Deep Learning with Convolutional Neural Networks for Radiologic Image Classification

Phillip Cheng, MD MS

Keck School of Medicine of USC

Objective

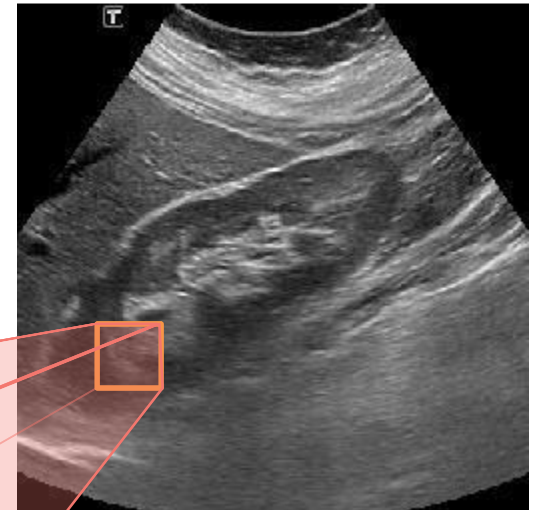
The purposes of this exhibit are

1. To illustrate the basic concepts of deep learning with convolutional neural networks.
2. To illustrate transfer learning with a deep convolutional neural network for classification of radiology images.

Computer Vision

Consider this ultrasound image of the right kidney:

It takes little training for a person to classify this image accurately as an image of the right kidney. Why is this task hard for a computer?



55	73	72	100	115	137	115	121	134	145	147	143
50	68	78	117	134	149	117	123	134	142	138	127
53	61	71	115	128	133	139	141	146	146	136	119
61	70	84	138	156	162	145	140	133	121	106	92
62	68	77	127	148	158	144	136	118	95	76	66
54	57	52	86	101	113	134	131	118	94	72	60
52	62	56	80	89	103	103	116	122	109	87	68
58	65	51	61	59	69	93	120	144	143	123	99
67	81	88	84	81	83	73	82	95	109	119	117
73	88	100	105	109	112	75	89	104	112	113	110
67	69	72	78	79	77	102	95	93	108	134	146
76	72	75	85	89	84	82	95	107	114	120	130
74	69	72	81	83	78	92	79	69	81	109	127

Instead of shades of gray, a computer “sees” a matrix of numbers representing pixel brightness. Computer vision typically involves computing the presence of numerical patterns (features) in this matrix, and applying *machine learning* algorithms to distinguish images based on these features.

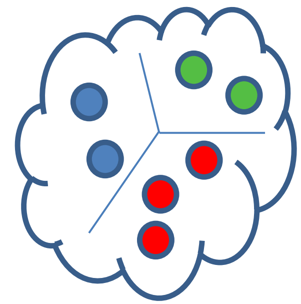
Problems with Features

Input image

- Lines
- Curves
- Blobs
- Edges
- Gradients
- Textures
- ...

Feature Decomposition

Classic machine learning



Mapping
(features → outputs)

"Right kidney"

Output

Engineering the best image features for distinguishing one class of images from another requires significant expertise. The optimization of distinguishing features traditionally has been a difficult problem in computer vision.

But what if the computer can learn on its own the best features to use? This is *representation learning*, the basis of deep learning. First some terminology...

Artificial Intelligence and Deep Learning

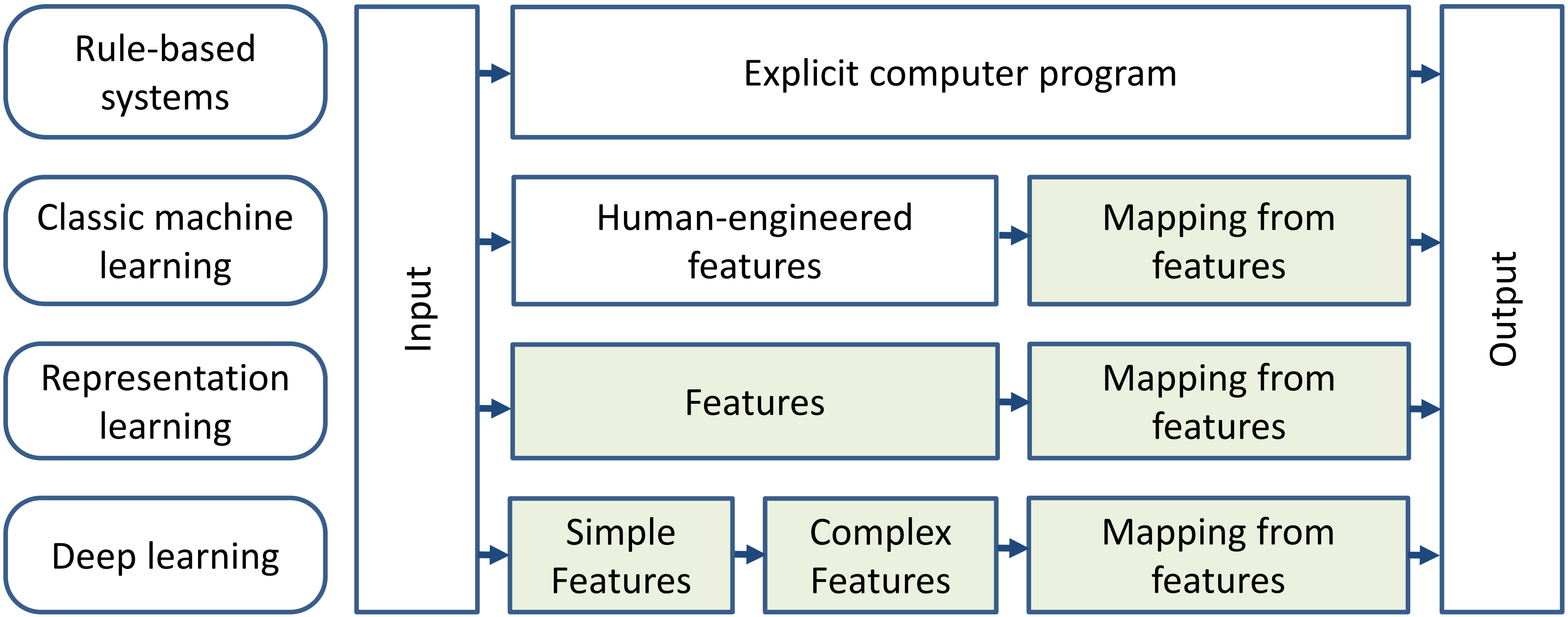
Artificial intelligence (AI) is a subfield of computer science devoted to creating systems to perform tasks ordinarily requiring human intelligence. In this presentation, we focus on algorithms for classifying data.

Machine learning is a subfield of artificial intelligence where computers are trained to perform tasks without explicit programming. Classically, humans engineer features by which a computer can learn distinguish patterns of data.

Representation learning is a type of machine learning where no feature engineering is used; instead, the computer *learns* the features by which to classify the provided data.

Deep learning is a type of representation learning where the learned features are compositional or hierarchical.

Machine Learning Paradigms

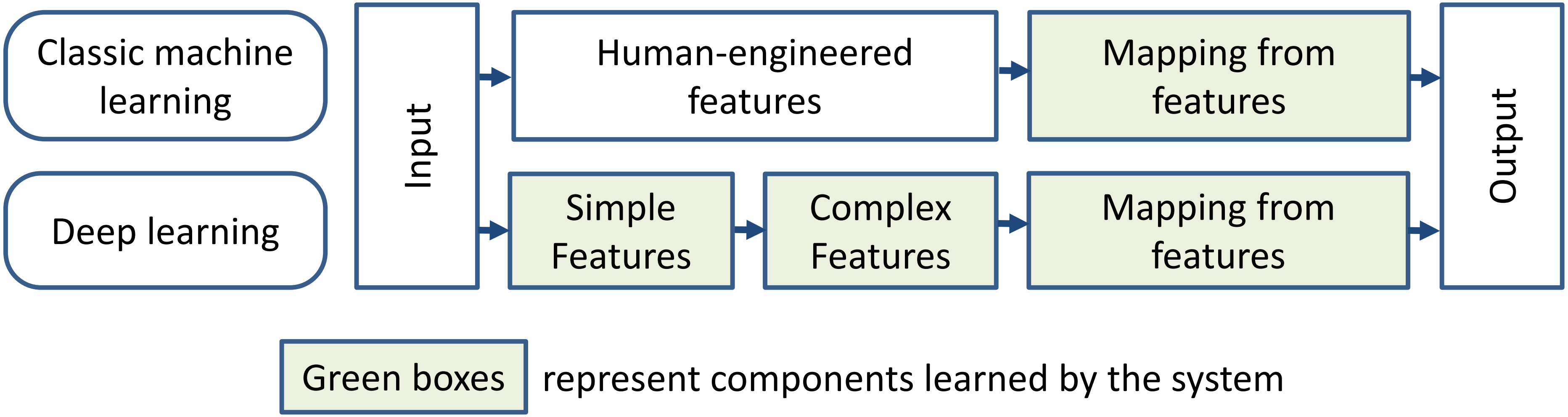


Green boxes represent components learned by the system

adapted from Goodfellow et al.

