# Lecture 3:
# Medical Images:
# Classification (Part 2),
# Segmentation

# Announcements

- A0 due tomorrow
- A1 will be released tomorrow, due in 2 weeks (Tue 10/18)
  - You will need to download several datasets to do the assignment. Make sure to start early!
  - 3 parts:
    - Medical image classification
    - Medical image segmentation in 2D
    - Medical image segmentation in 3D, with semi-supervised learning
- Tensorflow Review Session this Fri 1:30pm, helpful for A1

# Announcements - Course project

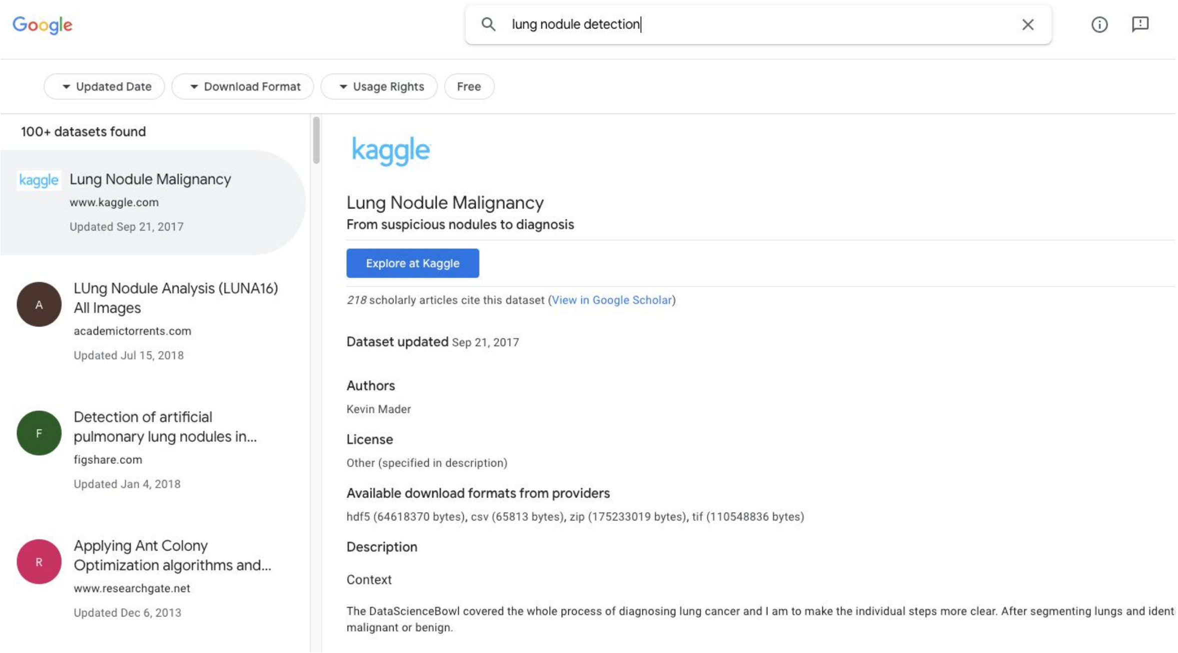
- Start thinking about your course project
    - Project proposal due Fri 10/21
    - See http://biods220.stanford.edu/finalproject.html for project components and requirements
    - **Released on Ed (#35): some project resources (open source datasets, and ideas curated from the Stanford Med School and broader community)**
        - Contributed project ideas are not vetted, you need to do your due diligence
            - Is the dataset easily accessible and well suited to machine learning? Access and play with the data before the project proposal, and make sure you can use GPU compute.
            - Is there a clearly defined task for which you can apply deep learning?
            - Can you evaluate your method?
            - Will need to answer these questions in the project proposal
        - If you are not sure, come to any of the teaching staff office hours. We are happy to discuss your project with you!

# Announcements - Course project

- Preview of graded components:
  - Proposal: Due Fri 10/21.
  - Milestone: Due Fri 11/18.
  - TA project advising sessions: after the milestone, details TBD.
  - Final project poster session: In person, during the final exam period for this course (Wed 12/14, 3:30-6:30pm)
  - Final report due: Fri 12/16.

# Google dataset search

datasetsearch.research.google.com



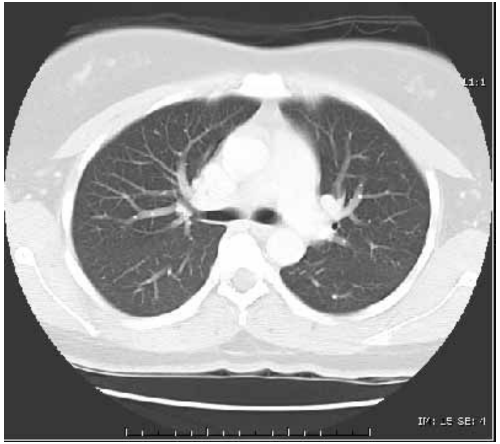
# Announcements - Review sessions

- Was in Alway M112 last Friday, but will be in **Alway M106** moving forward
- Due to incorrect location on Friday, did not get session recording
- Last year’s video recording of the material (almost identical, slightly re-arranged) is on Canvas (see pinned Ed post). Was a lecture last year, spun out into review session this year based on student feedback.
- Apparently the university may have also recorded in M112 on Friday, they are working on getting that recording out so it may also be shared.

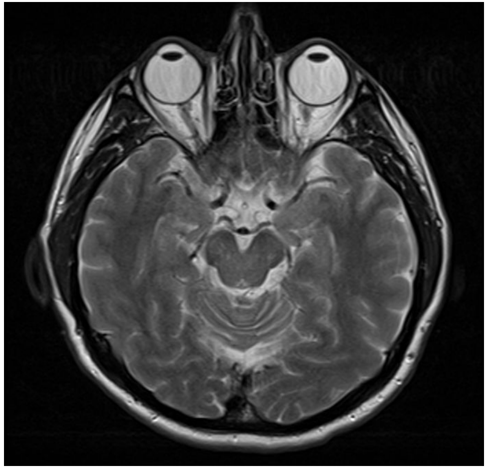
# Last time: Deep learning models for image classification

E.g.:


X-rays (invented 1895).



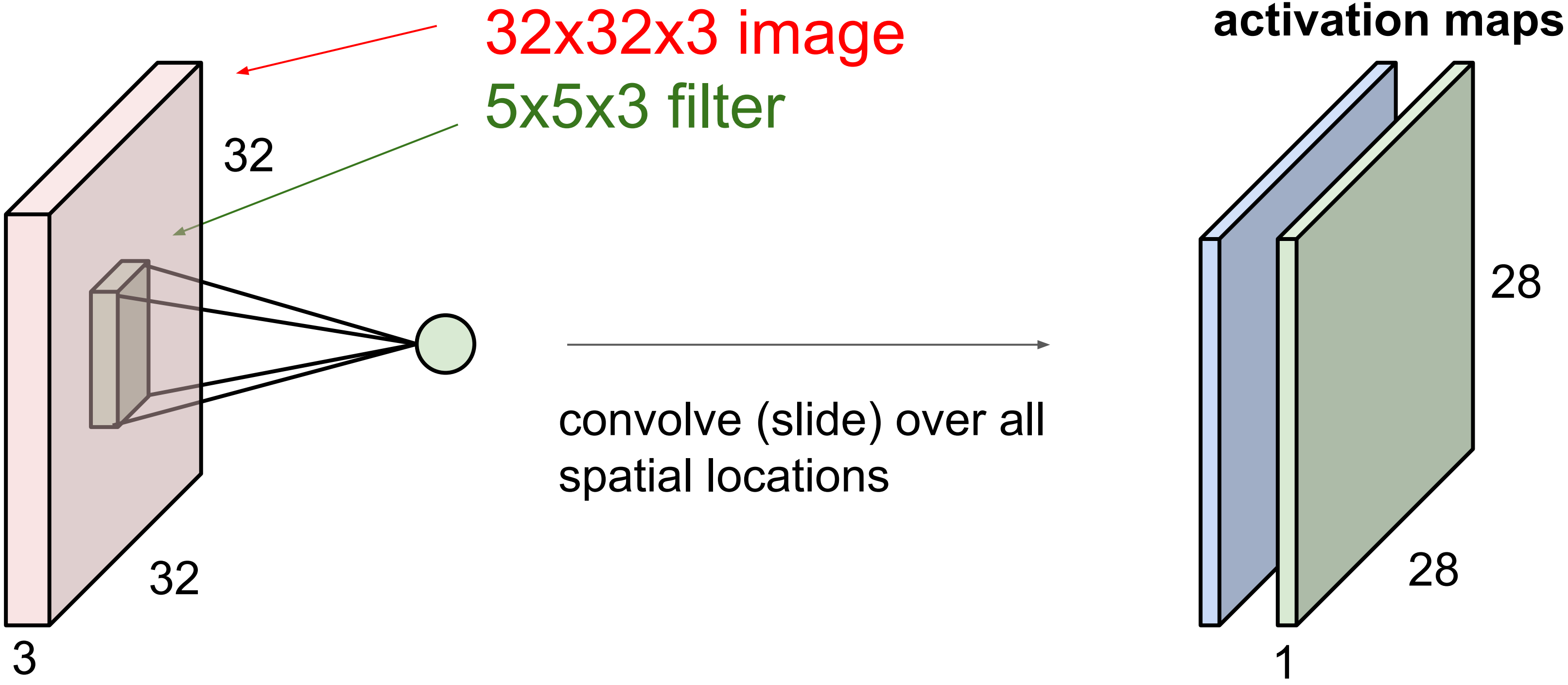
CT (invented 1972).



MRI (invented 1977).

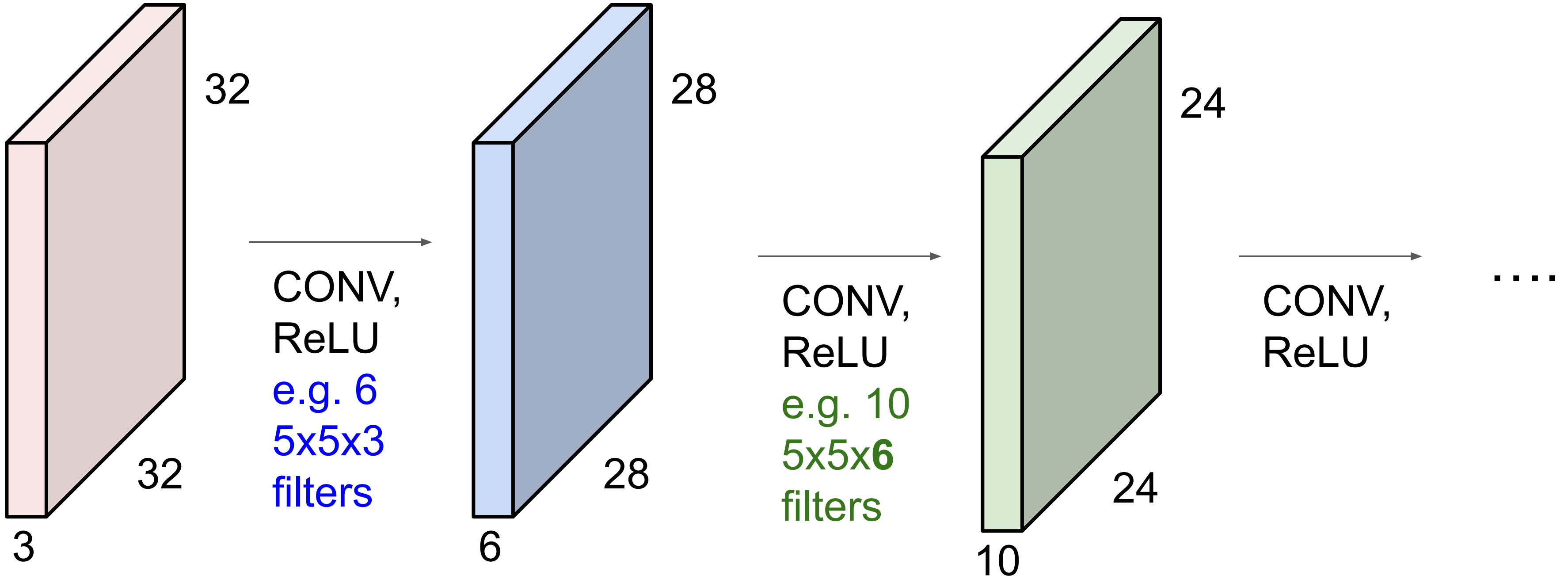
# Convolutional layer

consider a second, green filter




Slide credit: CS231n

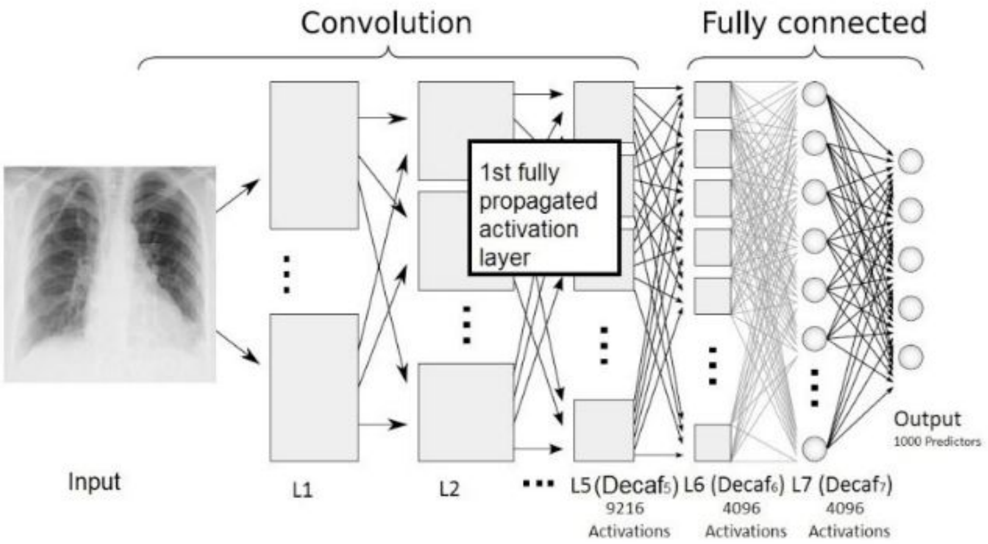
**Preview:** ConvNet (or CNN) is a sequence of Convolution Layers, interspersed with activation functions




Slide credit: CS231n

# Bar et al. 2015

- Did not train a deep learning model on the medical data
- Instead, extracted features from an AlexNet trained on ImageNet
  - 5th, 6th, and 7th layers
- Used extracted features with an SVM classifier
- Performed zero-mean unit-variance normalization of all features
- Evaluated combination with other hand-crafted image features

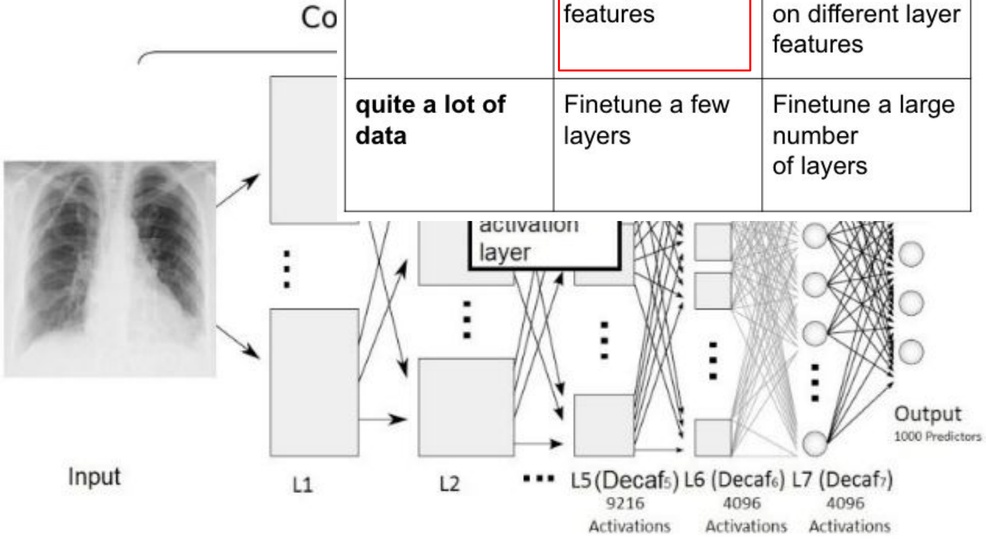



Bar et al. Deep learning with non-medical training used for chest pathology identification. SPIE, 2015.

# Bar et al. 2015

- Did not train a deep learning model on the medical data
- Instead, extracted features from an AlexNet trained on ImageNet
    - 5th, 6th, and 7th layers
- Used extracted features with an SVM classifier
- Performed zero-mean unit-variance normalization of all features
- Evaluated combination with other hand-crafted image features

| | **very similar dataset** | **very different dataset** |
|---|---|---|
| **very little data** | Use Linear Classifier on top layer features | You're in trouble… Try linear classifier on different layer features |
| **quite a lot of data** | Finetune a few layers | Finetune a large number of layers |




Bar et al. Deep learning with non-medical training used for chest pathology identification. SPIE, 2015.

# Bar et al. 2015

Table 1. Right Pleural Effusion Condition.

| | Low Level | | High Level | Deep | | | Fusion |
|---|---|---|---|---|---|---|---|
| | LBP | GIST | PiCoDes | Decaf L5 | Decaf L6 | Decaf L7 | PiCoDes+Decaf L5 |
| **Sensitivity** | 0.71 | 0.79 | 0.79 | 0.93 | 0.86 | 0.86 | **0.93** |
| **Specificity** | 0.77 | 0.92 | 0.91 | 0.84 | 0.86 | 0.80 | **0.84** |
| **AUC** | 0.75 | 0.93 | 0.91 | 0.92 | 0.91 | 0.84 | **0.93** |

Table 2. Healthy vs. Pathology.

| | Low Level | | High Level | Deep | | | Fusion |
|---|---|---|---|---|---|---|---|
| | LBP | GIST | PiCoDes | Decaf L5 | Decaf L6 | Decaf L7 | PiCoDes+Decaf L5 |
| **Sensitivity** | 0.65 | 0.68 | 0.59 | 0.73 | 0.89 | 0.76 | **0.81** |
| **Specificity** | 0.61 | 0.66 | 0.79 | 0.80 | 0.64 | 0.64 | **0.79** |
| **AUC** | 0.63 | 0.72 | 0.72 | 0.78 | 0.79 | 0.72 | **0.79** |

Table 3. Enlarged Heart Condition.

| | Low Level | | High Level | Deep | | | Fusion |
|---|---|---|---|---|---|---|---|
| | LBP | GIST | PiCoDes | Decaf L5 | Decaf L6 | Decaf L7 | PiCoDes+Decaf L5 |
| **Sensitivity** | 0.75 | 0.79 | 0.79 | 0.88 | 0.79 | 0.79 | **0.83** |
| **Specificity** | 0.78 | 0.81 | 0.84 | 0.78 | 0.88 | 0.77 | **0.84** |
| **AUC** | 0.80 | 0.82 | 0.87 | 0.87 | 0.84 | 0.79 | **0.89** |

Bar et al. Deep learning with non-medical training used for chest pathology identification. SPIE, 2015.

# Bar et al. 2015

Q: How might we interpret the AUC vs. CNN feature trends?

