Lecture 4:
Medical Images:
Classification (Part 2),
Segmentation, Detection

Announcements

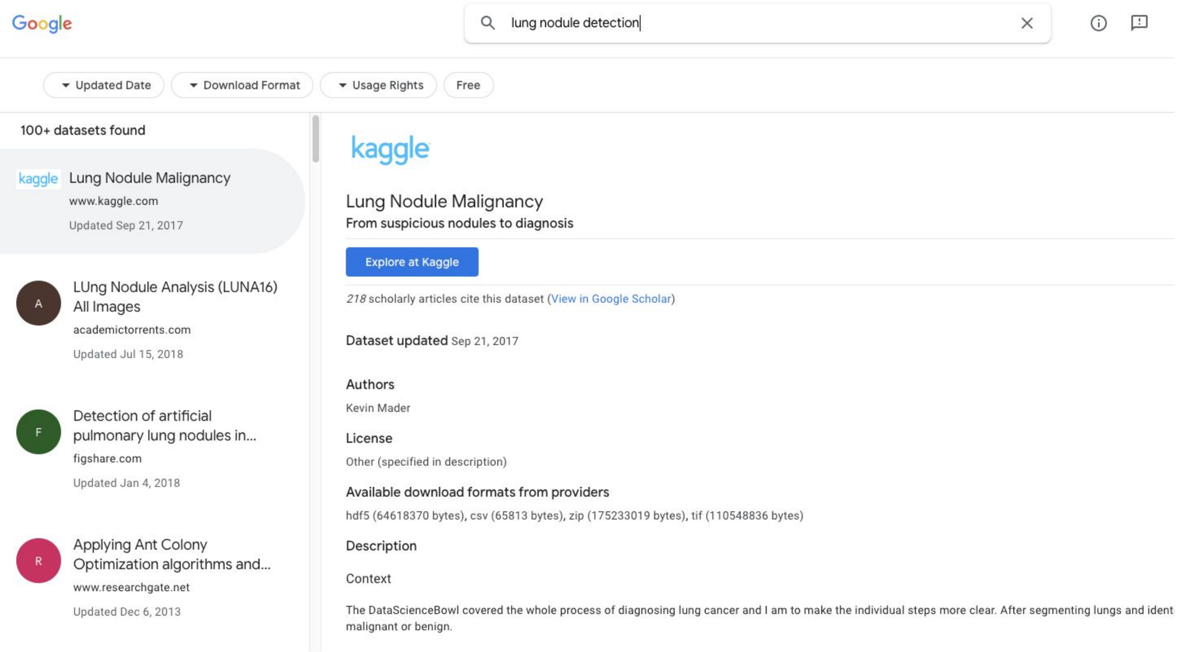
- A0 was due yesterday
- A1 was also released yesterday, due in 2 weeks (Tue 10/6)
  - You will need to download several datasets to do the assignment. Make sure to start early!
  - 3 parts:
    - Medical image classification
    - Medical image segmentation in 2D
    - Medical image segmentation in 3D, with semi-supervised learning
- Tensorflow Review Session this Fri 1pm, helpful for A1

Announcements - Course project

- Start thinking about your course project
    - Project proposal due Fri 10/9
    - See http://biods220.stanford.edu/finalproject.html for all course project components and requirements
    - We have released some project ideas (curated from the Stanford community) on Piazza
        - 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.
            - 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!

Google dataset search

datasetsearch.research.google.com



100+ datasets found

kaggle Lung Nodule Malignancy
www.kaggle.com
Updated Sep 21, 2017

LUng Nodule Analysis (LUNA16) All Images
academictorrents.com
Updated Jul 15, 2018

Detection of artificial pulmonary lung nodules in...
figshare.com
Updated Jan 4, 2018

Applying Ant Colony Optimization algorithms and...
www.researchgate.net
Updated Dec 6, 2013

kaggle

Lung Nodule Malignancy
From suspicious nodules to diagnosis

Explore at Kaggle

218 scholarly articles cite this dataset (View in Google Scholar)

Dataset updated Sep 21, 2017

Authors
Kevin Mader

License
Other (specified in description)

Available download formats from providers
hdf5 (64618370 bytes), csv (65813 bytes), zip (175233019 bytes), tif (110548836 bytes)

Description

Context

The DataScienceBowl covered the whole process of diagnosing lung cancer and I am to make the individual steps more clear. After segmenting lungs and ident
malignant or benign.

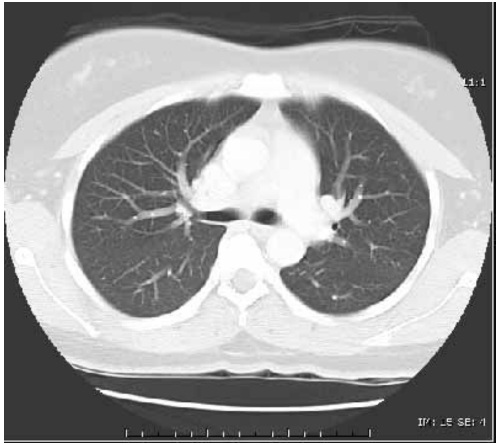
Announcements - Course project

- Preview of graded components:
  - Proposal: Due Fri 10/9.
  - Milestone: Due Fri 10/30.
  - Project milestone presentations (4-5 min): During Mon 11/2 class time.
  - TA project advising sessions: Sign-up by Fri 11/6.
  - Final project presentations (4-5 min): During Wed 11/18 class time.
  - Final report due: Fri 11/20.

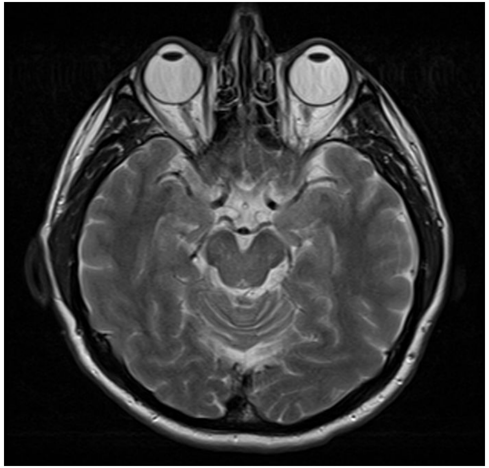
Last time: Deep learning models for image classification

E.g.:

X-rays (invented 1895).



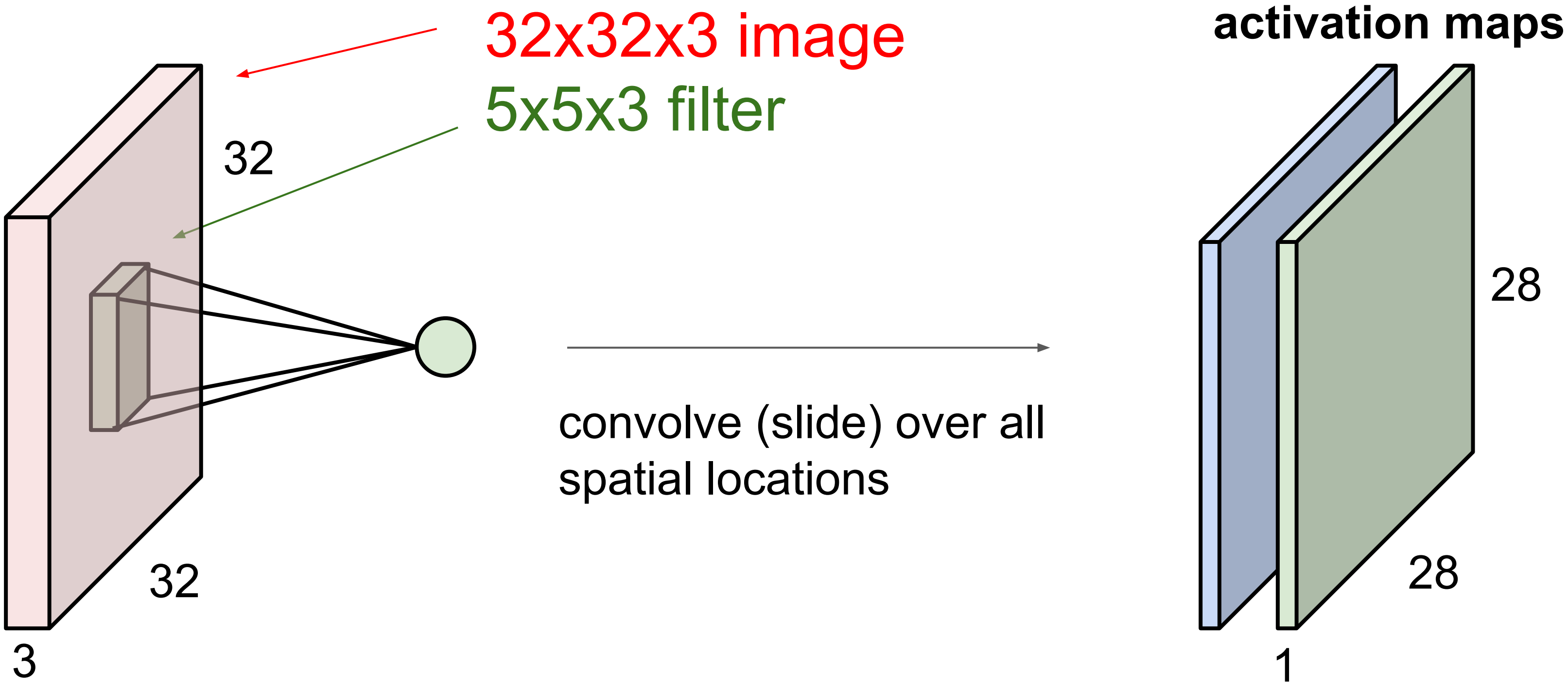
CT (invented 1972).



MRI (invented 1977).

consider a second, green filter

Convolutional layer



32x32x3 image

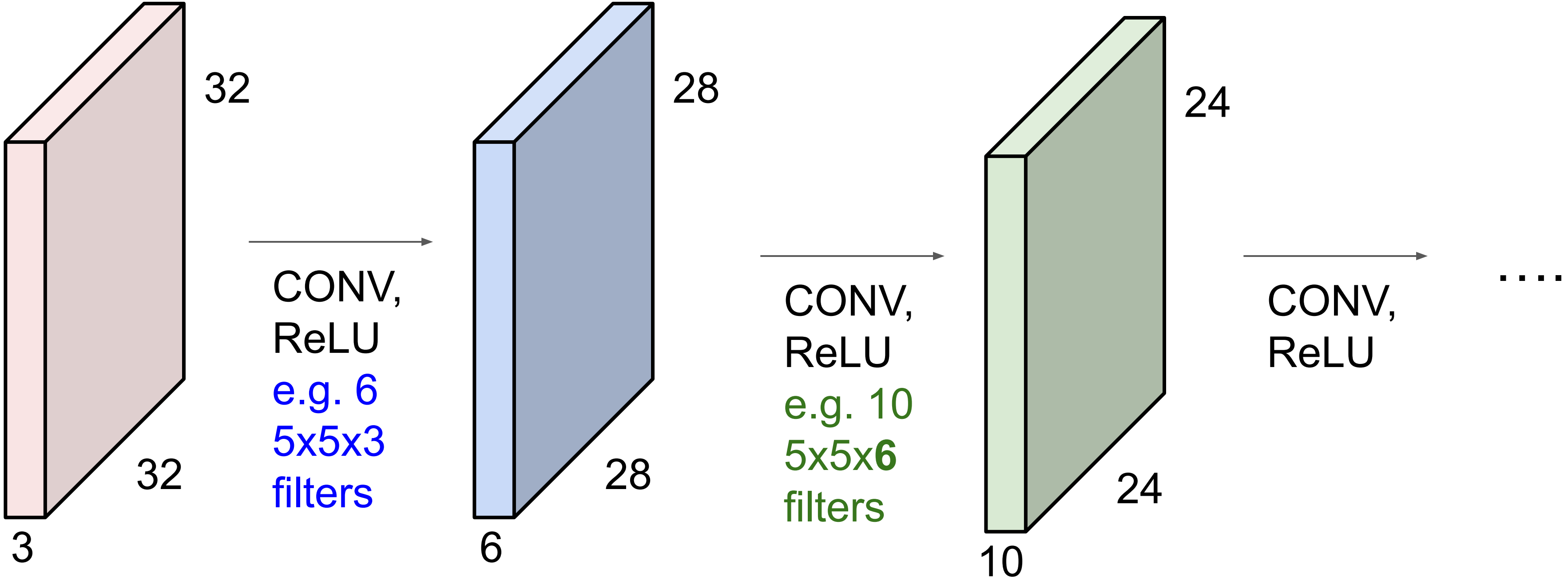
5x5x3 filter

activation maps

convolve (slide) over all spatial locations

Slide credit: CS231n

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



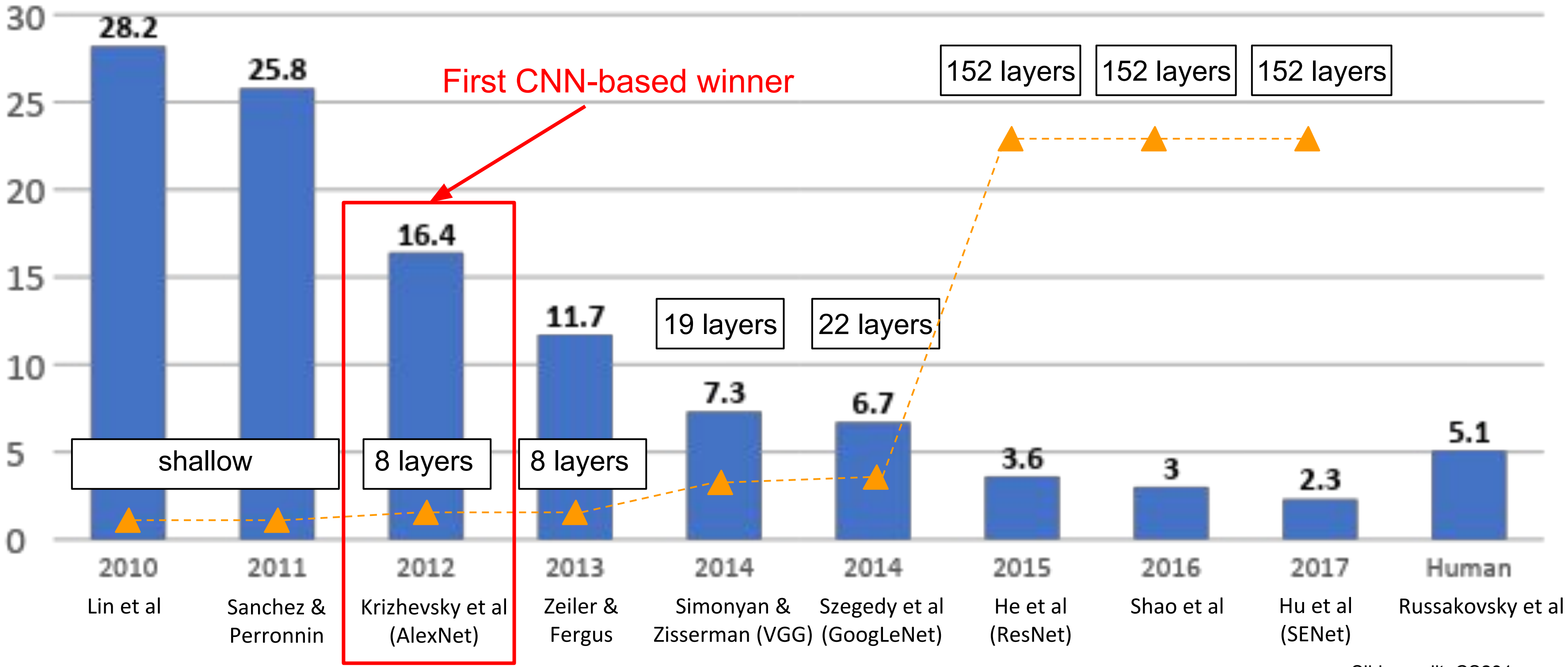
CONV, ReLU e.g. 6 5x5x3 filters

CONV, ReLU e.g. 10 5x5x**6** filters

CONV, ReLU

Slide credit: CS231n

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



Slide credit: CS231n

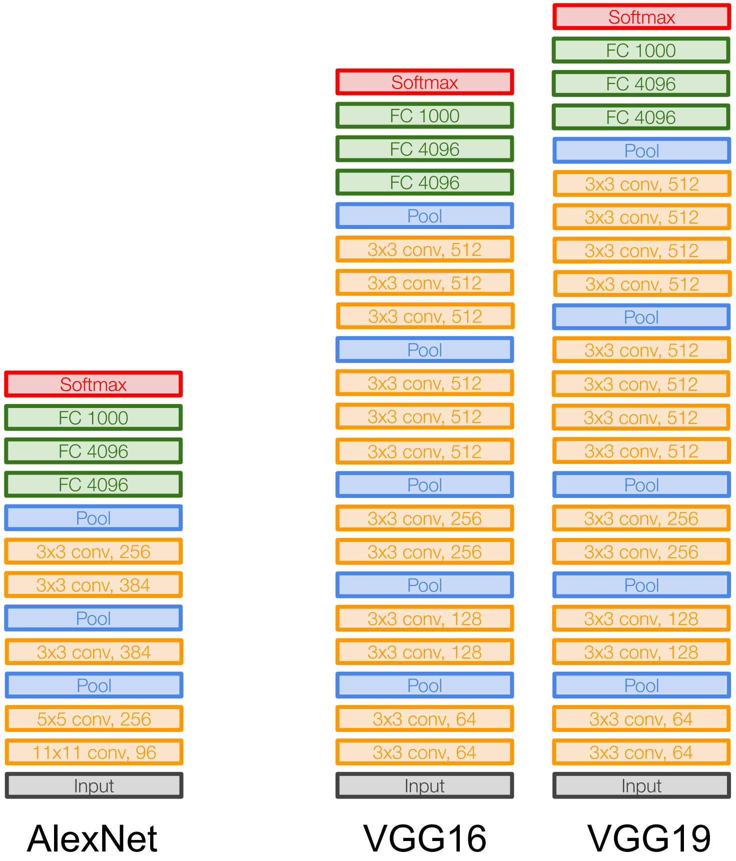
VGGNet

[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



AlexNet
- Softmax
- FC 1000
- FC 4096
- FC 4096
- Pool
- 3x3 conv, 256
- 3x3 conv, 384
- Pool
- 3x3 conv, 384
- Pool
- 5x5 conv, 256
- 11x11 conv, 96
- Input

VGG16
- Softmax
- FC 1000
- FC 4096
- FC 4096
- Pool
- 3x3 conv, 512
- 3x3 conv, 512
- 3x3 conv, 512
- Pool
- 3x3 conv, 512
- 3x3 conv, 512
- 3x3 conv, 512
- Pool
- 3x3 conv, 256
- 3x3 conv, 256
- Pool
- 3x3 conv, 128
- 3x3 conv, 128
- Pool
- 3x3 conv, 64
- 3x3 conv, 64
- Input

VGG19
- Softmax
- FC 1000
- FC 4096
- FC 4096
- Pool
- 3x3 conv, 512
- 3x3 conv, 512
- 3x3 conv, 512
- 3x3 conv, 512
- Pool
- 3x3 conv, 512
- 3x3 conv, 512
- 3x3 conv, 512
- 3x3 conv, 512
- Pool
- 3x3 conv, 256
- 3x3 conv, 256
- Pool
- 3x3 conv, 128
- 3x3 conv, 128
- Pool
- 3x3 conv, 64
- 3x3 conv, 64
- Input

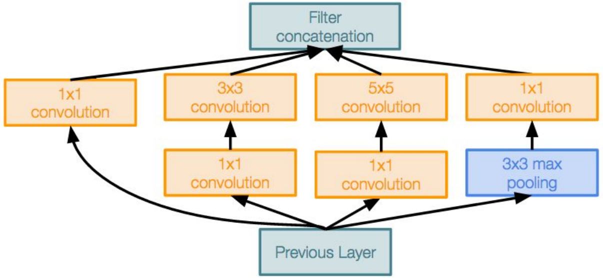
Slide credit: CS231n

GoogLeNet

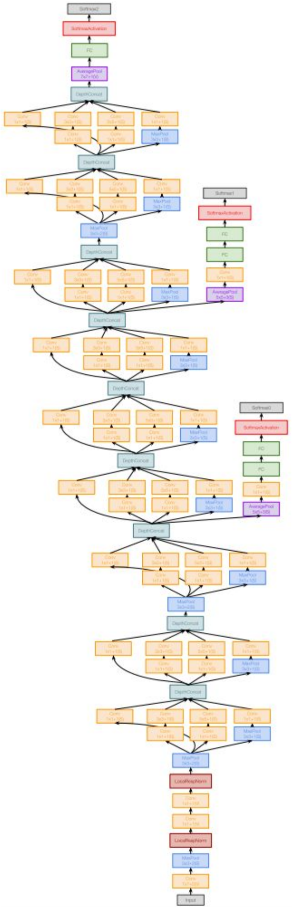
[Szegedy et al., 2014]

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”

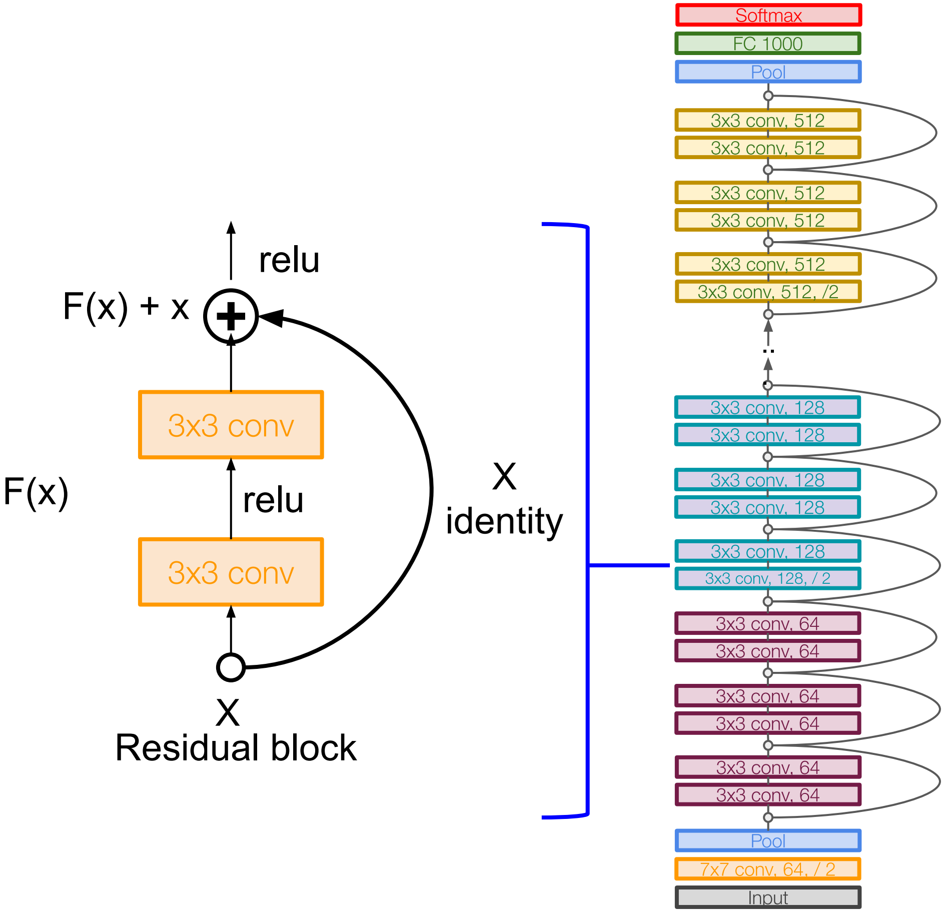
Slide credit: CS231n

ResNet

[He et al., 2015]

Full ResNet architecture:

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

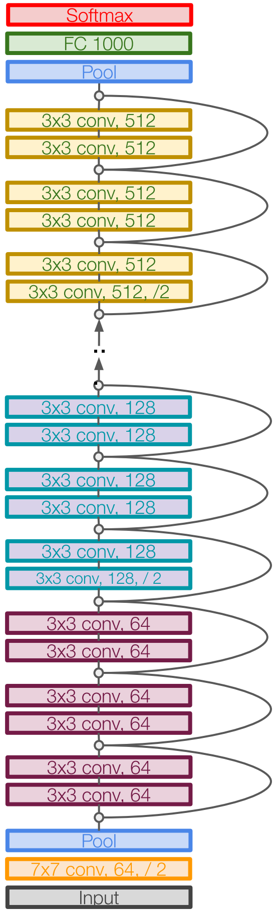


Slide credit: CS231n

ResNet

[He et al., 2015]

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

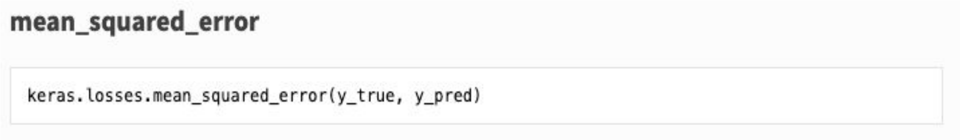


Slide credit: CS231n

Common loss functions

You will find these in tensorflow!