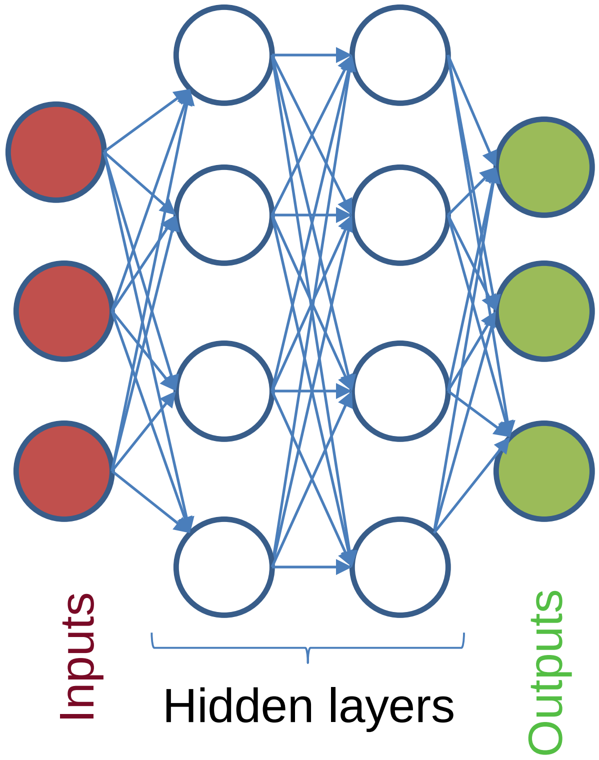
Why Deep Learning?

- Classic machine learning depends on carefully designed features, requiring human expertise and difficult to optimize.
- Deep learning bypasses feature engineering by taking advantage of lots of data and flexible hierarchical models.
- Deep learning has recently achieved striking performance improvements in diverse fields such as image classification, speech recognition, natural language processing, and game playing.



Neural Networks and Deep Learning

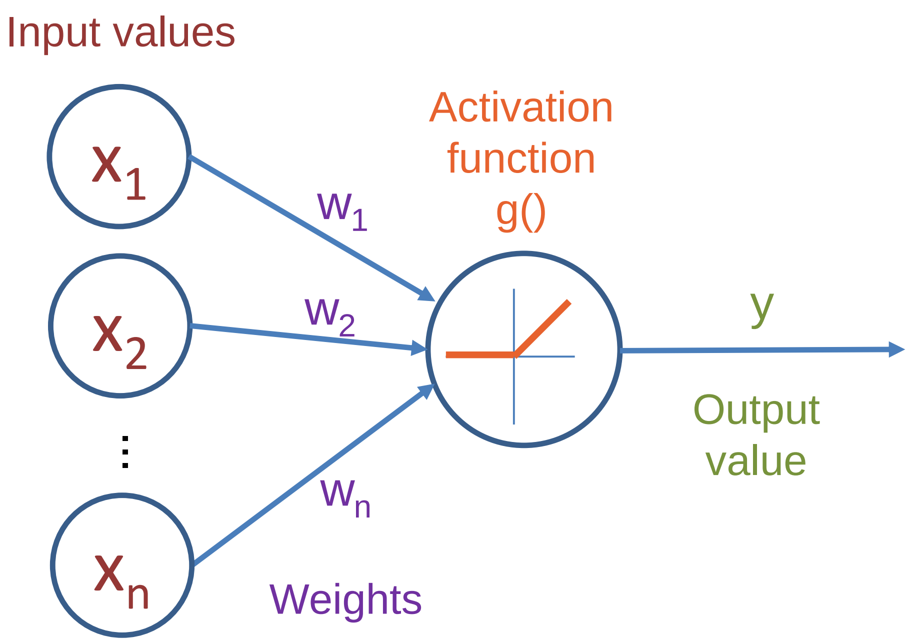
The basis for most deep learning research is the artificial neural network, a computational framework of interconnected nodes inspired by biological neural networks.



The “deep” aspect of deep learning refers to the multilayer architecture of these networks containing multiple “hidden layers” of nodes between input and output nodes.

Although neural networks have a long history dating to the 1950’s, training deep neural networks has only recently become feasible with improved training techniques, inexpensive parallel hardware and large amounts of labeled data.

Anatomy of an Artificial Neuron



Neurons in a neural network are linked by weighted connections. A neuron operates on a weighted sum of its inputs, $w_1 x_1 + w_2 x_2 + \cdots w_n x_n = \sum_i w_i x_i$.

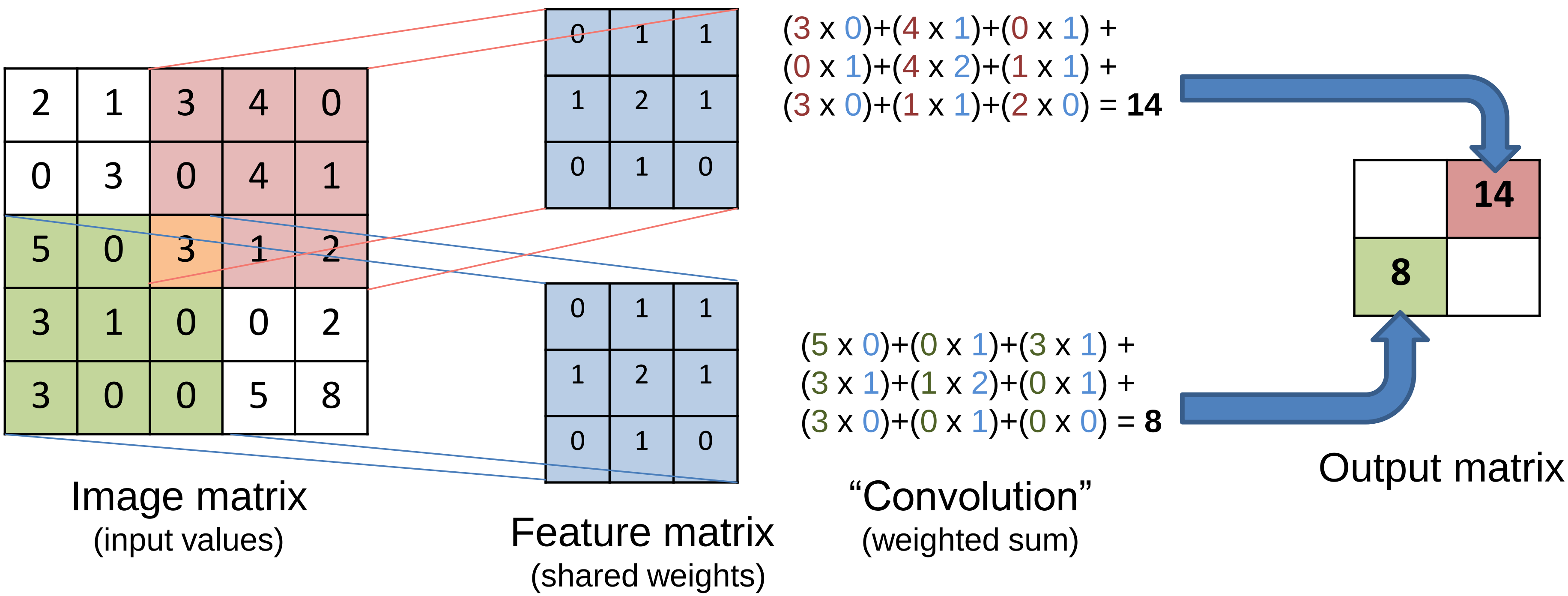
(an optional bias term is omitted here)

This sum is passed through a nonlinear activation function g(), typically a **Re**ctified **L**inear **U**nit (ReLU). The final output $\mathrm{y} = \mathrm{g}(\sum_i w_i x_i)$.