Table 1. Right Pleural Effusion Condition.

| | Low Level | | High Level | Deep | | | Fusion |
|---|---|---|---|---|---|---|---|
| | LBP | GIST | PiCoDes | Decaf L5 | Decaf L6 | Decaf L7 | PiCoDes+Decaf L5 |
| **Sensitivity** | 0.71 | 0.79 | 0.79 | 0.93 | 0.86 | 0.86 | **0.93** |
| **Specificity** | 0.77 | 0.92 | 0.91 | 0.84 | 0.86 | 0.80 | **0.84** |
| **AUC** | 0.75 | 0.93 | 0.91 | 0.92 | 0.91 | 0.84 | **0.93** |

Table 2. Healthy vs. Pathology.

| | Low Level | | High Level | Deep | | | Fusion |
|---|---|---|---|---|---|---|---|
| | LBP | GIST | PiCoDes | Decaf L5 | Decaf L6 | Decaf L7 | PiCoDes+Decaf L5 |
| **Sensitivity** | 0.65 | 0.68 | 0.59 | 0.73 | 0.89 | 0.76 | **0.81** |
| **Specificity** | 0.61 | 0.66 | 0.79 | 0.80 | 0.64 | 0.64 | **0.79** |
| **AUC** | 0.63 | 0.72 | 0.72 | 0.78 | 0.79 | 0.72 | **0.79** |

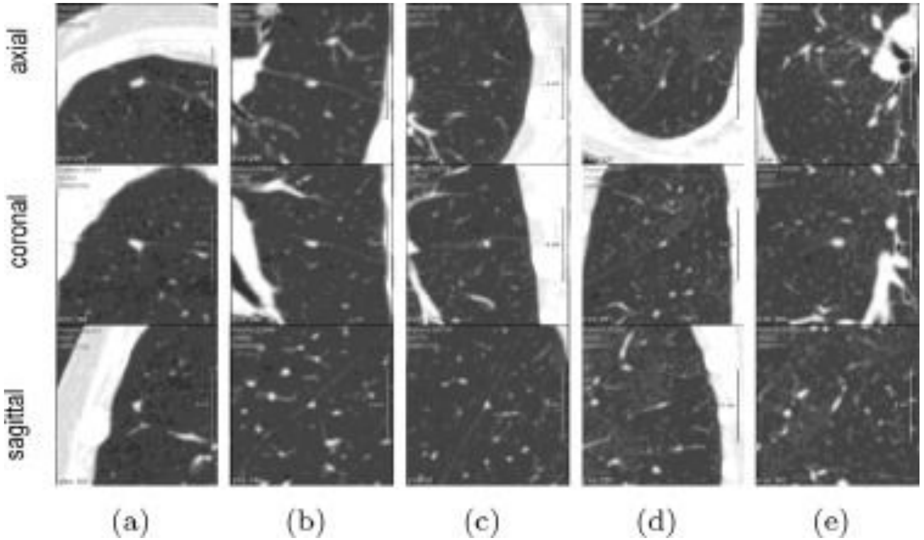
Table 3. Enlarged Heart Condition.

| | Low Level | | High Level | Deep | | | Fusion |
|---|---|---|---|---|---|---|---|
| | LBP | GIST | PiCoDes | Decaf L5 | Decaf L6 | Decaf L7 | PiCoDes+Decaf L5 |
| **Sensitivity** | 0.75 | 0.79 | 0.79 | 0.88 | 0.79 | 0.79 | **0.83** |
| **Specificity** | 0.78 | 0.81 | 0.84 | 0.78 | 0.88 | 0.77 | **0.84** |
| **AUC** | 0.80 | 0.82 | 0.87 | 0.87 | 0.84 | 0.79 | **0.89** |

Bar et al. Deep learning with non-medical training used for chest pathology identification. SPIE, 2015.

# Ciompi et al. 2015

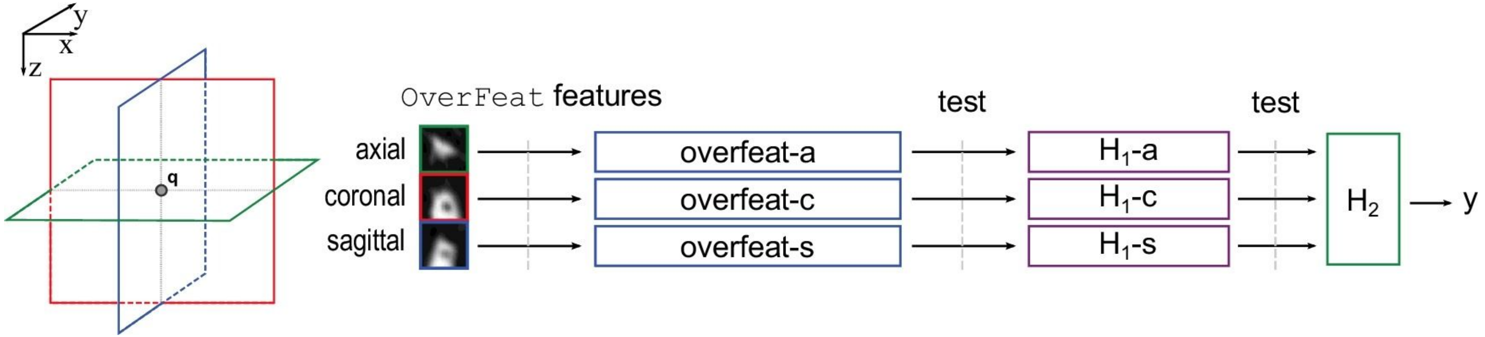
- Task: classification of lung nodules in **3D CT scans** as peri-fissural nodules (PFN, likely to be benign) or not
- Dataset: 568 nodules from 1729 scans at a single institution. (65 typical PFNs, 19 atypical PFNs, 484 non-PFNs).
- Data pre-processing: prescaling from CT hounsfield units (HU) into [0,255]. Replicate 3x across R,G,B channels to match input dimensions of ImageNet-trained CNNs.




Ciompi et al. Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. Medical Image Analysis, 2015.

# Ciompi et al. 2015

- Also extracted features from a deep learning model trained on ImageNet
  - Overfeat feature extractor (similar to AlexNet, but trained using additional losses for localization and detection)
  - To capture 3D information, extracted features from 3 different 2D views of each nodule, then input into 2-stage classifier (independent predictions on each view first, then outputs combined into second classifier).




Ciompi et al. Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. Medical Image Analysis, 2015.

# Gulshan et al. 2016

- **Task**: Binary classification of referable diabetic retinopathy from **retinal fundus photographs**
- **Input**: Retinal fundus photographs
- **Output**: Binary classification of referable diabetic retinopathy (y in {0,1})
    - Defined as moderate and worse diabetic retinopathy, referable diabetic macular edema, or both

Gulshan, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA, 2016.

# Gulshan et al. 2016

- **Dataset**:
  - 128,175 images, each graded by 3-7 ophthalmologists.
  - 54 total graders, each paid to grade between 20 to 62508 images.
- **Data preprocessing**:
  - Circular mask of each image was detected and rescaled to be 299 pixels wide
- **Model**:
  - Inception-v3 CNN, with ImageNet pre-training
  - Multiple BCE losses corresponding to different binary prediction problems, which were then used for final determination of referable diabetic retinopathy

Gulshan, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA, 2016.

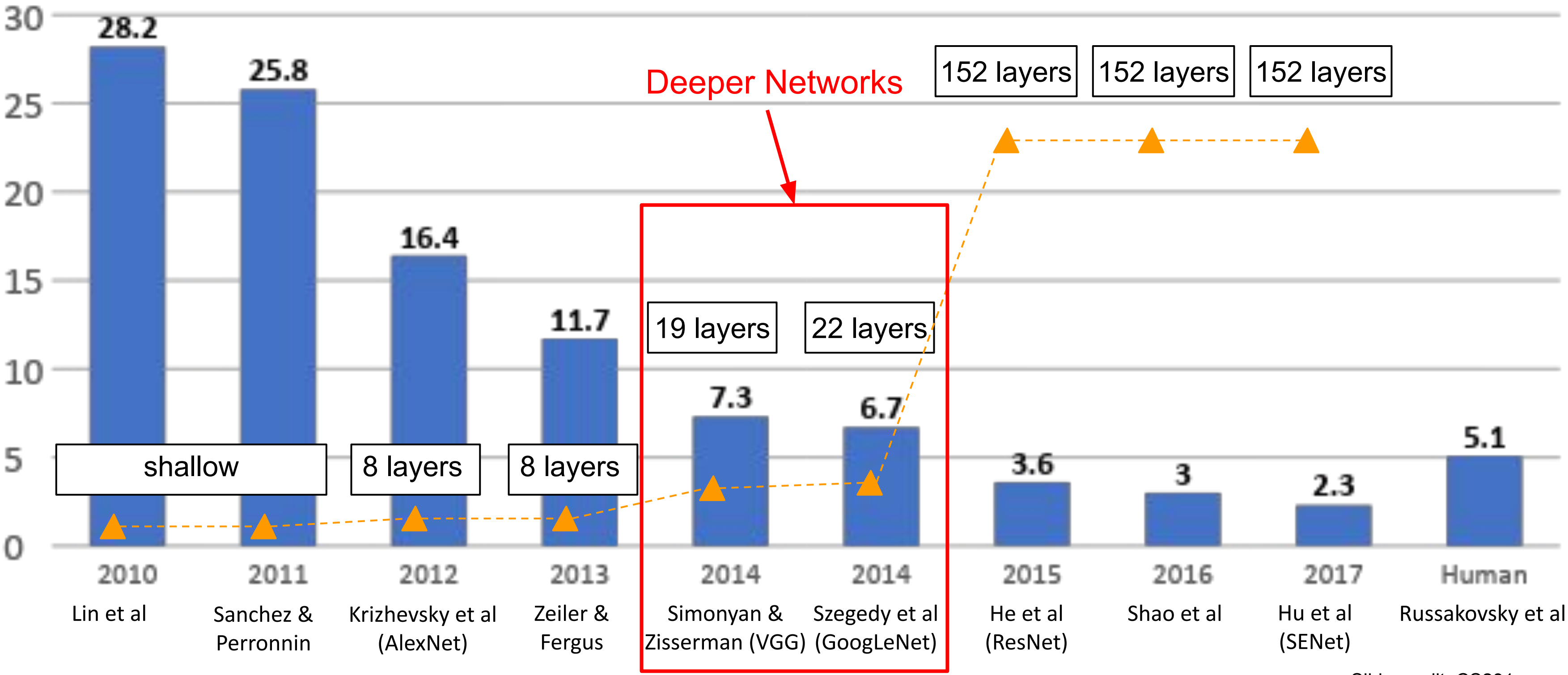
# Gulshan et al. 2016

Graders provided finer-grained labels which were then consolidated into (easier) binary prediction problems

- **Dataset**:
  - 128,175 images, each graded by 3-7 ophthalmologists.
  - 54 total graders, each paid to grade between 20 to 62508 images.
- **Data preprocessing**:
  - Circular mask of each image was detected and rescaled to be 299 pixels wide
- **Model**:
  - Inception-v3 CNN, with ImageNet pre-training
  - Multiple BCE losses corresponding to different binary prediction problems, which were then used for final determination of referable diabetic retinopathy

Gulshan, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA, 2016.

# ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners




Slide credit: CS231n

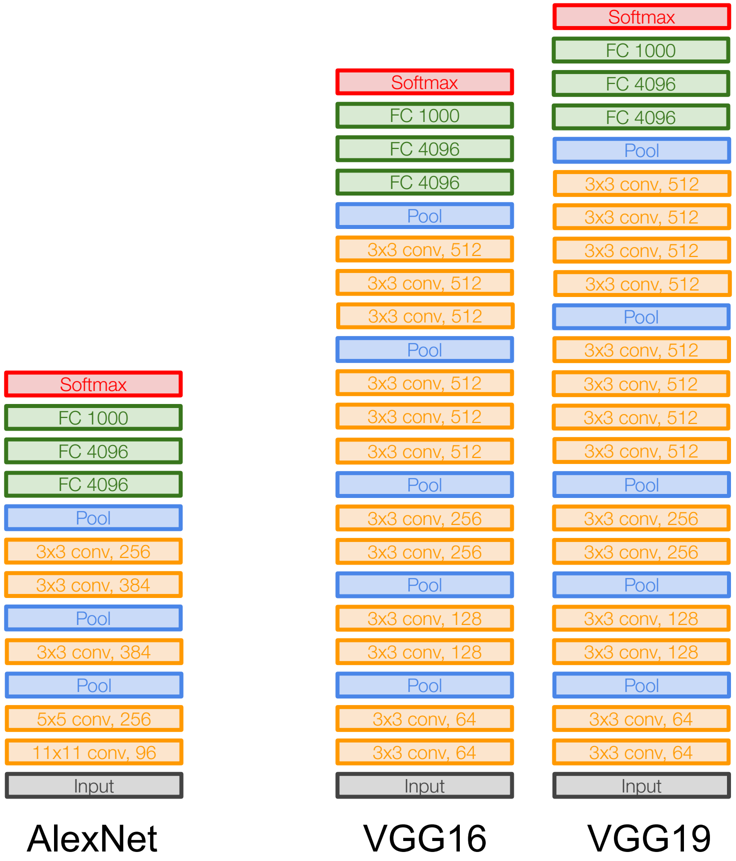
# VGGNet

*[Simonyan and Zisserman, 2014]*

Small filters, Deeper networks

8 layers (AlexNet)
-> 16 - 19 layers (VGG16Net)

Only 3x3 CONV stride 1, pad 1
and 2x2 MAX POOL stride 2

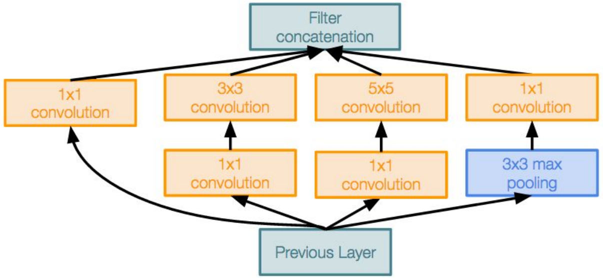



Slide credit: CS231n

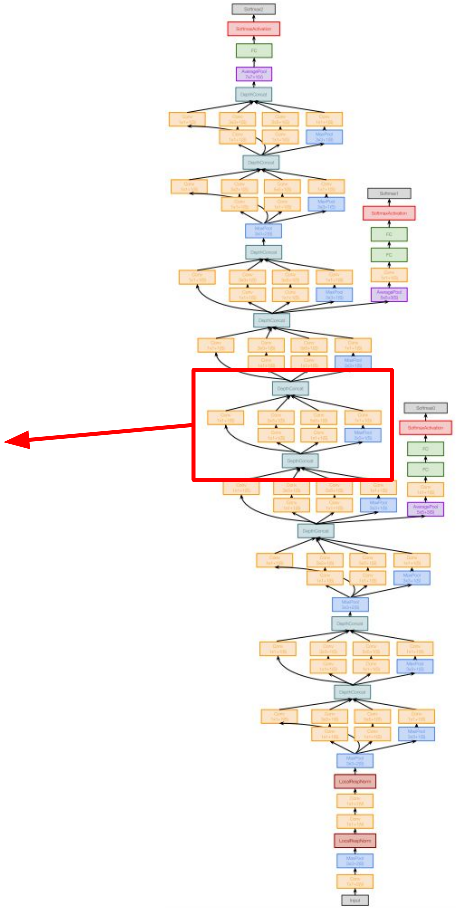
# GoogLeNet

*[Szegedy et al., 2014]*

"Inception module": design a good local network topology (network within a network) and then stack these modules on top of each other




Inception module



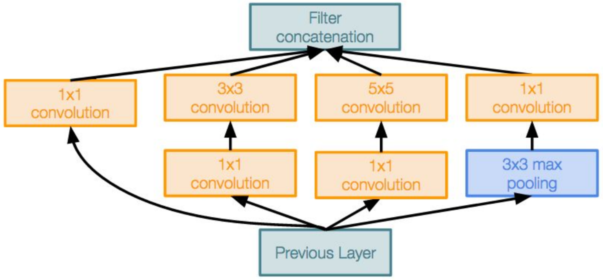
Slide credit: CS231n

# GoogLeNet

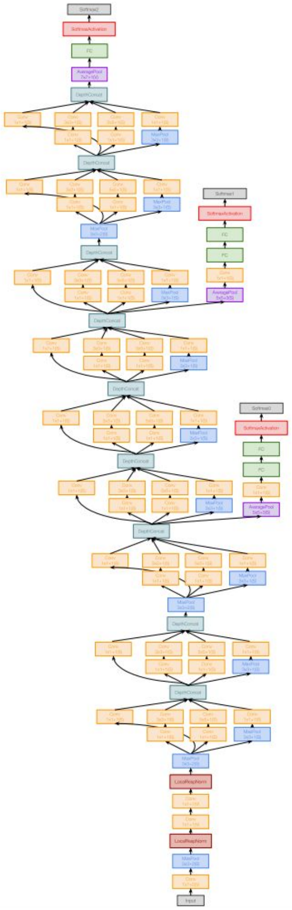
*[Szegedy et al., 2014]*

Deeper networks, with computational efficiency

- 22 layers
- Efficient "Inception" module
- Avoids expensive FC layers using a global averaging layer
- 12x less params than AlexNet




Inception module



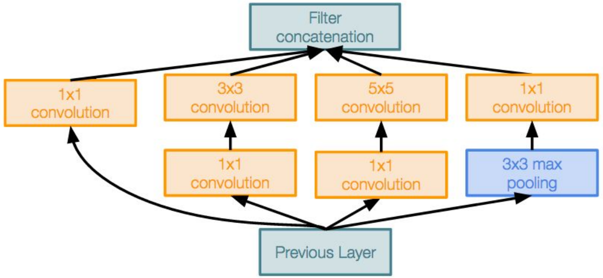
Slide credit: CS231n

# GoogLeNet

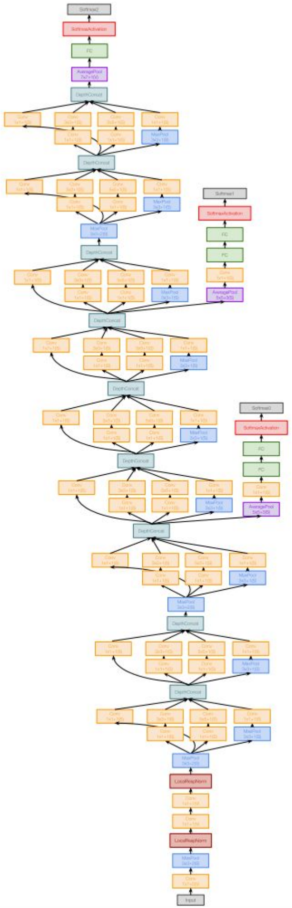
*[Szegedy et al., 2014]*

Deeper networks, with computational efficiency

- 22 layers
- Efficient "Inception" module
- Avoids expensive FC layers using a global averaging layer
- 12x less params than AlexNet




Inception module



Also called "Inception Network"

Slide credit: CS231n

# Gulshan et al. 2016

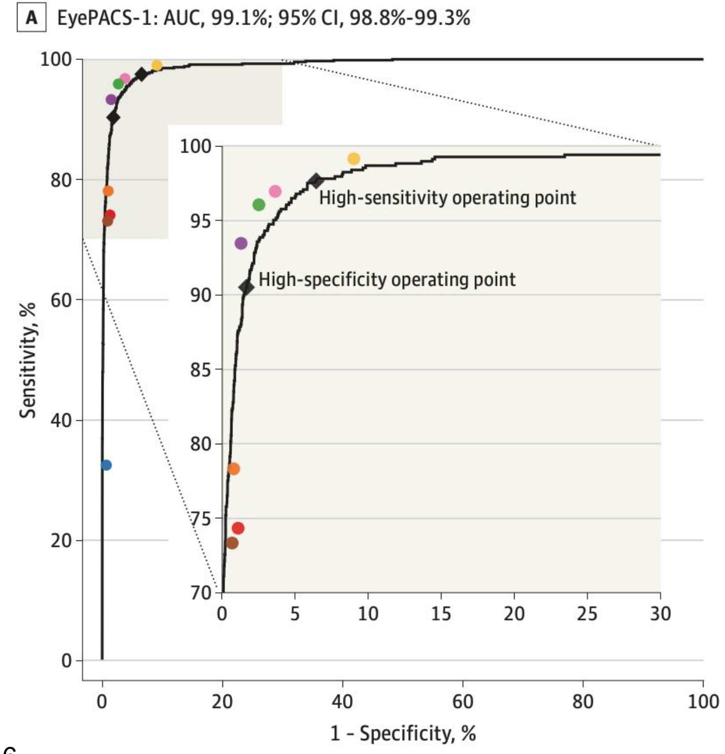
Graders provided finer-grained labels which were then consolidated into (easier) binary prediction problems

- **Dataset**:
  - 128,175 images, each graded by 3-7 ophthalmologists.
  - 54 total graders, each paid to grade between 20 to 62508 images.
- **Data preprocessing**:
  - Circular mask of each image was detected and rescaled to be 299 pixels wide
- **Model**:
  - Inception-v3 CNN, with ImageNet pre-training
  - Multiple BCE losses corresponding to different binary prediction problems, which were then used for final determination of referable diabetic retinopathy

Gulshan, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA, 2016.

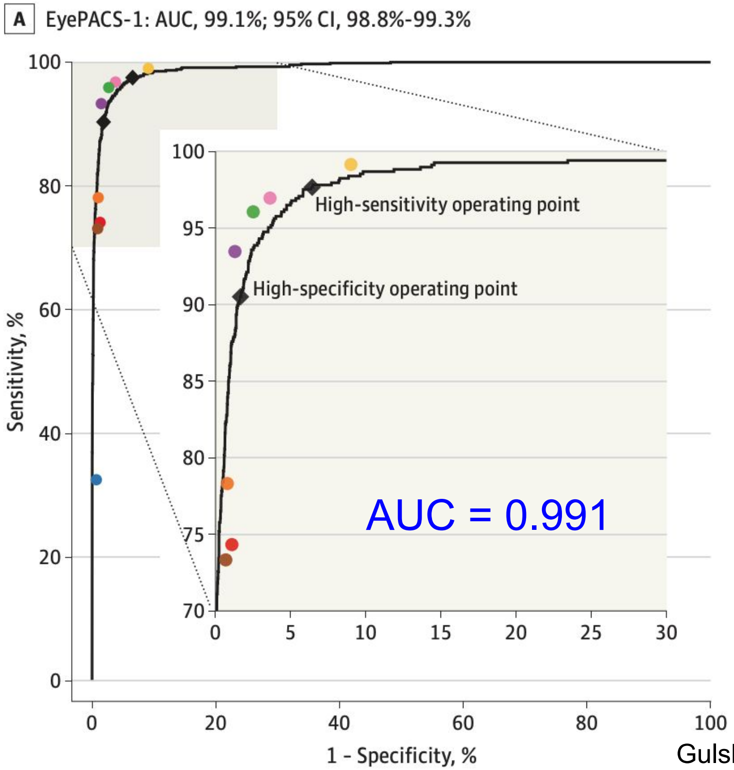
# Gulshan et al. 2016

- **Results**:
    - Evaluated using ROC curves, AUC, sensitivity and specificity analysis




Gulshan, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA, 2016.