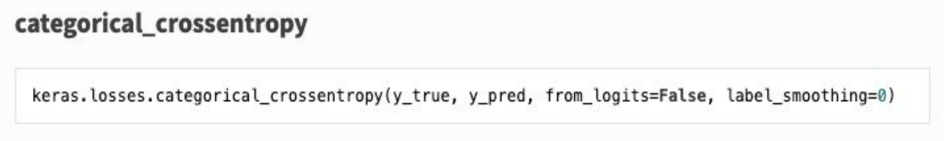
In Keras:



mean_squared_error

```
keras.losses.mean_squared_error(y_true, y_pred)
```

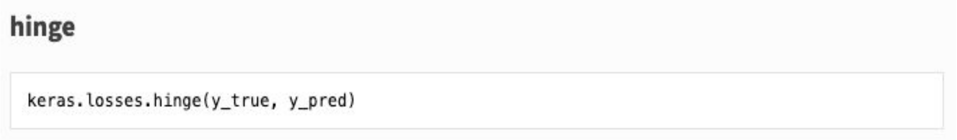
Mean squared error (MSE) is another name for regression loss



categorical_crossentropy

```
keras.losses.categorical_crossentropy(y_true, y_pred, from_logits=False, label_smoothing=0)
```

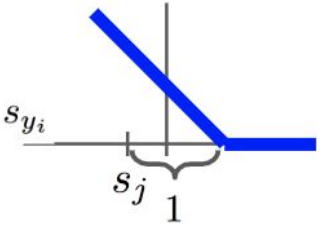
Covers both BCE and Softmax loss (remember softmax is a multiclass extension of BCE)



hinge

```
keras.losses.hinge(y_true, y_pred)
```

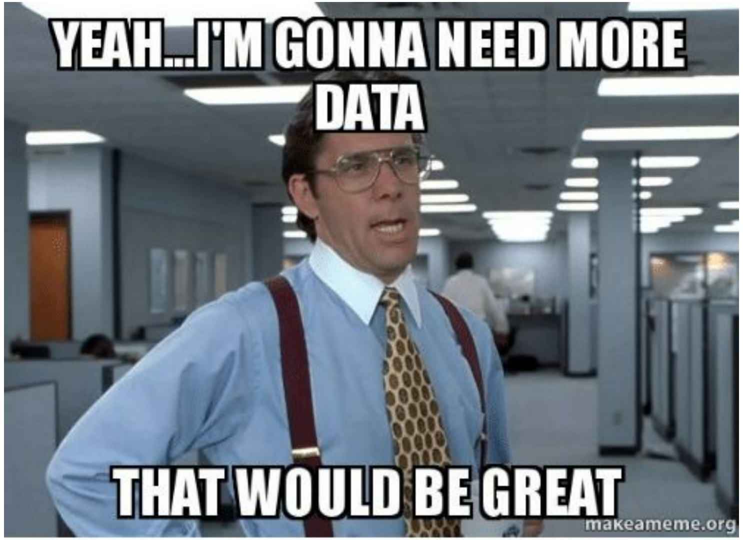
Hinge is another name for SVM loss, due to the loss function shape.



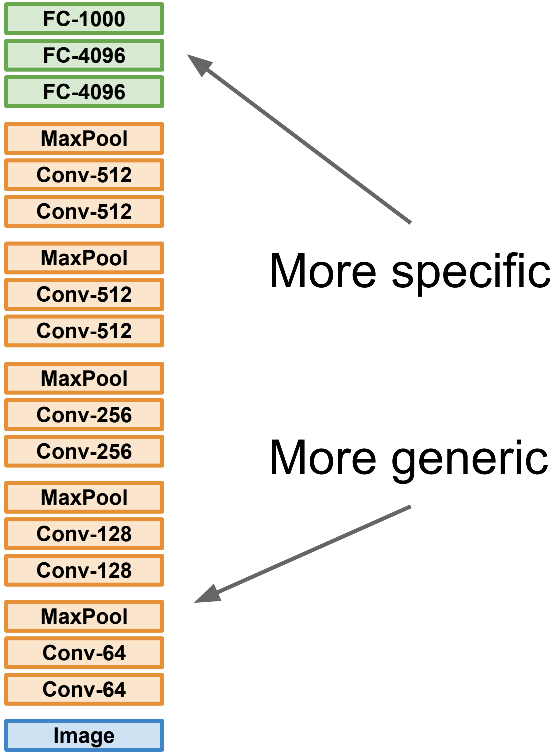
https://keras.io/losses/

How much data do you need for deep learning?

A: A lot.



Transfer learning from a large dataset to your dataset...



FC-1000

FC-4096

FC-4096

MaxPool

Conv-512

Conv-512

MaxPool

Conv-512

Conv-512

MaxPool

Conv-256

Conv-256

MaxPool

Conv-128

Conv-128

MaxPool

Conv-64

Conv-64

Image

More specific

More generic

	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

Often good idea to try this first, try fine-tuning all layers of the network

Slide credit: CS231n

Today:

Medical Images: Classification

- Deep learning models for image classification
- Data considerations for image classification models
- Evaluating image classification models
- Case studies

Medical Images: Advanced Vision Models (Detection and Segmentation)

Evaluating image classification models

Evaluation metrics

Confusion matrix

		Prediction	
		0	1
Ground Truth	0	TN	FP
	1	FN	TP

Accuracy: (TP + TN) / total

Evaluation metrics

Confusion matrix

		Prediction	
		0	1
Ground Truth	0	TN	FP
	1	FN	TP

Accuracy: (TP + TN) / total

Q: When might evaluating purely accuracy be problematic?

Evaluation metrics

Confusion matrix

		Prediction	
		0	1
Ground Truth	0	TN	FP
	1	FN	TP

Accuracy: (TP + TN) / total

Q: When might evaluating purely accuracy be problematic?

A: Imbalanced datasets.

Evaluation metrics

Confusion matrix

		Prediction	
		0	1
Ground Truth	0	TN	FP
	1	FN	TP

Accuracy: (TP + TN) / total

Sensitivity / Recall (true positive rate)**:**
TP / total positives

Specificity (true negative rate)**:**
TN / total negatives

Precision (positive predictive value)**:**
TP / total predicted positives

Negative predictive value:
TN / total predicted negatives

Evaluation metrics

As we vary our classifier’s score threshold to predict a positive, we can trade-off different values of these metrics

Confusion matrix

		Prediction	
		0	1
Ground Truth	0	TN	FP
	1	FN	TP

Accuracy: (TP + TN) / total

Sensitivity / Recall (true positive rate)**:**
TP / total positives

Specificity (true negative rate)**:**
TN / total negatives

Precision (positive predictive value)**:**
TP / total predicted positives

Negative predictive value:
TN / total predicted negatives

Evaluation metrics

Q: As prediction threshold increases, how does that generally affect sensitivity? Specificity?

Confusion matrix

		Prediction	
		0	1
Ground Truth	0	TN	FP
	1	FN	TP

Accuracy: (TP + TN) / total

Sensitivity / Recall (true positive rate)**:**
TP / total positives

Specificity (true negative rate)**:**
TN / total negatives

Precision (positive predictive value)**:**
TP / total predicted positives

Negative predictive value:
TN / total predicted negatives

Evaluation metrics

Q: As prediction threshold increases, how does that generally affect sensitivity? Specificity?
A: Sensitivity goes down, specificity up

Confusion matrix

		Prediction	
		0	1
Ground Truth	0	TN	FP
	1	FN	TP

Accuracy: (TP + TN) / total

Sensitivity / Recall (true positive rate)**:**
TP / total positives

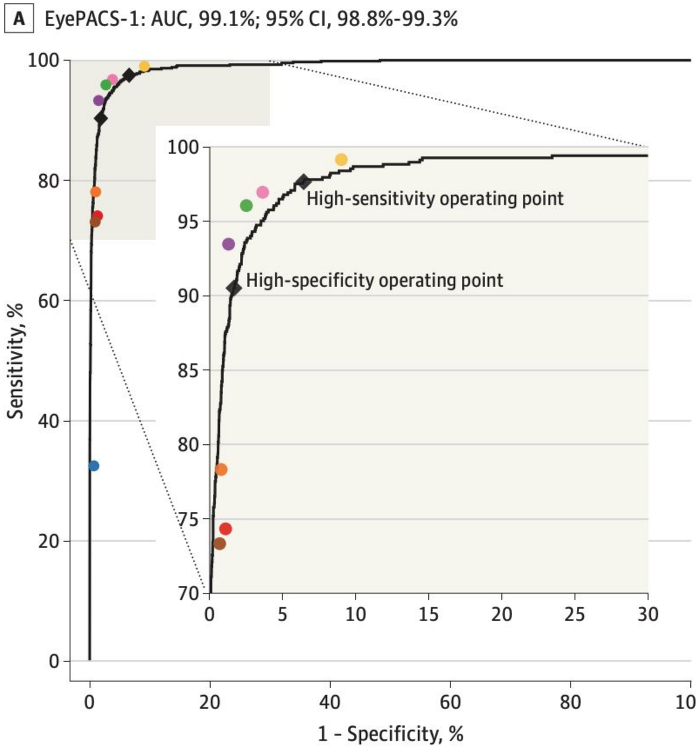
Specificity (true negative rate)**:**
TN / total negatives

Precision (positive predictive value)**:**
TP / total predicted positives

Negative predictive value:
TN / total predicted negatives

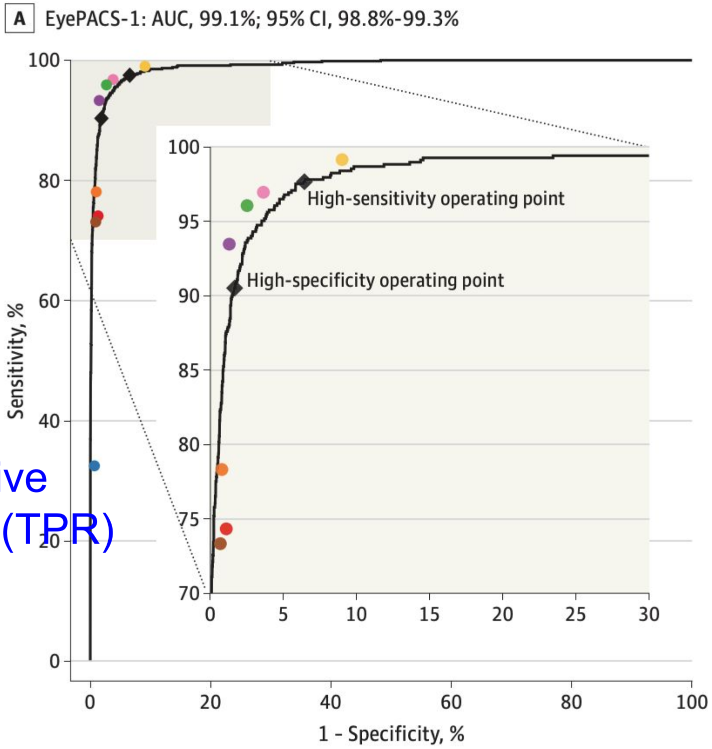
Evaluation metrics

- **Receiver Operating Characteristic (ROC) curve**:
  - Plots sensitivity and specificity (specifically, 1 - specificity) as prediction threshold is varied
  - Gives trade-off between sensitivity and specificity
  - Also report summary statistic AUC (area under the curve)



Evaluation metrics

- **Receiver Operating Characteristic (ROC) curve**:
  - Plots sensitivity and specificity (specifically, 1 - specificity) as prediction threshold is varied
  - Gives trade-off between sensitivity and specificity
  - Also report summary statistic AUC (area under the curve)



True Positive Rate (TPR)

False Positive Rate (FPR)

Evaluation metrics

- Sometimes also see **precision recall curve**
    - More informative when dataset is heavily imbalanced (specificity = true negative rate less meaningful in this case)

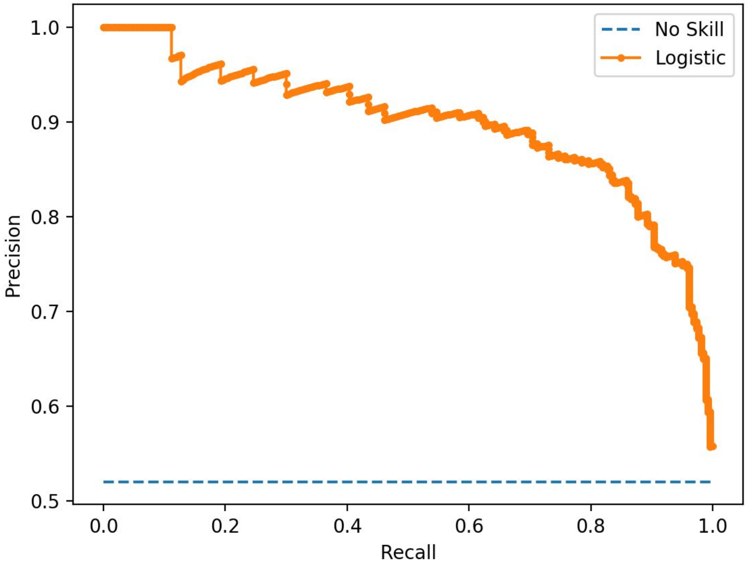


Figure credit: https://3qeqpr26caki16dnhd19sv6by6v-wpengine.netdna-ssl.com/wp-content/uploads/2018/08/Precision-Recall-Plot-for-a-No-Skill-Classifier-and-a-Logistic-Regression-Model4.png

Evaluation metrics

- Selecting optimal trade-off points
  - Maximize **Youden's Index**
    - J = sensitivity + specificity - 1
    - Gives equal weight to optimizing true positives and true negatives

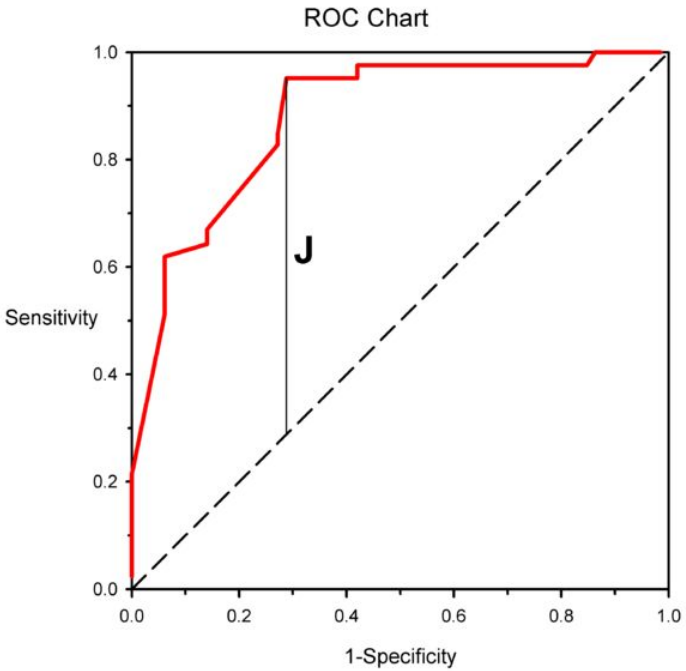


Figure credit: https://en.wikipedia.org/wiki/File:ROC_Curve_Youden_J.png

Evaluation metrics

- Selecting optimal trade-off points
    - Maximize **Youden's Index**
        - J = sensitivity + specificity - 1
        - Gives equal weight to optimizing true positives and true negatives

Also equal to distance above chance line for a balanced dataset: sensitivity - (1 - specificity) = sensitivity + specificity - 1

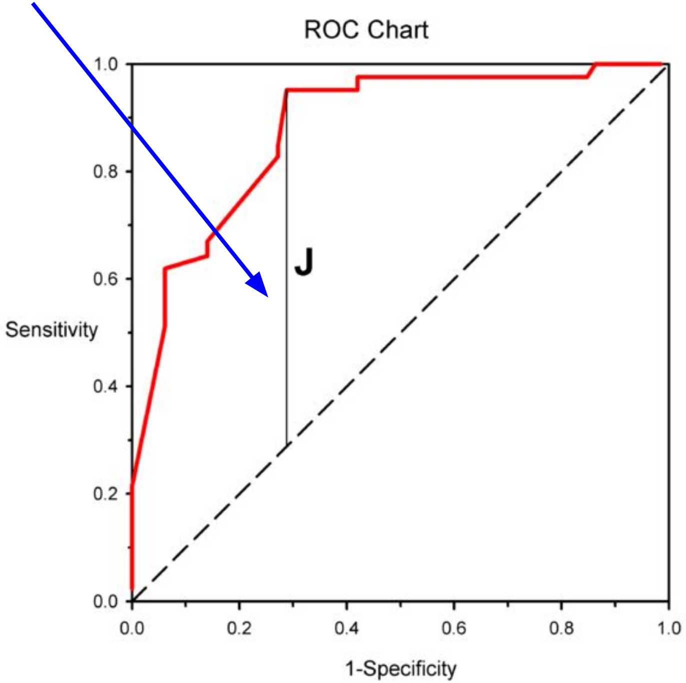


Figure credit: https://en.wikipedia.org/wiki/File:ROC_Curve_Youden_J.png

Evaluation metrics

- Selecting optimal trade-off points
    - Maximize **Youden's Index**
        - J = sensitivity + specificity - 1
        - Gives equal weight to optimizing true positives and true negatives
    - Sometimes also see F-measure (or F1 score)
        - F1 = 2*(precision*recall) / (precision + recall)
        - Harmonic mean of precision and recall

Also equal to distance above chance line for a balanced dataset: sensitivity - (1 - specificity) = sensitivity + specificity - 1

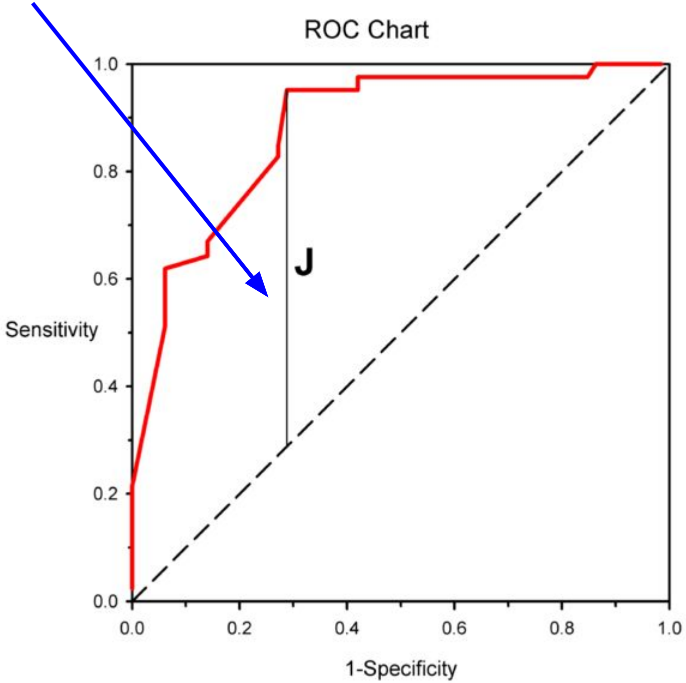


Figure credit: https://en.wikipedia.org/wiki/File:ROC_Curve_Youden_J.png

Evaluation metrics

- Selecting optimal trade-off points
  - Maximize **Youden's Index**
    - J = sensitivity + specificity - 1
    - Gives equal weight to optimizing true positives and true negatives
  - Sometimes also see F-measure (or F1 score)
    - F1 = 2*(precision*recall) / (precision + recall)
    - Harmonic mean of precision and recall

But selected trade-off points could also depend on application

Also equal to distance above chance line for a balanced dataset: sensitivity - (1 - specificity) = sensitivity + specificity - 1

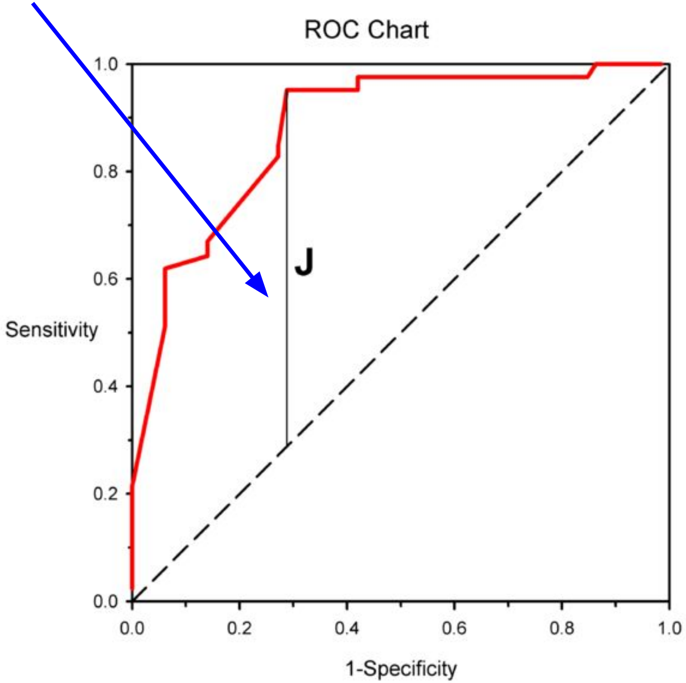


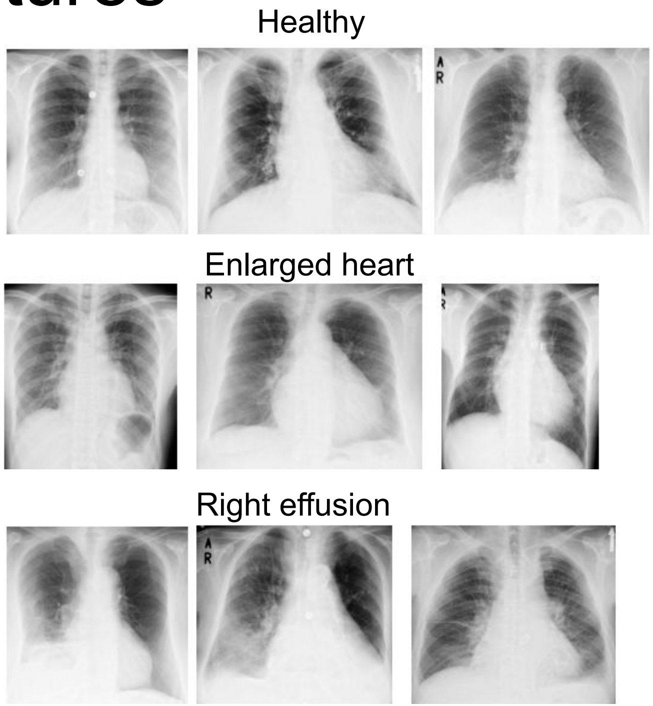
Figure credit: https://en.wikipedia.org/wiki/File:ROC_Curve_Youden_J.png

Case Studies of CNNs for Medical Imaging Classification

Early steps of deep learning in medical imaging: using ImageNet CNN features

Bar et al. 2015

- Input: Chest **x-ray images**
- Output: Several binary classification tasks
    - Right pleural effusion or not
    - Enlarged heart or not
    - Healthy or abnormal
- Very small dataset: 93 frontal chest x-ray images

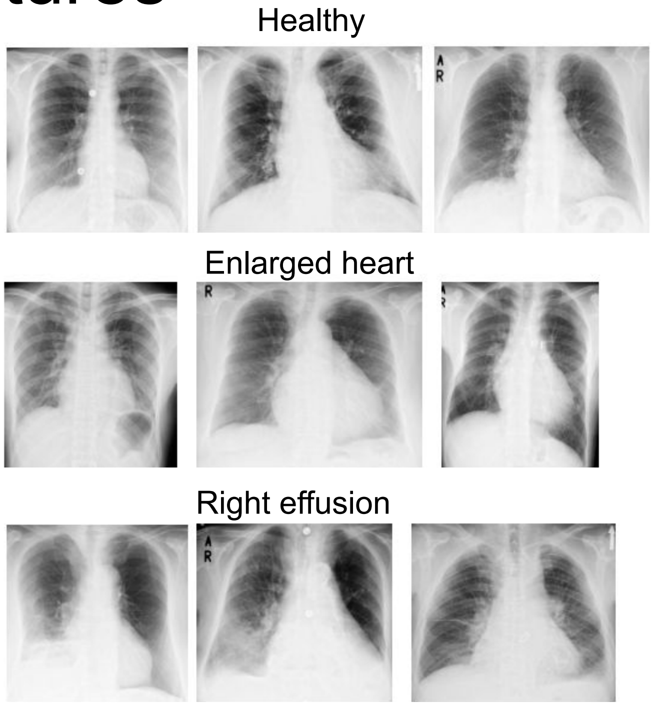


Bar et al. Deep learning with non-medical training used for chest pathology identification. SPIE, 2015.