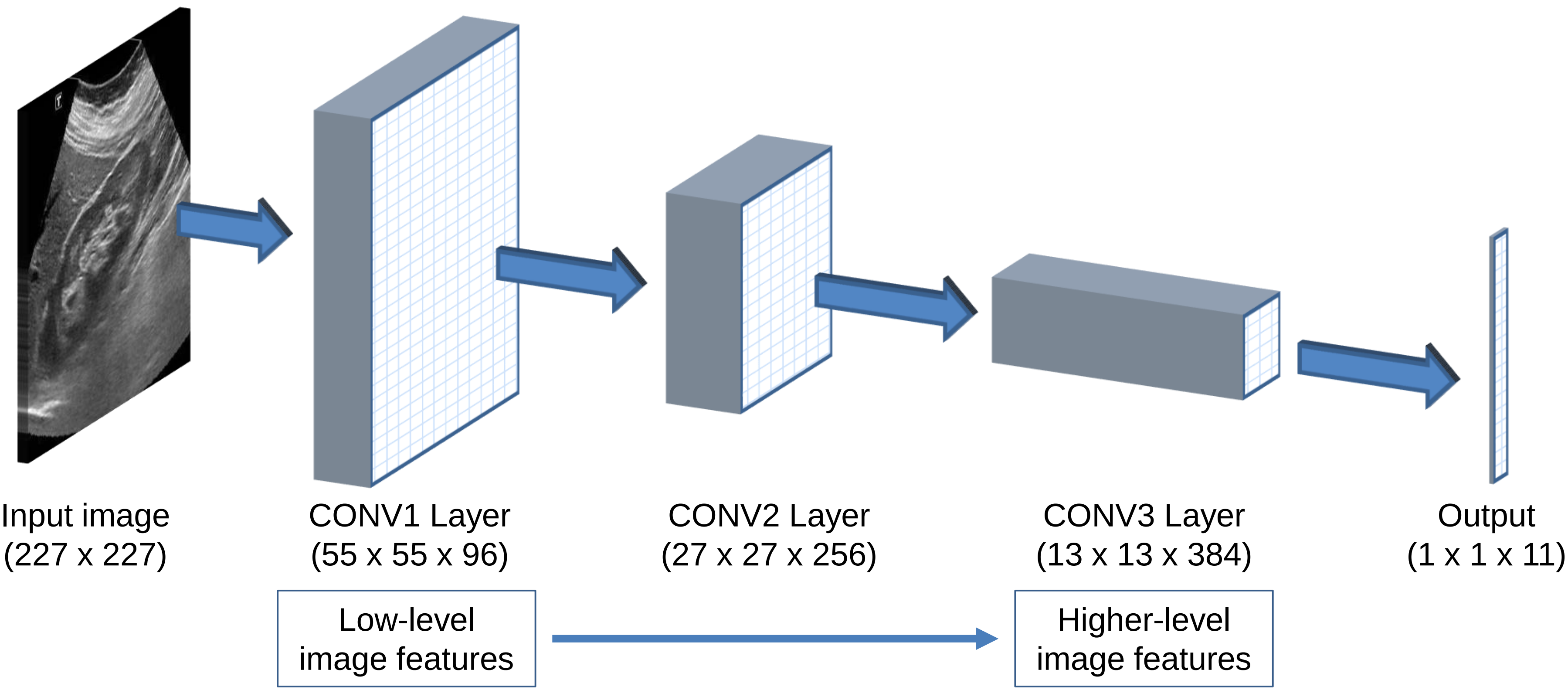
Although an individual artificial neuron is simple, a computing architecture consisting of thousands of neurons can represent very complex functions for complex tasks such as image classification. Training such a network involves iteratively adjusting the weights of the connections based on training examples, through a process called **backpropagation**.

Convolution

For a neural network to operate on pixel values, we organize input weights into a matrix to represent a feature. The weight matrix for a feature is usually small, but since a feature may occur anywhere in the image, we can apply the same weight matrix to multiple locations in the image. This sharing of weight parameters simplifies training, and is the basis of the *convolutional* neural network.



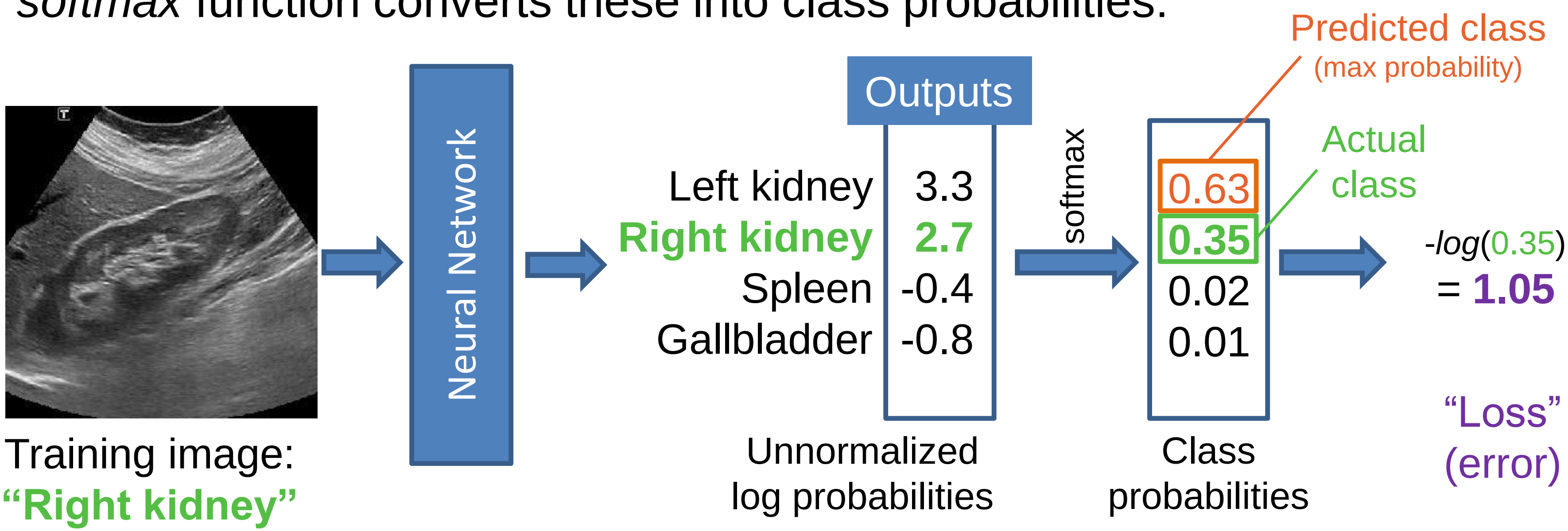
Convolutional Neural Networks



Each convolution operation representing an image feature produces a matrix, usually smaller than the input. A convolutional “layer” in a convolutional neural network (CNN) produces multiple output matrices stacked in a volume. This volume can serve as the input for another convolutional layer which detects more complex “higher-level” features in the image.

Softmax Classifier

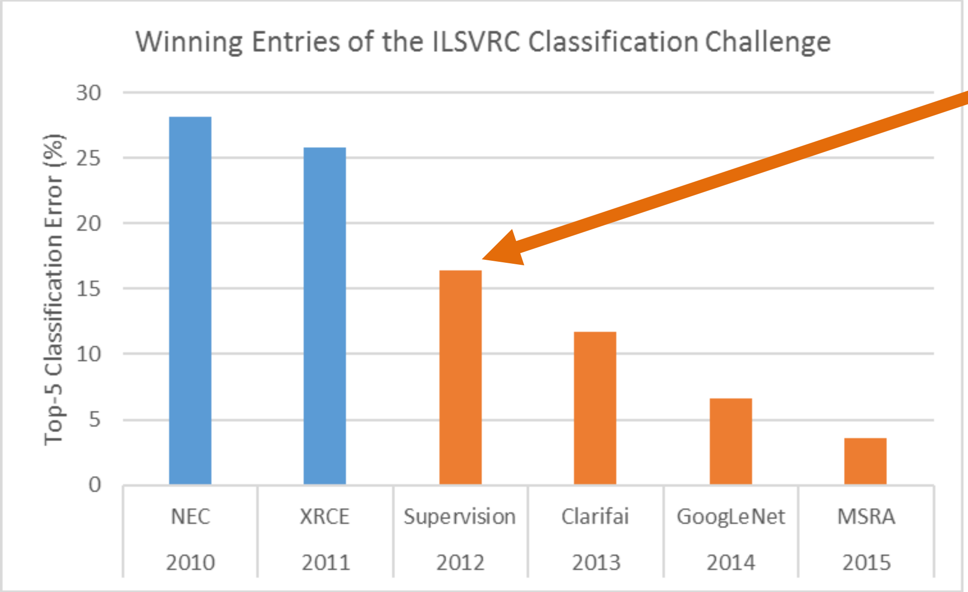
For classification, the output nodes of a neural network can be regarded as unnormalized log probabilities for each class. The *softmax* function converts these into class probabilities:



During training, a “loss” value is computed to represent the error between the network’s output predicted class and the actual class of the input. This error is **backpropagated** from the final layer to adjust the weights throughout the network in a manner to minimize the loss.