# Gulshan et al. 2016

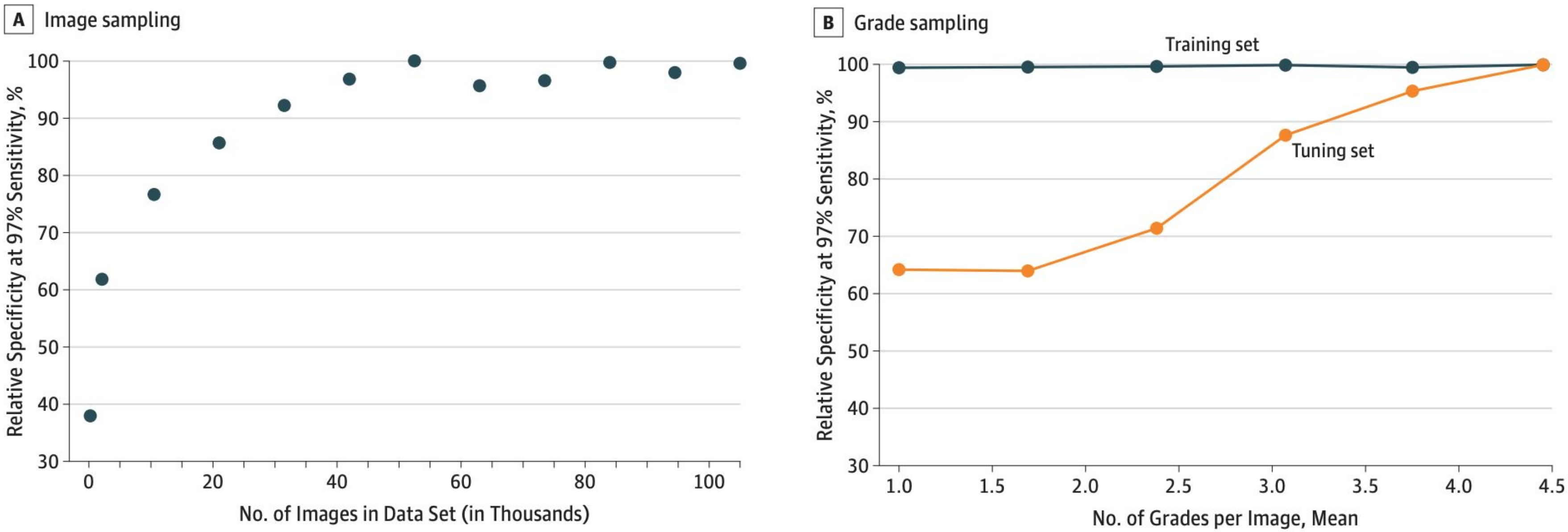



Looked at different operating points

- High-specificity point approximated ophthalmologist specificity for comparison. Should also use high-specificity to make decisions about high-risk actions.
- High-sensitivity point should be used for screening applications.

Gulshan, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA, 2016.

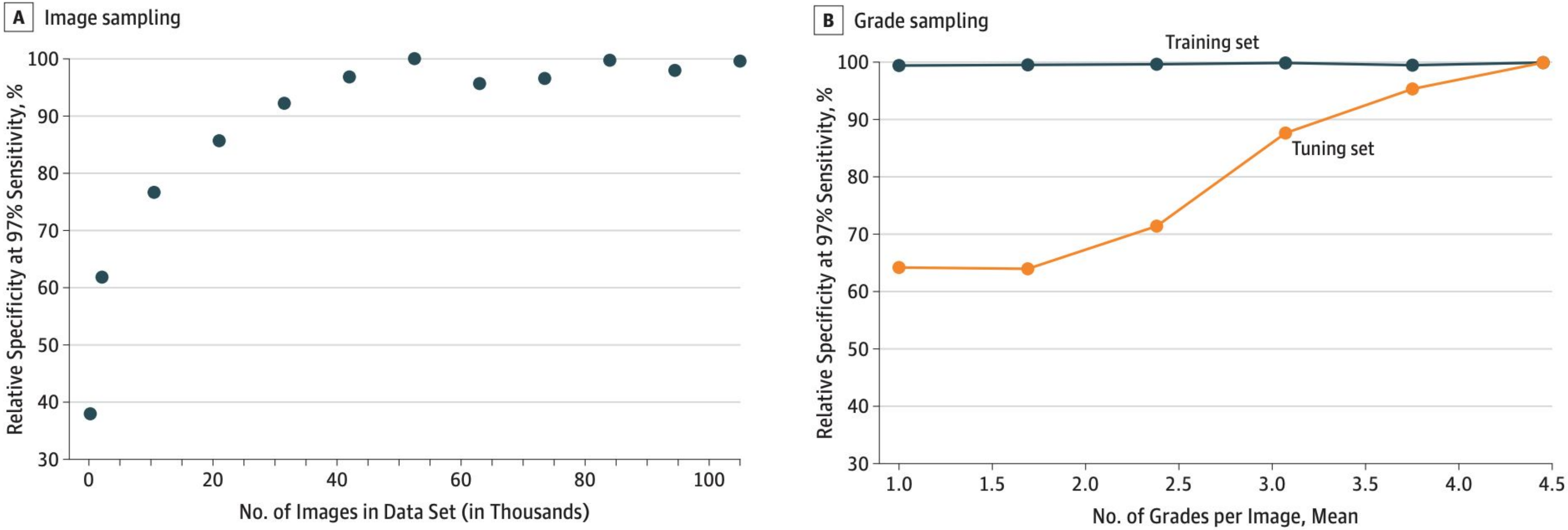
# Gulshan et al. 2016




Gulshan, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA, 2016.

# Gulshan et al. 2016

Q: What could explain the difference in trends for reducing # grades / image on training set vs. tuning set, on tuning set performance?




Gulshan, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA, 2016.

# Considering multiple possible sources of data

E.g., some with noisier / less accurate labels than others, from different hospital sites, etc.

- Expected diversity of data during deployment should be reflected in both training and test sets
  - Need to see these during training to learn how to handle them
  - Need to see these during testing to accurately evaluate the model

# Considering multiple possible sources of data

E.g., some with noisier / less accurate labels than others, from different hospital sites, etc.

- Expected diversity of data during deployment should be reflected in both training and test sets
  - Need to see these during training to learn how to handle them
  - Need to see these during testing to accurately evaluate the model

- Want test set labels to be as accurate as possible

# Considering multiple possible sources of data

E.g., some with noisier / less accurate labels than others, from different hospital sites, etc.

- Expected diversity of data during deployment should be reflected in both training and test sets
    - Need to see these during training to learn how to handle them
    - Need to see these during testing to accurately evaluate the model

- Want test set labels to be as accurate as possible

- Noisy labels is often still useful during training -- can provide useful signal in aggregate. Much larger amount, but noisy, data is *sometimes* better than small but clean data.
    - "Weakly supervised learning" is a major area of research focused on learning with large amounts of noisy or imprecise labels

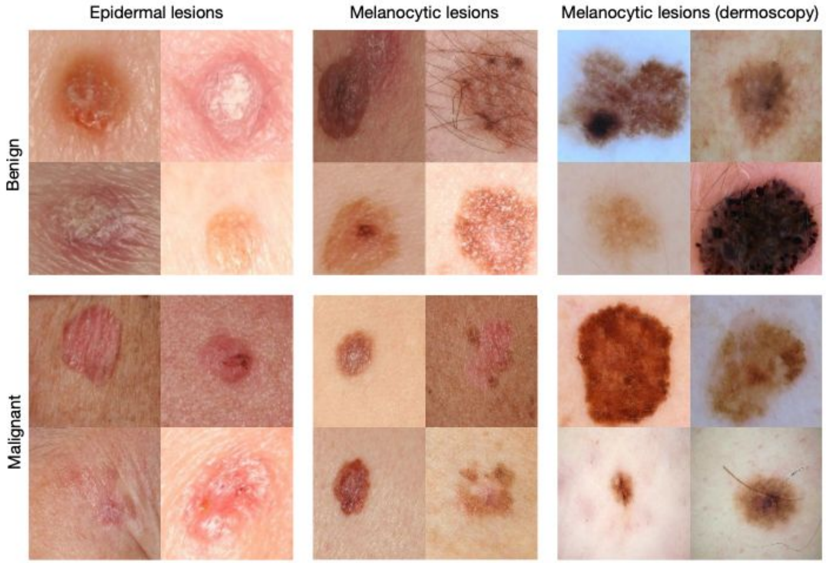
# Preview: advanced approaches for handling limited labeled data

- Semi-supervised learning
- Self-supervised learning
- Weakly supervised learning

Will talk more about these in later lectures...

# Esteva et al. 2017

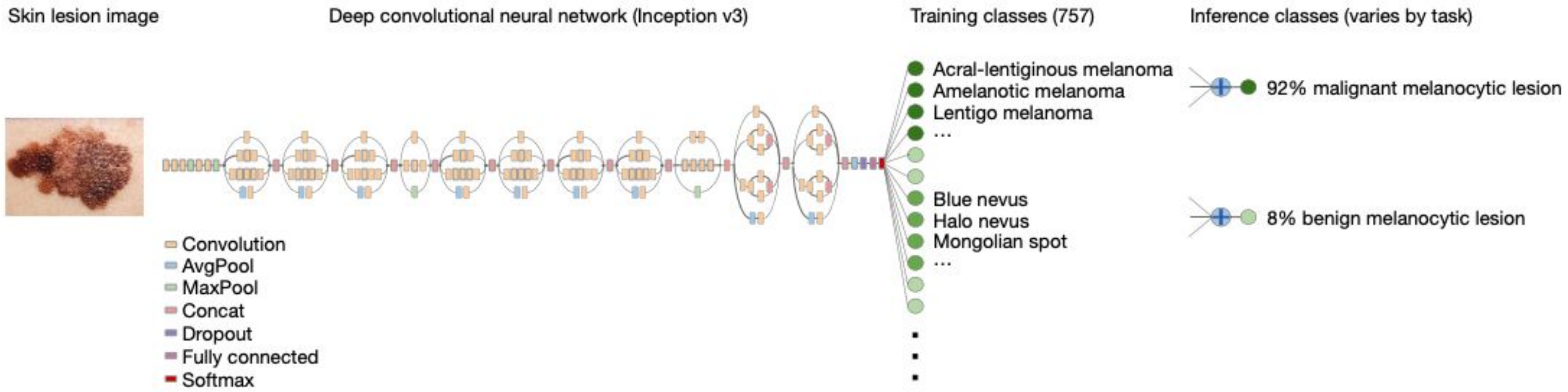
- Two binary classification tasks: malignant vs. benign lesions of epidermal or melanocytic origin
- Inception-v3 (GoogLeNet) CNN with ImageNet pre-training
- Fine-tuned on dataset of 129,450 lesions (from several sources) comprising 2,032 diseases
- Evaluated model vs. 21 or more dermatologists in various settings




Esteva*, Kuprel*, et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature, 2017.

# Esteva et al. 2017

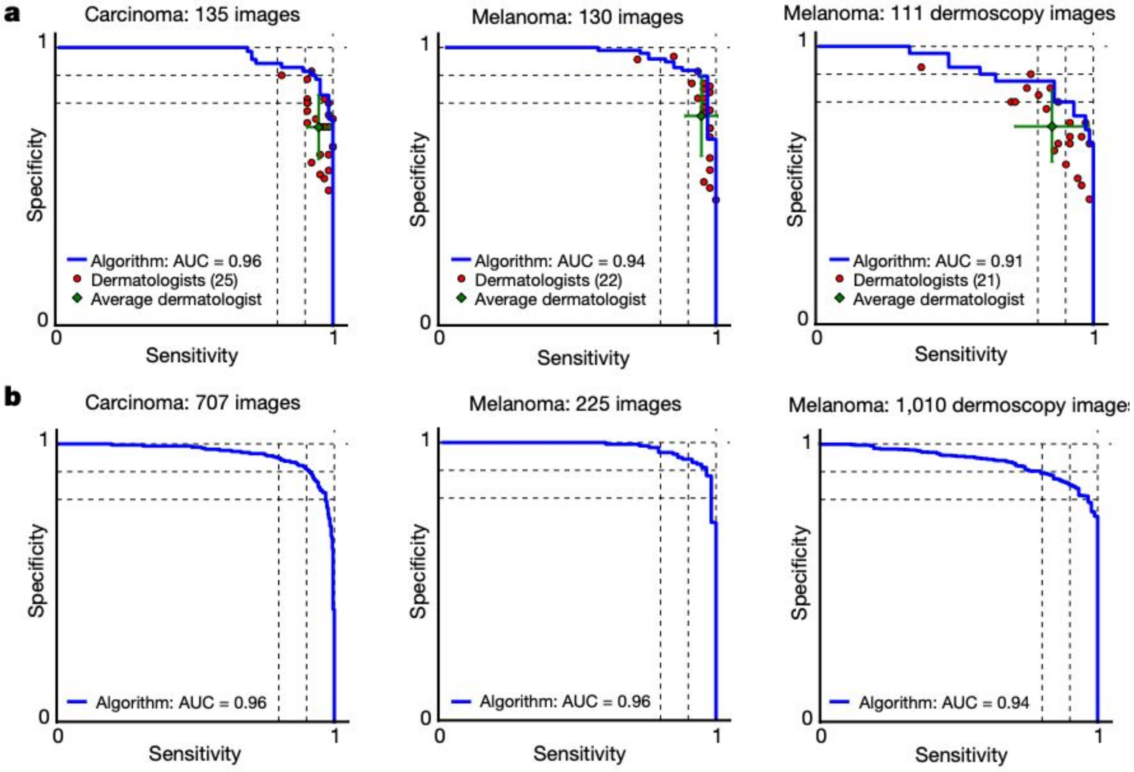
- Train on finer-grained classification (757 classes) but perform binary classification at inference time by summing probabilities of fine-grained sub-classes
- The stronger fine-grained supervision during the training stage improves inference performance!




Esteva*, Kuprel*, et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature, 2017.

# Esteva et al. 2017

- Evaluation of algorithm vs. dermatologists




Esteva*, Kuprel*, et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature, 2017.

# Lakhani and Sundaram 2017

- Binary classification of pulmonary tuberculosis from x-rays
- Four de-identified datasets
- 1007 chest x-rays (68% train, 17.1% validation, 14.9% test)
- Tried training CNNs from scratch as well as fine-tuning from ImageNet

**AUC Test Dataset**

| Parameter | Untrained | Pretrained | Untrained with Augmentation* | Pretrained with Augmentation* |
|---|---|---|---|---|
| AlexNet | 0.90 (0.84, 0.95) | 0.98 (0.95, 1.00) | 0.95 (0.90, 0.98) | 0.98 (0.94, 0.99) |
| GoogLeNet | 0.88 (0.81, 0.92) | 0.97 (0.93, 0.99) | 0.94 (0.89, 0.97) | 0.98 (0.94, 1.00) |
| Ensemble | | | | 0.99 (0.96, 1.00) |

Note.—Data in parentheses are 95% confidence interval.

* Additional augmentation of 90, 180, 270 rotations, and Contrast Limited Adaptive Histogram Equalization processing.

Lakhani and Sundaram. Deep learning at chest radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. Radiology, 2017.

# Lakhani and Sundaram 2017

- Binary classification of pulmonary tuberculosis from x-rays
- Four de-identified datasets
- 1007 chest x-rays (68% train, 17.1% validation, 14.9% test)
- Tried training CNNs from scratch as well as fine-tuning from ImageNet

**AUC Test Dataset**

| Parameter | Untrained | Pretrained | Untrained with Augmentation* | Pretrained with Augmentation* |
|---|---|---|---|---|
| AlexNet | 0.90 (0.84, 0.95) | 0.98 (0.95, 1.00) | 0.95 (0.90, 0.98) | 0.98 (0.94, 0.99) |
| GoogLeNet | 0.88 (0.81, 0.92) | 0.97 (0.93, 0.99) | 0.94 (0.89, 0.97) | 0.98 (0.94, 1.00) |
| Ensemble | | | | 0.99 (0.96, 1.00) |

Note.—Data in parentheses are 95% confidence interval.

* Additional augmentation of 90, 180, 270 rotations, and Contrast Limited Adaptive Histogram Equalization processing.

All training images were resized to 256x256 and underwent base data augmentation of random 227x227 cropping and mirror images. Additional data augmentation experiments in results table.

Lakhani and Sundaram. Deep learning at chest radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. Radiology, 2017.

# Lakhani and Sundaram 2017

- Binary classification of pulmonary tuberculosis from x-rays
- Four de-identified datasets
- 1007 chest x-rays (68% train, 17.1% validation, 14.9% test)
- Tried training CNNs from scratch as well as fine-tuning from ImageNet

**AUC Test Dataset**

| Parameter | Untrained | Pretrained | Untrained with Augmentation* | Pretrained with Augmentation* |
|---|---|---|---|---|
| AlexNet | 0.90 (0.84, 0.95) | 0.98 (0.95, 1.00) | 0.95 (0.90, 0.98) | 0.98 (0.94, 0.99) |
| GoogLeNet | 0.88 (0.81, 0.92) | 0.97 (0.93, 0.99) | 0.94 (0.89, 0.97) | 0.98 (0.94, 1.00) |
| Ensemble | | | | 0.99 (0.96, 1.00) |

Note.—Data in parentheses are 95% confidence interval.

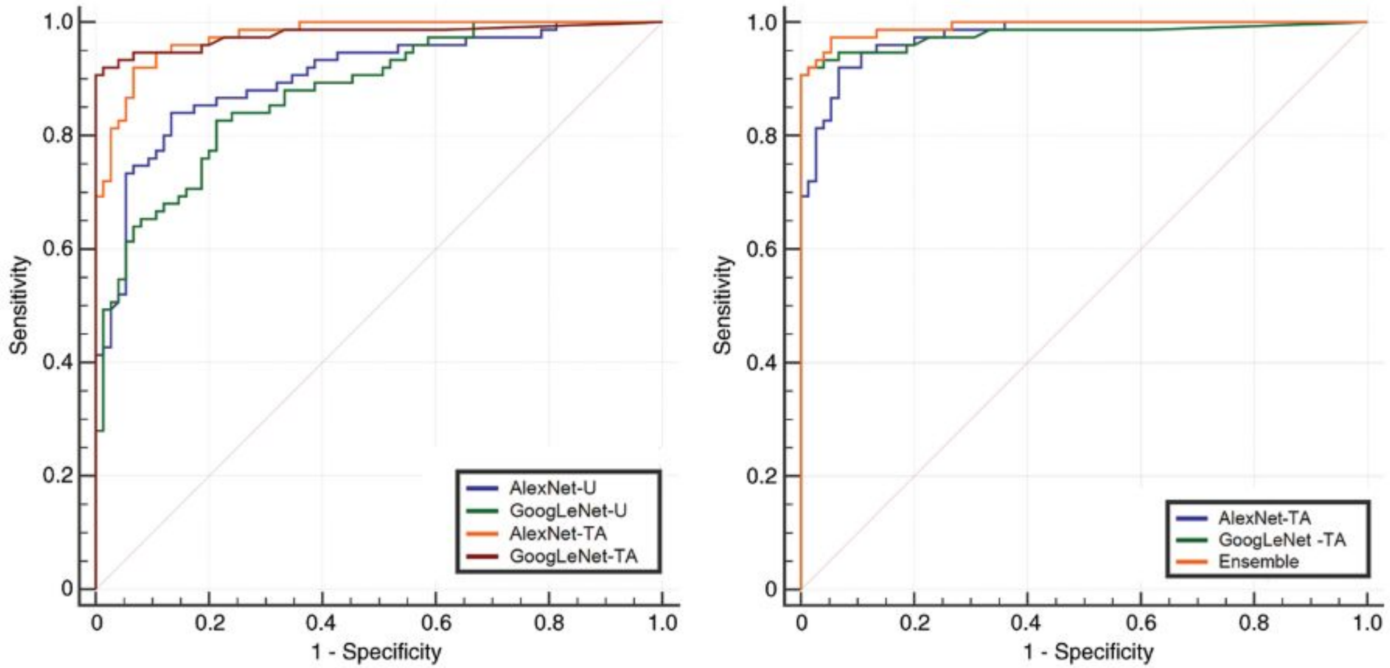
* Additional augmentation of 90, 180, 270 rotations, and Contrast Limited Adaptive Histogram Equalization processing.

All training images were resized to 256x256 and underwent base data augmentation of random 227x227 cropping and mirror images. Additional data augmentation experiments in results table.

Often resize to match input size of pre-trained networks. Also fine approach to making high-res dataset easier to work with!

Lakhani and Sundaram. Deep learning at chest radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. Radiology, 2017.