Early steps of deep learning in medical imaging: using ImageNet CNN features

Bar et al. 2015

- Input: Chest **x-ray images**
- Output: Several binary classification tasks
  - Right pleural effusion or not
  - Enlarged heart or not
  - Healthy or abnormal
- Very small dataset: 93 frontal chest x-ray images

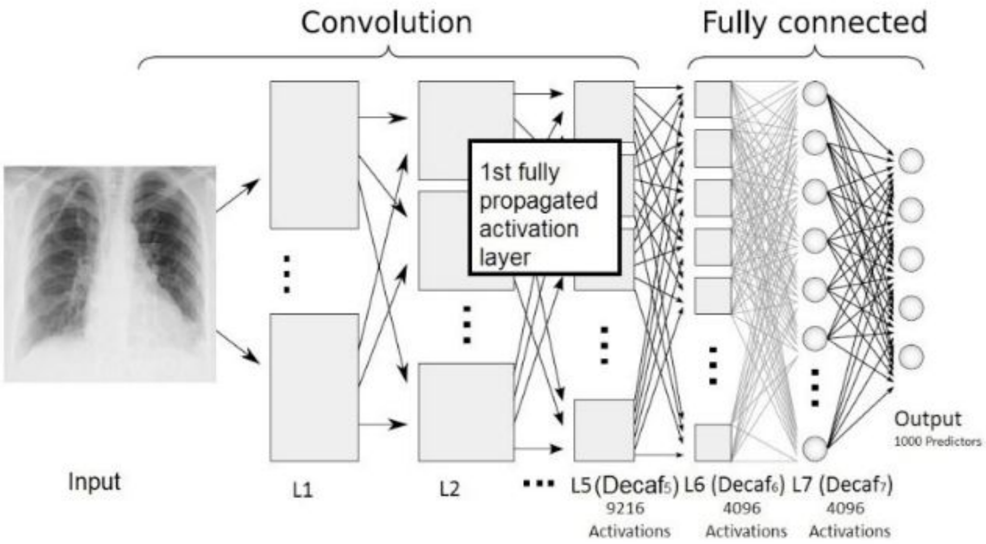


Q: How might we approach this problem?

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


Convolution
Fully connected
1st fully propagated activation layer
Input
L1
L2
...
L5 (Decaf5) 9216 Activations
L6 (Decaf6) 4096 Activations
L7 (Decaf7) 4096 Activations
Output
1000 Predictors

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

Bar et al. 2015

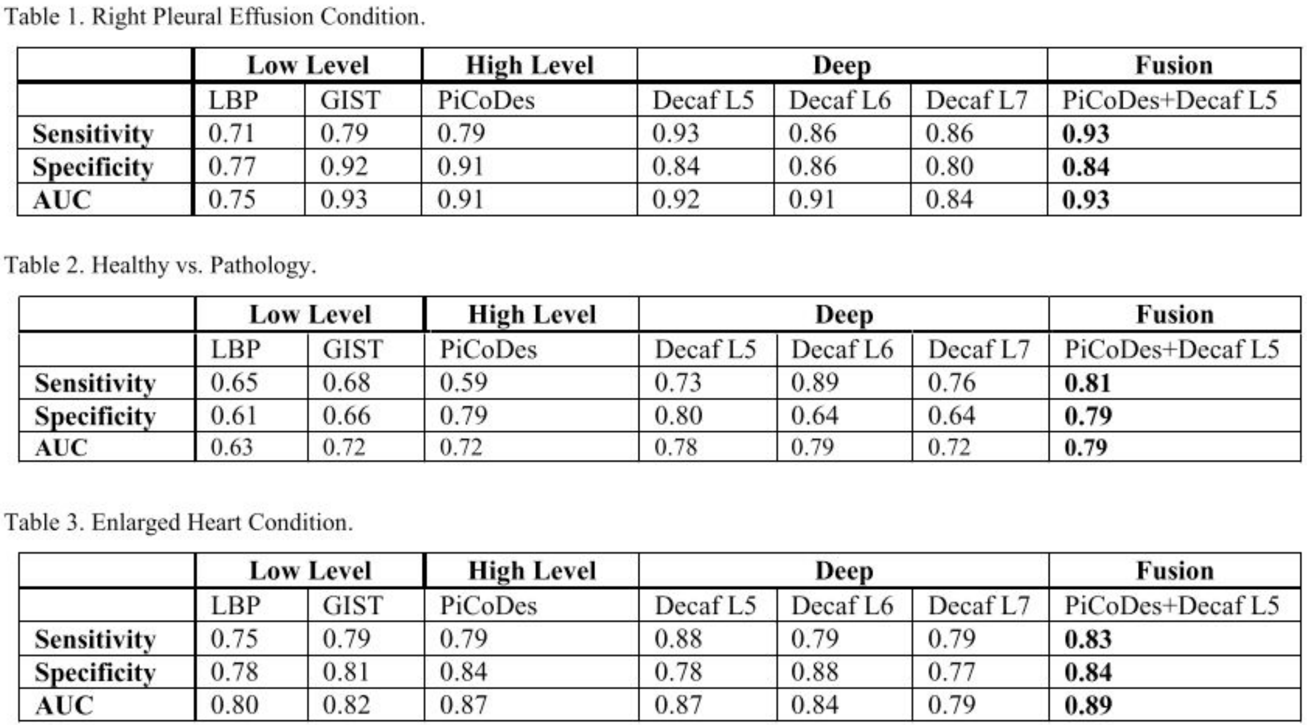


Table 1. Right Pleural Effusion Condition.

	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?

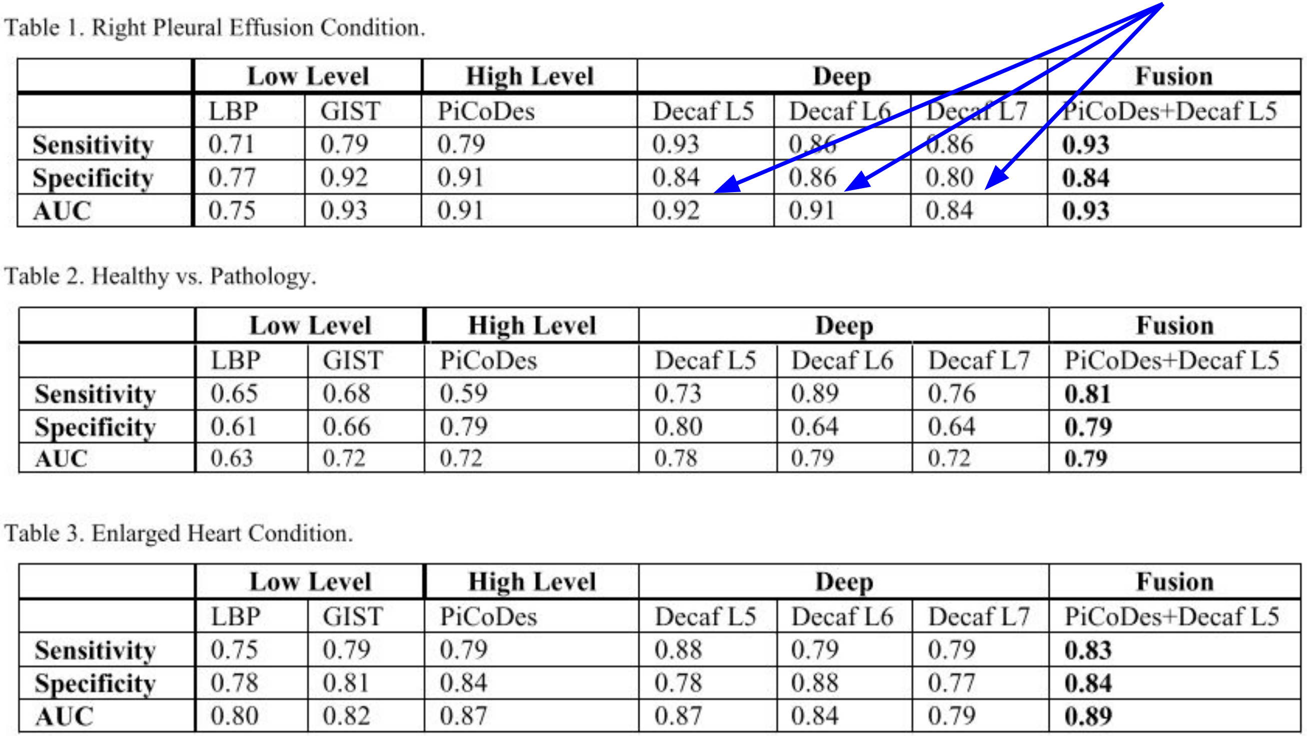


Table 1. Right Pleural Effusion Condition.

	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**

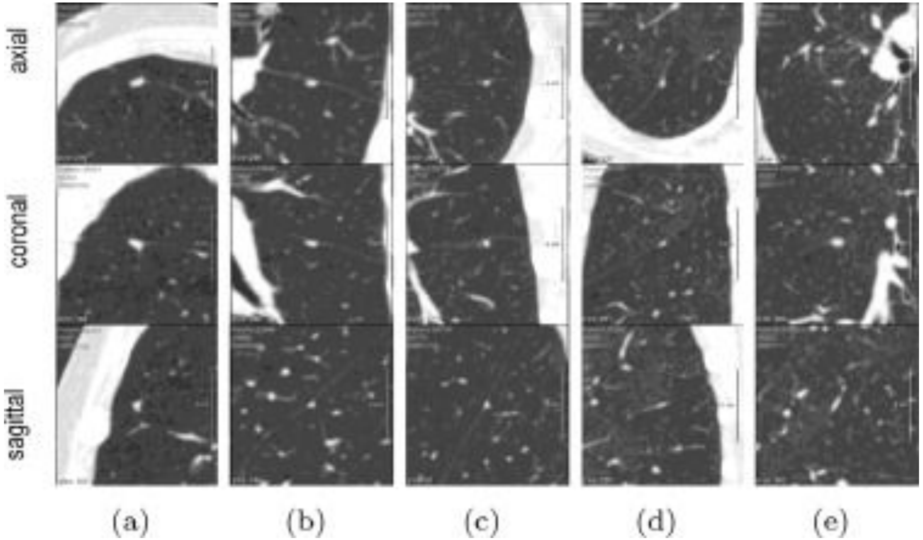
Table 3. Enlarged Heart Condition.

	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

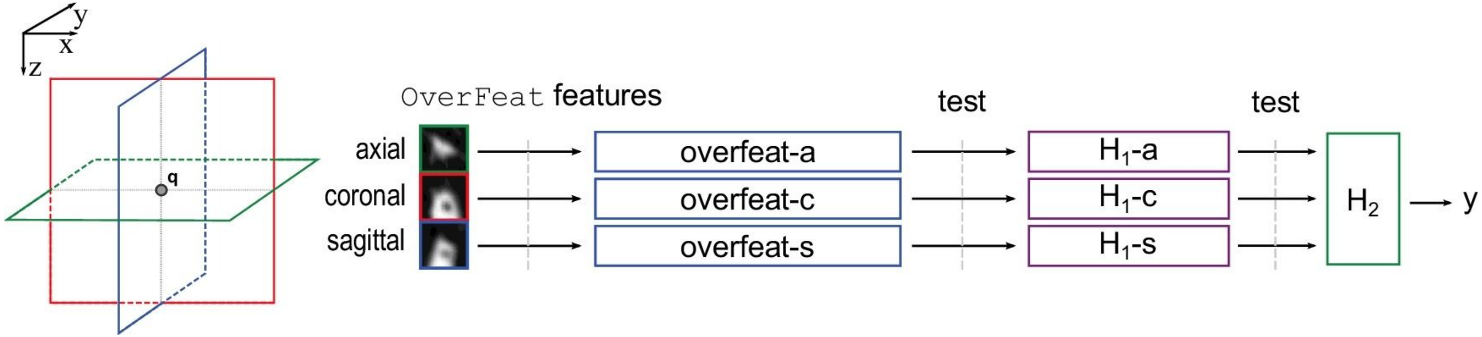
- Task: classification of lung nodules in **3D CT scans** as peri-fissural nodules (PFN, likely to be benign) or not
- Dataset: 568 nodules from 1729 scans at a single institution. (65 typical PFNs, 19 atypical PFNs, 484 non-PFNs).
- Data pre-processing: prescaling from CT hounsfield units (HU) into [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

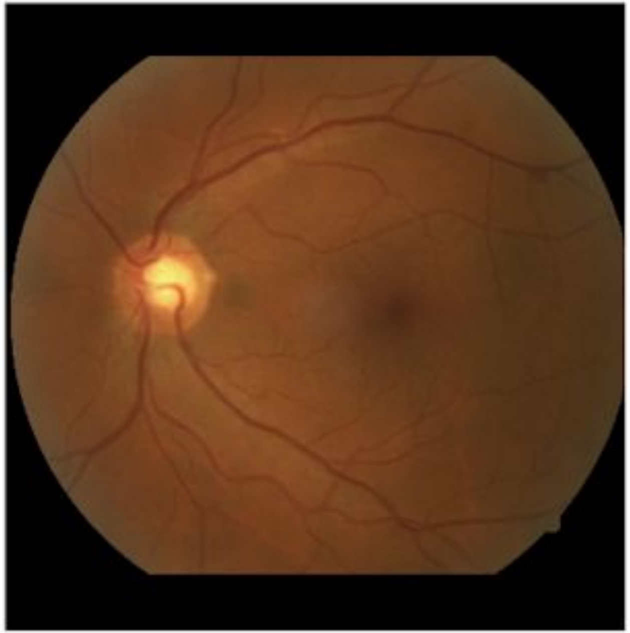
- Also extracted features from a deep learning model trained on ImageNet
  - Overfeat feature extractor (similar to AlexNet, but trained using additional losses for localization and detection)
  - To capture 3D information, extracted features from 3 different 2D views of each nodule, then input into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

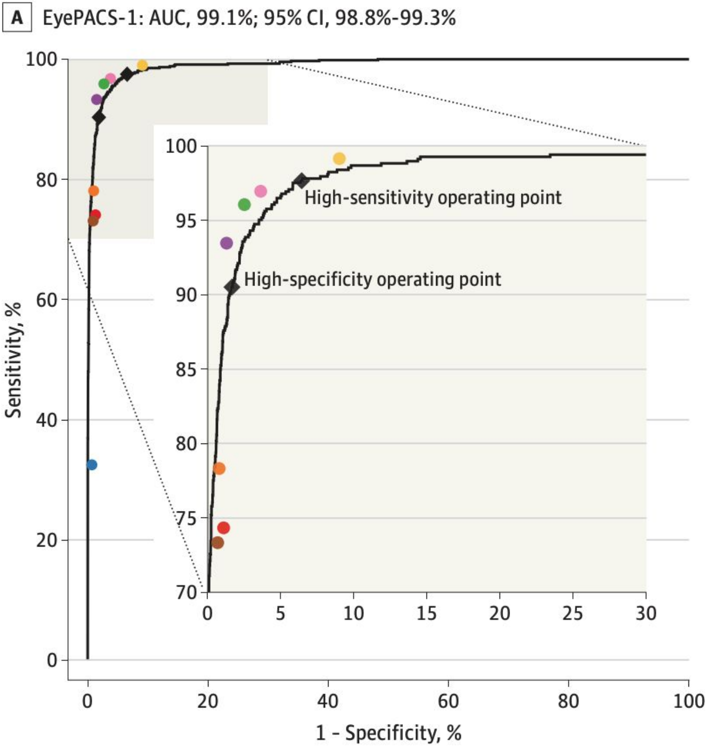
Graders provided finer-grained labels which were then consolidated into (easier) binary prediction problems

- **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.

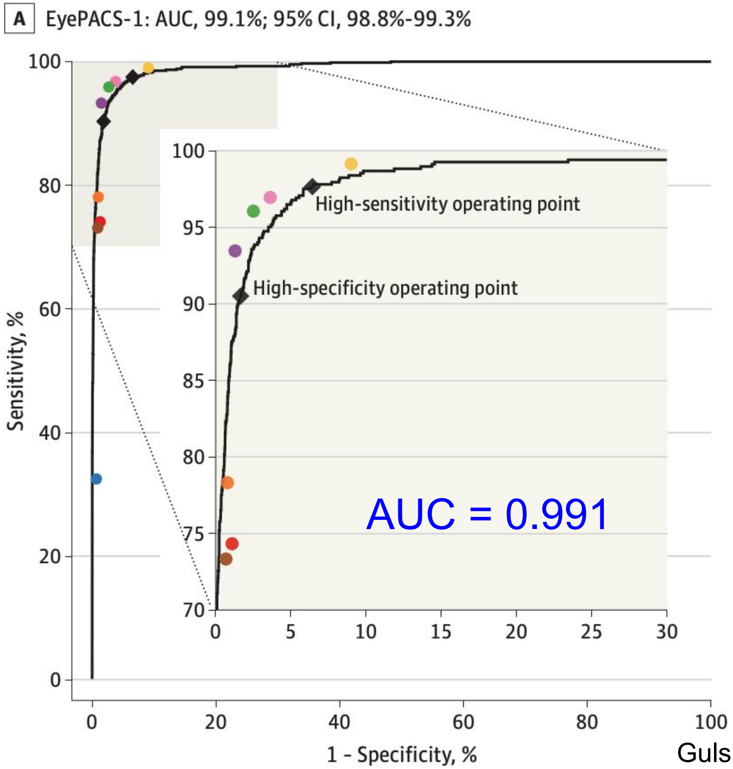
Gulshan et al. 2016

- **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



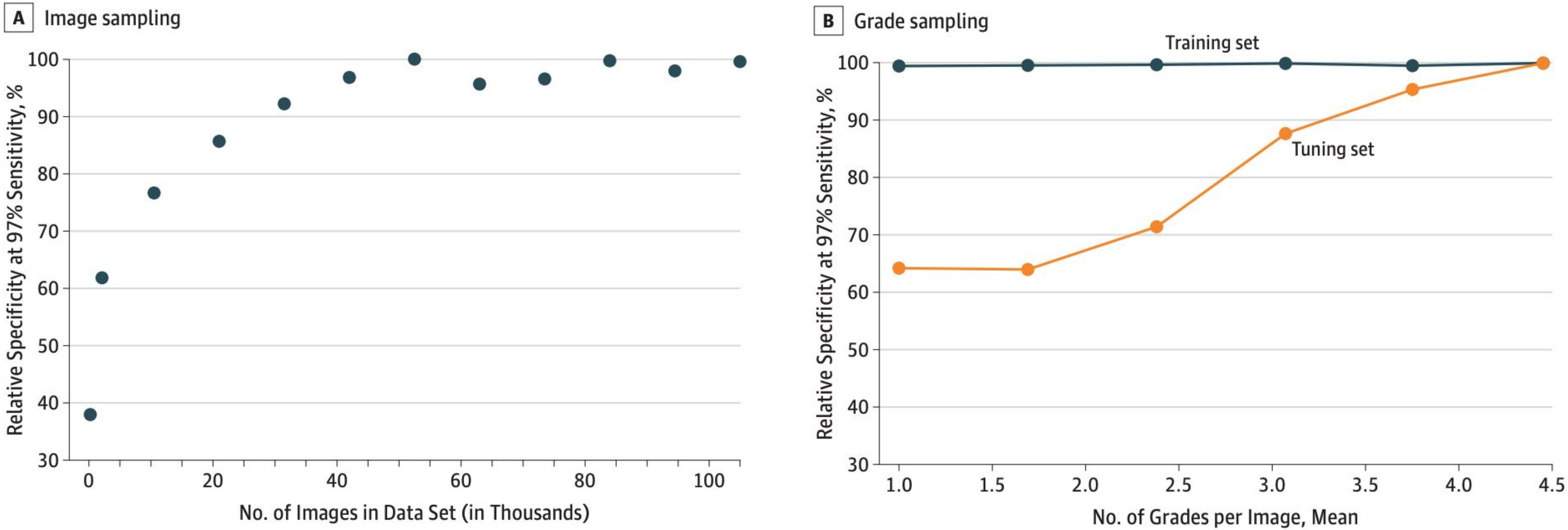
AUC = 0.991

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.

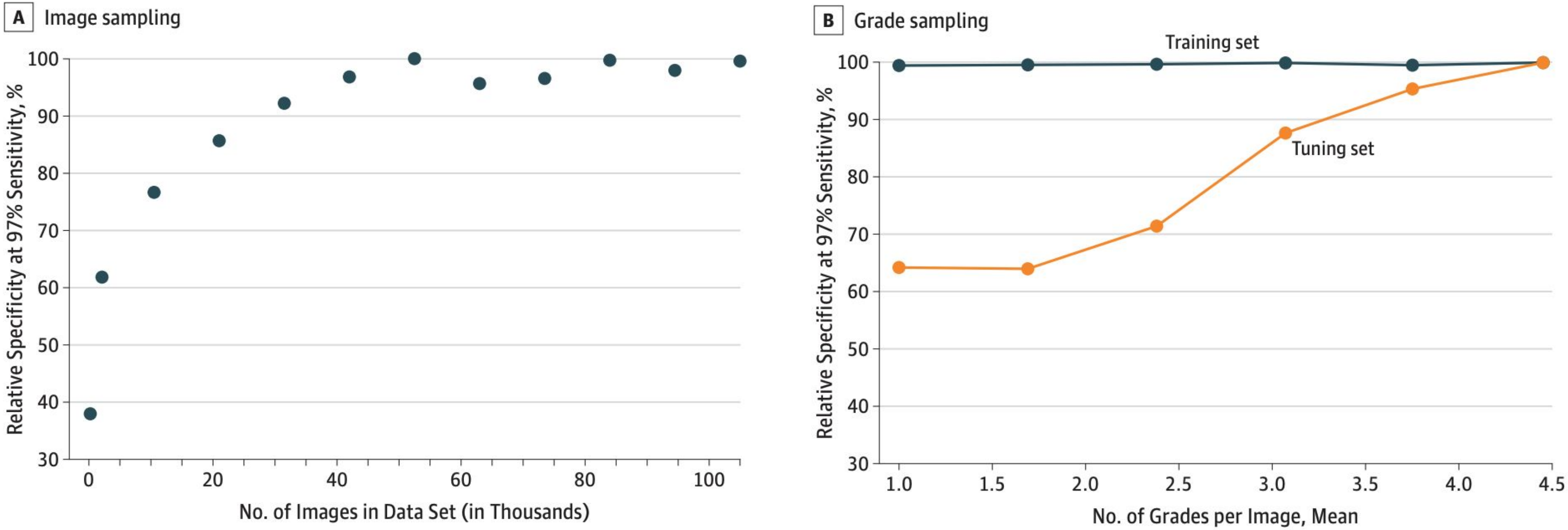
Gulshan et al. 2016



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

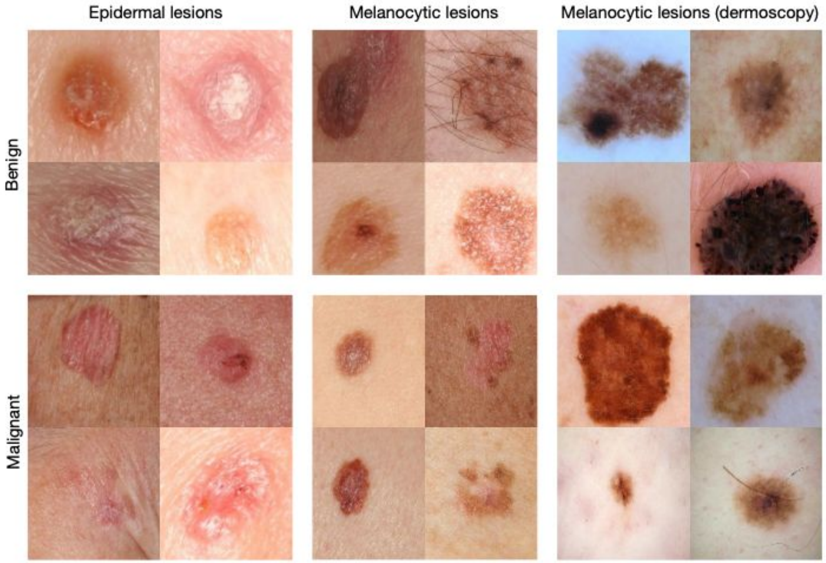
Q: What could explain the difference in trends for reducing # grades / image on training set vs. tuning set, on tuning set performance?