The ImageNet Challenge

Since 2010, the annual ImageNet classification challenge has been used to determine the state-of-the-art in computerized image classification. The ImageNet training data set consists of more than 1,000,000 photographs in 1000 object categories.



Winning entry = convolutional neural network

In 2012, Krizhevsky et al. from the Univ. of Toronto achieved a performance breakthrough (markedly decreased error) using a deep convolutional neural network.

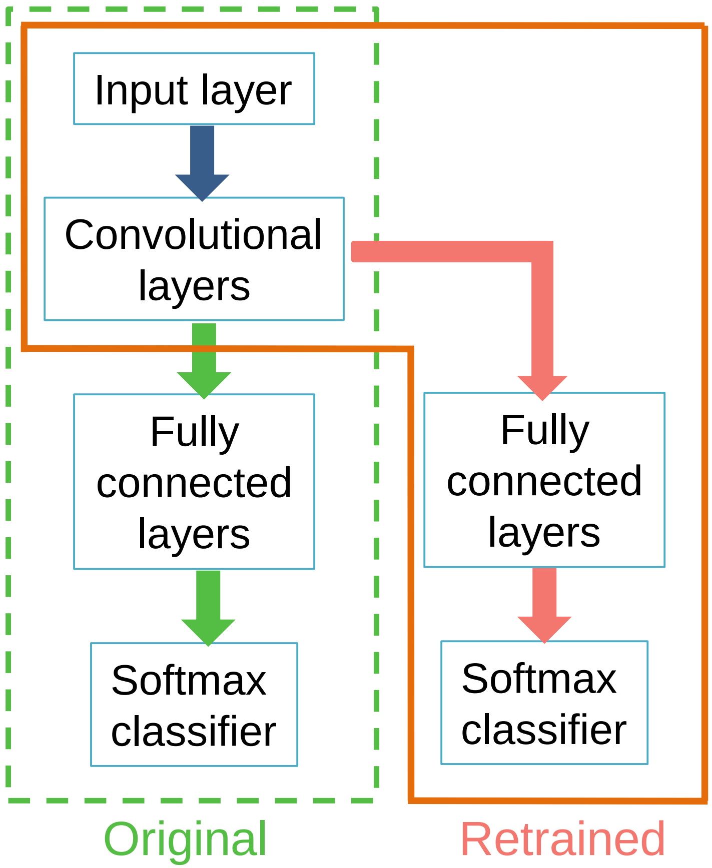
Since 2012, all winning entries (and most entries overall) have used convolutional neural networks.

Transfer Learning

Training convolutional neural networks for medical images can be challenging due to the relative lack of large labeled medical image data sets for training and testing.

One approach to solve this problem is *transfer learning*, where a network is initialized using weights derived from training on a large dataset; only a portion of the network (typically the final layers) needs to be retrained for a new smaller dataset.

The underlying assumption is that distinguishing image features may be shared among seemingly disparate data sets.

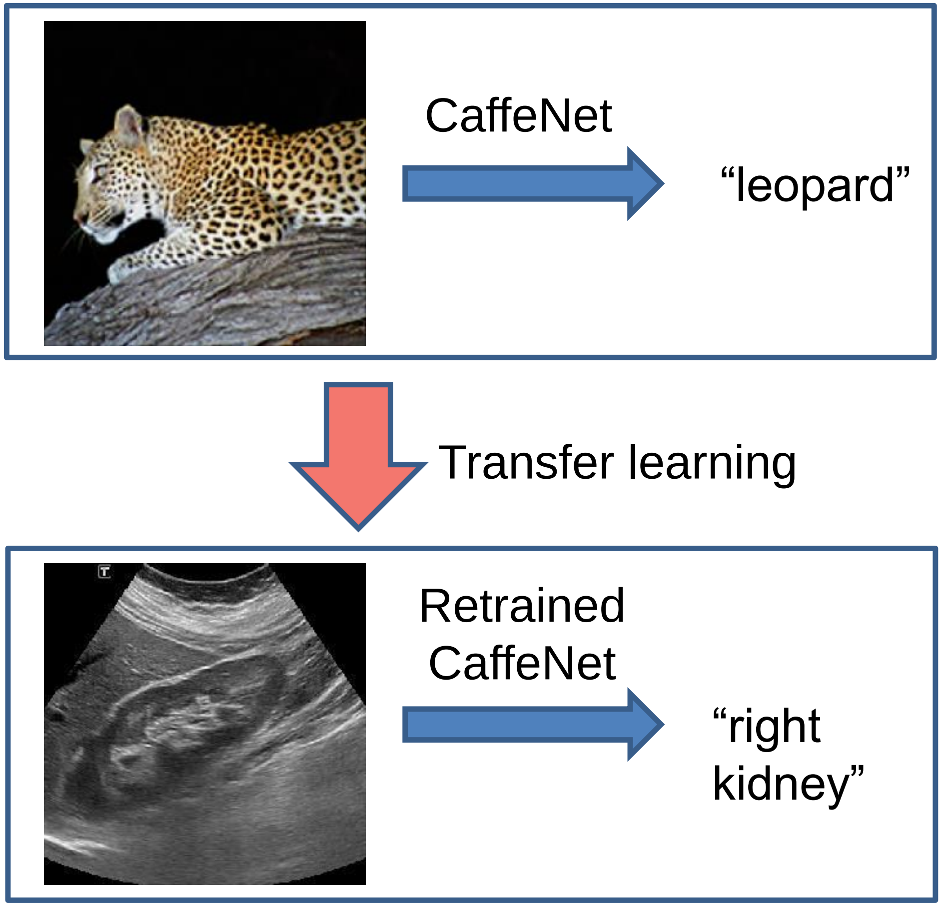


Transfer Learning in Radiology

We sought to adapt a deep neural network, originally trained for the ImageNet classification challenge, to learn to classify radiology images – specifically abdominal ultrasound images.

This experiment assesses several potential obstacles to transfer learning in radiology:

- Photographs may have different basic image features from medical imaging modalities such as ultrasound.
- The ImageNet images are color whereas most medical images are grayscale.



An Ultrasound Data Set

We constructed a data set of abdominal ultrasound images to evaluate the effectiveness of transfer learning in classifying grayscale medical images.