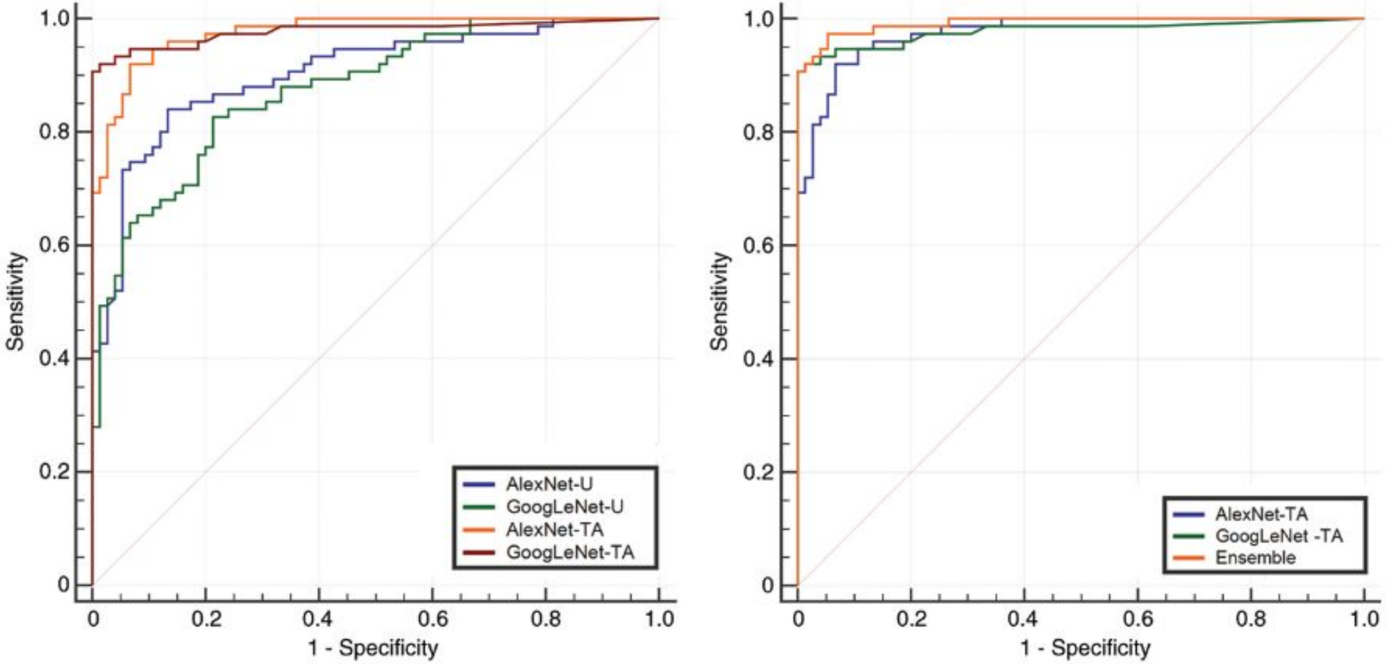
# Lakhani and Sundaram 2017




Lakhani and Sundaram. Deep learning at chest radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. Radiology, 2017.

# Lakhani and Sundaram 2017

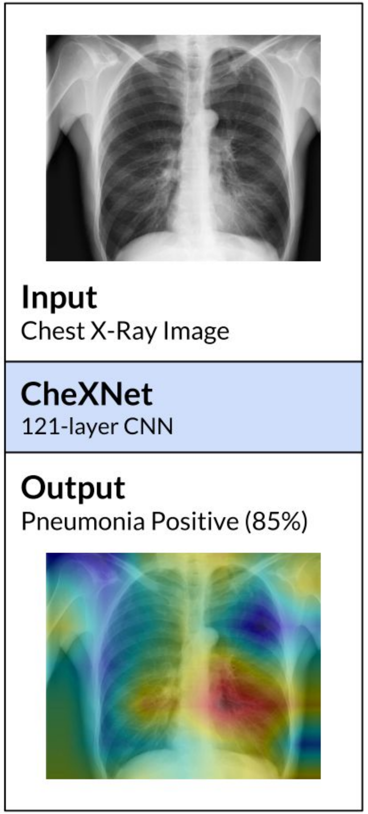
Performed further analysis at optimal threshold determined by the Youden Index.




Lakhani and Sundaram. Deep learning at chest radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. Radiology, 2017.

# Rajpurkar et al. 2017

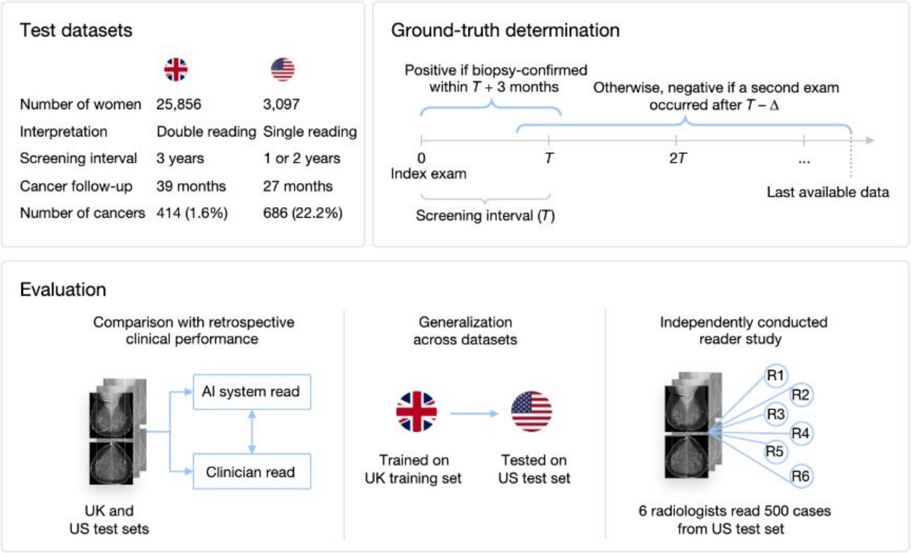
- Binary classification of pneumonia presence in chest X-rays
- Used ChestX-ray14 dataset with over 100,000 frontal X-ray images with 14 diseases
- 121-layer DenseNet CNN
- Compared algorithm performance with 4 radiologists
- Also applied algorithm to other diseases to surpass previous state-of-the-art on ChestX-ray14




Rajpurkar et al. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. 2017.

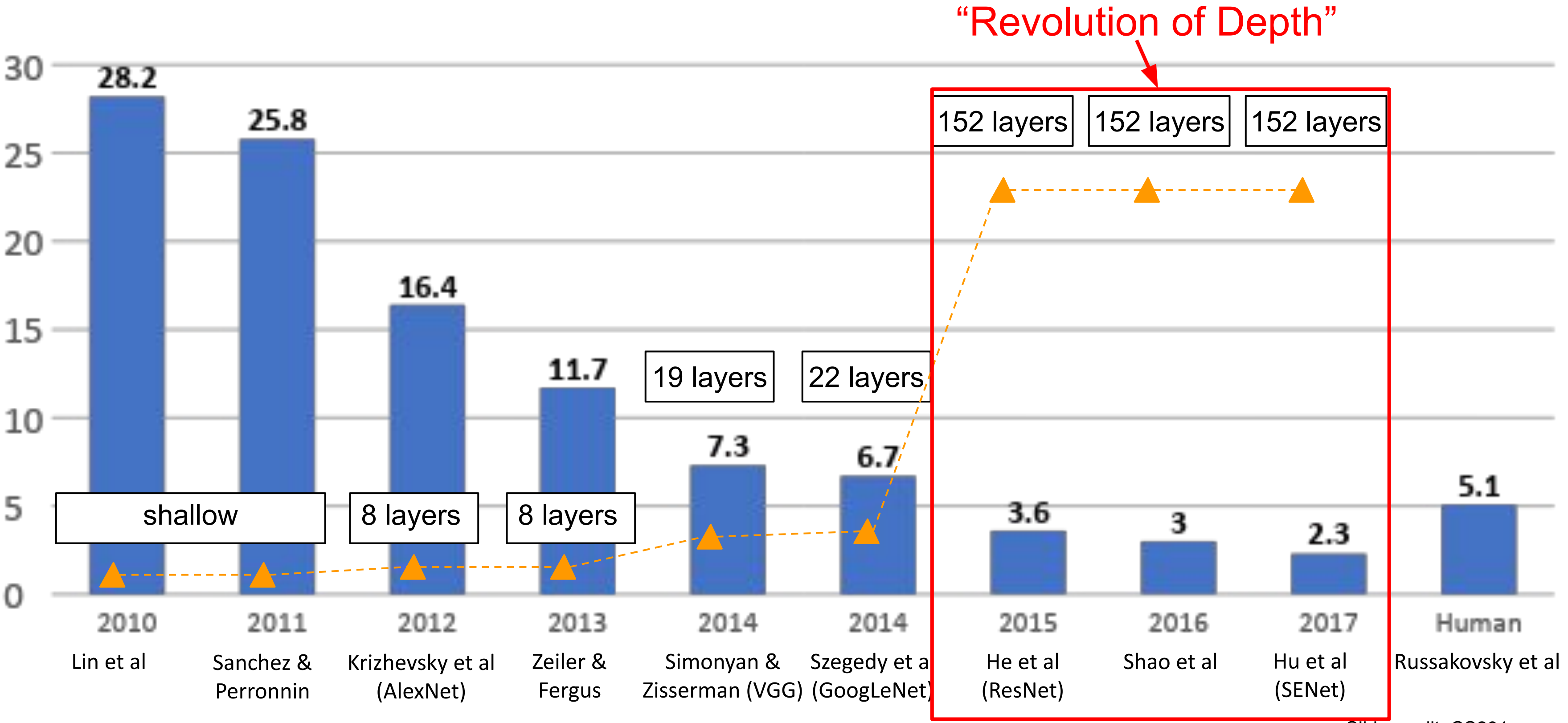
# McKinney et al. 2020

- Binary classification of breast cancer in mammograms
- Used an ensemble of models including ResNets
- International dataset and evaluation, across UK and US




McKinney et al. International evaluation of an AI system for breast cancer screening. Nature, 2020.

# ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



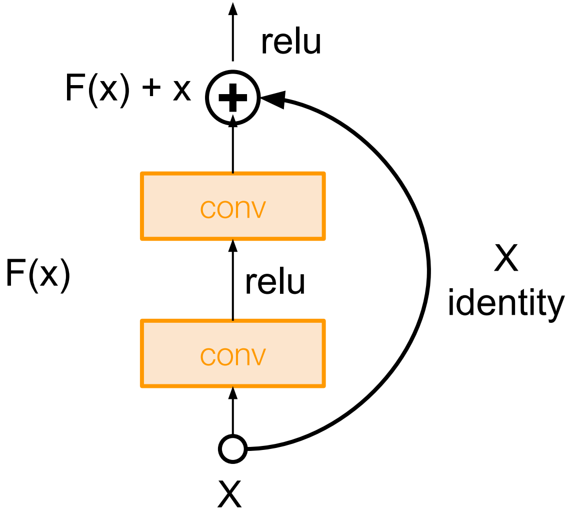

Slide credit: CS231n

# ResNet

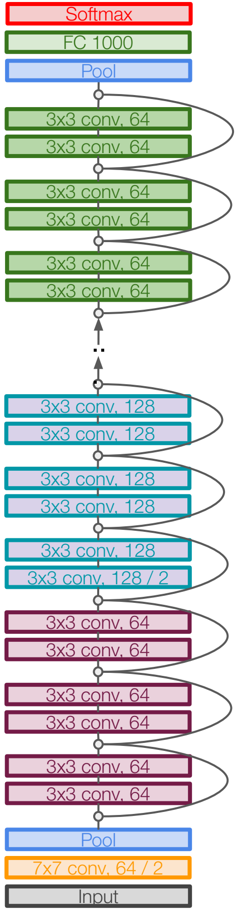
*[He et al., 2015]*

Very deep networks using residual connections

- 152-layer model for ImageNet
- Won all major classification and detection benchmark challenges in 2015




Residual block

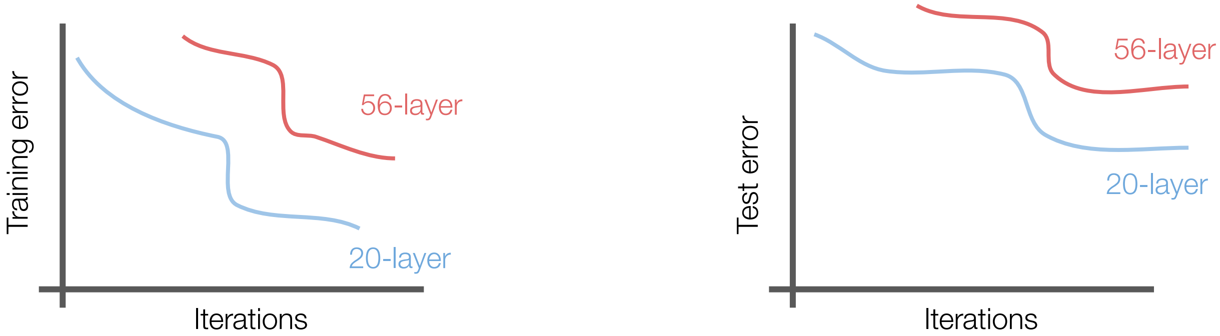



Slide credit: CS231n

# ResNet

*[He et al., 2015]*

What happens when we continue stacking deeper layers on a “plain” convolutional neural network?



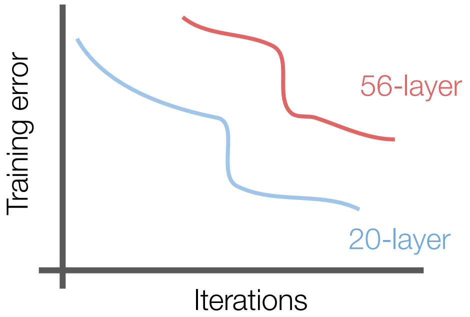

Q: What’s strange about these training and test curves?
[Hint: look at the order of the curves]

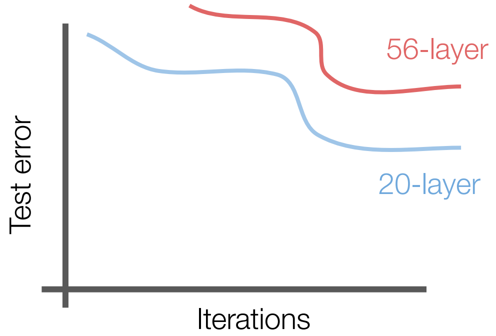
Slide credit: CS231n

# ResNet

*[He et al., 2015]*

What happens when we continue stacking deeper layers on a “plain” convolutional neural network?




56-layer model performs worse on both training and test error
-> The deeper model performs worse, but it’s not caused by overfitting!

Slide credit: CS231n

# ResNet

*[He et al., 2015]*

Hypothesis: the problem is an *optimization* problem, deeper models are harder to optimize

Slide credit: CS231n

# ResNet

*[He et al., 2015]*

Hypothesis: the problem is an *optimization* problem, deeper models are harder to optimize

The deeper model should be able to perform at least as well as the shallower model.

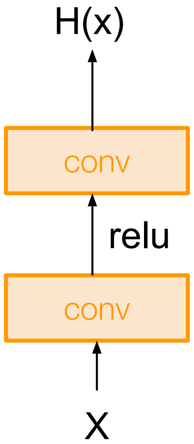
A solution by construction is copying the learned layers over from the shallower model and setting all additional layers to the **identity** function.

Slide credit: CS231n

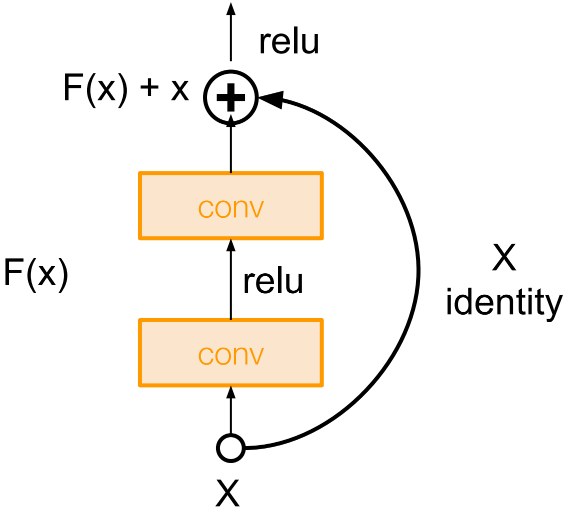
# ResNet

*[He et al., 2015]*

Solution: Structure each network layer to fit a “residual function” with respect to the identity function, then add the two functions together




“Plain” layers




Residual block

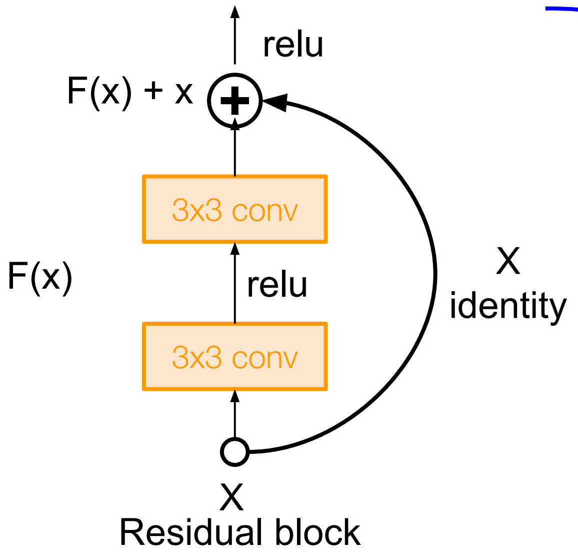
Slide credit: CS231n

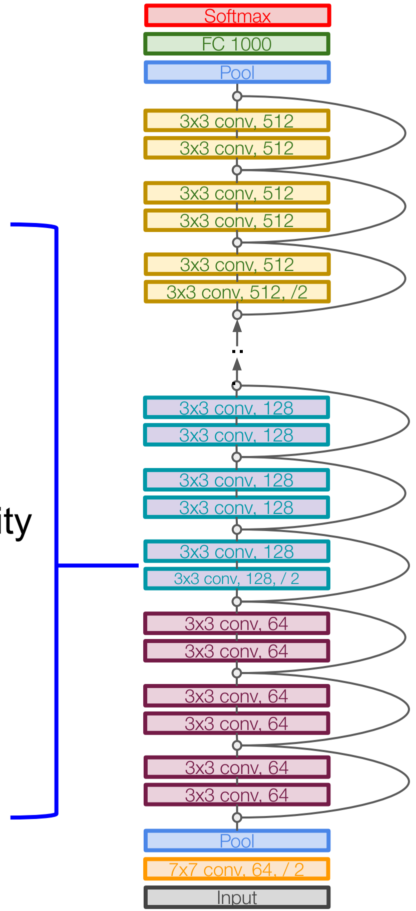
# ResNet

*[He et al., 2015]*

Full ResNet architecture:

- Stack residual blocks
- Every residual block has two 3x3 conv layers



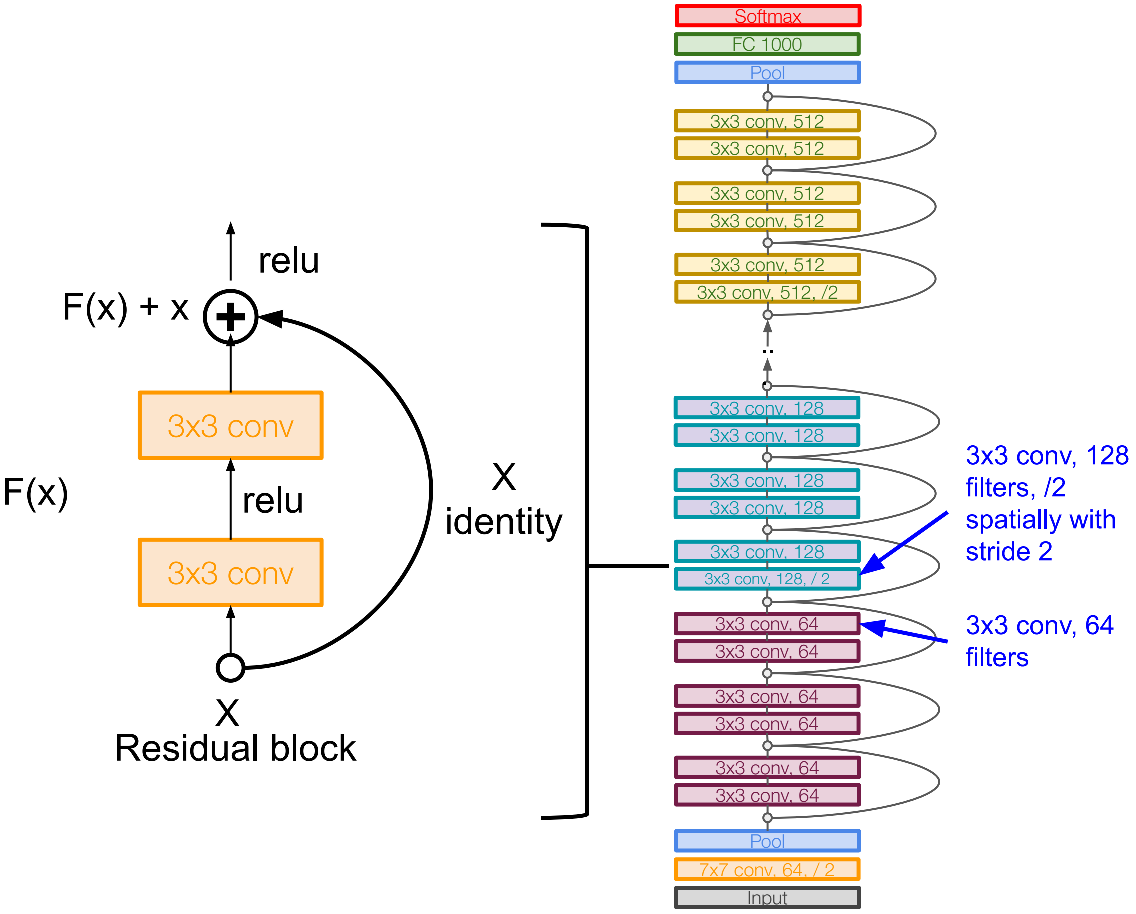

Slide credit: CS231n

# ResNet

*[He et al., 2015]*

Full ResNet architecture:

- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2 (/2 in each dimension)



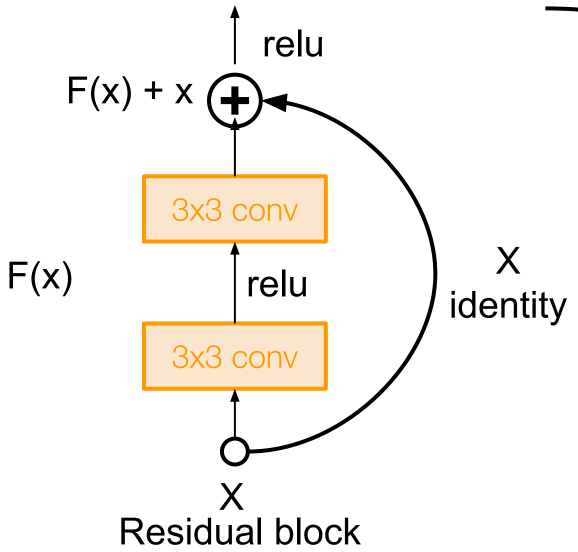

Slide credit: CS231n

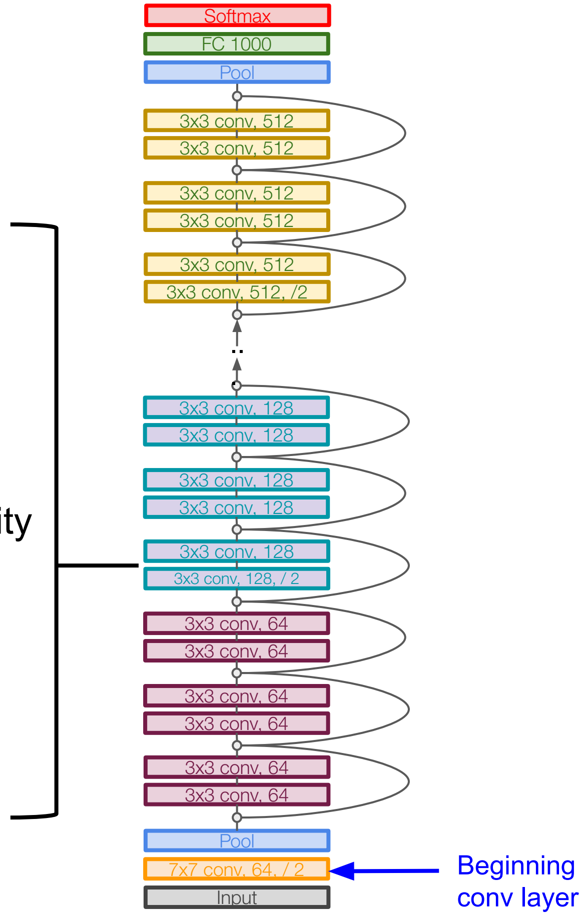
# ResNet

*[He et al., 2015]*

Full ResNet architecture:

- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2 (/2 in each dimension)
- Additional conv layer at the beginning



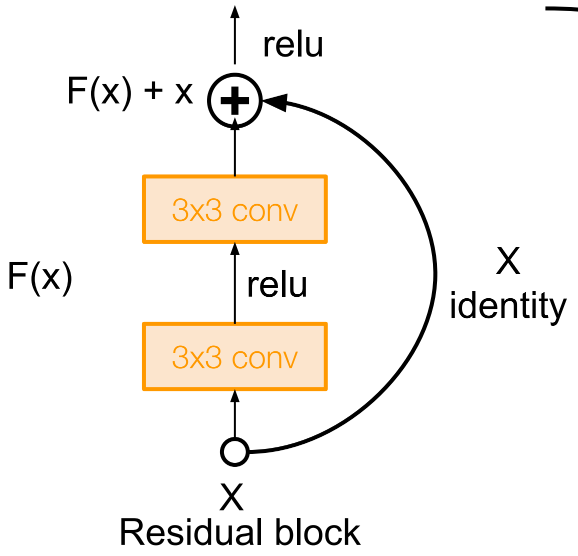



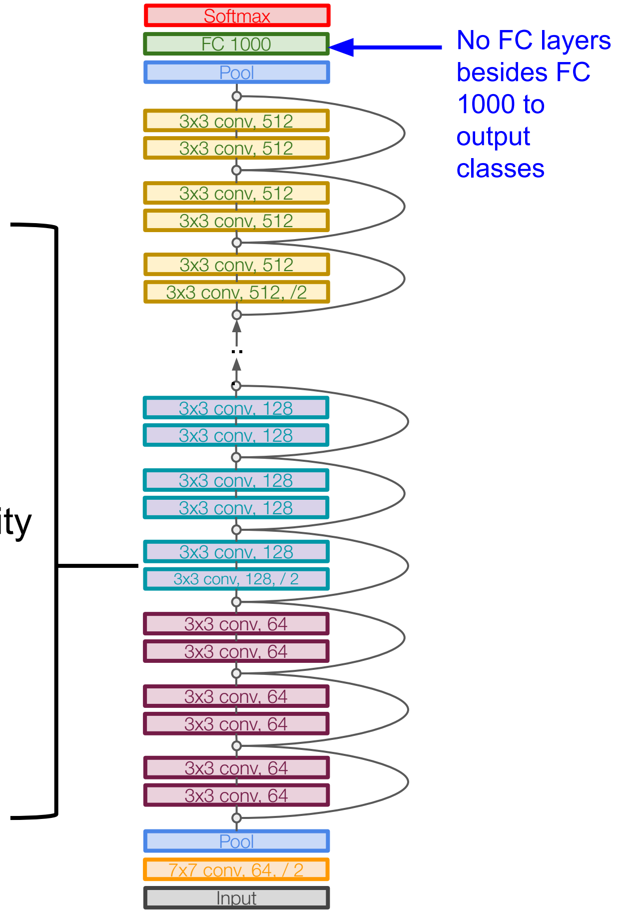

Slide credit: CS231n

# ResNet

*[He et al., 2015]*

Full ResNet architecture:
- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2 (/2 in each dimension)
- Additional conv layer at the beginning
- No FC layers at the end (only FC 1000 to output classes)



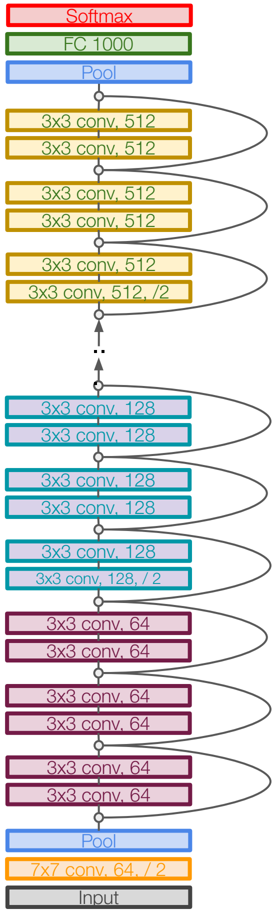




Slide credit: CS231n

# ResNet

*[He et al., 2015]*

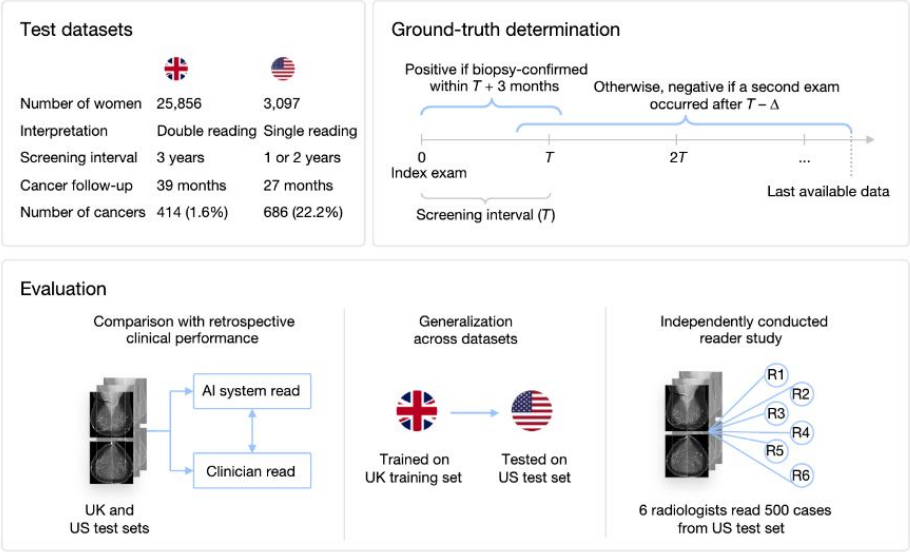
Total depths of 34, 50, 101, or 152 layers for ImageNet




Slide credit: CS231n

# McKinney et al. 2020

- Binary classification of breast cancer in mammograms
- Used an ensemble of models including ResNets
- International dataset and evaluation, across UK and US

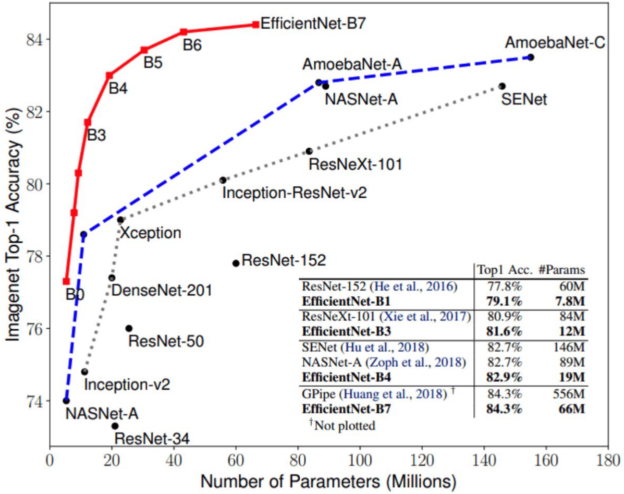



McKinney et al. International evaluation of an AI system for breast cancer screening. Nature, 2020.

# More recent CNN architectures

- MobileNet (Sandler et al. 2018) - architecture with separable convolutions for light-weight CNNs
- NASNet (Zoph et al. 2016) and AmoebaNet (Real et al. 2019) - architectures discovered through “neural architecture search” via reinforcement learning or evolutionary algorithms

- EfficientNet (Tan et al. 2020) - family of architectures designed using “compound scaling” that simultaneously scale width, depth, and resolution of neural networks with a fixed ratio



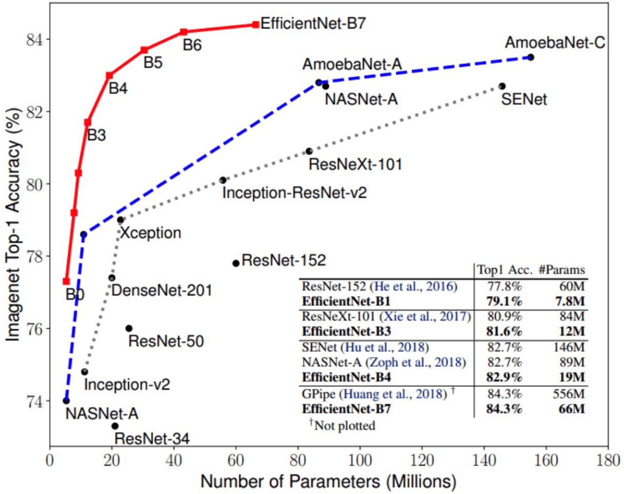

| | Top1 Acc. | #Params |
|---|---|---|
| ResNet-152 (He et al., 2016) | 77.8% | 60M |
| **EfficientNet-B1** | **79.1%** | **7.8M** |
| ResNeXt-101 (Xie et al., 2017) | 80.9% | 84M |
| **EfficientNet-B3** | **81.6%** | **12M** |
| SENet (Hu et al., 2018) | 82.7% | 146M |
| NASNet-A (Zoph et al., 2018) | 82.7% | 89M |
| **EfficientNet-B4** | **82.9%** | **19M** |
| GPipe (Huang et al., 2018) † | 84.3% | 556M |
| **EfficientNet-B7** | **84.3%** | **66M** |

†Not plotted

# More recent CNN architectures

Worth exploring for class projects!

- MobileNet (Sandler et al. 2018) - architecture with separable convolutions for light-weight CNNs
- NASNet (Zoph et al. 2016) and AmoebaNet (Real et al. 2019) - architectures discovered through “neural architecture search” via reinforcement learning or evolutionary algorithms
- EfficientNet (Tan et al. 2020) - family of architectures designed using “compound scaling” that simultaneously scale width, depth, and resolution of neural networks with a fixed ratio




| | Top1 Acc. | #Params |
|---|---|---|
| ResNet-152 (He et al., 2016) | 77.8% | 60M |
| **EfficientNet-B1** | **79.1%** | **7.8M** |
| ResNeXt-101 (Xie et al., 2017) | 80.9% | 84M |
| **EfficientNet-B3** | **81.6%** | **12M** |
| SENet (Hu et al., 2018) | 82.7% | 146M |
| NASNet-A (Zoph et al., 2018) | 82.7% | 89M |
| **EfficientNet-B4** | **82.9%** | **19M** |
| GPipe (Huang et al., 2018) † | 84.3% | 556M |
| **EfficientNet-B7** | **84.3%** | **66M** |

†Not plotted

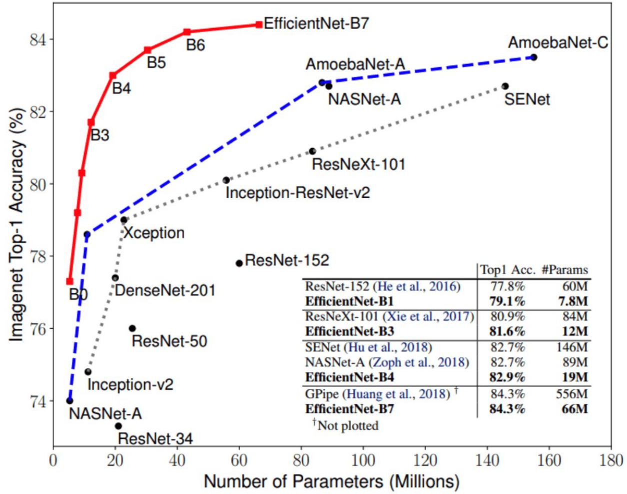
# More recent CNN architectures

Worth exploring for class projects!

- MobileNet (Sandler et al. 2018) - architecture with separable convolutions for light-weight CNNs
- NASNet (Zoph et al. 2016) and AmoebaNet (Real et al. 2019) - architectures discovered through “neural architecture search” via reinforcement learning or evolutionary algorithms

Preview: Transformers, a new class of deep learning architecture, was originally designed for NLP/sequence data but has recently also been applied for computer vision tasks. Stay tuned!

- EfficientNet (Tan et al. 2020) - family of architectures designed using “compound scaling” that simultaneously scale width, depth, and resolution of neural networks with a fixed ratio

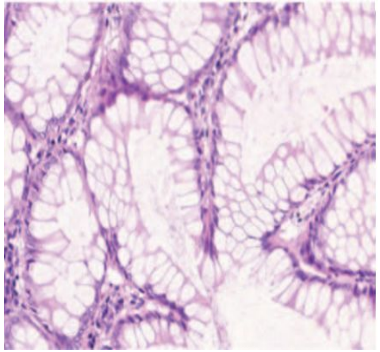
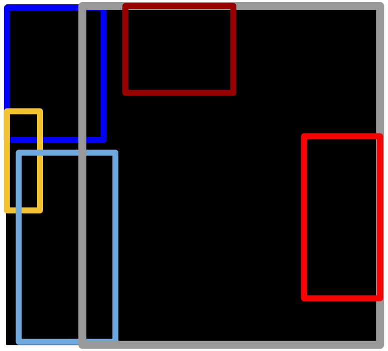



| | Top1 Acc. | #Params |
|---|---|---|
| ResNet-152 (He et al., 2016) | 77.8% | 60M |
| **EfficientNet-B1** | **79.1%** | **7.8M** |
| ResNeXt-101 (Xie et al., 2017) | 80.9% | 84M |
| **EfficientNet-B3** | **81.6%** | **12M** |
| SENet (Hu et al., 2018) | 82.7% | 146M |
| NASNet-A (Zoph et al., 2018) | 82.7% | 89M |
| **EfficientNet-B4** | **82.9%** | **19M** |
| GPipe (Huang et al., 2018) † | 84.3% | 556M |
| **EfficientNet-B7** | **84.3%** | **66M** |

†Not plotted

# Advanced Vision Models: Segmentation and Detection

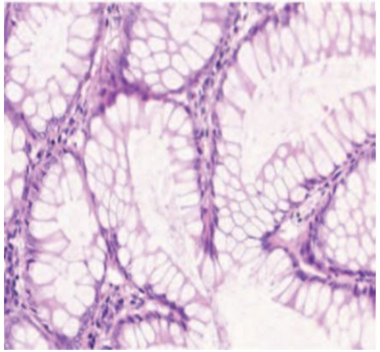
# Richer visual recognition tasks: segmentation and detection

| **Classification** | **Semantic Segmentation** | **Detection** | **Instance Segmentation** |
|---|---|---|---|
|  |  |  |  |
| Output:<br>one category label for image (e.g., colorectal glands) | Output:<br>category label for each pixel in the image | Output:<br>Spatial bounding box for each **instance** of a category object in the image | Output:<br>Category label and instance label for each pixel in the image |

Figures: Chen et al. 2016. https://arxiv.org/pdf/1604.02677.pdf

# Richer visual recognition tasks: segmentation and detection

## Classification

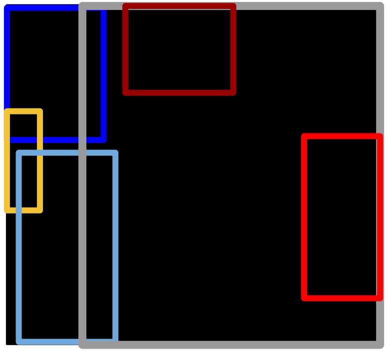


Output:
one category label for image (e.g., colorectal glands)

## Semantic Segmentation

Output:
category label for each pixel in the image

## Detection



Output:
Spatial bounding box for each **instance** of a category object in the image

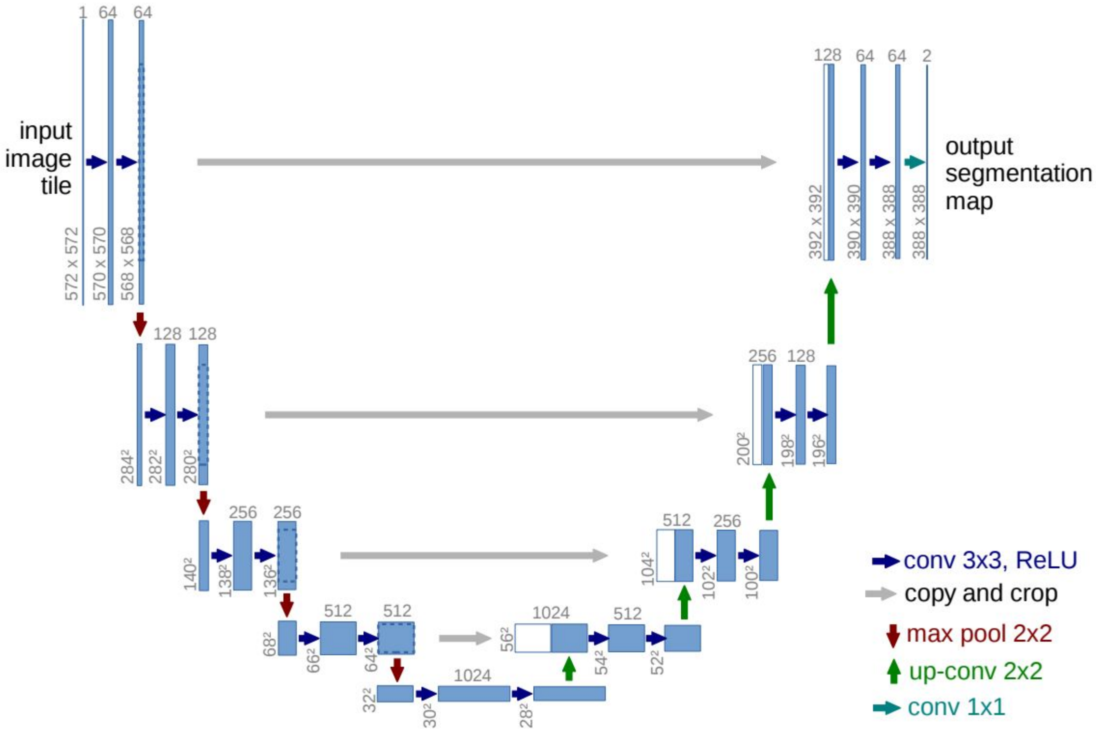
## Instance Segmentation

Output:
Category label and instance label for each pixel in the image

Distinguishes between different instances of an object

Figures: Chen et al. 2016. [https://arxiv.org/pdf/1604.02677.pdf](https://arxiv.org/pdf/1604.02677.pdf)