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

Esteva et al. 2017

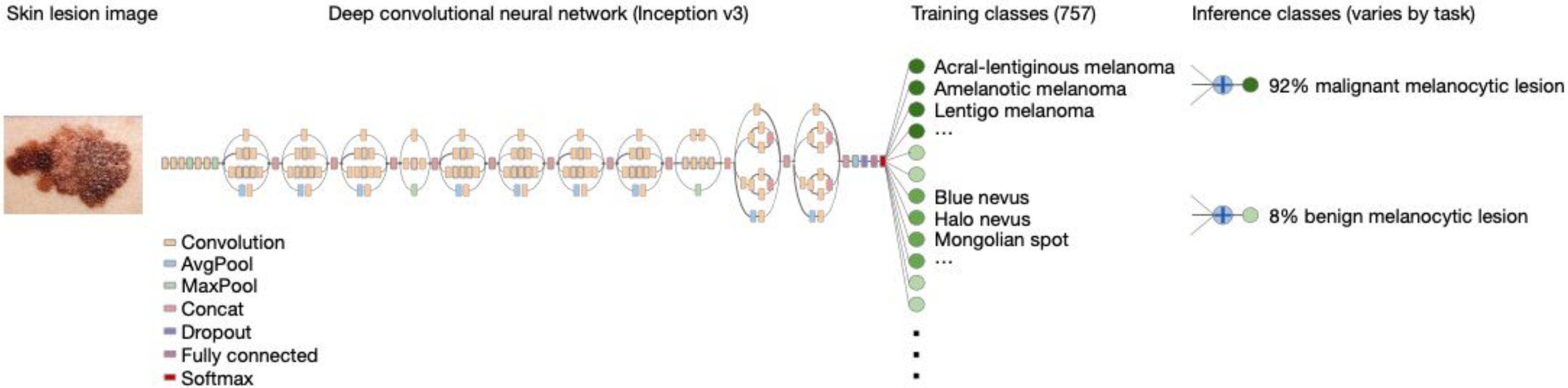
- Two binary classification tasks on **dermatology images**: malignant vs. benign lesions of epidermal or melanocytic origin
- Inception-v3 (GoogLeNet) CNN with ImageNet pre-training
- Fine-tuned on dataset of 129,450 lesions (from several sources) comprising 2,032 diseases
- Evaluated model vs. 21 or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

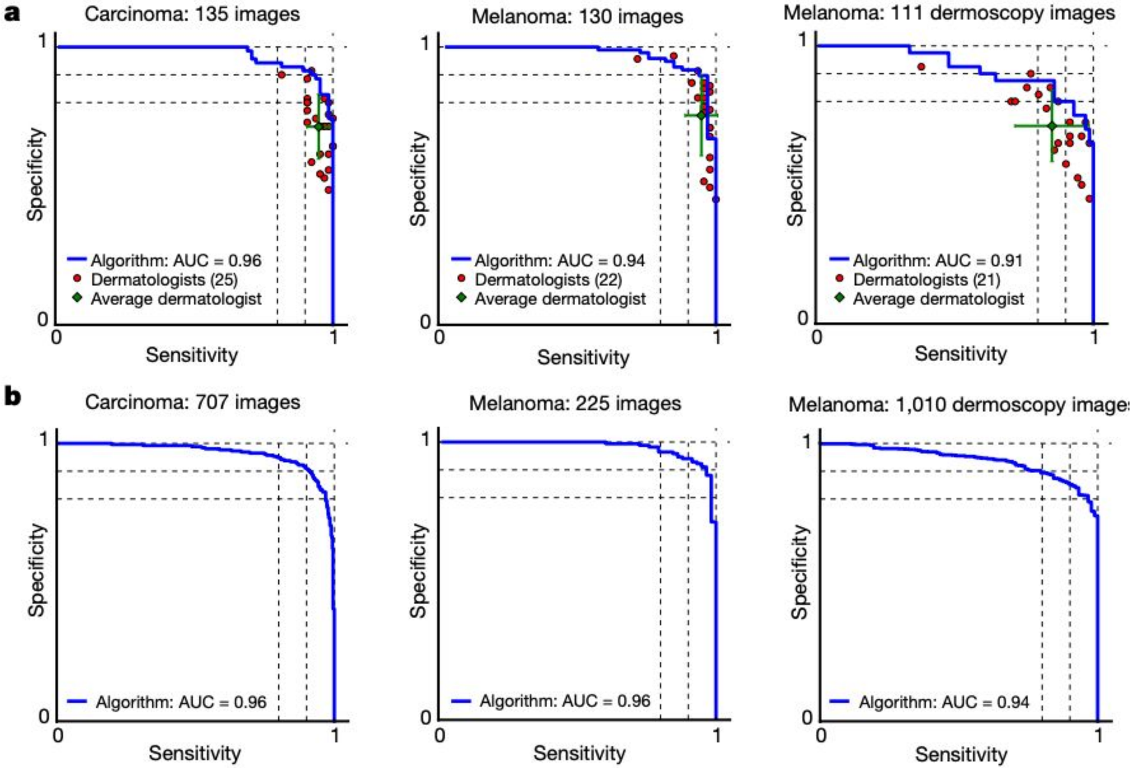
- Train on finer-grained classification (757 classes) but perform binary classification at inference time by summing probabilities of fine-grained sub-classes
- The stronger fine-grained supervision during the training stage improves inference performance!



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

Esteva et al. 2017

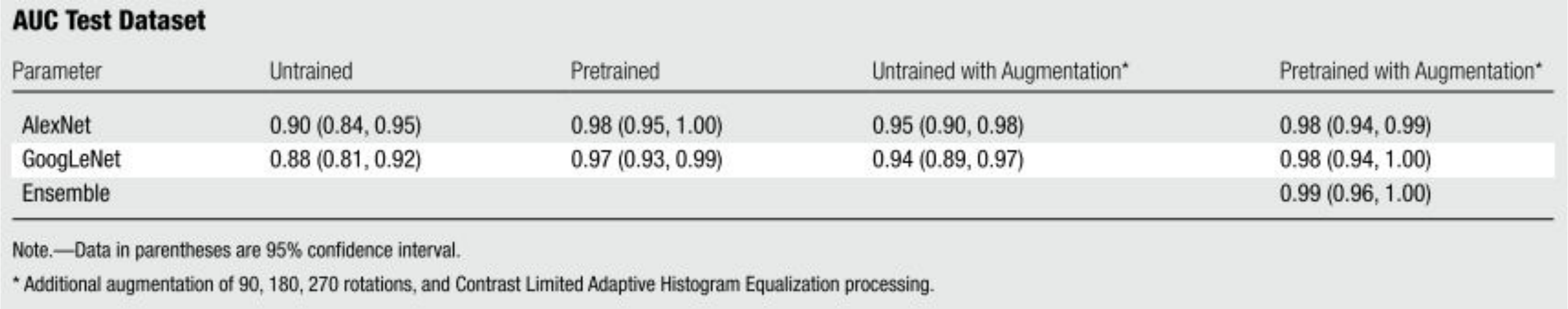
- Evaluation of algorithm vs. dermatologists



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)

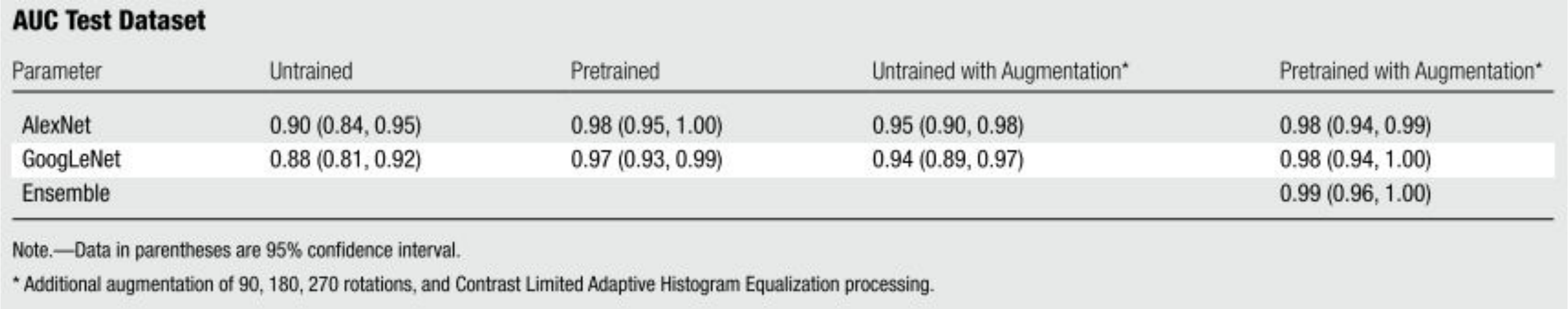
Note.—Data in parentheses are 95% confidence interval.

* 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.

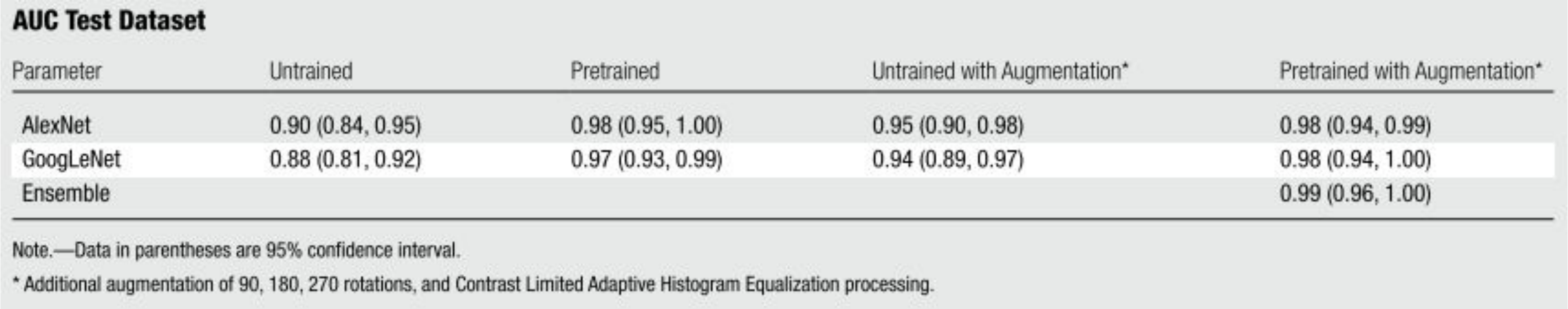
* Additional augmentation of 90, 180, 270 rotations, and Contrast Limited Adaptive Histogram Equalization processing.

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.

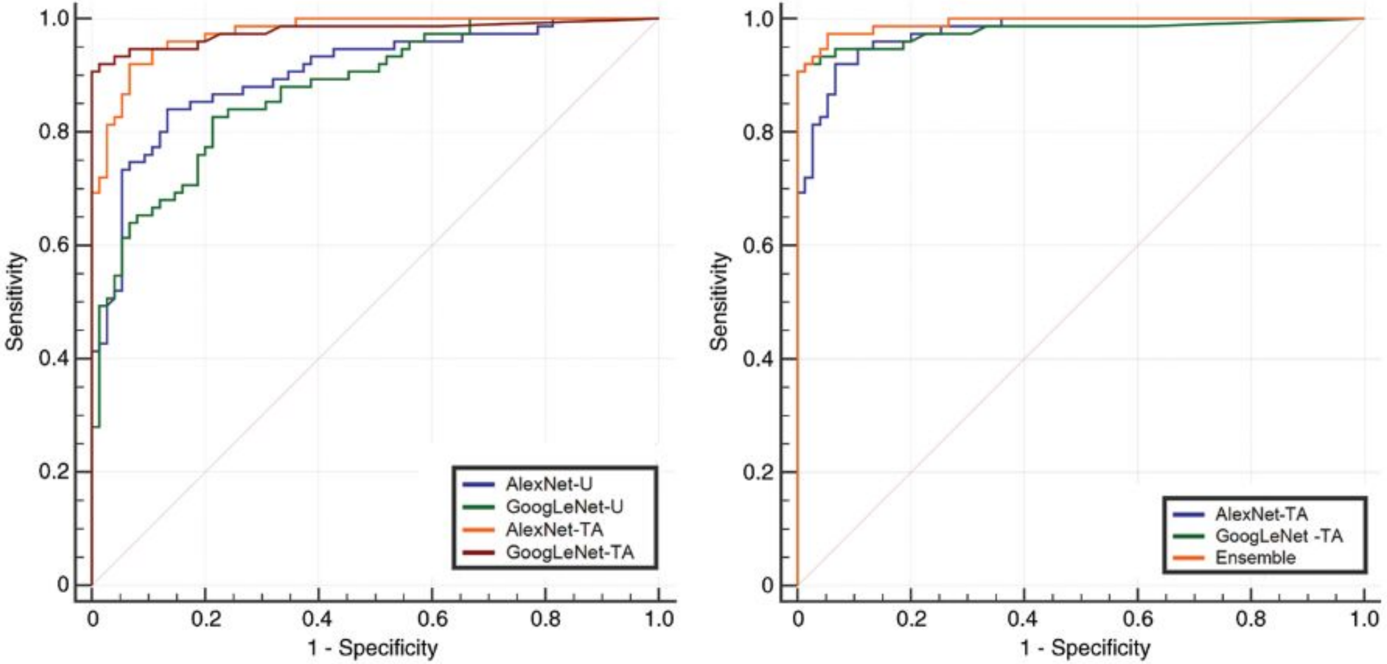
* Additional augmentation of 90, 180, 270 rotations, and Contrast Limited Adaptive Histogram Equalization processing.

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.

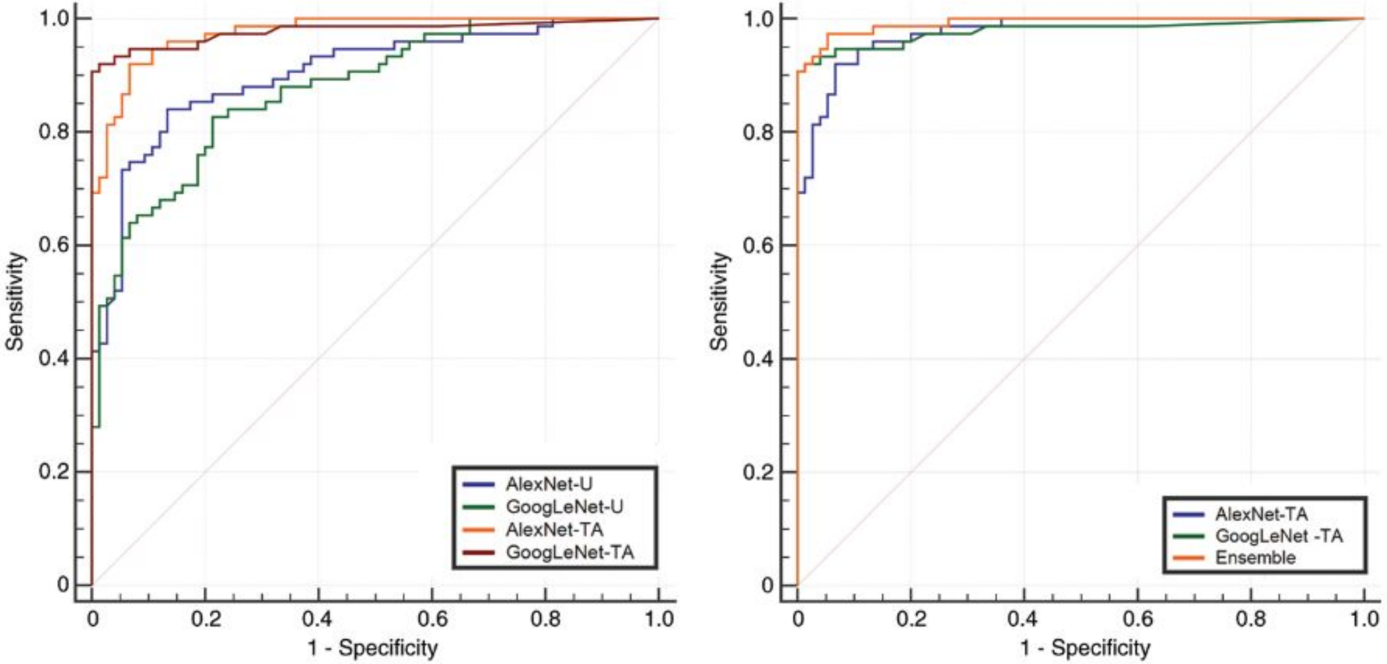
Lakhani and Sundaram 2017



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

Lakhani and Sundaram 2017

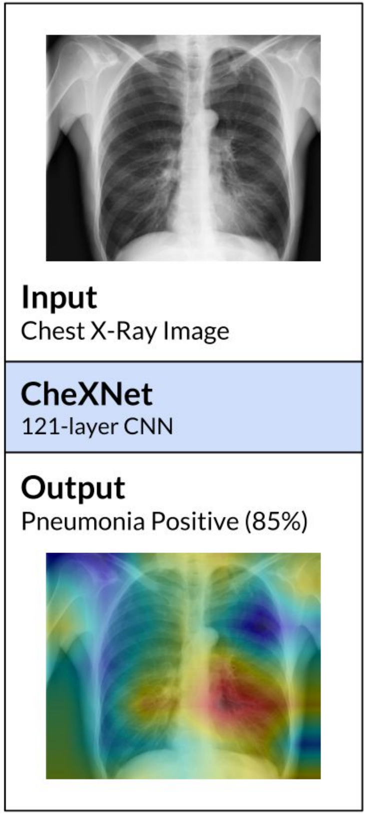
Performed further analysis at optimal threshold determined by the Youden Index.



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



Input
Chest X-Ray Image

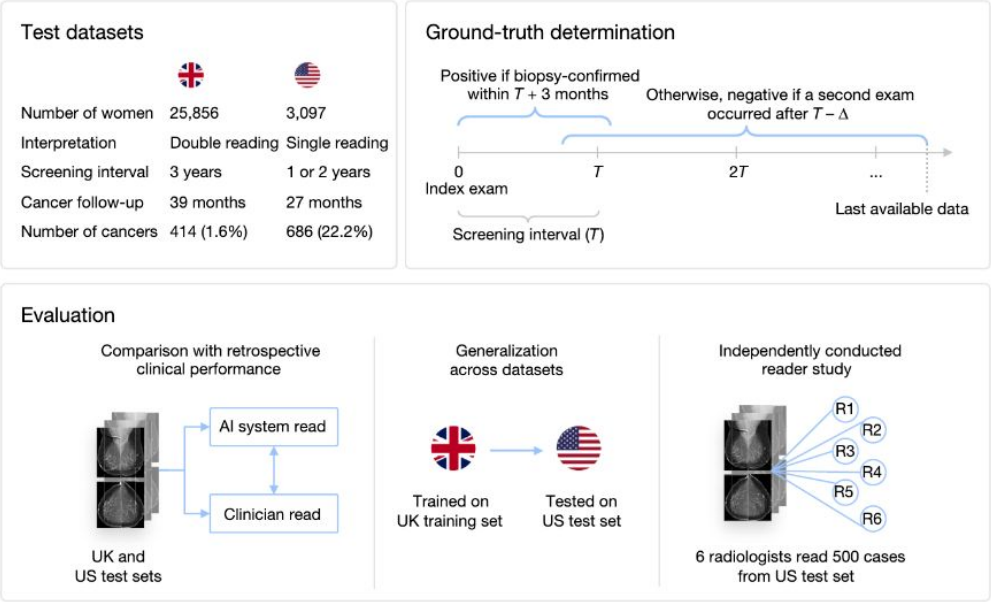
CheXNet
121-layer CNN

Output
Pneumonia Positive (85%)

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**
- International dataset and evaluation, across UK and US



Test datasets

	🇬🇧	🇺🇸
Number of women	25,856	3,097
Interpretation	Double reading	Single reading
Screening interval	3 years	1 or 2 years
Cancer follow-up	39 months	27 months
Number of cancers	414 (1.6%)	686 (22.2%)

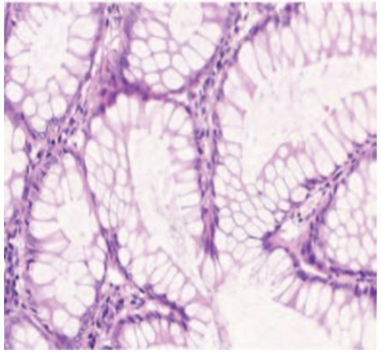
Ground-truth determination
Positive if biopsy-confirmed within T + 3 months
Otherwise, negative if a second exam occurred after T − Δ
0
Index exam
T
2T
...
Last available data
Screening interval (T)
Evaluation
Comparison with retrospective clinical performance
AI system read
Clinician read
UK and US test sets
Generalization across datasets
Trained on UK training set
Tested on US test set
Independently conducted reader study
R1
R2
R3
R4
R5
R6
6 radiologists read 500 cases from US test set

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

Advanced Vision Models: Segmentation and Detection

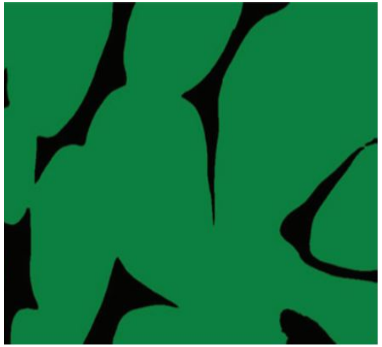
Richer visual recognition tasks: segmentation and detection

Classification



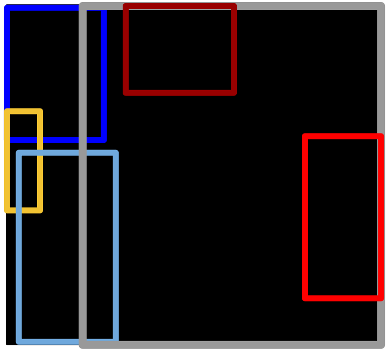
Output:
one category label for image (e.g., colorectal glands)

Semantic Segmentation



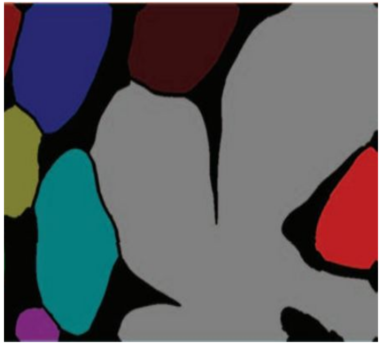
Output:
category label for each pixel in the image

Detection



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

Instance Segmentation

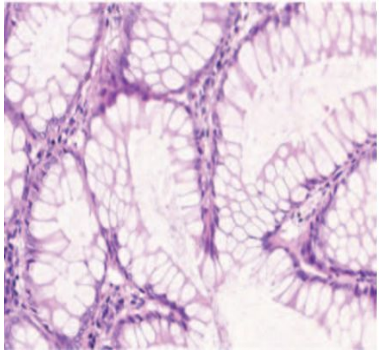


Output:
Category label and instance label for each pixel in the image

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

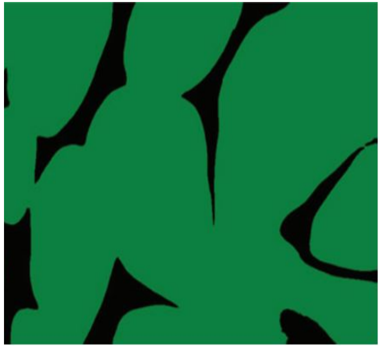
Richer visual recognition tasks: segmentation and detection

Classification



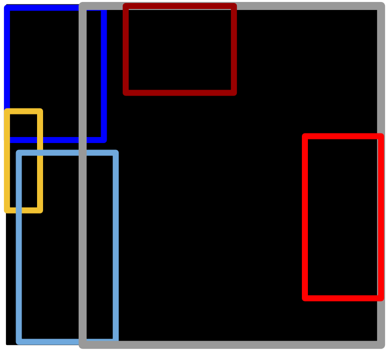
Output:
one category label for image (e.g., colorectal glands)

Semantic Segmentation



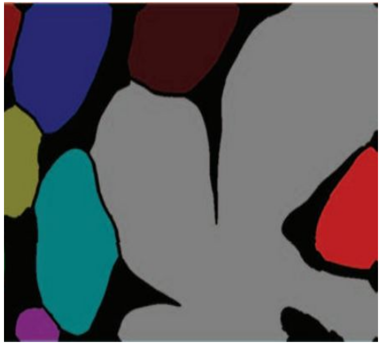
Output:
category label for each pixel in the image