5518 grayscale images from 185 consecutive abdominal ultrasound studies were categorized into 11 categories based on the technologist annotation:

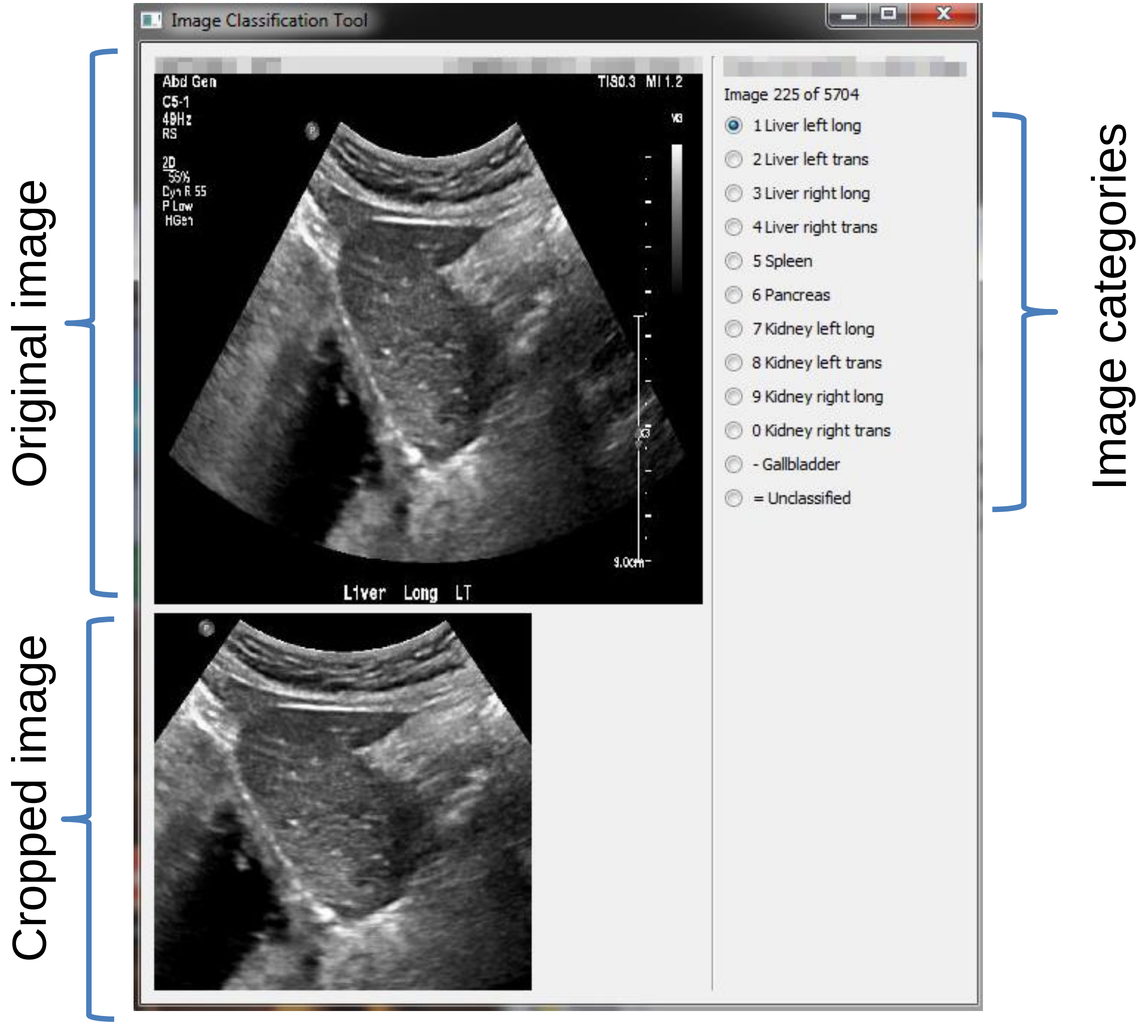
- Liver left longitudinal
- Liver left transverse
- Liver right longitudinal
- Liver right transverse
- Spleen
- Pancreas
- Kidney left longitudinal
- Kidney left transverse
- Kidney right longitudinal
- Kidney right transverse
- Gallbladder

Other images were excluded:

- Images with color or spectral Doppler
- Images with annotations or measurements
- Images with very limited or no anatomy of the labeled organ

All images were cropped to a central square, and downsampled to 256 x 256 resolution.

Ground Truth Labeling



A typically arduous task in preparation of large data sets is the labeling of “ground truth” categories of the clinical images, which is performed manually.

To facilitate manual labeling of this data set, we built a simple custom graphical user interface using the PyQt framework.

Training and Testing

Entire data set

5518 ultrasound images
(185 studies)

4095 images
(136 studies)

Training Set

1423 images
(49 studies)

Test Set

Common practice in machine learning studies is to randomly divide the available data into a training set used to train and optimize the model, and a test set to evaluate the model.

Since images in a given patient's study may be correlated, we group the images by study when dividing the data set.

Category	Training	Test	Total
Liver left longitudinal	482	191	673
Liver left transverse	464	164	628
Liver right longitudinal	531	171	702
Liver right transverse	653	223	876
Spleen	137	48	185
Pancreas	273	104	377
Kidney left longitudinal	183	65	248
Kidney left transverse	318	99	417
Kidney right longitudinal	193	67	260
Kidney right transverse	285	93	378
Gallbladder	576	198	774
Total	**4095**	**1423**	**5518**

Training Hardware and Software

Hardware

- Training of deep learning systems is often performed with graphical processing units (GPUs) which speed up matrix computations through parallel processing.
- However, hardware requirements may be modest for transfer learning.
- Our (suboptimal!) hardware:
  - 64-bit Windows desktop PC, Intel Core i7 4770
  - 8 GB RAM
  - No GPU

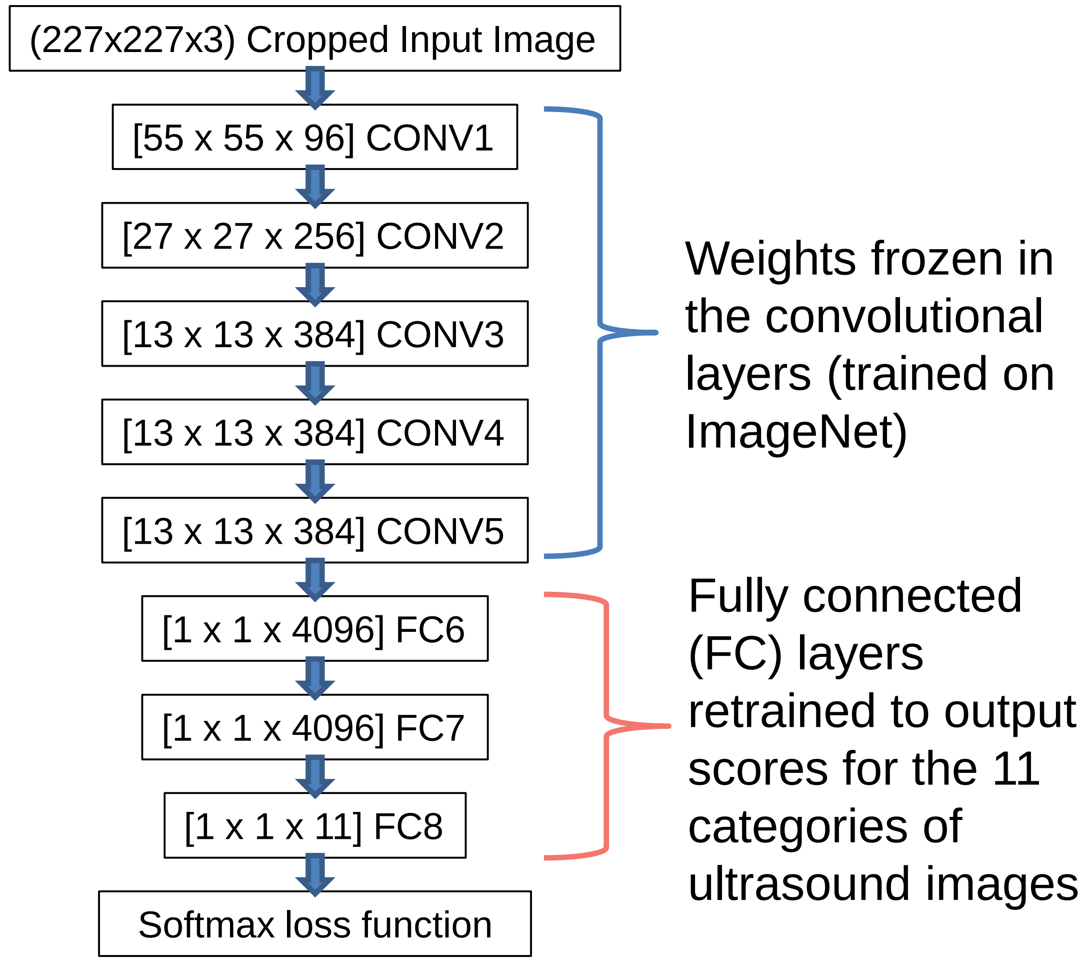
Software

- Many open source frameworks are now available for constructing and training multilayer neural networks.
  - e.g. Torch7, TensorFlow, Theano, CNTK, Caffe, ...
- We used **Caffe**, an open source framework originally developed at UC Berkeley.
  - High performance for convolutional neural networks
  - “Model zoo” of pretrained neural networks