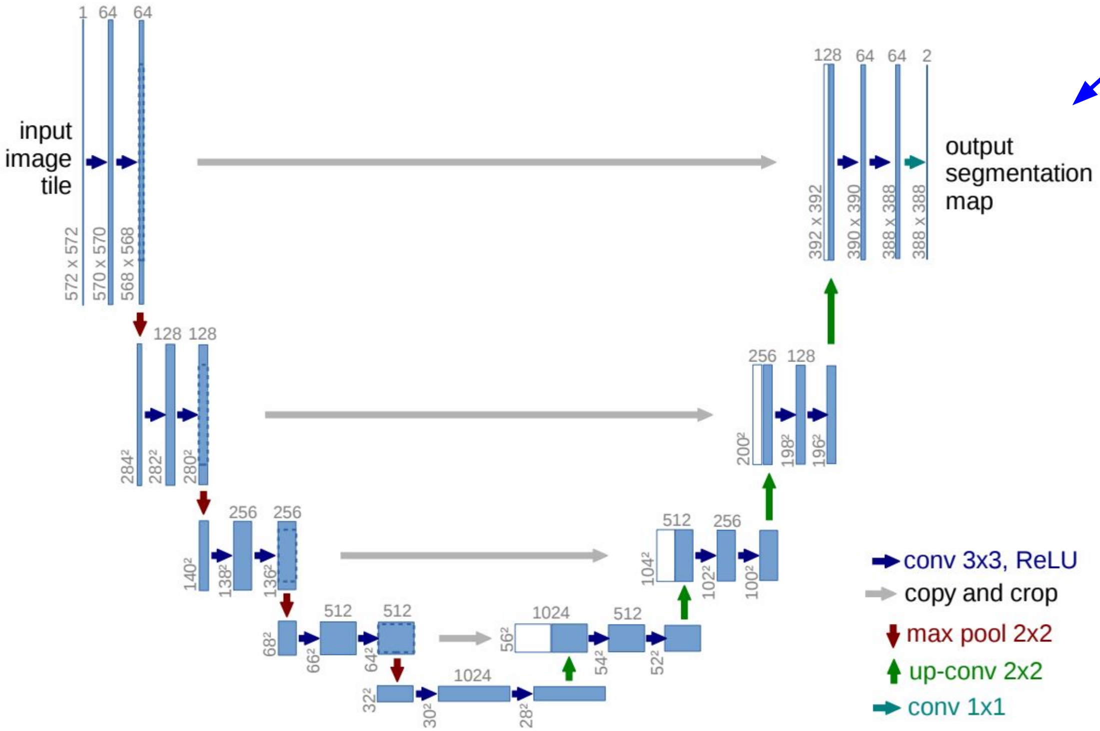
# Semantic segmentation: U-Net




Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

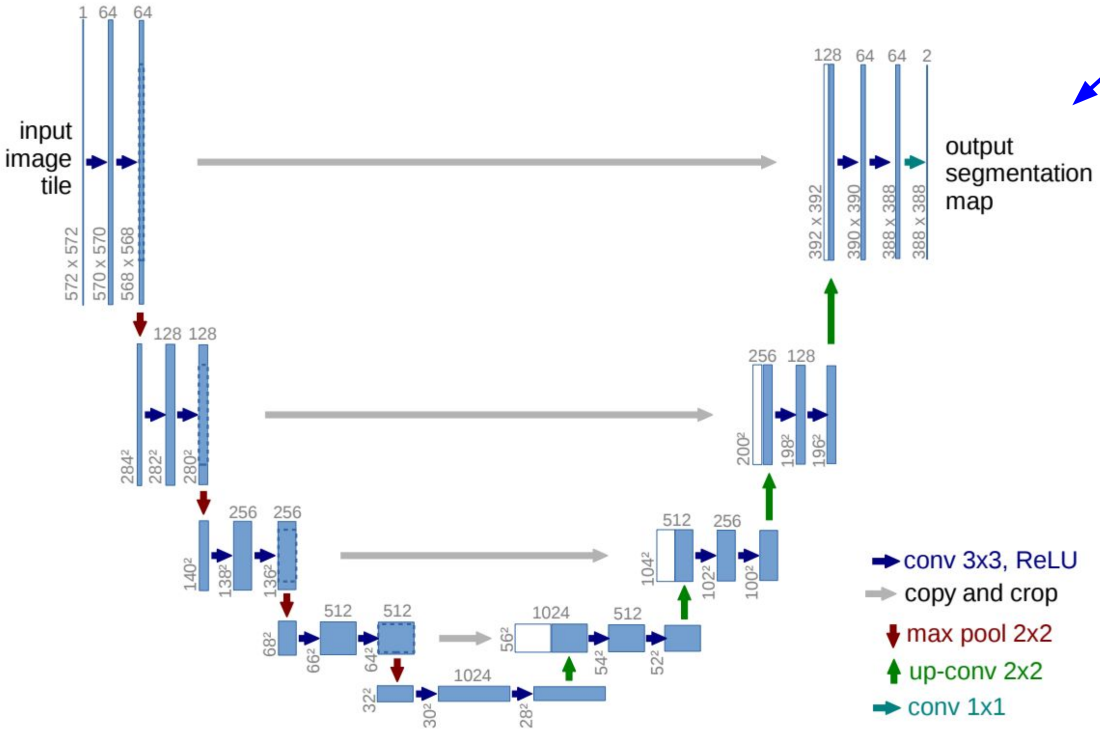
# Semantic segmentation: U-Net

Output is an image mask: width x height x # classes



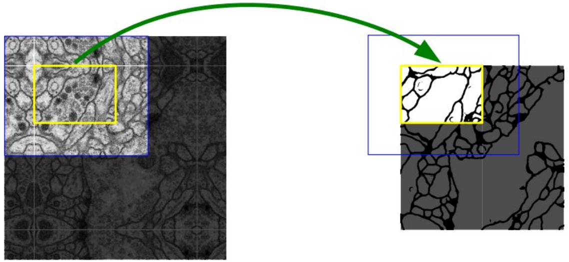

Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

# Semantic segmentation: U-Net



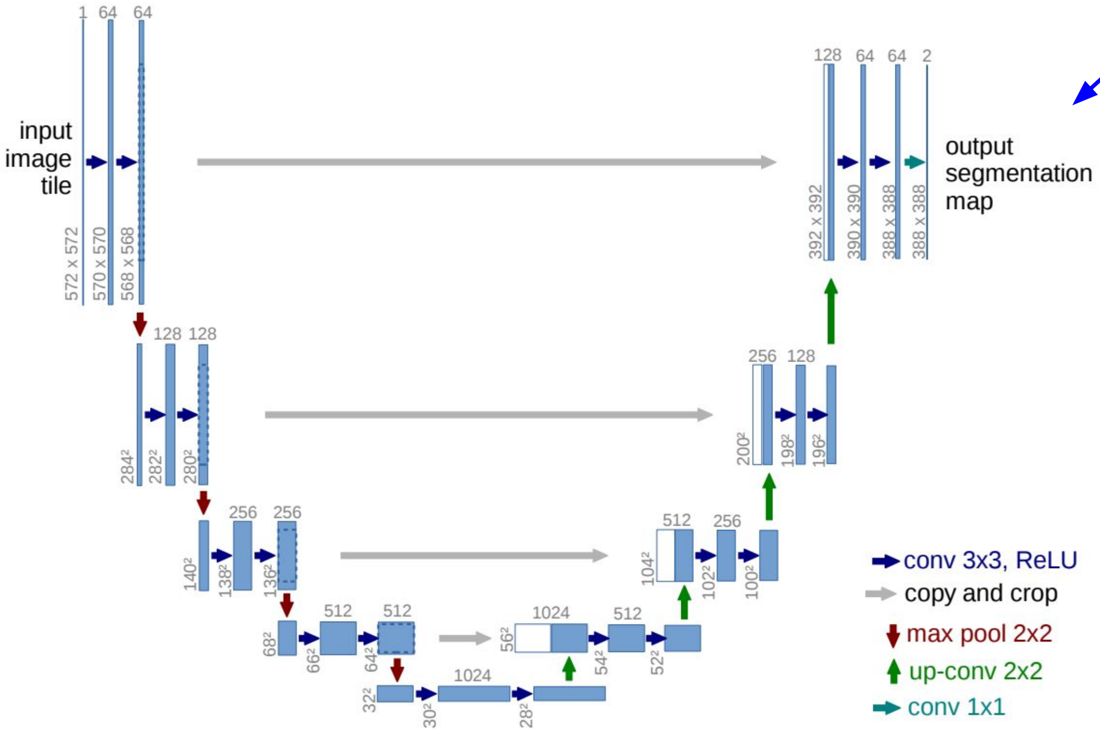

Output is an image mask: width x height x # classes

Output image size somewhat smaller than original, due to convolutional operations w/o padding



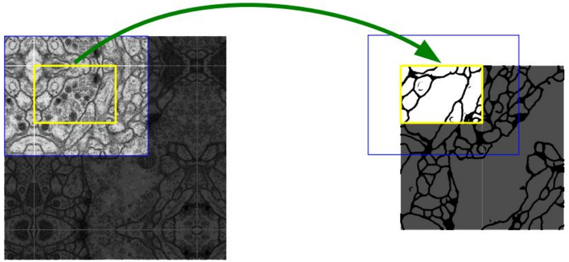
Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

# Semantic segmentation: U-Net




Output is an image mask: width x height x # classes

Output image size somewhat smaller than original, due to convolutional operations w/o padding

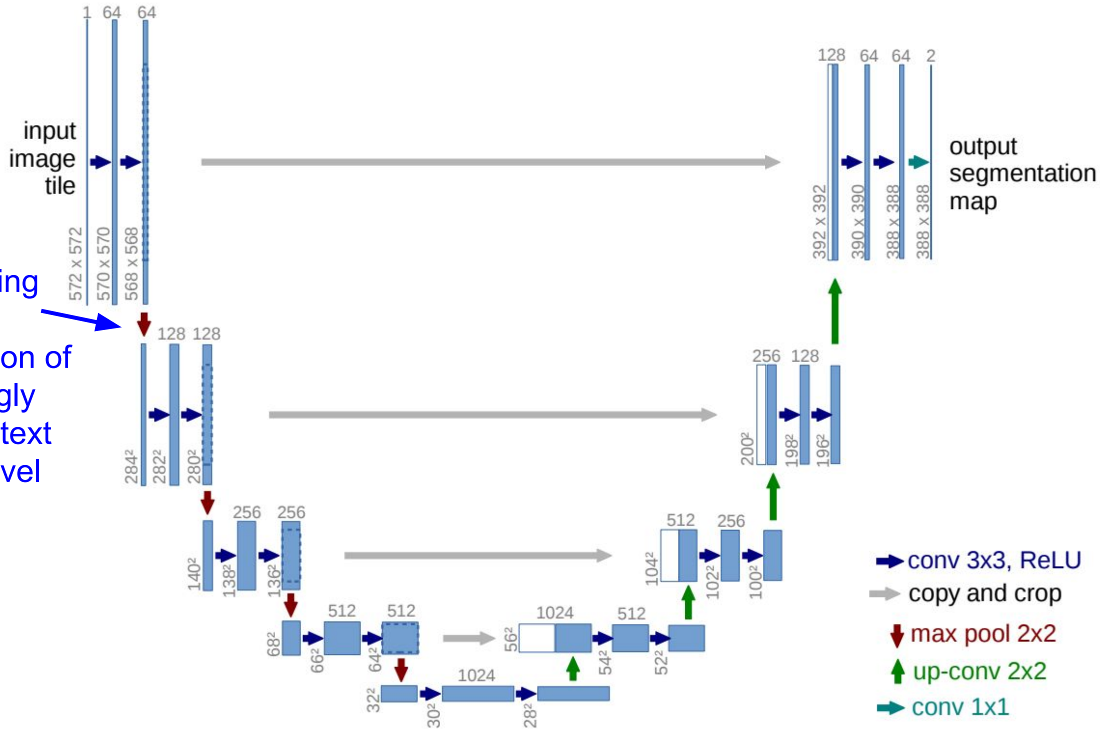


Gives more “true” context for reasoning over each image area. Can tile to make predictions for arbitrarily large images

Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

# Semantic segmentation: U-Net

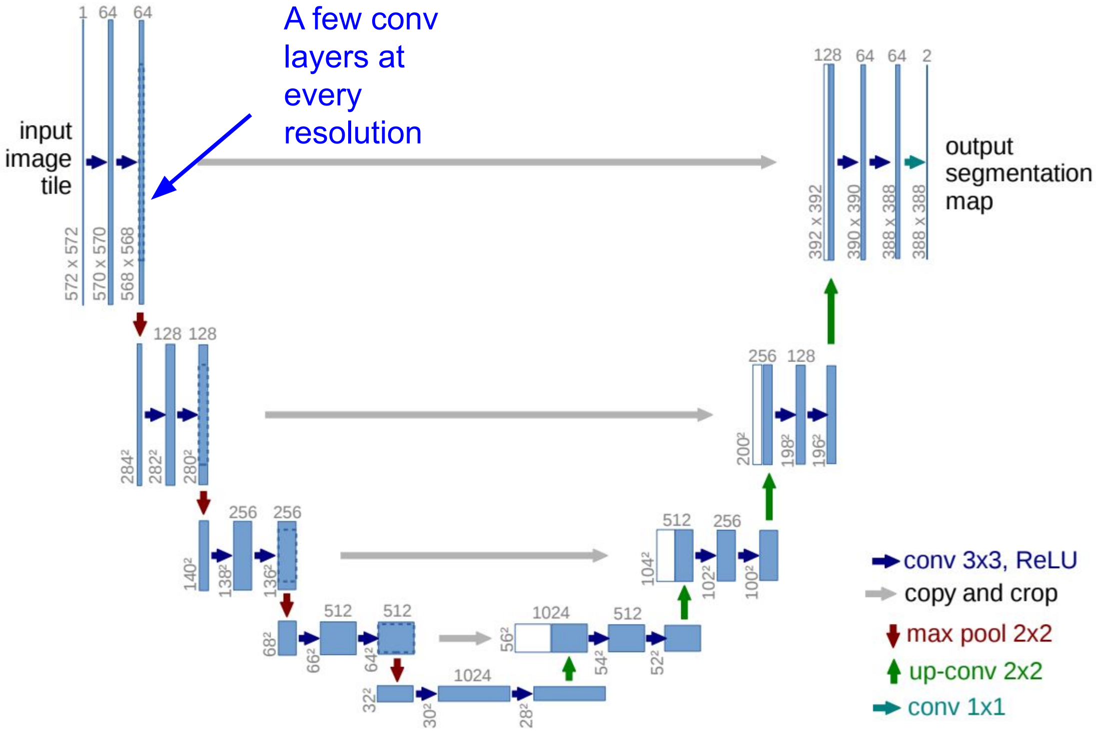
Max pooling enables aggregation of increasingly more context (higher level features)




Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

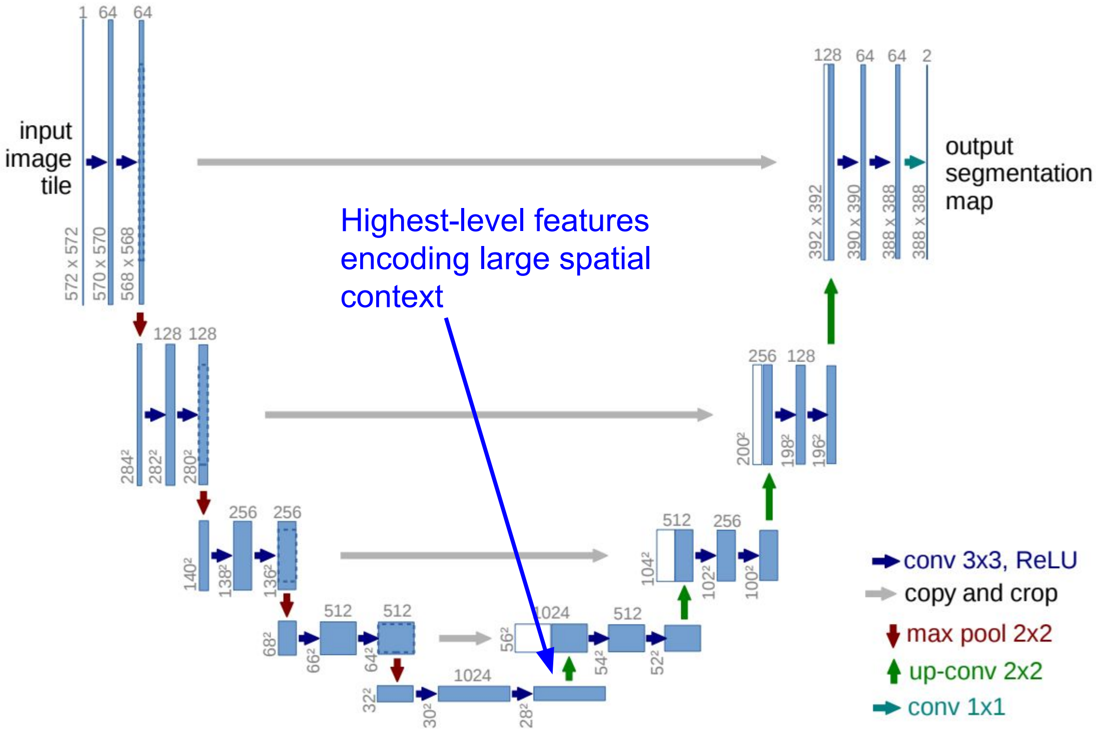
# Semantic segmentation: U-Net

A few conv
layers at
every
resolution



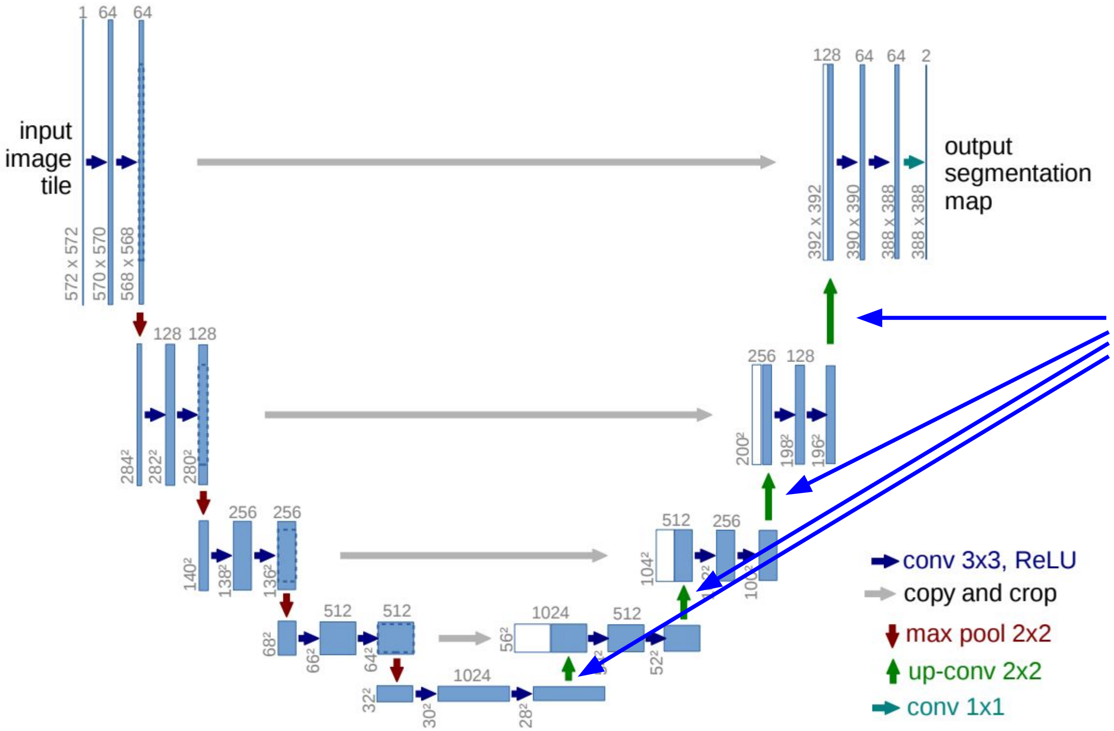

Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

# Semantic segmentation: U-Net




Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

# Semantic segmentation: U-Net

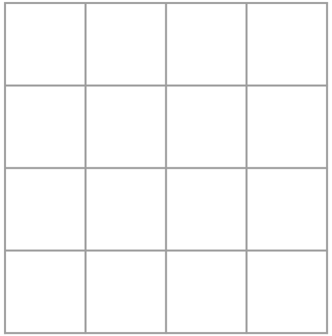



Up-convolutions to go from the global information encoded in highest-level features, back to individual pixel predictions

