepresented in this dataset (only 407 images)
- Our future work involves looking at data augmentation (a method of generating artificial data) and fine-tuning to improve results

Work done with funding from School of Science and Technology Dean Stauffer in Summer 2017

Image Attributions:
developer.apple.com
image-net.org
research.googleblog.com
wikipedia.org