Pretrained Network: CaffeNet

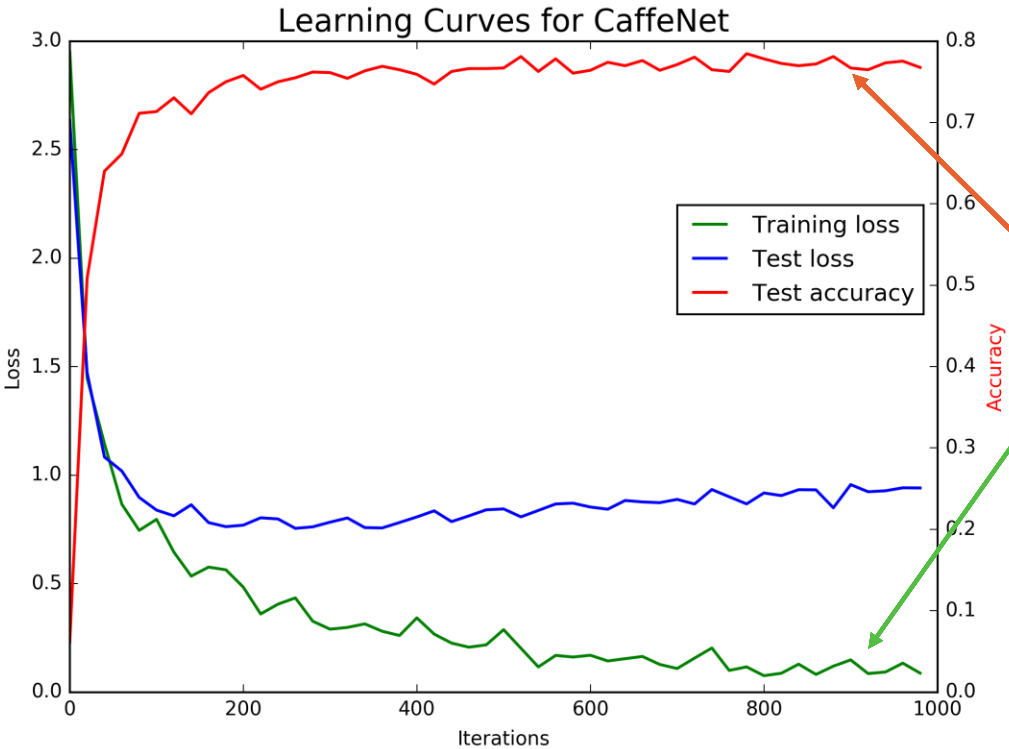
For this experiment, we used CaffeNet, a slightly modified version of the deep convolutional neural network which won the 2012 ImageNet classification challenge.

Weights of the network for the convolutional layers were trained on the ImageNet data set and frozen. Only the final layers were retrained for the ultrasound data set.



Training Curves

Training the network involves repeatedly running training images through it, and using errors to adjust the weights of the network connections.



We can monitor the progress of training by plotting the average training loss, which decreases toward zero (each iteration batch = 256 images).

We can also plot the accuracy of the model on the test set; the accuracy reaches a high plateau quickly.

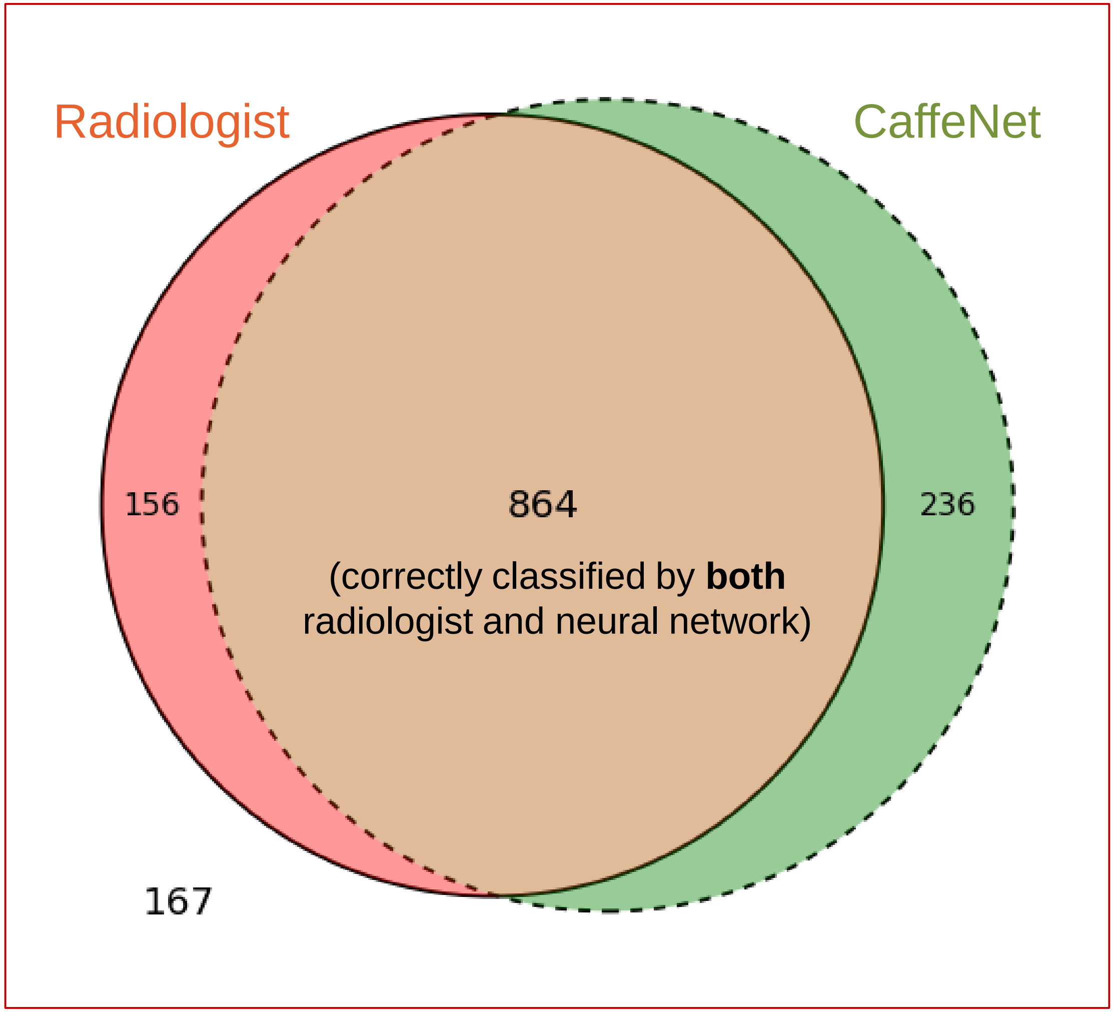
(note that test set performance was examined only after training was complete so as not to bias the training)

Accuracy Comparison

The trained CaffeNet network has a final classification accuracy of 77.3% on the test set.