Detection



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

Instance Segmentation

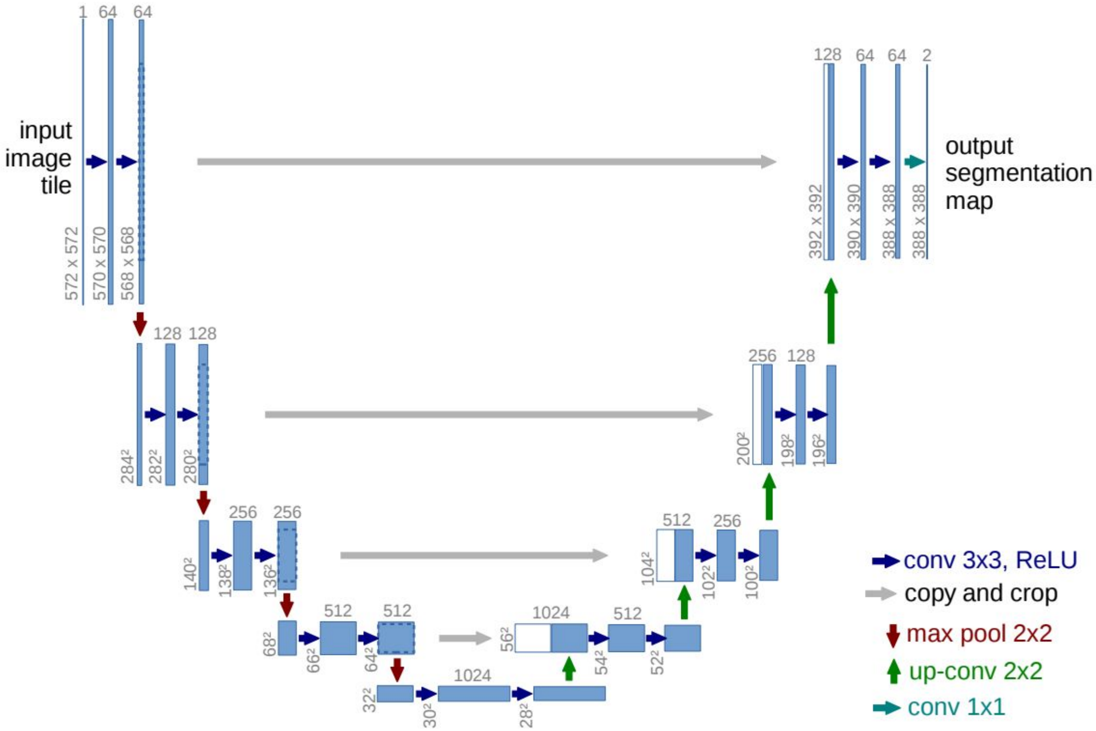


Output:
Category label and instance label for each pixel in the image

Distinguishes between different instances of an object

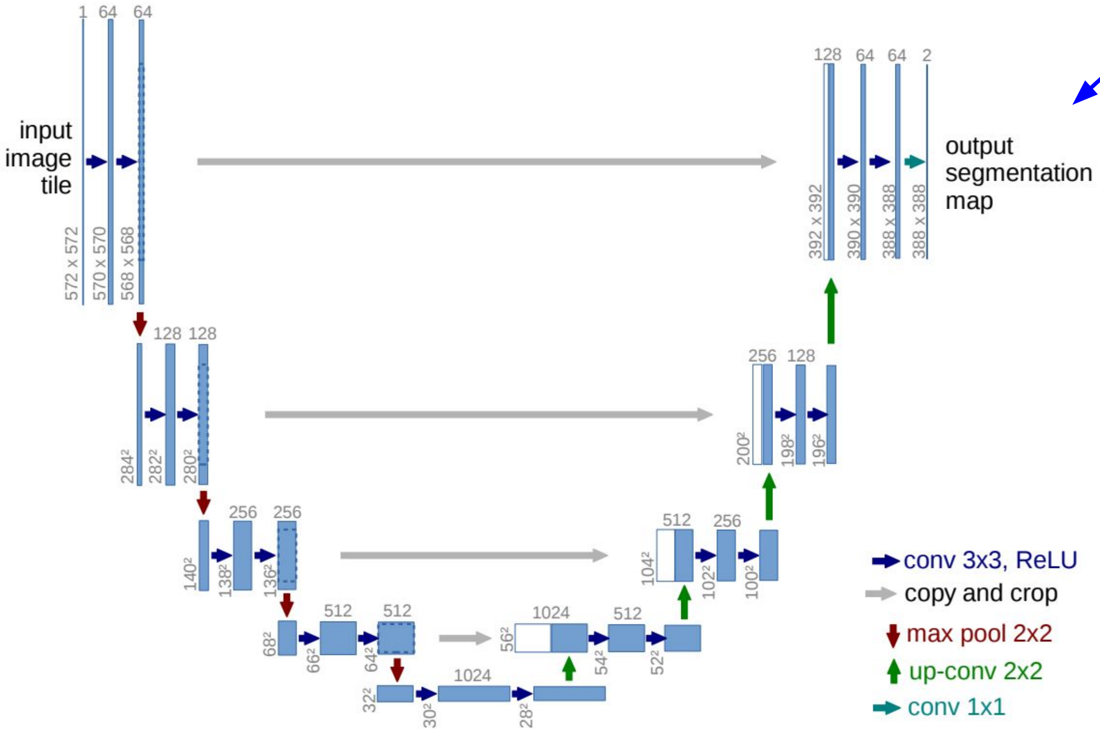
Figures: Chen et al. 2016. https://arxiv.org/pdf/1604.02677.pdf

Semantic segmentation: U-Net



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

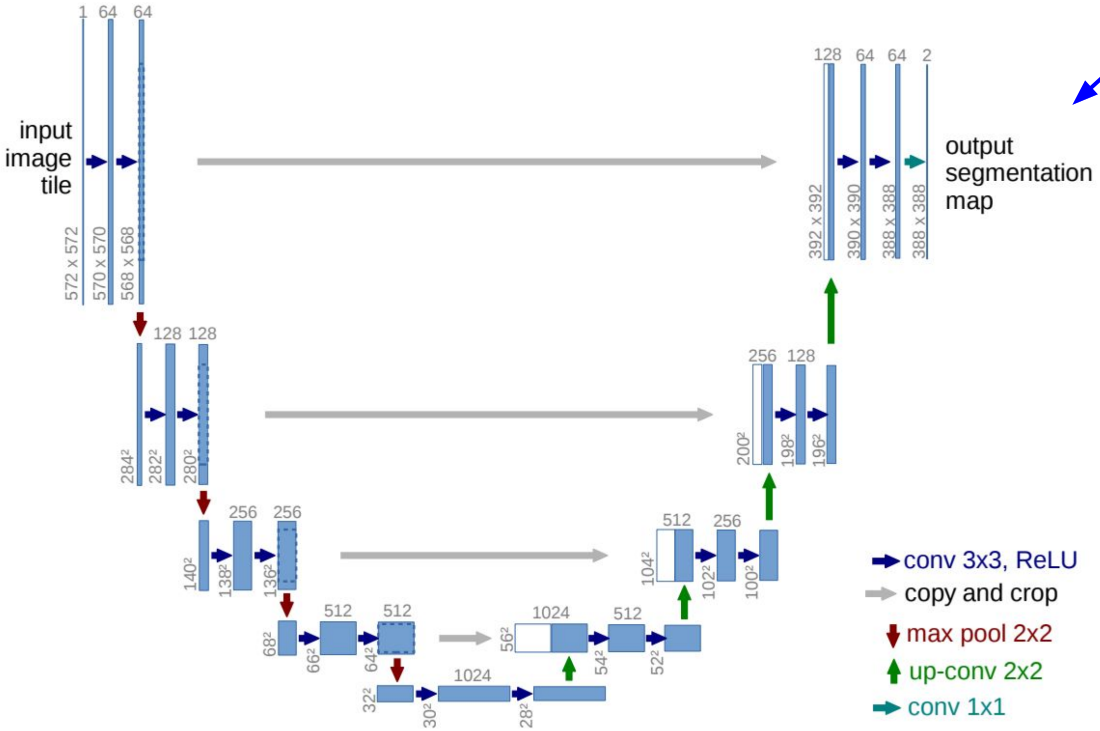
Semantic segmentation: U-Net



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

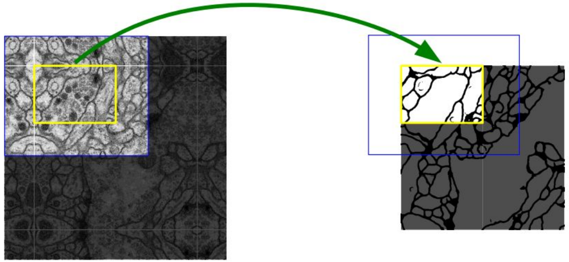
Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

Semantic segmentation: U-Net



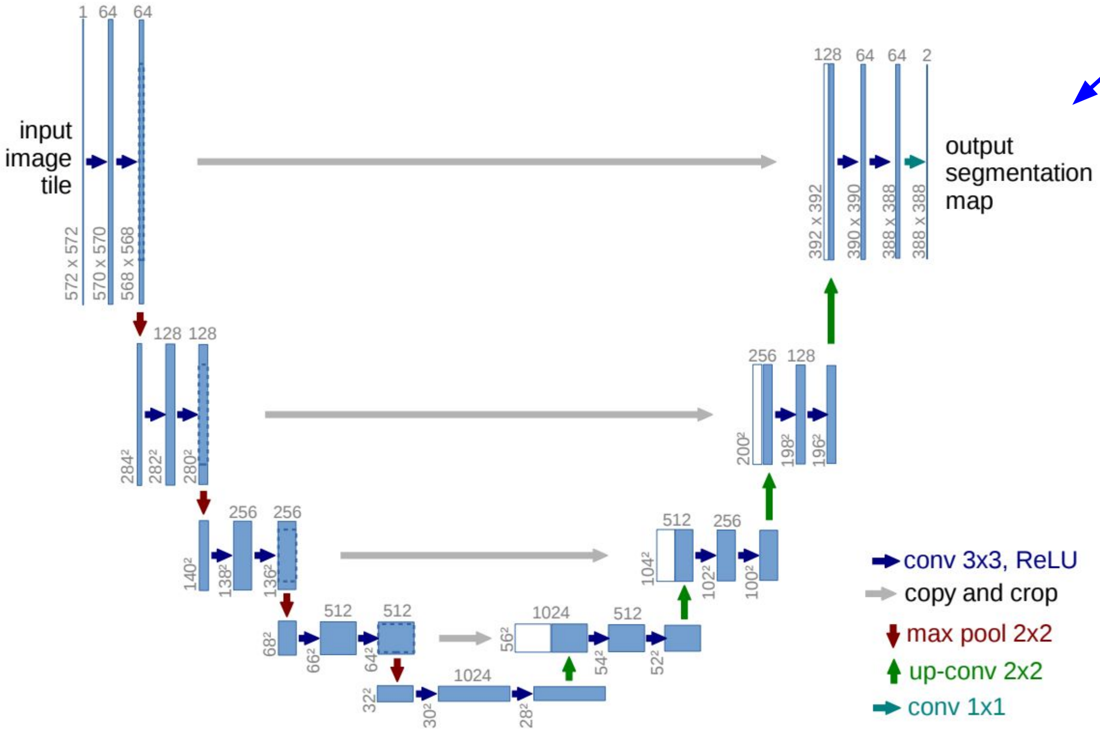
Output is an image mask: width x height x # classes

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



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

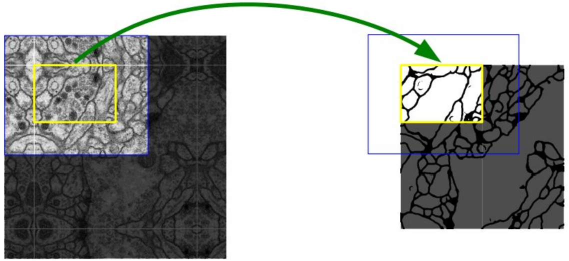
Semantic segmentation: U-Net



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

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

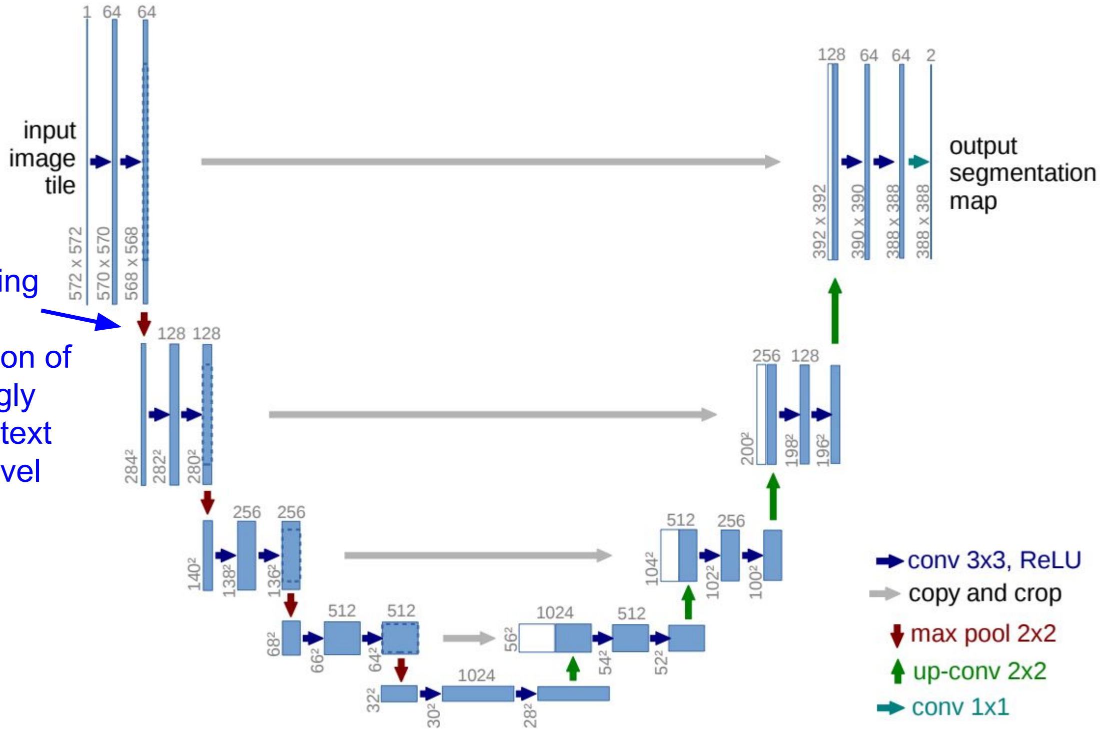
Output image size a little smaller than original, due to convolutional operations w/o padding



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

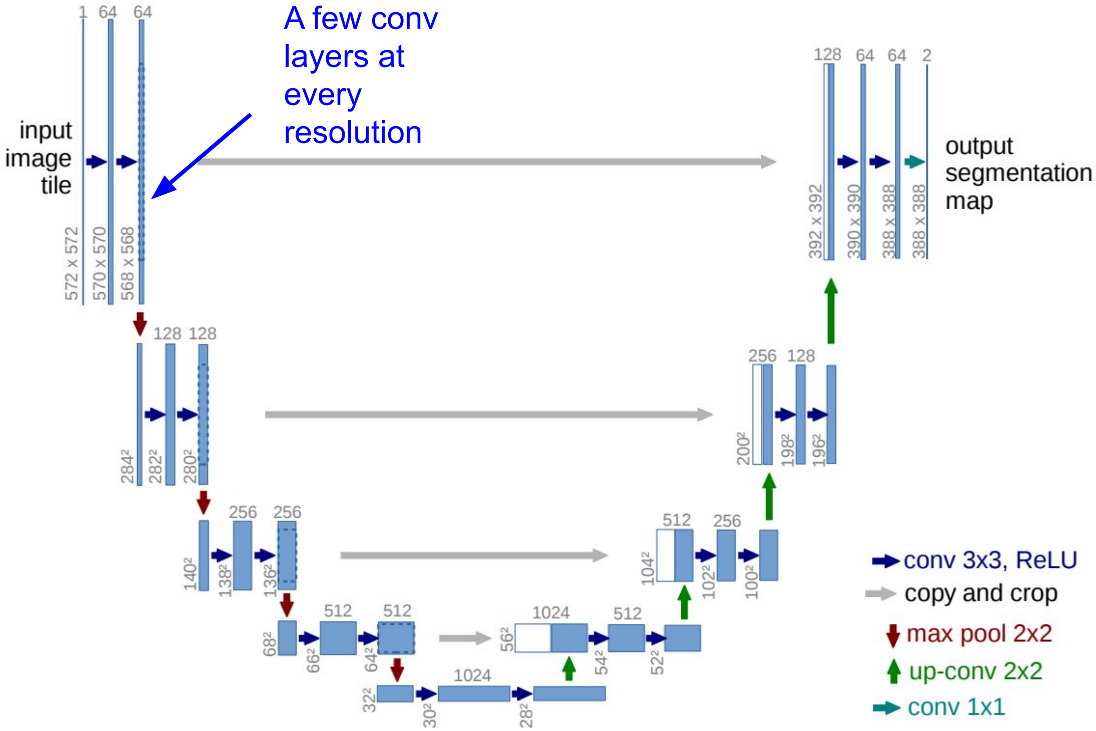
Semantic segmentation: U-Net

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



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

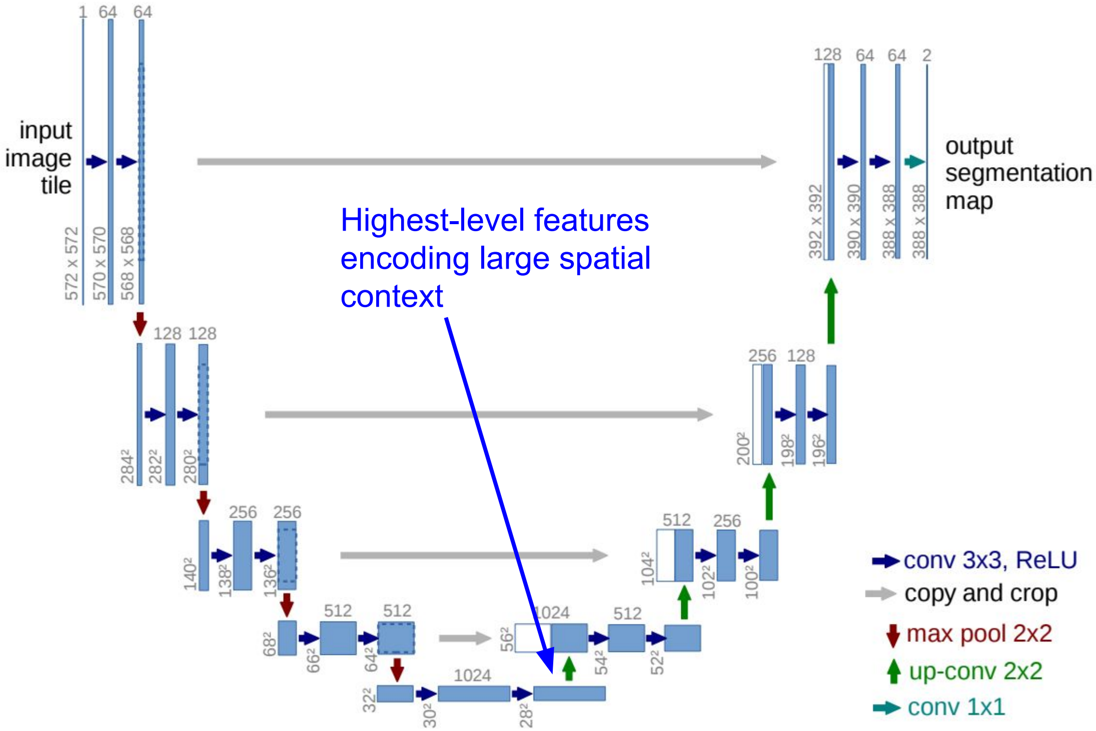
Semantic segmentation: U-Net



A few conv layers at every resolution

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

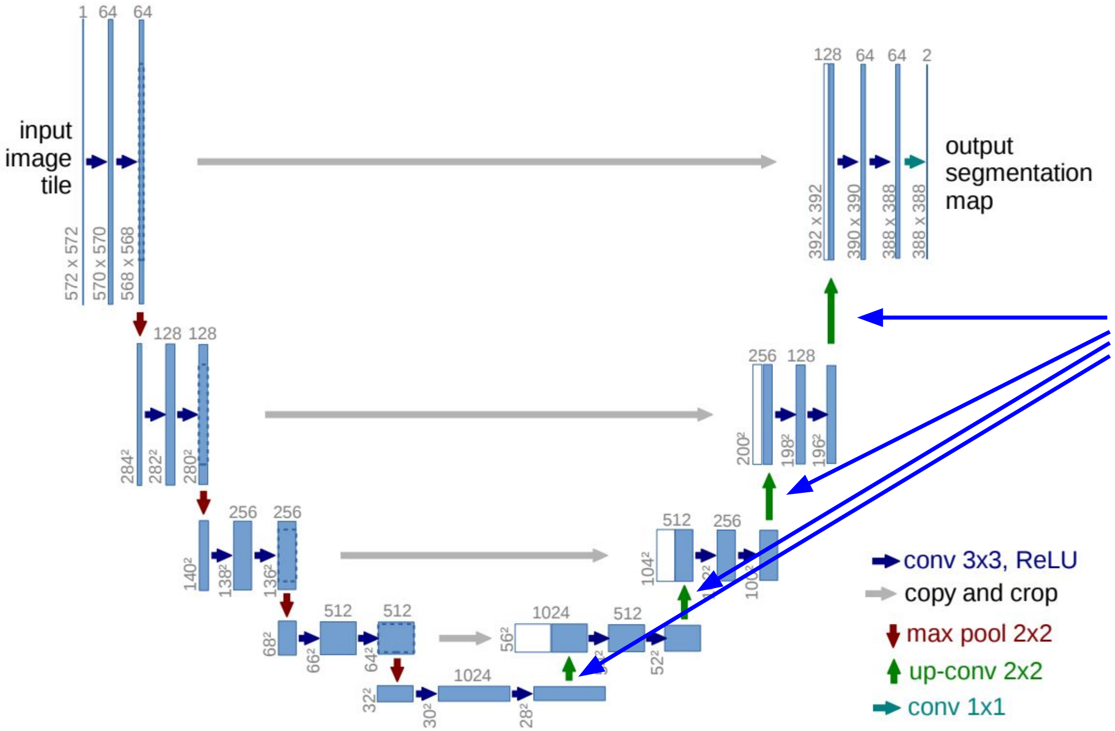
Semantic segmentation: U-Net



Highest-level features encoding large spatial context

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

Semantic segmentation: U-Net

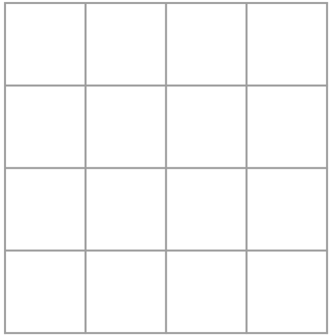


Up-convolutions to go from the global information encoded in highest-level features, back to individual pixel predictions

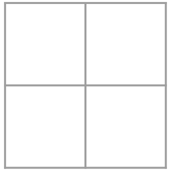
Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