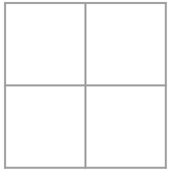
Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

# Up-convolutions

**Recall:** Normal 3 x 3 convolution, <u>stride 2</u> pad 1



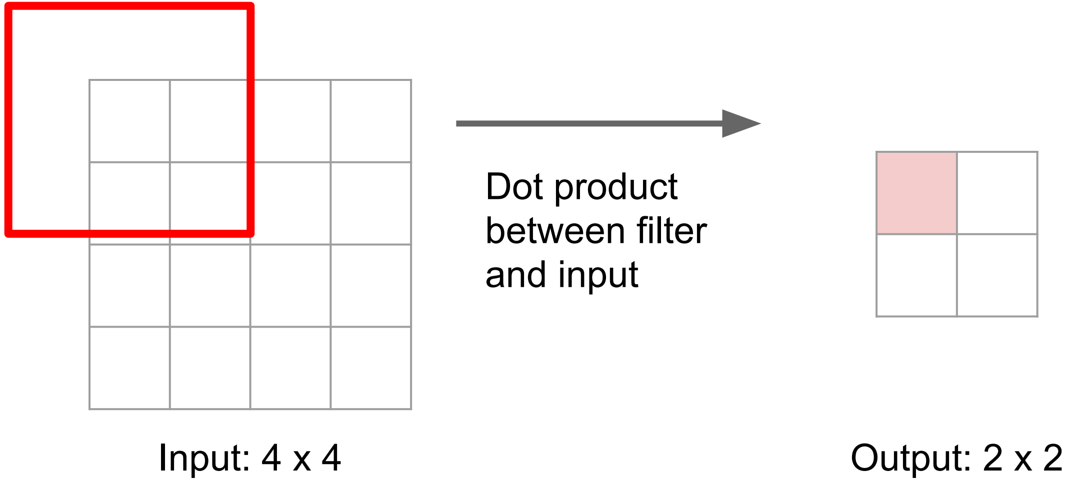
Input: 4 x 4



Output: 2 x 2

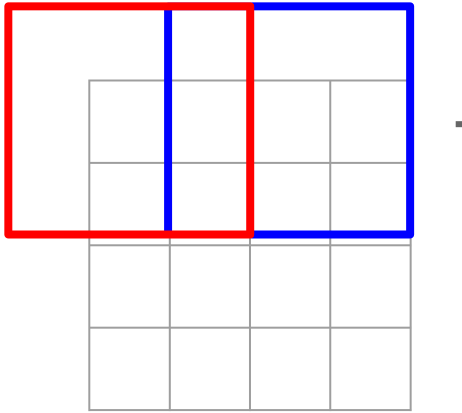
# Up-convolutions

**Recall:** Normal 3 x 3 convolution, stride 2 pad 1



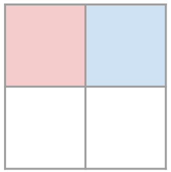
# Up-convolutions

**Recall:** Normal 3 x 3 convolution, stride 2 pad 1



Input: 4 x 4

Dot product between filter and input



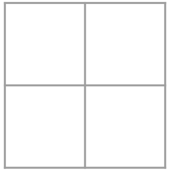
Output: 2 x 2

Filter moves 2 pixels in the input for every one pixel in the output

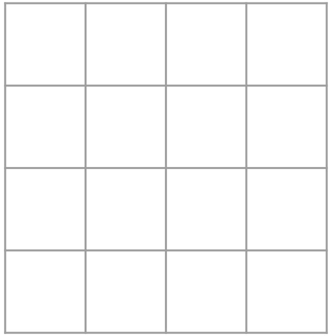
Stride gives ratio between movement in input and output

# Up-convolutions

3 x 3 **transpose** convolution, stride 2 pad 1



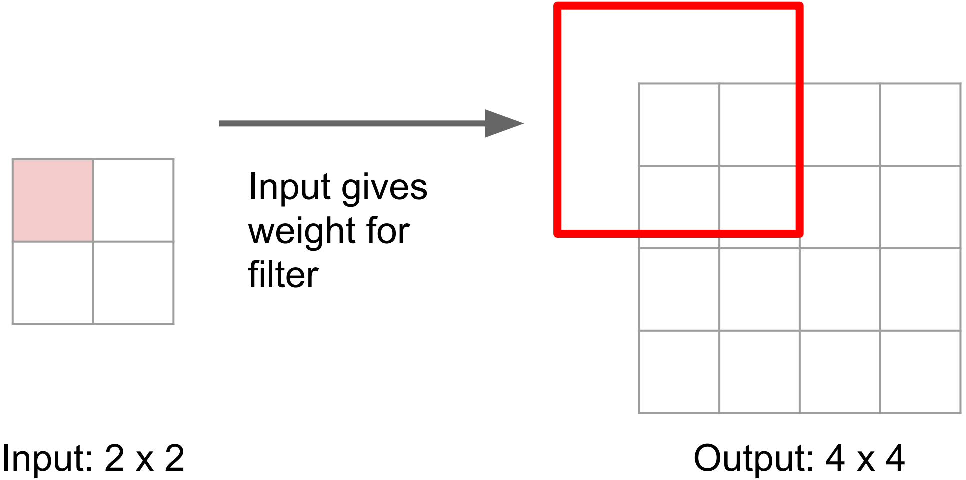
Input: 2 x 2



Output: 4 x 4

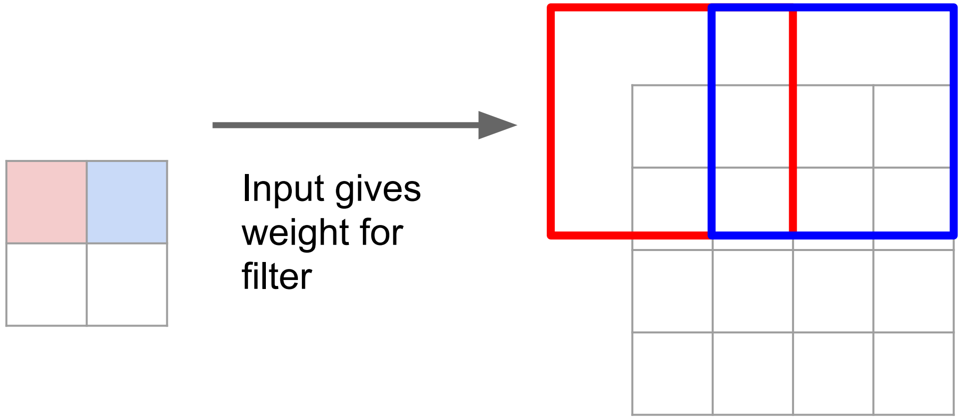
# Up-convolutions

3 x 3 **up-convolution**, stride 2 pad 1



# Up-convolutions

3 x 3 **up-convolution**, stride 2 pad 1




Input: 2 x 2

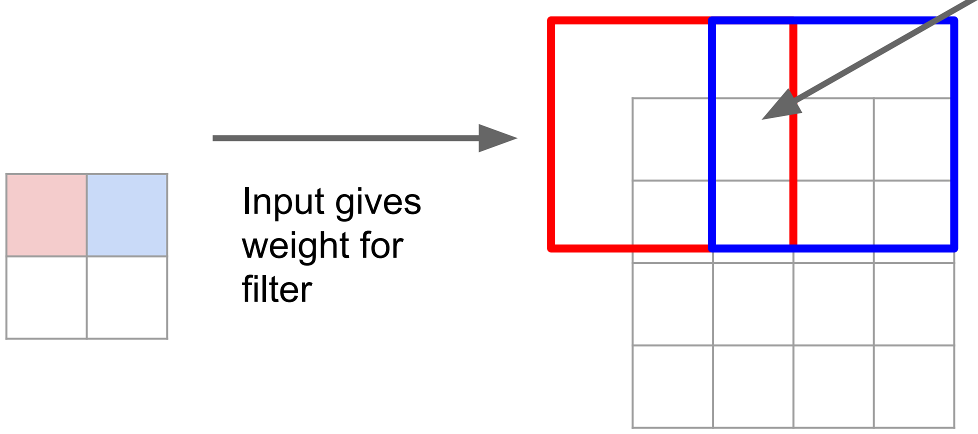
Output: 4 x 4

Filter moves 2 pixels in the <u>output</u> for every one pixel in the <u>input</u>

Stride gives ratio between movement in output and input

# Up-convolutions

3 x 3 **up-convolution**, stride 2 pad 1



Input: 2 x 2

Input gives weight for filter

Output: 4 x 4

Sum where output overlaps

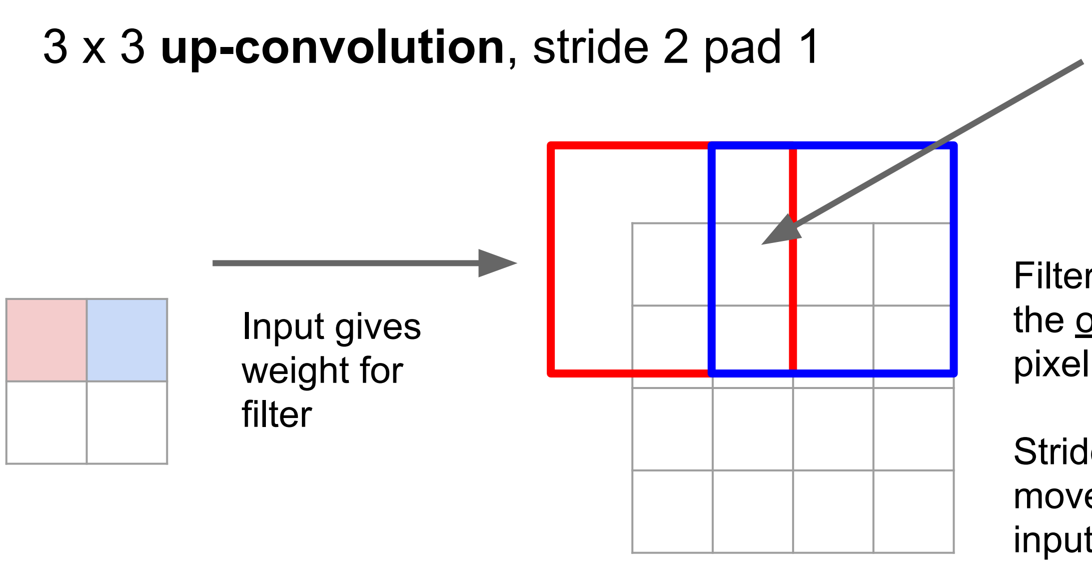
Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

# Up-convolutions

**Other names:**
-Transpose convolution
-Fractionally strided convolution
-Backward strided convolution

3 x 3 **up-convolution**, stride 2 pad 1



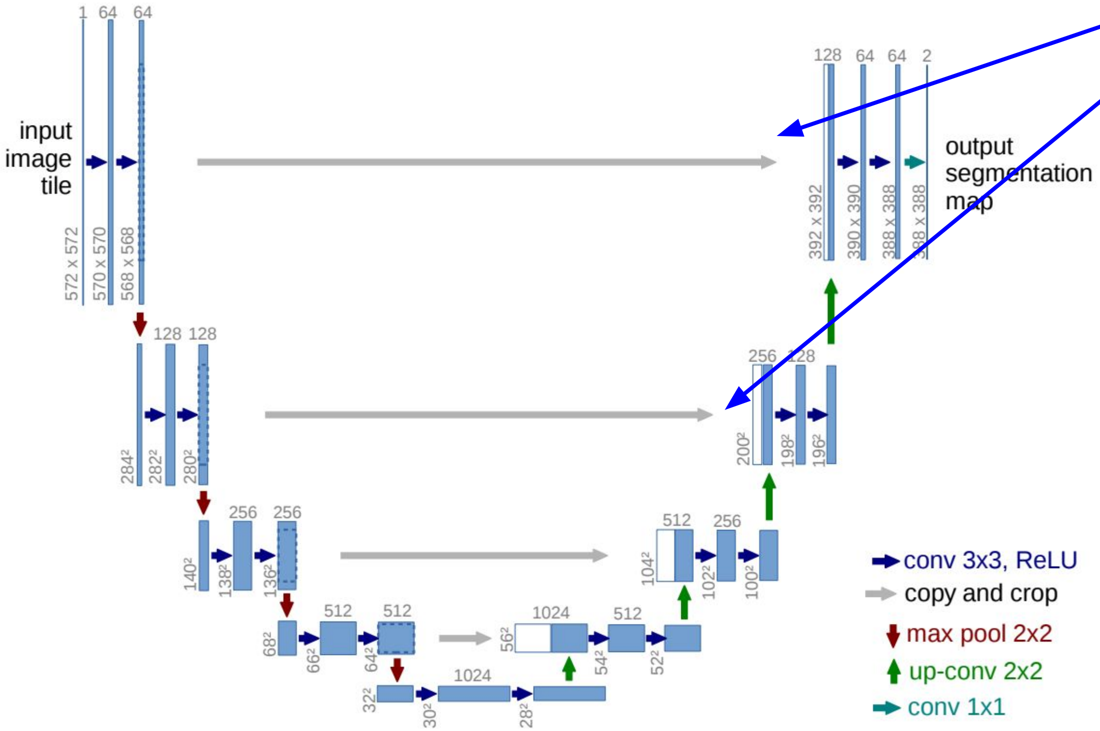

Input: 2 x 2

Output: 4 x 4

Filter moves 2 pixels in the <u>output</u> for every one pixel in the <u>input</u>

Stride gives ratio between movement in output and input

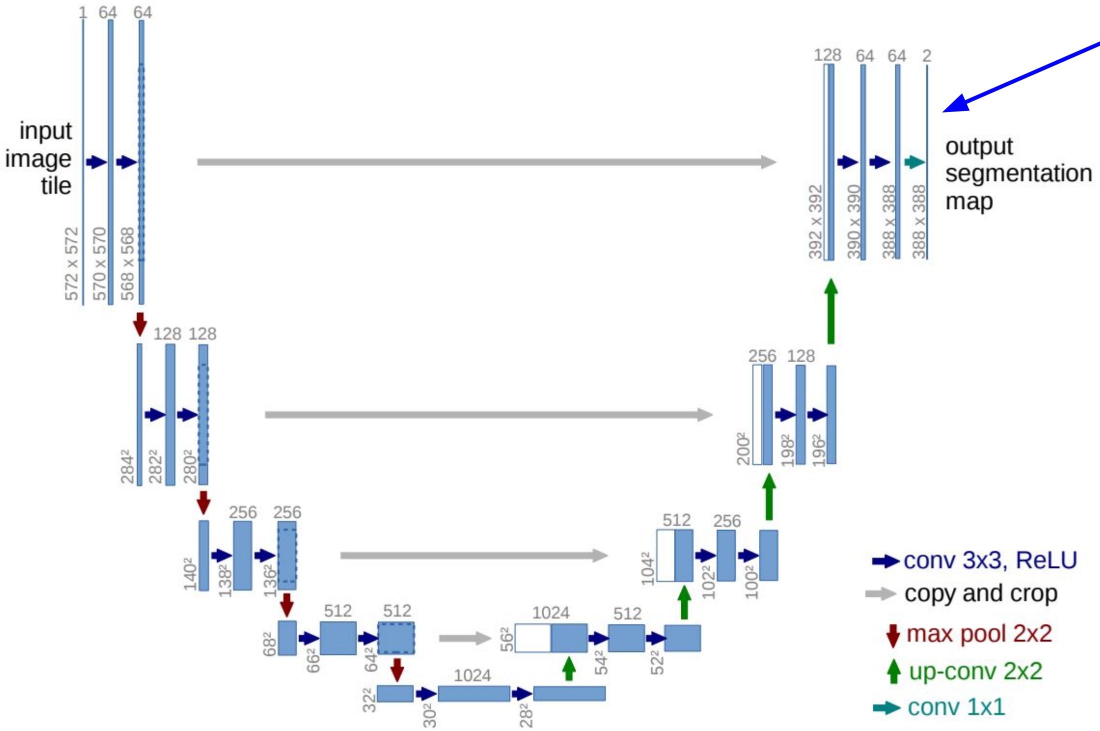
# Semantic segmentation: U-Net




Concatenate with same-resolution feature map during downsampling process to combine high-level information with low-level (local) information

Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

# Semantic segmentation: U-Net




Train with classification loss (e.g. binary cross entropy) on every pixel, sum over all pixels to get total loss

Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

# Semantic segmentation: IOU evaluation

**Intersection over Union:**

# pixels included in both target and prediction maps

$$IoU = \frac{target \cap prediction}{target \cup prediction}$$

Total # pixels in the union of both masks

# Semantic segmentation: IOU evaluation

**Intersection over Union:**

# pixels included in both target and prediction maps

$$IoU = \frac{target \cap prediction}{target \cup prediction}$$

Total # pixels in the union of both masks

Can compute this over all masks in the evaluation set, or at individual mask and image levels to get finer-grained understanding of performance.

# Semantic segmentation: IOU evaluation

**Intersection over Union:**

# pixels included in both target and prediction maps

$$IoU = \frac{target \cap prediction}{target \cup prediction}$$

Total # pixels in the union of both masks

Can compute this over all masks in the evaluation set, or at individual mask and image levels to get finer-grained understanding of performance.

Also known as Jaccard Index

# Semantic segmentation: Pixel Accuracy evaluation

$$\text{Pixel Accuracy (PA)} = \frac{\#\ \text{correctly classified pixels}}{\#\ \text{total pixels}}$$

# Semantic segmentation: Pixel Accuracy evaluation

TP + TN

$$\text{Pixel Accuracy (PA)} = \frac{\text{\# correctly classified pixels}}{\text{\# total pixels}}$$

Total pixels
in image

# Semantic segmentation: Pixel Accuracy evaluation

TP + TN

$$\text{Pixel Accuracy (PA)} = \frac{\#\ \text{correctly classified pixels}}{\#\ \text{total pixels}}$$

Total pixels
in image

Q: What is a potential
problem with this?

# Semantic segmentation: Pixel Accuracy evaluation

TP + TN

$$\text{Pixel Accuracy (PA)} = \frac{\#\ \text{correctly classified pixels}}{\#\ \text{total pixels}}$$

Total pixels in image

Q: What is a potential problem with this?

A: Think about what happens when there is class imbalance.

# Semantic segmentation: Dice coefficient evaluation

$$\text{Dice Coefficient} = \frac{2 * (\text{target} \cap \text{prediction})}{\#\ \text{target mask pixels} + \#\ \text{prediction mask pixels}}$$

# Semantic segmentation: Dice coefficient evaluation

2 * intersection

$$\text{Dice Coefficient} = \frac{2 * (\text{target} \cap \text{prediction})}{\#\ \text{target mask pixels} + \#\ \text{prediction mask pixels}}$$

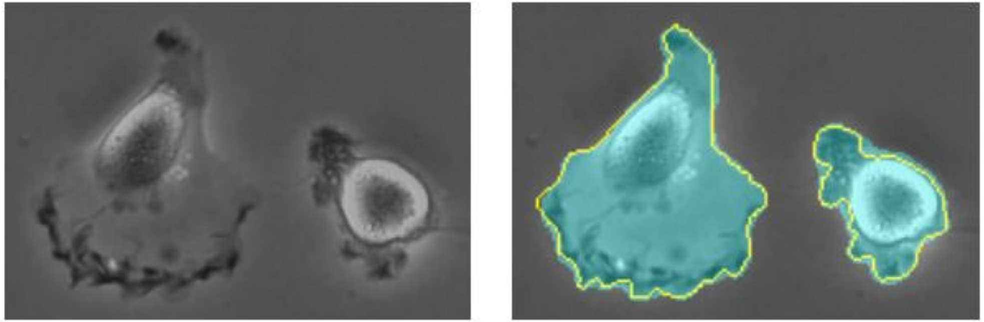
Sum of target mask size
+ prediction mask size

Very similar to IOU / Jaccard, can derive one from the other

# Semantic segmentation: summary of evaluation metrics

- Most commonly use IOU / Jaccard or Dice Coefficient
- Sometimes will also see pixel accuracy
- If multi-class segmentation task, typically report all these metrics per-class, and then a mean over all classes

# Semantic segmentation: U-Net cell segmentation



| Name | PhC-U373 | DIC-HeLa |
|---|---|---|
| IMCB-SG (2014) | 0.2669 | 0.2935 |
| KTH-SE (2014) | 0.7953 | 0.4607 |
| HOUS-US (2014) | 0.5323 | - |
| second-best 2015 | 0.83 | 0.46 |
| u-net (2015) | **0.9203** | **0.7756** |

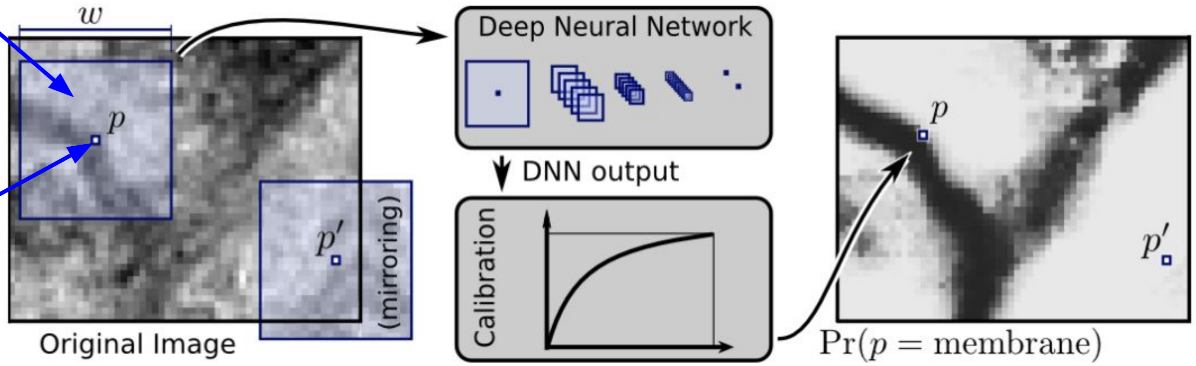
Very small dataset: 30 training images of size 512x512, in the ISBI 2012 Electron Microscopy (EM) segmentation challenge. Used excessive data augmentation to compensate.