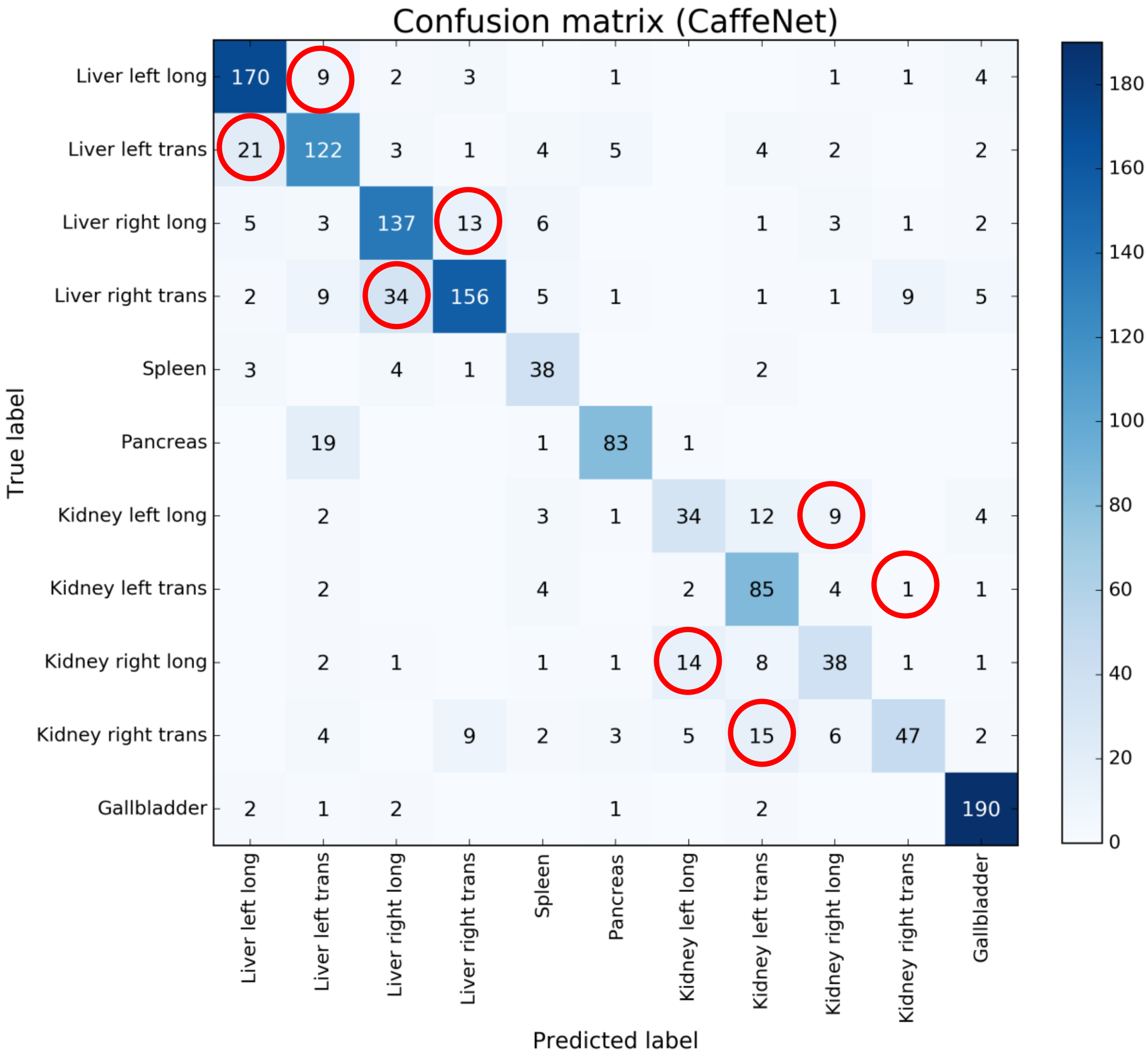
When we asked a trained radiologist to perform the same classification task on the test set, the accuracy was 71.7%.

The Venn diagram shows that there is a large overlap of cases that are classified correctly by both radiologist and neural network. 167 cases were not classified correctly by either.



Numbers within circles represent correctly classified cases

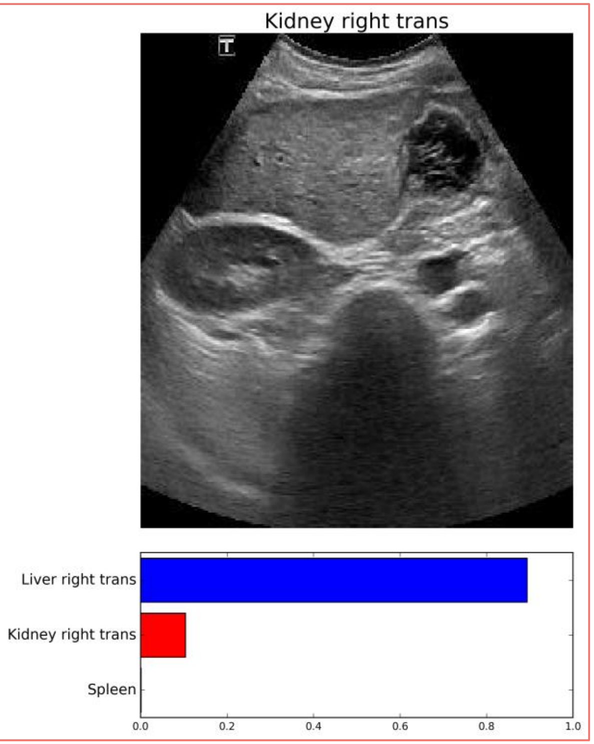
Confusion Matrix



One way to understand the performance of the neural network is to generate a confusion matrix. The matrix contains counts of the number of images corresponding to each combination of predicted and true labels (diagonal elements are correctly classified images).

The network had particular difficulty distinguishing transverse and longitudinal images of the liver, and views of the left and right kidney.

Misclassified Examples



We may also gain insight from examining examples of images misclassified by the network, but correctly classified by the radiologist.

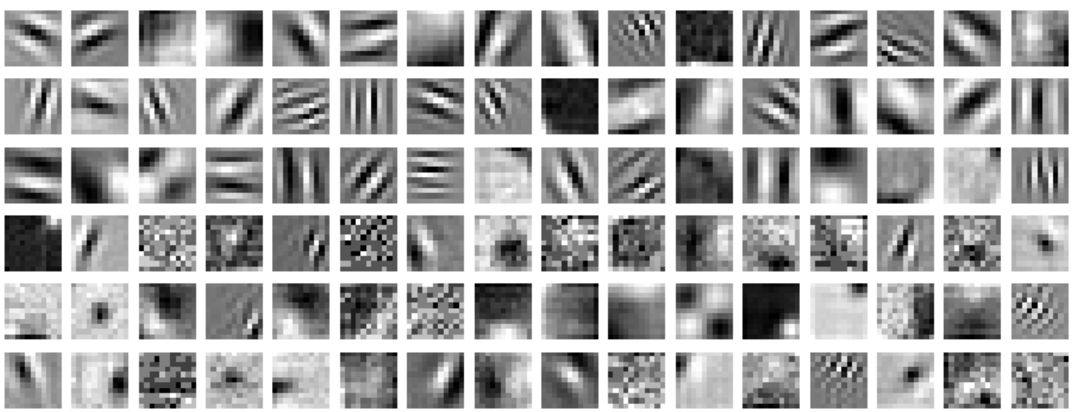
These two images were both classified by the network as transverse views of the right hepatic lobe, though the actual label was transverse image of the right kidney.

These examples illustrate inherent ambiguity in the data set, since both images *do* incorporate a portion of the right hepatic lobe.

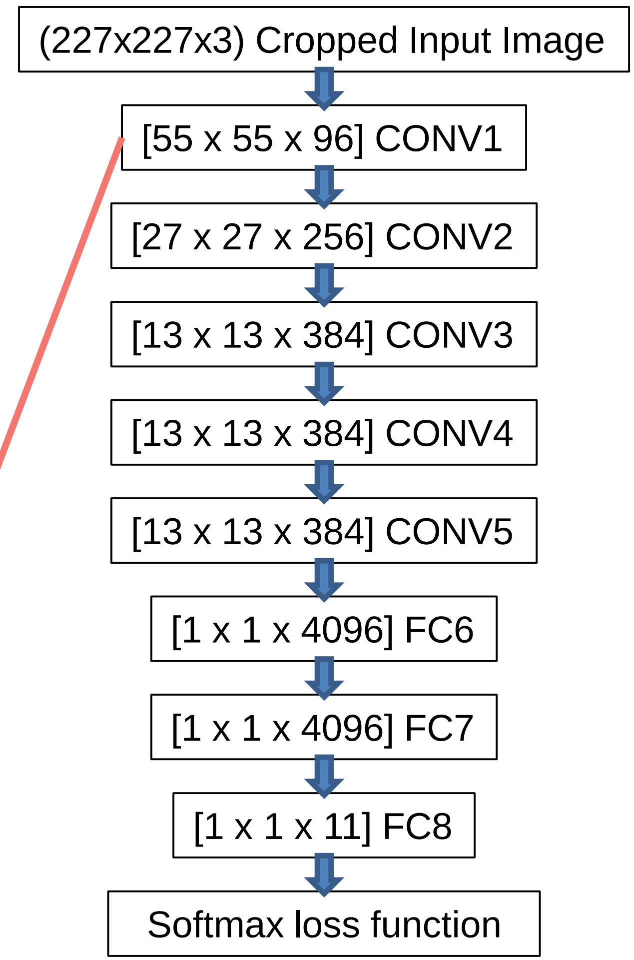
Examining the Trained Network

Neural networks have a reputation for being inscrutable “black boxes” due to their complexity.