Up-convolutions

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



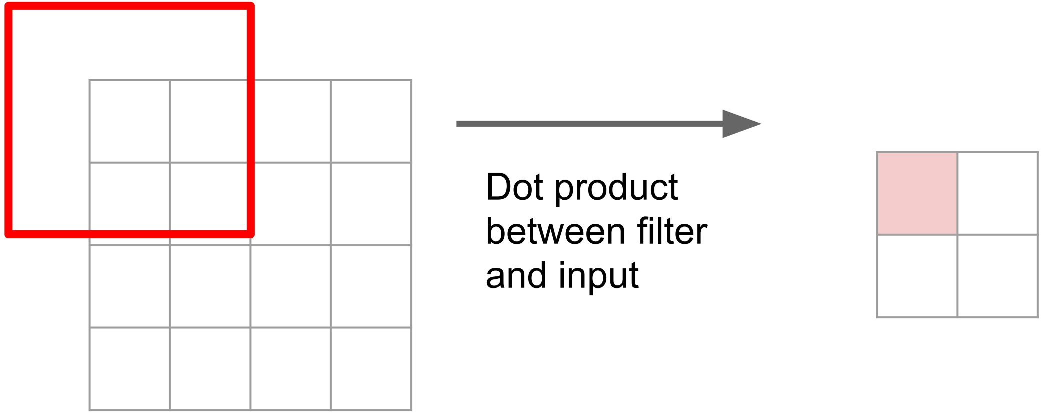
Input: 4 x 4



Output: 2 x 2

Up-convolutions

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

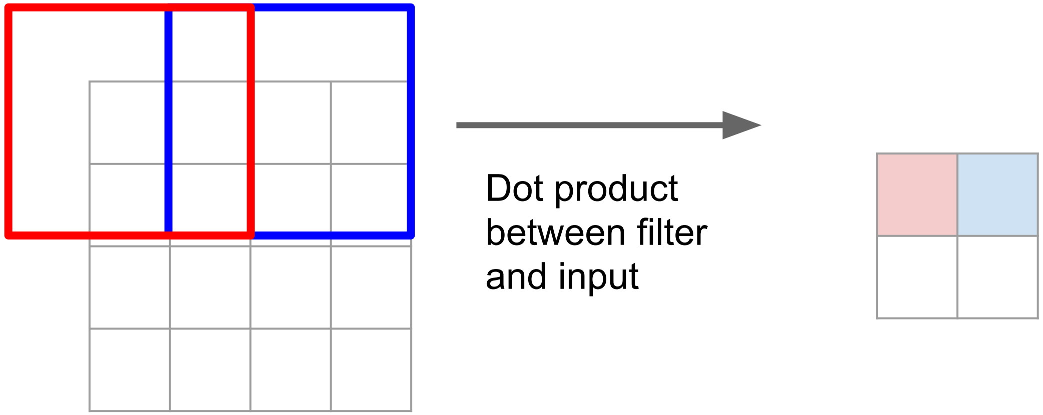


Input: 4 x 4

Output: 2 x 2

Up-convolutions

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



Dot product between filter and input

Input: 4 x 4

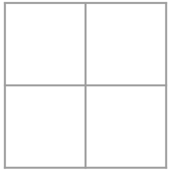
Output: 2 x 2

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

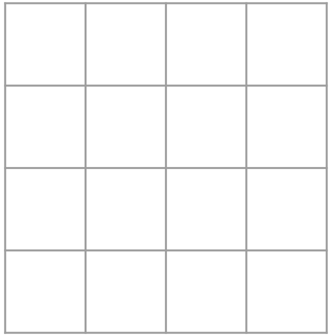
Stride gives ratio between movement in input and output

Up-convolutions

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



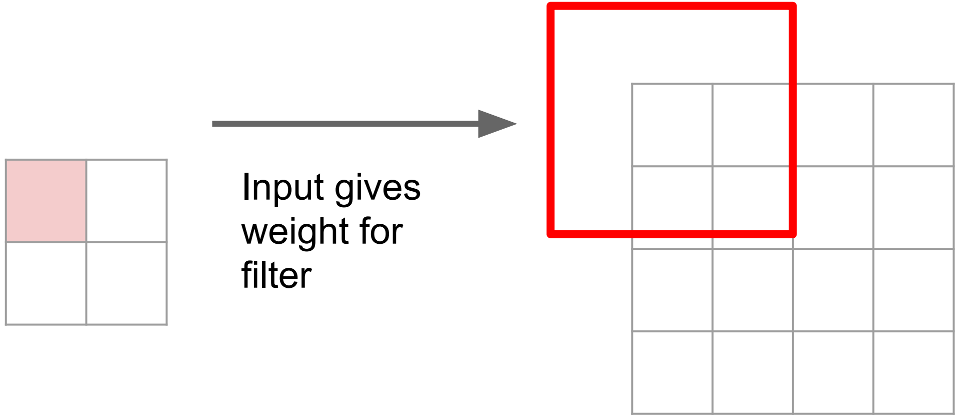
Input: 2 x 2



Output: 4 x 4

Up-convolutions

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



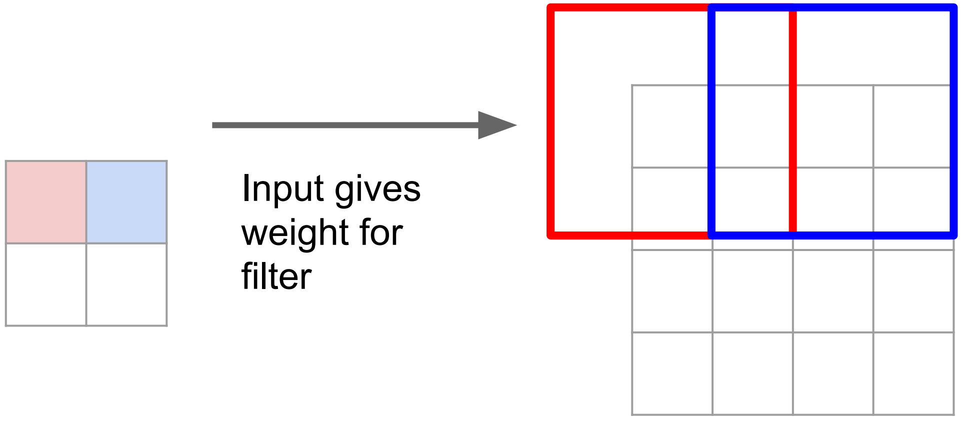
Input gives weight for filter

Input: 2 x 2

Output: 4 x 4

Up-convolutions

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



Input gives weight for filter

Input: 2 x 2

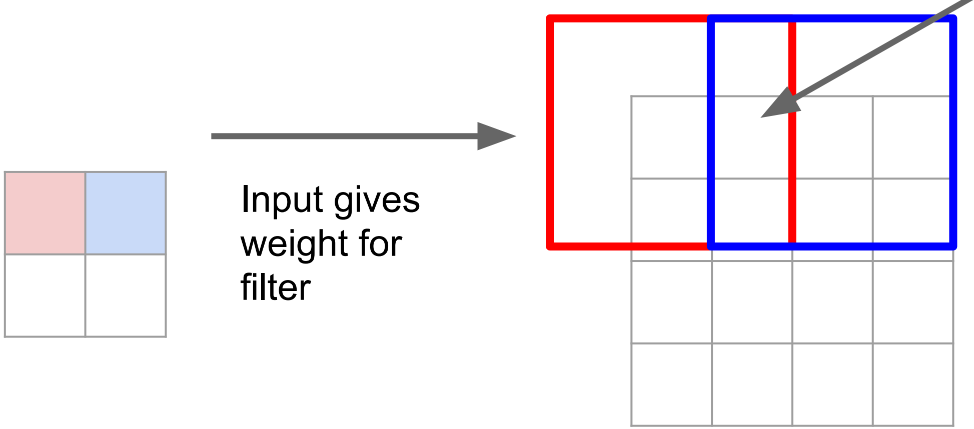
Output: 4 x 4

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 gives weight for filter

Sum where output overlaps

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

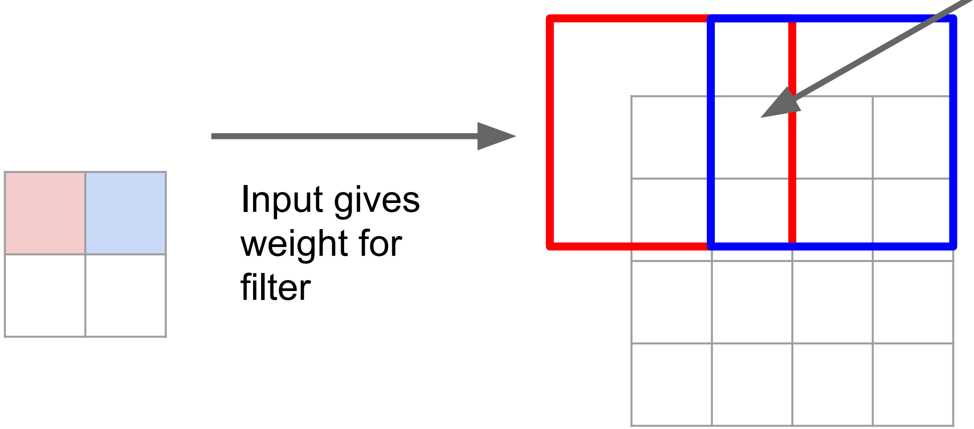
Input: 2 x 2

Output: 4 x 4

Up-convolutions

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

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



Input gives weight for filter

Sum where output overlaps

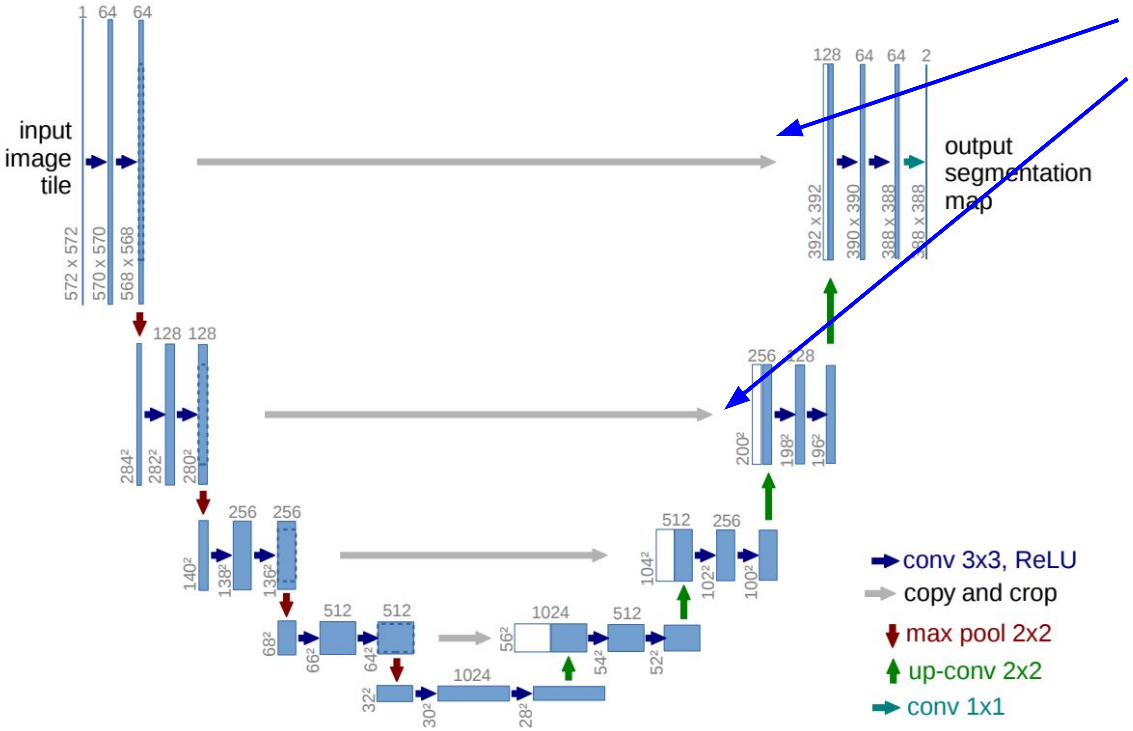
Input: 2 x 2

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

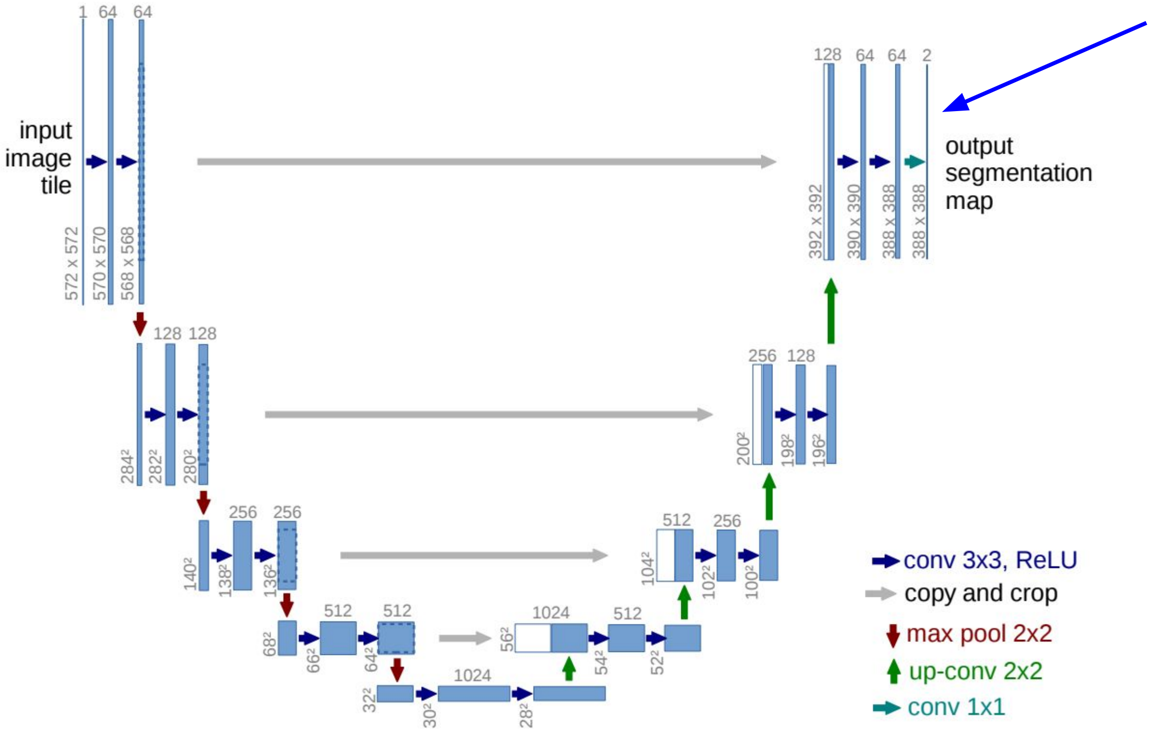
Semantic segmentation: U-Net



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:

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

# pixels included in both target and prediction maps

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

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

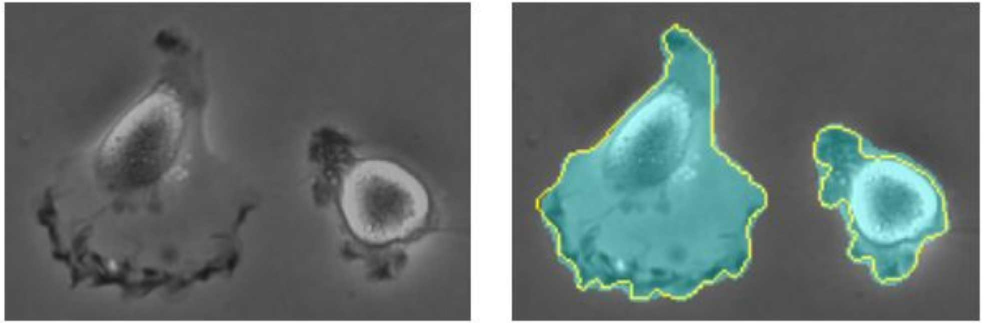
Sum of target mask size
+ prediction mask size

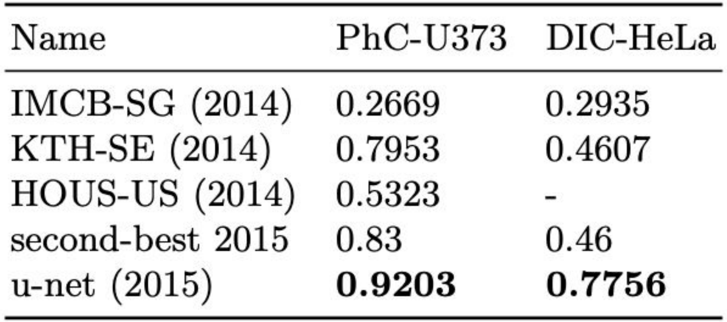
Very similar to IOU / Jaccard, can derive one from the other

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**

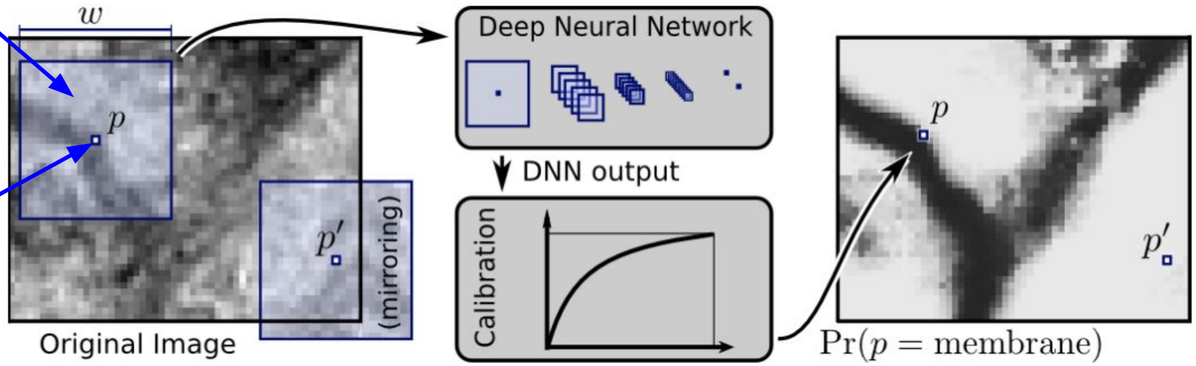
Very small dataset: 30 training images of size 512x512, in the ISBI 2012 Electron Microscopy (EM) segmentation challenge. Used excessive data augmentation to compensate.

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

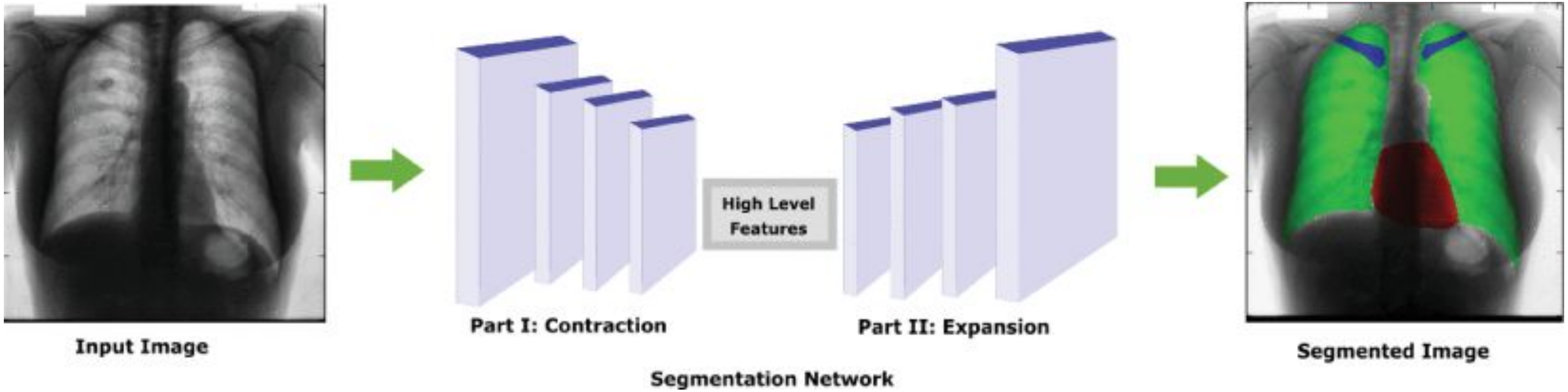


Note: a simple approach to segmentation can also be applying a classification CNN on image patches in a dense, sliding-window fashion (e.g. Ciresan et al.). But fully convolutional approaches such as U-Net generally achieve better performance.

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

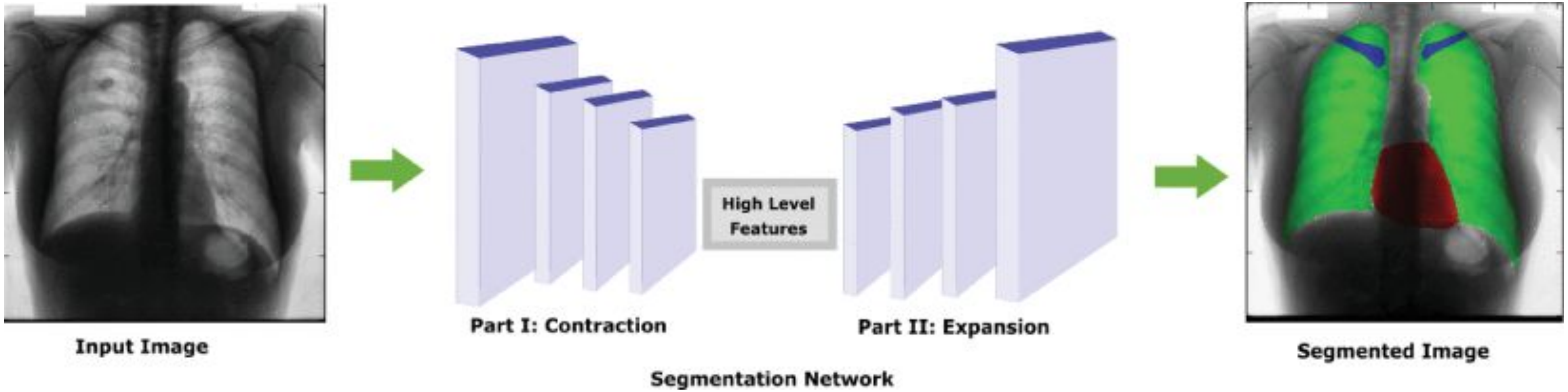


Novikov et al. Fully Convolutional Architectures for Multiclass Segmentation in Chest Radiographs. IEEE Trans. on Medical Imaging, 2018.

Novikov et al. 2018

Q: What loss function would be appropriate here?

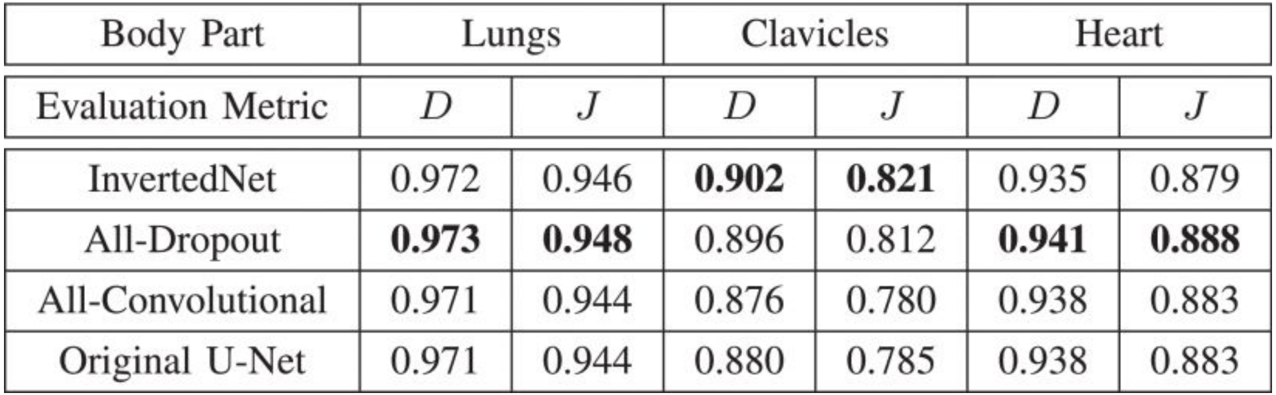
- Chest x-ray segmentation of lungs, clavicles, and heart
- JSRT dataset of 247 chest-xrays at 2048x2048 resolution. (But downsampled to 128x128 and 256x256!)
- Used a U-Net based segmentation network with a few modifications



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.

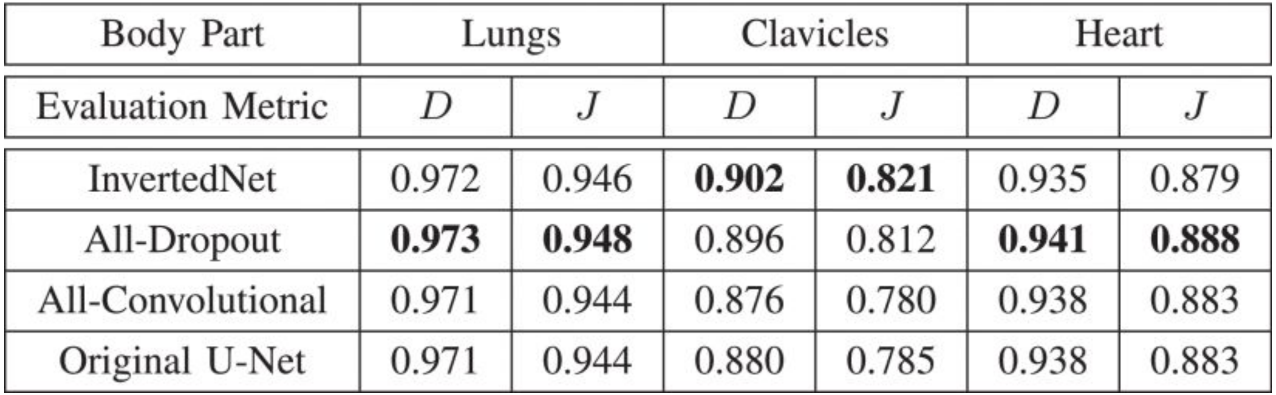
Novikov et al. 2018

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.