Ronneberger et al. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

# Aside: segmentation through sliding-window pixel classification

Image patch: input to classification network

Classification output is prediction for the center pixel of the patch

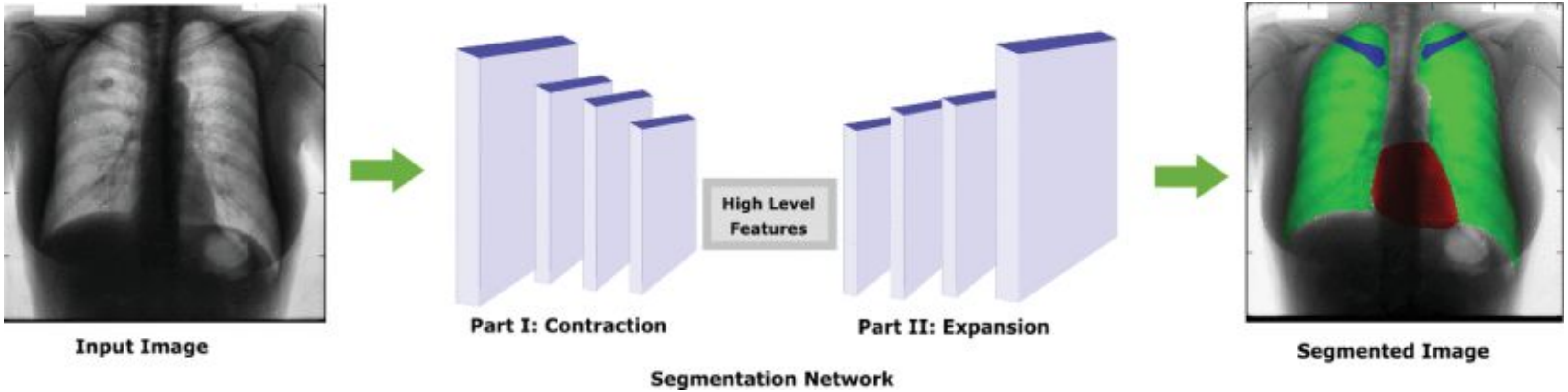



Note: a simple approach to segmentation can also be applying a classification CNN on image patches in a dense, sliding-window fashion (e.g. Ciresan et al.). But fully convolutional approaches such as U-Net generally achieve better performance.

Ciresan et al. Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. NeurIPS, 2012.

# Novikov et al. 2018

- Chest x-ray segmentation of lungs, clavicles, and heart
- JSRT dataset of 247 chest-xrays at 2048x2048 resolution. (But downsampled to 128x128 and 256x256!)
- Used a U-Net based segmentation network with a few modifications

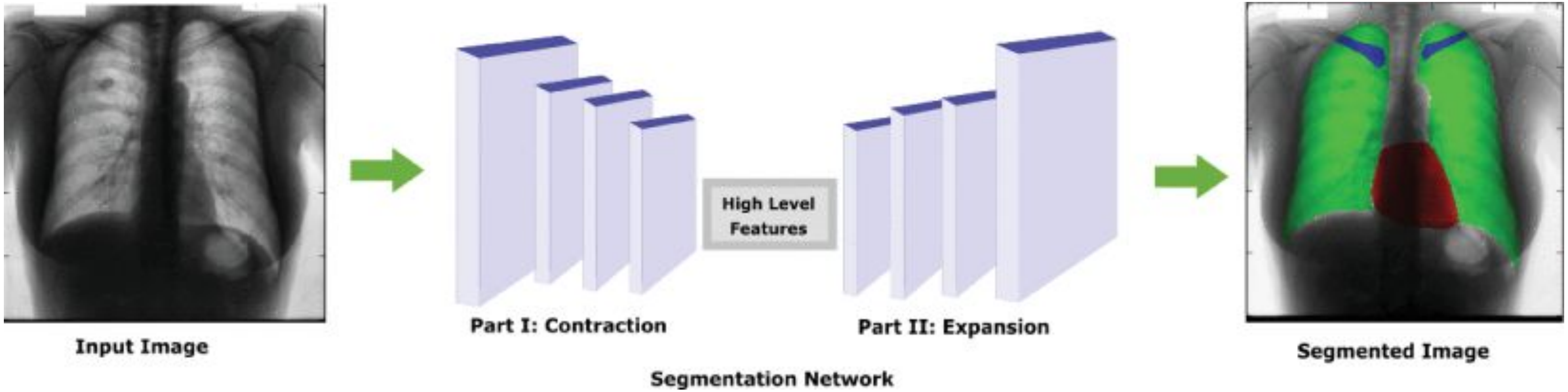



Novikov et al. Fully Convolutional Architectures for Multiclass Segmentation in Chest Radiographs. IEEE Trans. on Medical Imaging, 2018.

# Novikov et al. 2018

Q: What loss function would be appropriate here?

- Chest x-ray segmentation of lungs, clavicles, and heart
- JSRT dataset of 247 chest-xrays at 2048x2048 resolution. (But downsampled to 128x128 and 256x256!)
- Used a U-Net based segmentation network with a few modifications




Novikov et al. Fully Convolutional Architectures for Multiclass Segmentation in Chest Radiographs. IEEE Trans. on Medical Imaging, 2018.

# Novikov et al. 2018

- Multi-class segmentation -> tried both a per-pixel softmax loss as well as a loss based on the Dice coefficient.
- Class imbalance -> weight loss terms corresponding to each ground-truth class by inverse of class frequency: (# class pixels) / (total # pixels in data)

| Body Part | Lungs | | Clavicles | | Heart | |
|---|---|---|---|---|---|---|
| Evaluation Metric | $D$ | $J$ | $D$ | $J$ | $D$ | $J$ |
| InvertedNet | 0.972 | 0.946 | **0.902** | **0.821** | 0.935 | 0.879 |
| All-Dropout | **0.973** | **0.948** | 0.896 | 0.812 | **0.941** | **0.888** |
| All-Convolutional | 0.971 | 0.944 | 0.876 | 0.780 | 0.938 | 0.883 |
| Original U-Net | 0.971 | 0.944 | 0.880 | 0.785 | 0.938 | 0.883 |

Novikov et al. Fully Convolutional Architectures for Multiclass Segmentation in Chest Radiographs. IEEE Trans. on Medical Imaging, 2018.

# Novikov et al. 2018

Image ground truth class mask

$$L_{\text{dice}}(y, \hat{y}) = 1 - \frac{2\sum_{i,j} y_{i,j}\hat{y}_{i,j}}{\sum_{i,j} y_{i,j} + \sum_{i,j} \hat{y}_{i,j}}$$

Image pixel class probabilities

- Multi-class segmentation -> tried both a per-pixel softmax loss as well as a loss based on the Dice coefficient. Note: this Dice loss is often useful to try!
- Class imbalance -> weight loss terms corresponding to each ground-truth class by inverse of class frequency: (# class pixels) / (total # pixels in data)

| Body Part | Lungs | | Clavicles | | Heart | |
|---|---|---|---|---|---|---|
| Evaluation Metric | $D$ | $J$ | $D$ | $J$ | $D$ | $J$ |
| InvertedNet | 0.972 | 0.946 | **0.902** | **0.821** | 0.935 | 0.879 |
| All-Dropout | **0.973** | **0.948** | 0.896 | 0.812 | **0.941** | **0.888** |
| All-Convolutional | 0.971 | 0.944 | 0.876 | 0.780 | 0.938 | 0.883 |
| Original U-Net | 0.971 | 0.944 | 0.880 | 0.785 | 0.938 | 0.883 |

Novikov et al. Fully Convolutional Architectures for Multiclass Segmentation in Chest Radiographs. IEEE Trans. on Medical Imaging, 2018.

# Novikov et al. 2018

Image ground truth class mask

$$L_{\text{dice}}(y, \hat{y}) = 1 - \frac{2\sum_{i,j} y_{i,j}\hat{y}_{i,j}}{\sum_{i,j} y_{i,j} + \sum_{i,j} \hat{y}_{i,j}}$$

Image pixel class probabilities

- Multi-class segmentation -> tried both a per-pixel softmax loss as well as a loss based on the Dice coefficient. Note: this Dice loss is often useful to try!
- Class imbalance -> weight loss terms corresponding to each ground-truth class by inverse of class frequency: (# class pixels) / (total # pixels in data)

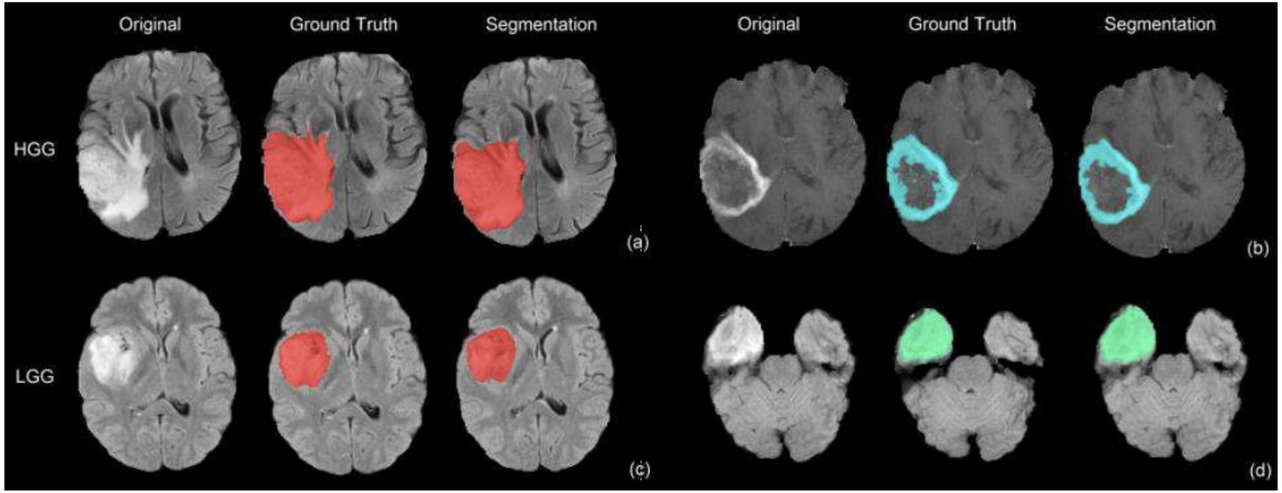
| Body Part | Lungs | | Clavicles | | Heart | |
|---|---|---|---|---|---|---|
| Evaluation Metric | $D$ | $J$ | $D$ | $J$ | $D$ | $J$ |
| InvertedNet | 0.972 | 0.946 | **0.902** | **0.821** | 0.935 | 0.879 |
| All-Dropout | **0.973** | **0.948** | 0.896 | 0.812 | **0.941** | **0.888** |
| All-Convolutional | 0.971 | 0.944 | 0.876 | 0.780 | 0.938 | 0.883 |
| Original U-Net | 0.971 | 0.944 | 0.880 | 0.785 | 0.938 | 0.883 |

Dice and Jaccard evaluation

Novikov et al. Fully Convolutional Architectures for Multiclass Segmentation in Chest Radiographs. IEEE Trans. on Medical Imaging, 2018.

# Dong et al. 2017

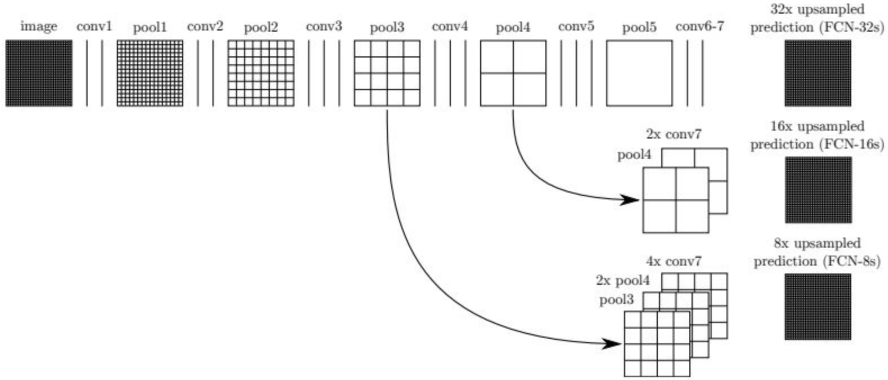
- Segmentation of tumors in brain MR image slices
- BRATS 2015 dataset: 220 high-grade brain tumor and 54 low-grade brain tumor MRIs
- U-Net architecture, Dice loss function




Dong et al. Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks. MIUA, 2017.

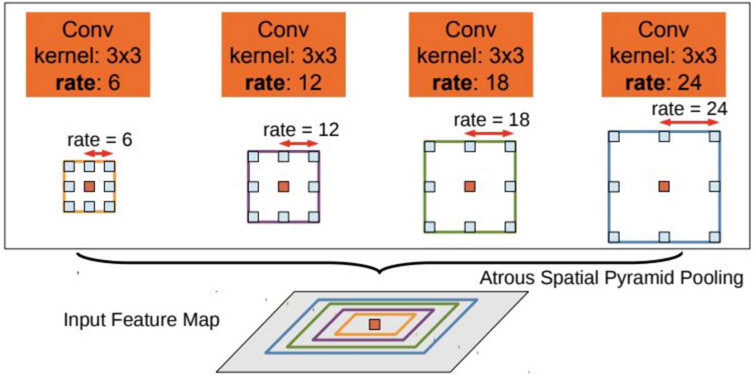
# Other segmentation architectures

- **Fully convolutional networks (FCN)**
- Pre-cursor to U-Net, similar in structure but simpler upsampling pathway




Shelhamer*, Long*, et al. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

- **DeepLab (v1-v3)**
- Uses "atrous convolutions" to control a filter's field of view
- Parallel atrous convolutions with different rates for multi-scale features



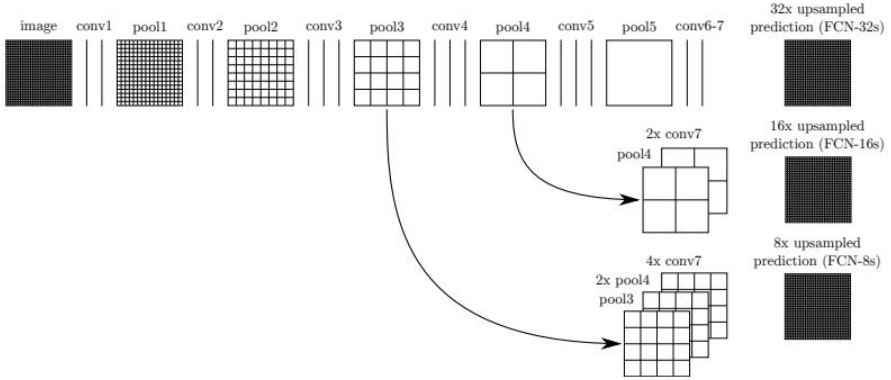

Chen et al. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE TPAMI, 2017.
Chen et al. Rethinking Atrous Convolution for Semantic Image Segmentation. 2917.

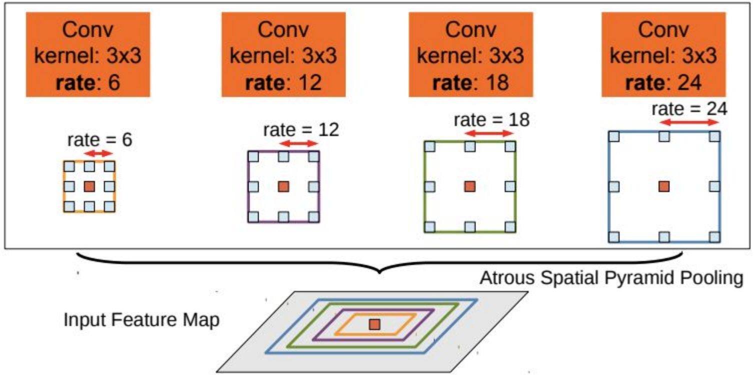
# Other segmentation architectures

Can try DeepLab v3+ for segmentation projects!

- **Fully convolutional networks (FCN)**
- Pre-cursor to U-Net, similar in structure but simpler upsampling pathway

- **DeepLab (v1-v3+)**
- Uses "atrous convolutions" to control a filter's field of view
- Parallel atrous convolutions with different rates for multi-scale features



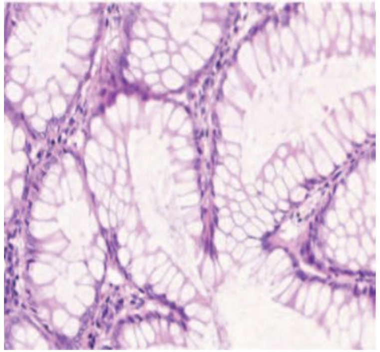




Shelhamer*, Long*, et al. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Chen et al. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE TPAMI, 2017.
Chen et al. Rethinking Atrous Convolution for Semantic Image Segmentation. 2917.

# Richer visual recognition tasks: segmentation and detection

**Classification**

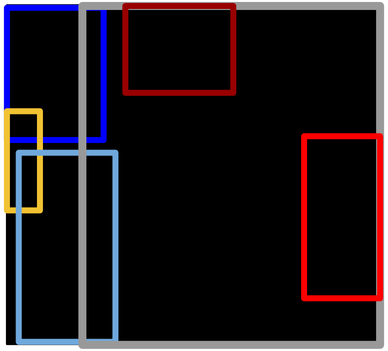


Output:
one category label for image (e.g., colorectal glands)

**Semantic Segmentation**

Output:
category label for each pixel in the image

**Detection**



Output:
Spatial bounding box for each **instance** of a category object in the image

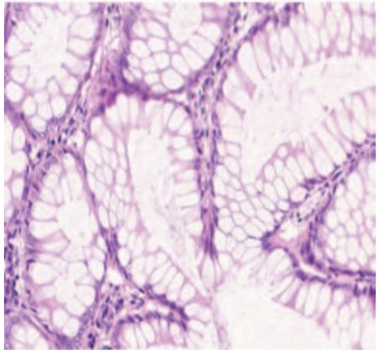
**Instance Segmentation**

Output:
Category label and instance label for each pixel in the image

Figures: Chen et al. 2016. https://arxiv.org/pdf/1604.02677.pdf

# Richer visual recognition tasks: segmentation and detection

**Classification**

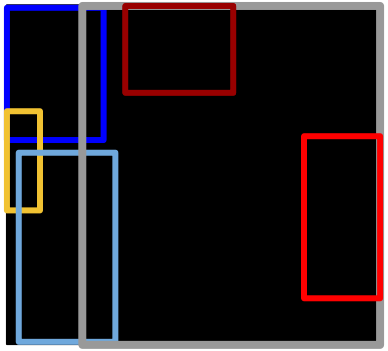


Output:
one category label for image (e.g., colorectal glands)

**Semantic Segmentation**

Output:
category label for each pixel in the image

**Detection**



Output:
Spatial bounding box for each **instance** of a category object in the image

Next Time:

**Instance Segmentation**

Output:
Category label and instance label for each pixel in the image

Distinguishes between different instances of an object

Figures: Chen et al. 2016. https://arxiv.org/pdf/1604.02677.pdf

# Summary

Finished up medical image classification

Beyond classification to richer visual recognition tasks

- Semantic segmentation
- Object detection
- Instance segmentation

Next time: Advanced vision models (Object detection, Instance segmentation, 3D and video)