However, certain parts of the network are amenable to visualization. For instance, plots of the weights of the first convolutional layer appear as follows:



The features appear as structured edges and blobs in varying orientations, as one might expect for low-level image features. There are resemblances here to the receptive fields in the human visual cortex.

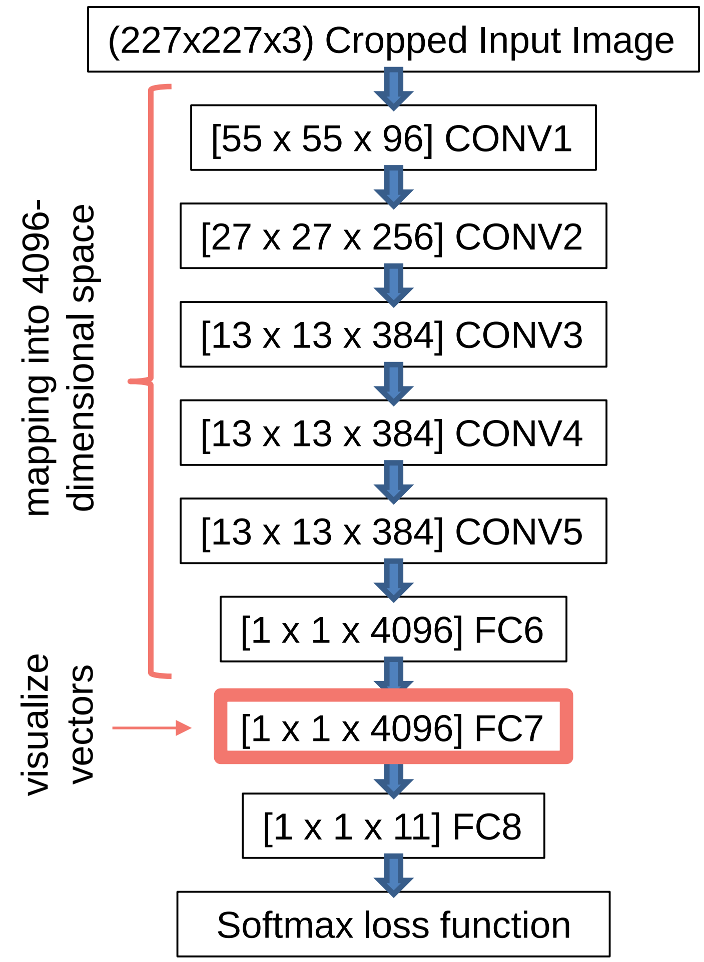


Dimensionality Reduction

We can also examine the last layer of the network prior to the final classification layer. The neural network can then be regarded as mapping an image to a 4096-element feature vector to be used for classification.

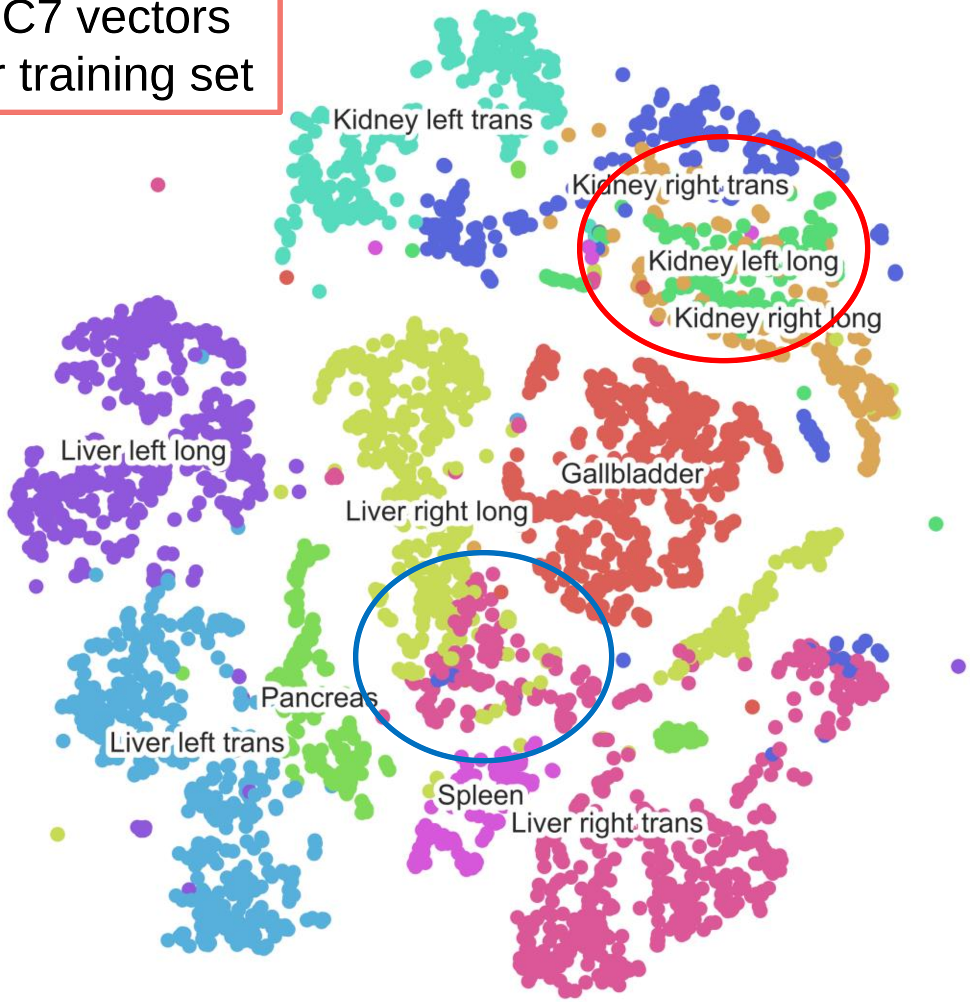
Since it is impossible to directly visualize such a high-dimensional vector, we apply *dimensionality reduction* techniques to project the vectors into a 2-dimensional space that we can visualize.

A common dimensionality reduction technique for this setting is t-stochastic neighbor embedding (t-SNE), which tends to preserve Euclidean distances; i.e., nearby vectors in the high-dimensional space are close to each other in the low-dimensional projection.



t-SNE visualization

FC7 vectors for training set



This map depicts the distribution of the 4096-element vectors to which the training cases were mapped.

Areas of overlap correspond to potential areas of classification confusion. For instance, there is significant overlap between:

- longitudinal views of the left and right kidney;
- longitudinal and transverse views of the right hepatic lobe.

Maps like these provide insight into the performance of the neural network classification.

Future Challenges

- This experiment does not assess a typical clinical imaging task. Many groups are now building data sets for clinical classification problems.
- No pretrained deep networks are available for 3D image datasets such as CT or MRI. Convolutional neural networks for 3D images need to be trained from scratch with large labeled data sets.
- Classification is only a first step in automated processing of a medical image. However, other tasks (e.g. segmentation, feature localization) appear to be amenable to deep neural networks.
- **Deep learning itself in its current form is not a panacea**.
  - Data hungry; humans by comparison do not require nearly as many labeled examples to perform accurate classification. Learning from limited or unlabeled data is an important consideration in the medical domain.
  - Relatively opaque models, difficult to delineate limitations or debug errors.
  - Need to find a way to combine representation learning with complex reasoning to produce more intelligent systems.

Summary

Deep learning uses hierarchical abstractions to learn features from data.

Transfer learning with convolutional neural networks can be effectively applied to classification of radiology images.

phillip.cheng@med.usc.edu

Keck School of Medicine of USC

References

- Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press (in preparation); 2016.
- Karpathy, A. CS231n Course Notes: Transfer Learning. http://cs231n.github.io/transfer-learning.
- Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. *Adv Neural inf Process Syst NIPS 2012*. Lake Tahoe, Nevada; 2012.
- Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv*: 1408.5093; 2014.
- LeCun Y, Bengio Y, Hinton G. Deep Learning. *Nature* 2015; 521(7553): 436-444.
- Sussakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge. *Int J Comput Vis.* 2015; 115(3): 211-252.
- Van der Maaten L, Hinton G. Visualizing Data using t-SNE. *J Mach Learn Res.* 2008; 9(Nov): 2579-2605.