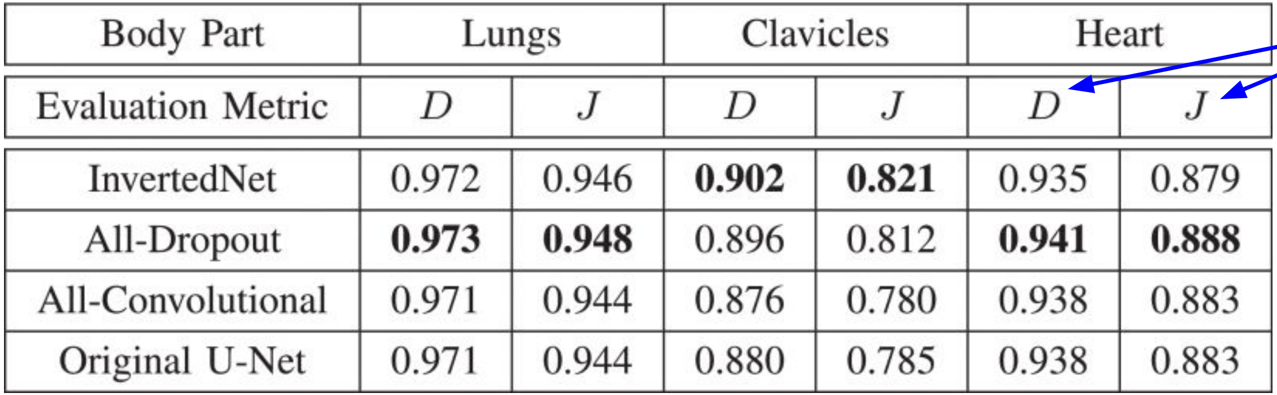
Novikov et al. 2018

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)



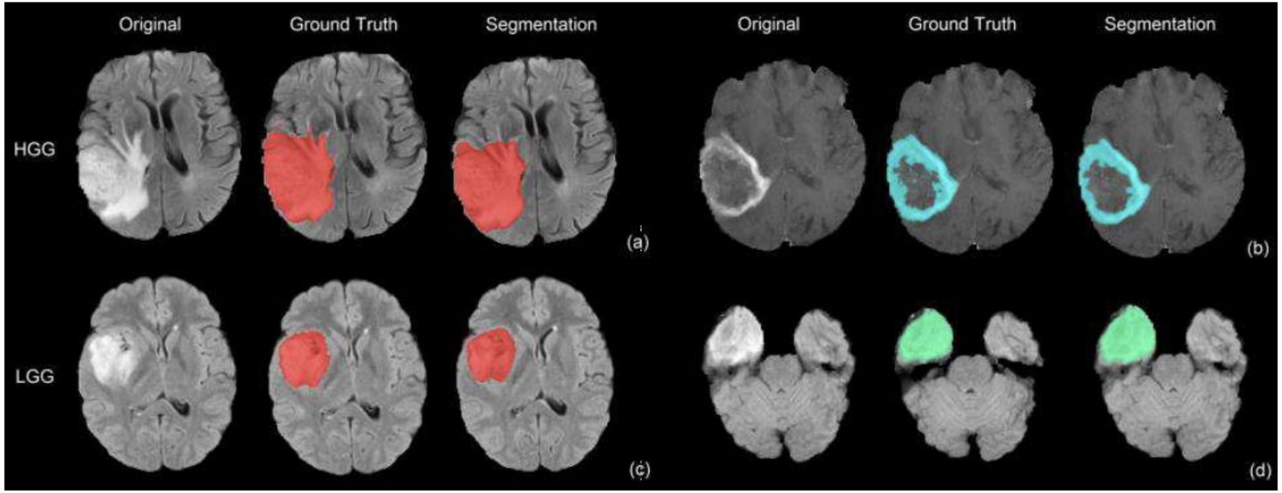
Body Part	Lungs		Clavicles		Heart	
Evaluation Metric	*D*	*J*	*D*	*J*	*D*	*J*
InvertedNet	0.972	0.946	**0.902**	**0.821**	0.935	0.879
All-Dropout	**0.973**	**0.948**	0.896	0.812	**0.941**	**0.888**
All-Convolutional	0.971	0.944	0.876	0.780	0.938	0.883
Original U-Net	0.971	0.944	0.880	0.785	0.938	0.883

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

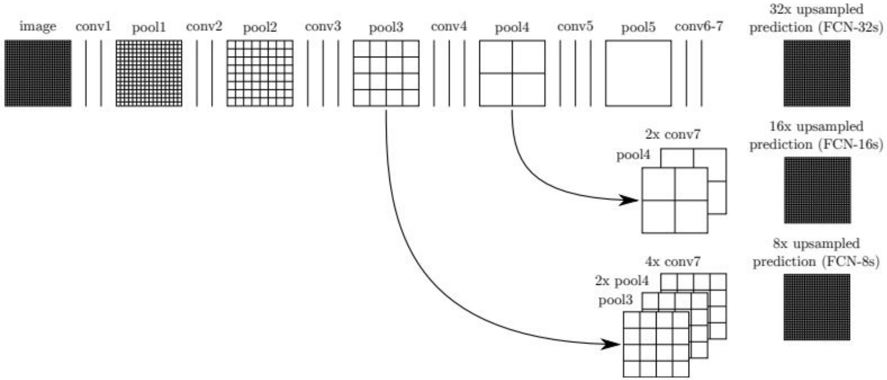
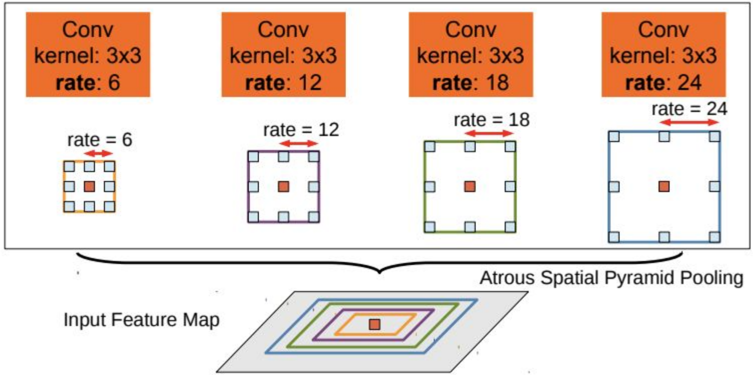


image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
32x upsampled prediction (FCN-32s)
2x conv7
pool4
16x upsampled prediction (FCN-16s)
4x conv7
2x pool4
pool3
8x upsampled prediction (FCN-8s)

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


Conv kernel: 3x3 rate: 6
Conv kernel: 3x3 rate: 12
Conv kernel: 3x3 rate: 18
Conv kernel: 3x3 rate: 24
rate = 6
rate = 12
rate = 18
rate = 24
Atrous Spatial Pyramid Pooling
Input Feature Map

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

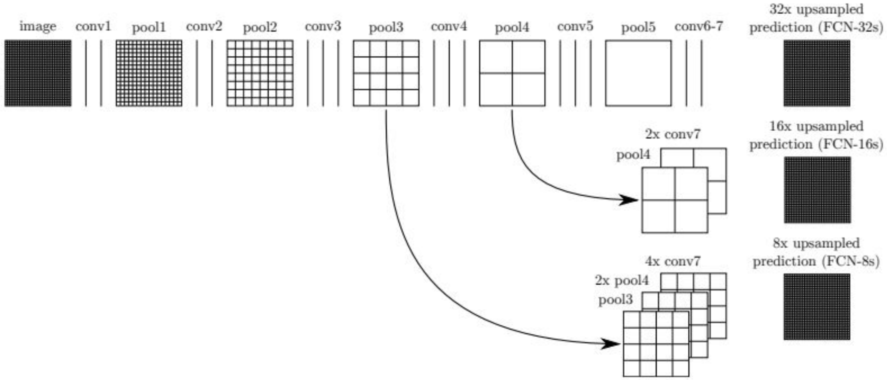
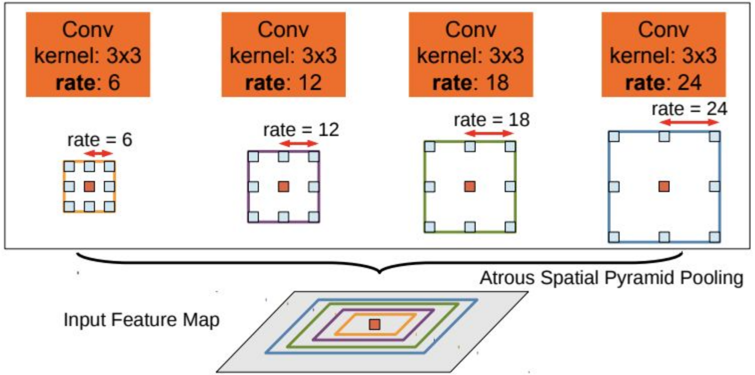


image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
32x upsampled prediction (FCN-32s)
2x conv7
pool4
16x upsampled prediction (FCN-16s)
4x conv7
2x pool4
pool3
8x upsampled prediction (FCN-8s)

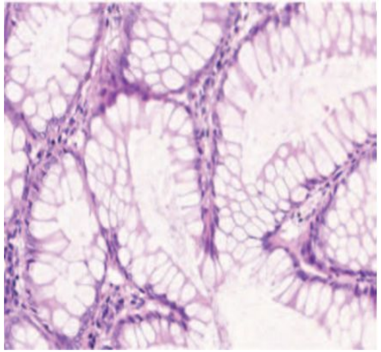

Conv kernel: 3x3 rate: 6
Conv kernel: 3x3 rate: 12
Conv kernel: 3x3 rate: 18
Conv kernel: 3x3 rate: 24
rate = 6
rate = 12
rate = 18
rate = 24
Atrous Spatial Pyramid Pooling
Input Feature Map

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.

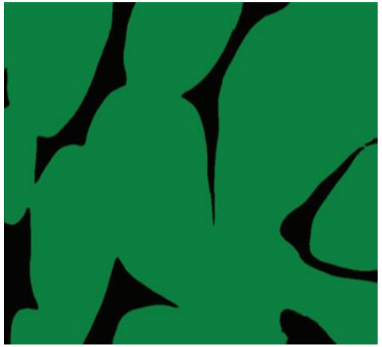
Richer visual recognition tasks: segmentation and detection

Classification



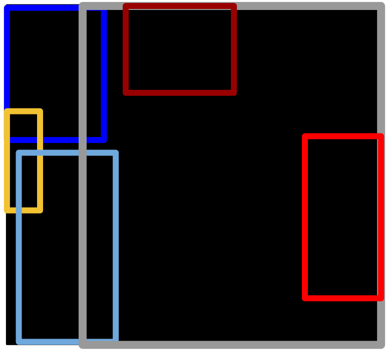
Output:
one category label for image (e.g., colorectal glands)

Semantic Segmentation



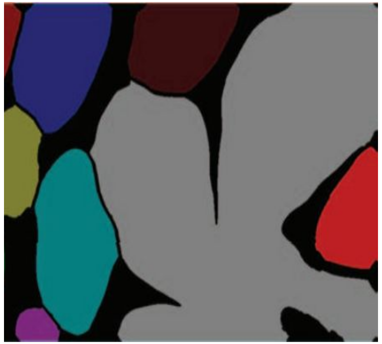
Output:
category label for each pixel in the image

Detection



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

Instance Segmentation

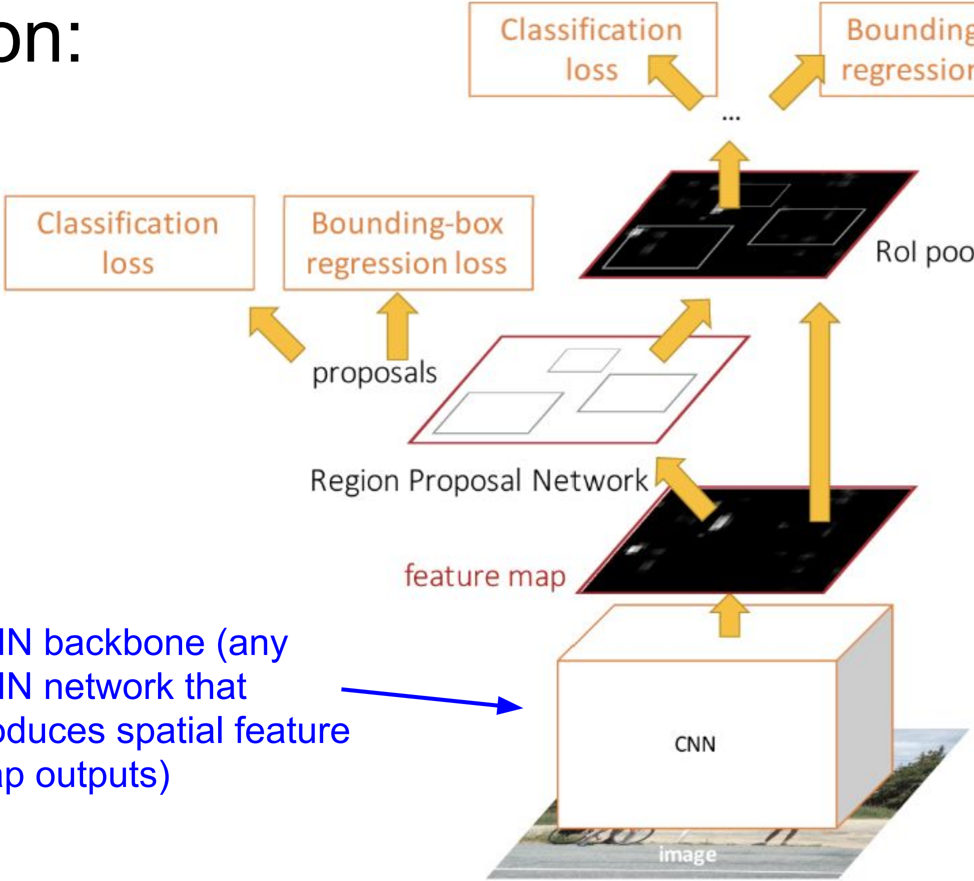


Output:
Category label and instance label for each pixel in the image

Distinguishes between different instances of an object

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

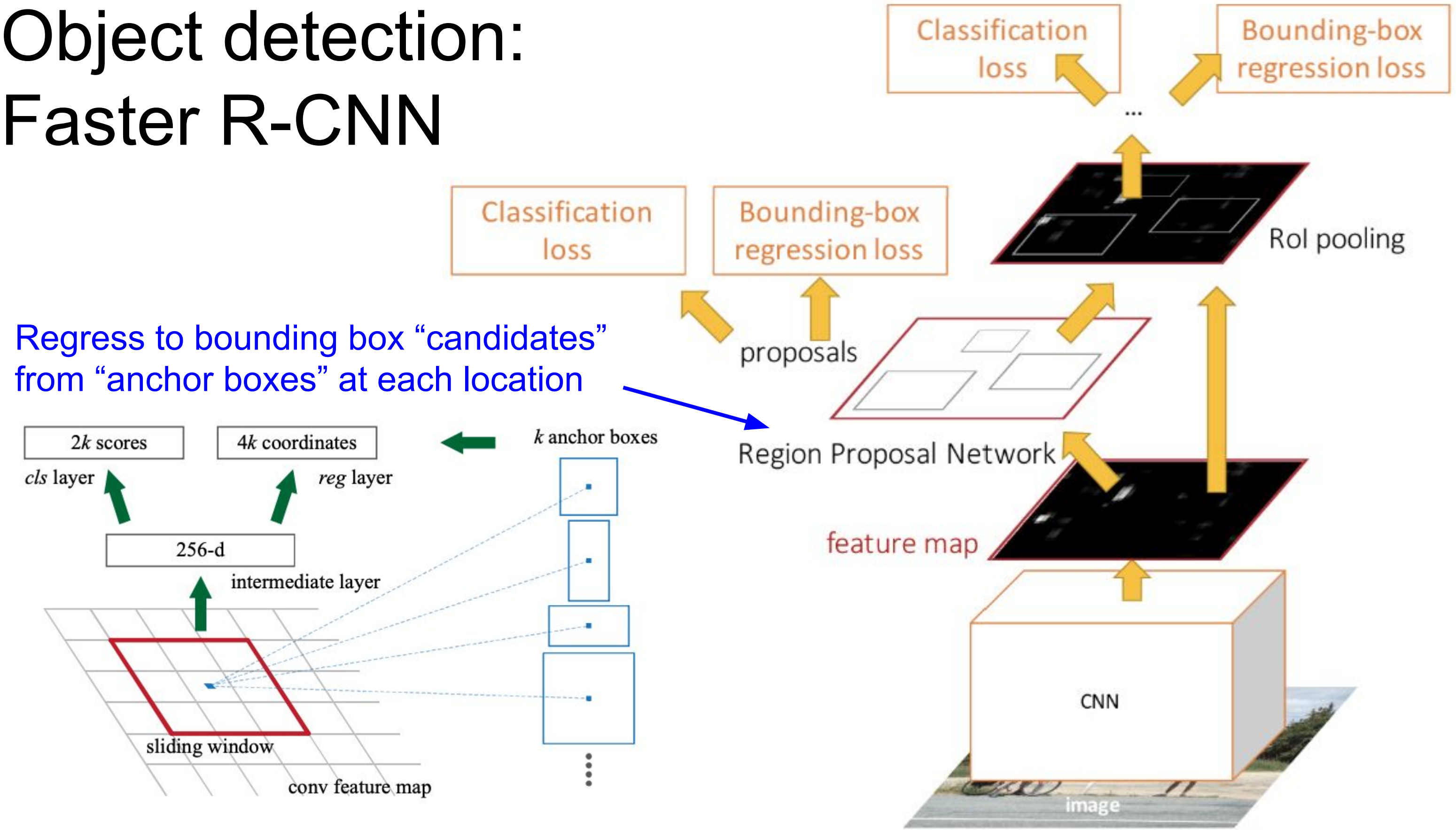
Object detection: Faster R-CNN



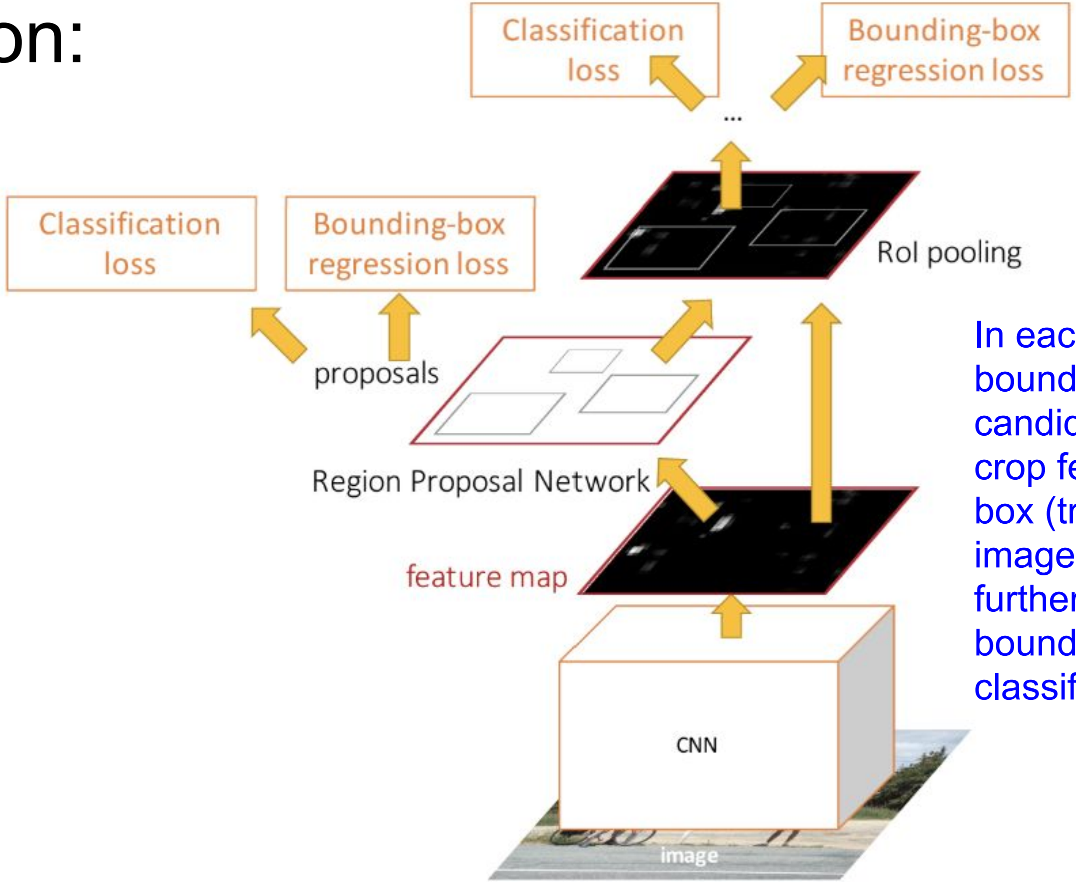
CNN backbone (any CNN network that produces spatial feature map outputs)

Object detection: Faster R-CNN

Regress to bounding box "candidates" from "anchor boxes" at each location



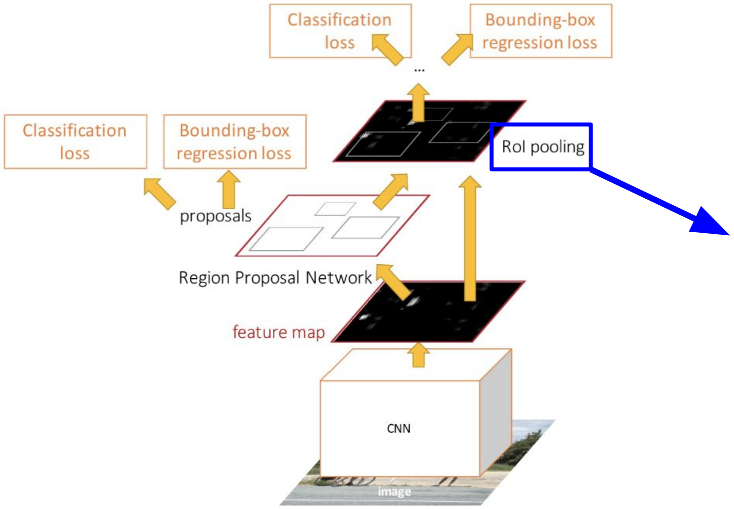
Object detection: Faster R-CNN



In each of top bounding box candidate locations, crop features within box (treat as own image) and perform further refinement of bounding box + classification

Cropping Features: RoI Pool

Divide into grid of (roughly) equal subregions, corresponding to fixed-size input required for final classification / bounding box regression networks



“Snap” to grid cells

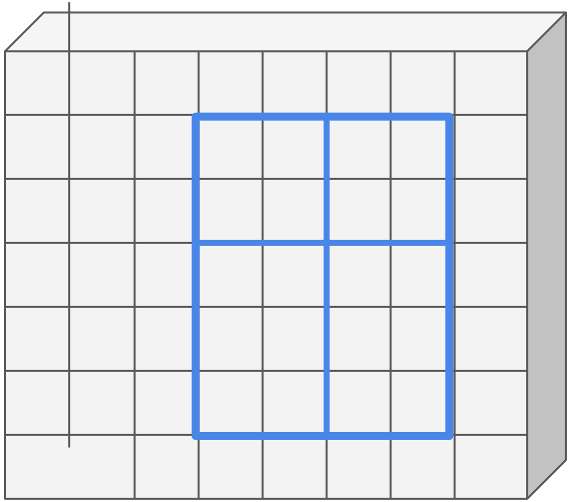
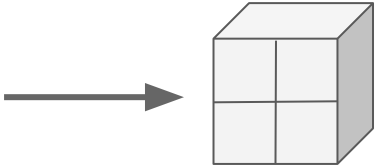


Image features

Max-pool within each subregion



Girshick, “Fast R-CNN”, ICCV 2015.

Evaluation of object detection

Standard output of object detection

For each class, a set of bounding box predictions with associated confidences:

- E.g., (x, y, h, w, c)

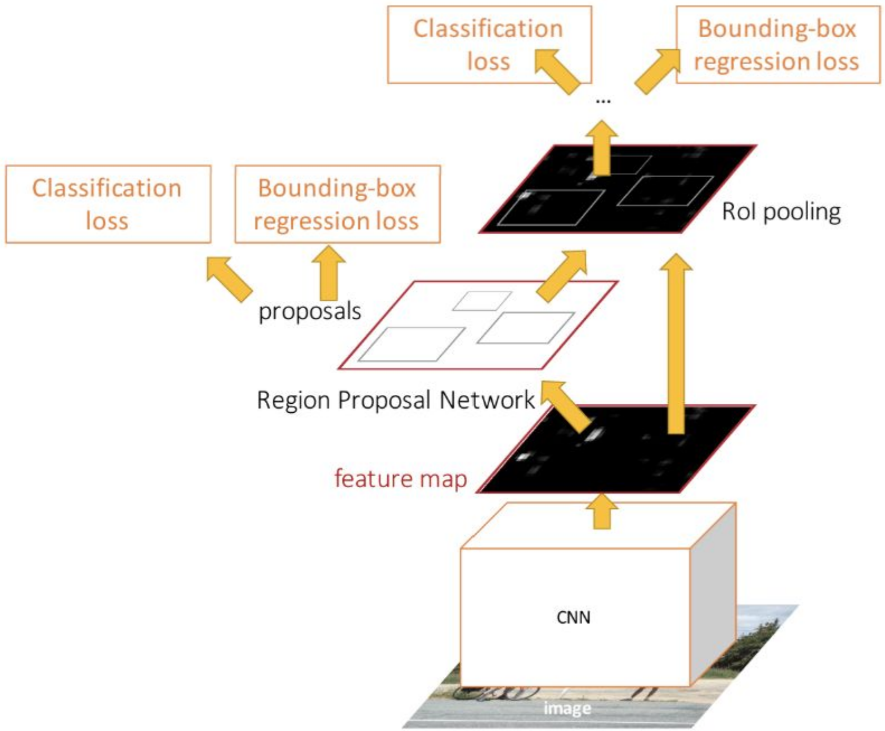
Evaluation of object detection

Standard output of object detection

For each class, a set of bounding box predictions with associated confidences:

- E.g., $(\underbrace{x, y, h, w}_{\text{Bounding box}}, c)$ ← Class confidence

Obtain from model outputs



Remember: ROC and precision recall curves

- **Receiver Operating Characteristic (ROC) curve**:
  - Plots sensitivity and specificity (specifically, 1 - specificity) as prediction threshold is varied
  - Gives trade-off between sensitivity and specificity
  - Also report summary statistic AUC (area under the curve)

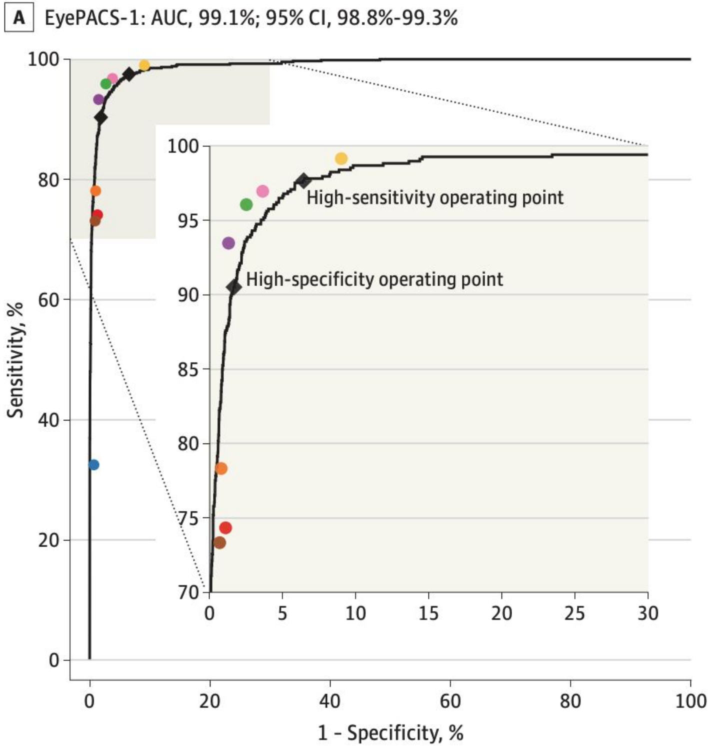
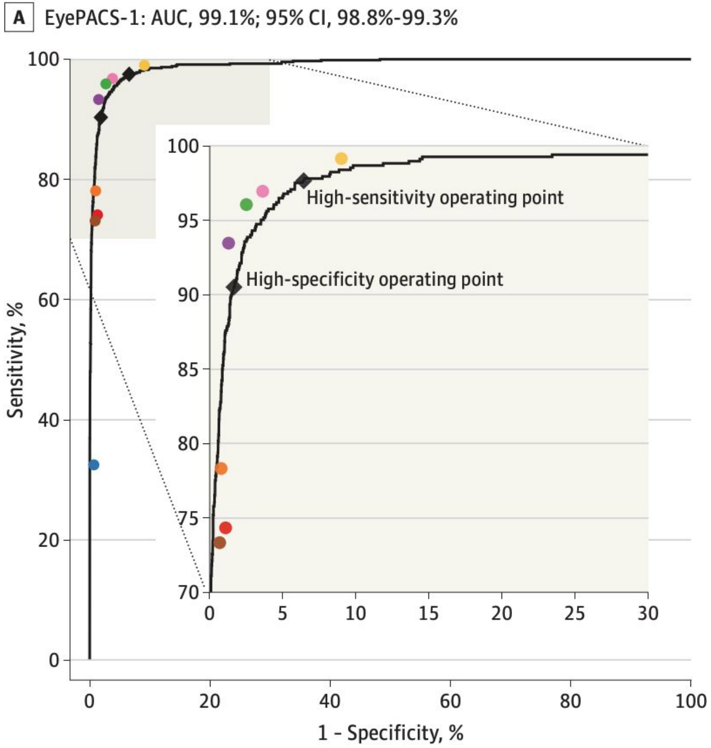


Figure credit: Gulshan et al. 2016

Remember: ROC and precision recall curves

- **Receiver Operating Characteristic (ROC) curve**:
    - Plots sensitivity and specificity (specifically, 1 - specificity) as prediction threshold is varied
    - Gives trade-off between sensitivity and specificity
    - Also report summary statistic AUC (area under the curve)



Plot curve is based on TP, TN, FP, FN when varying the prediction threshold -- i.e., class confidence threshold

Figure credit: Gulshan et al. 2016

Remember: ROC and precision recall curves

Confusion matrix

		Prediction	
		0	1
Ground Truth	0	TN	FP
	1	FN	TP

Accuracy: (TP + TN) / total

Sensitivity / Recall (true positive rate)**:**
TP / total positives

Specificity (true negative rate)**:**
TN / total negatives

Precision (positive predictive value)**:**
TP / total predicted positives

Negative predictive value:
TN / total predicted negatives

Remember: ROC and precision recall curves

- Sometimes also see **precision recall curve**
    - More informative when dataset is heavily imbalanced (specificity = true negative rate less meaningful in this case)

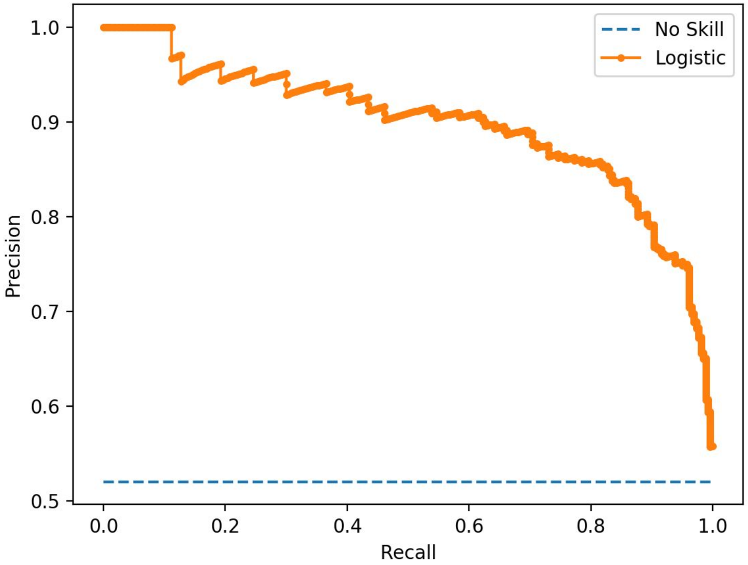


Figure credit: https://3qeqpr26caki16dnhd19sv6by6v-wpengine.netdna-ssl.com/wp-content/uploads/2018/08/Precision-Recall-Plot-for-a-No-Skill-Classifier-and-a-Logistic-Regression-Model4.png

Remember: ROC and precision recall curves

- Sometimes also see **precision recall curve**
    - More informative when dataset is heavily imbalanced (specificity = true negative rate less meaningful in this case)

Object detection is typically heavily imbalanced (most of the data is background) -> PR curves most common evaluation

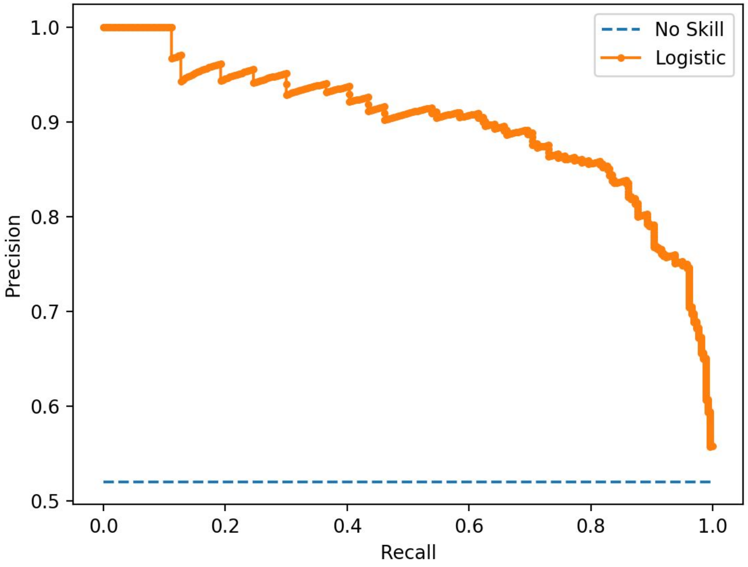
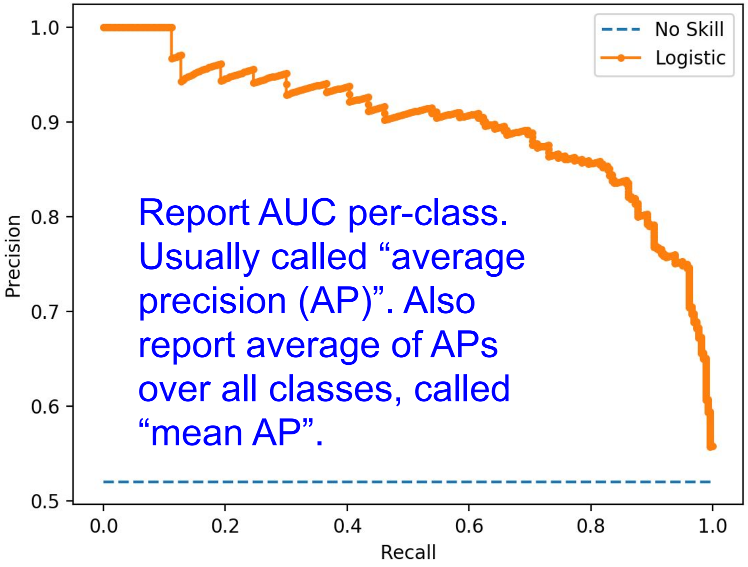


Figure credit: https://3qeqpr26caki16dnhd19sv6by6v-wpengine.netdna-ssl.com/wp-content/uploads/2018/08/Precision-Recall-Plot-for-a-No-Skill-Classifier-and-a-Logistic-Regression-Model4.png

Remember: ROC and precision recall curves

- Sometimes also see **precision recall curve**
  - More informative when dataset is heavily imbalanced (specificity = true negative rate less meaningful in this case)

Object detection is typically heavily imbalanced (most of the data is background) -> PR curves most common evaluation



Report AUC per-class. Usually called “average precision (AP)”. Also report average of APs over all classes, called “mean AP”.

Figure credit: https://3qeqpr26caki16dnhd19sv6by6v-wpengine.netdna-ssl.com/wp-content/uploads/2018/08/Precision-Recall-Plot-for-a-No-Skill-Classifier-and-a-Logistic-Regression-Model4.png

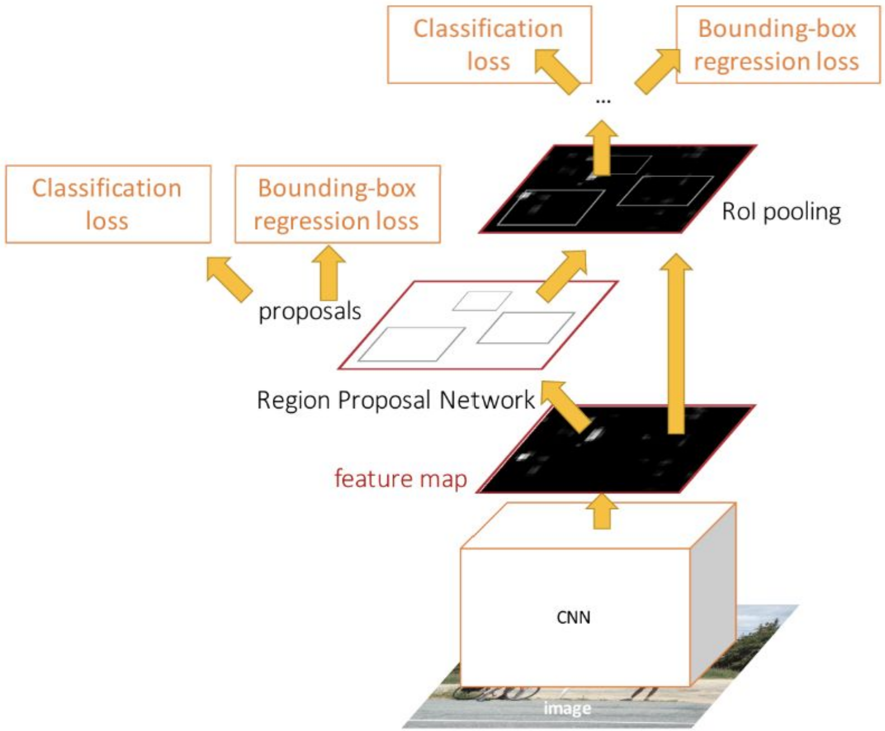
Evaluation of object detection

Obtain from model outputs

Standard output of object detection

For each class, a set of bounding box predictions with associated confidences:

- E.g., $(\underbrace{x, y, h, w}_{\text{Bounding box}}, c)$ Class confidence



Evaluation of object detection

Standard output of object detection

For each class, a set of bounding box predictions with associated confidences:

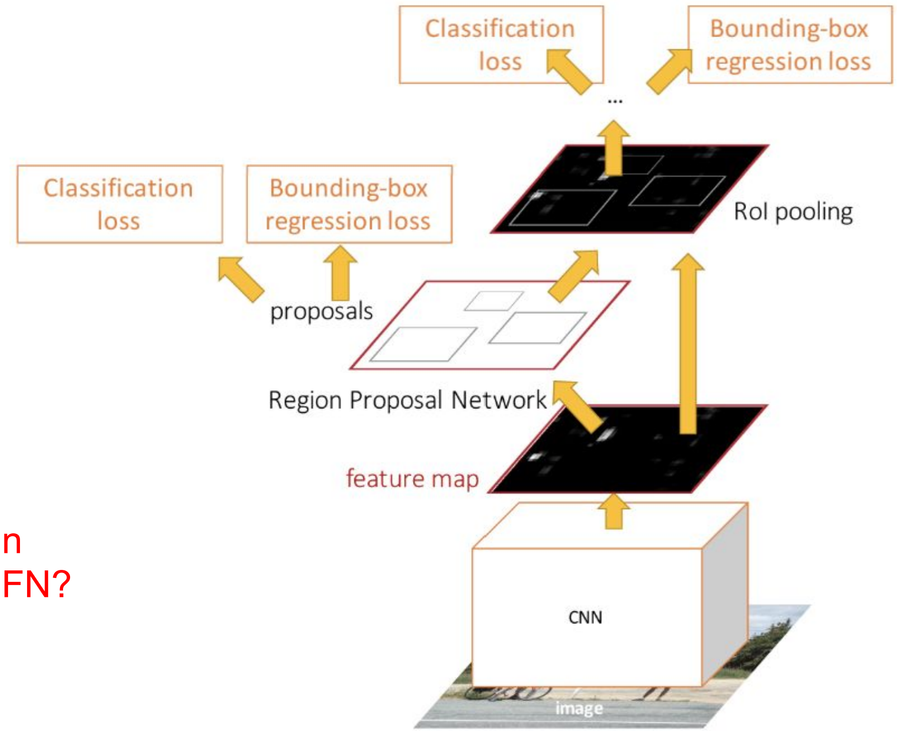
- E.g., (x, y, h, w, c)

Bounding box

Class confidence

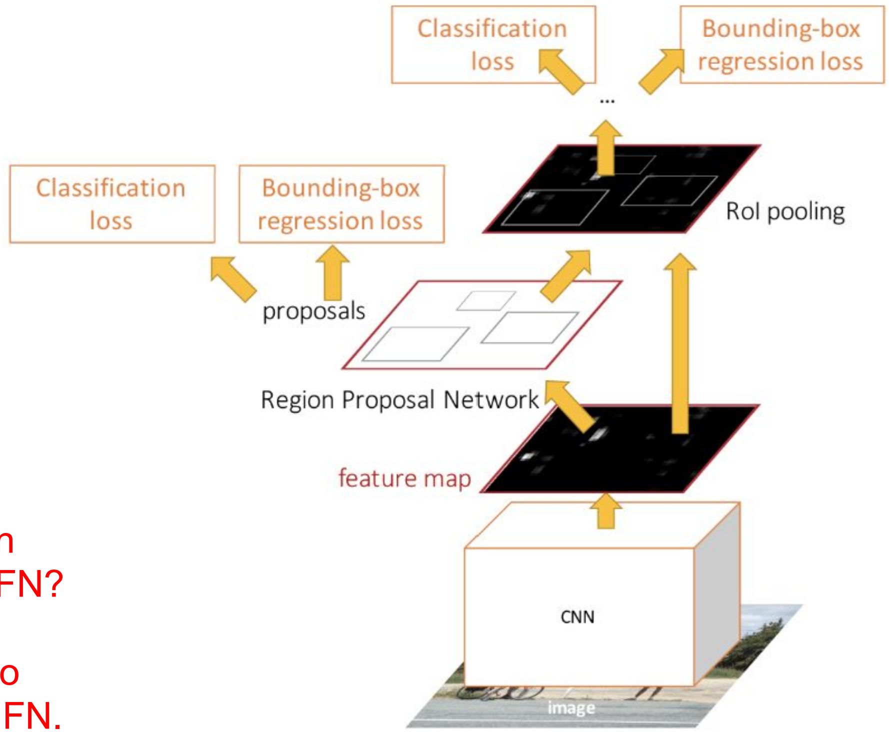
We have the class confidences to vary the threshold in plotting the PR curve. But how do we get TP, TN, FP, FN?

Obtain from model outputs



Evaluation of object detection

Obtain from model outputs



Standard output of object detection

For each class, a set of bounding box predictions with associated confidences:

- E.g., $(\underbrace{x, y, h, w}_{\text{Bounding box}}, c)$

  Class confidence

We have the class confidences to vary the threshold in plotting the PR curve. But how do we get TP, TN, FP, FN?

A: Choose an IOU threshold with ground truth boxes to determine if bounding box prediction is TP, TN, FP, or FN. Then can plot PR curve and obtain AP metric.

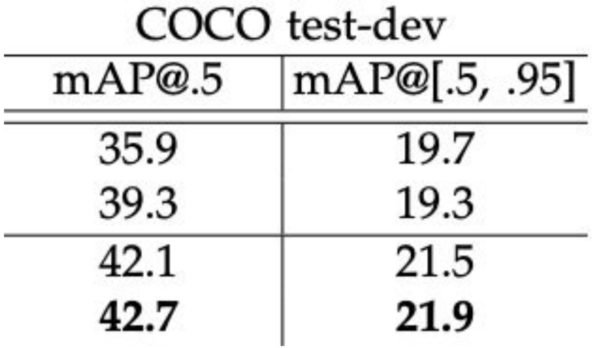
Evaluation of object detection

Standard output of object detection

For each class, a set of bounding box predictions with associated confidences:

- E.g., $(\underbrace{x, y, h, w}_{\text{Bounding box}}, c)$

Class confidence



COCO test-dev	
mAP@.5	mAP@[.5, .95]
35.9	19.7
39.3	19.3
42.1	21.5
42.7	**21.9**

We have the class confidences to vary the threshold in plotting the PR curve. But how do we get TP, TN, FP, FN?

A: Choose an IOU threshold with ground truth boxes to determine if bounding box prediction is TP, TN, FP, or FN. Then can plot PR curve and obtain AP metric.

Evaluation of object detection

Standard output of object detection

For each class, a set of bounding box predictions with associated confidences:

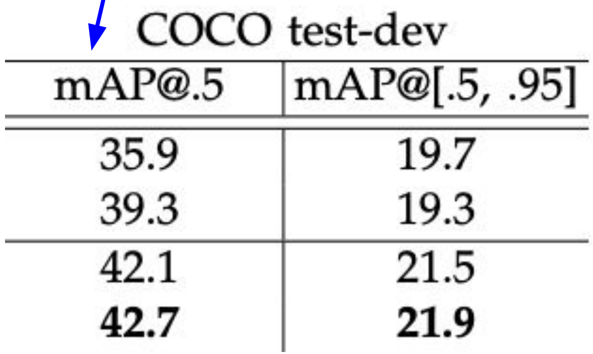
- E.g., $(\underbrace{x, y, h, w}_{\text{Bounding box}}, c)$

Class confidence

We have the class confidences to vary the threshold in plotting the PR curve. But how do we get TP, TN, FP, FN?

A: Choose an IOU threshold with ground truth boxes to determine if bounding box prediction is TP, TN, FP, or FN. Then can plot PR curve and obtain AP metric.

mAP (over all classes), with IOU threshold of 0.5. Often report mAP at multiple IOUs.



COCO test-dev

mAP@.5	mAP@[.5, .95]
35.9	19.7
39.3	19.3
42.1	21.5
42.7	**21.9**

Evaluation of object detection

Standard output of object detection

For each class, a set of bounding box predictions with associated confidences:

- E.g., (x, y, h, w, c)

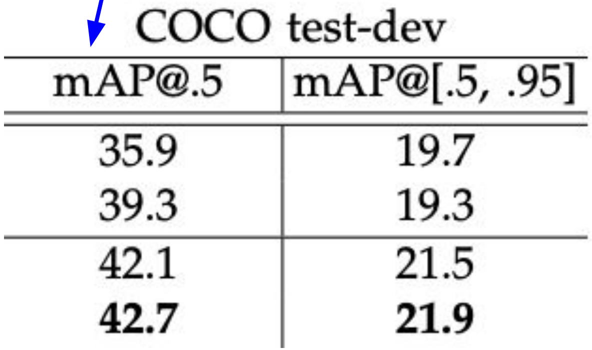
Bounding box

Class confidence

We have the class confidences to vary the threshold in plotting the PR curve. But how do we get TP, TN, FP, FN?

A: Choose an IOU threshold with ground truth boxes to determine if bounding box prediction is TP, TN, FP, or FN. Then can plot PR curve and obtain AP metric.

mAP (over all classes), with IOU threshold of 0.5. Often report mAP at multiple IOUs.



COCO test-dev

mAP@.5	mAP@[.5, .95]
35.9	19.7
39.3	19.3
42.1	21.5
42.7	**21.9**

If IOU threshold not specified in experiments description for a paper, may need to look in dataset evaluation documentation. Default is often 0.5.

Evaluation of object detection

Standard output of object detection

For each class, a set of bounding box predictions with associated confidences:

- E.g., $(\underbrace{x, y, h, w}_{\text{Bounding box}}, c)$

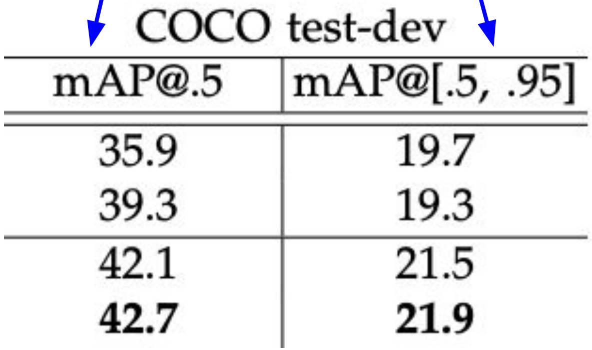
Class confidence

We have the class confidences to vary the threshold in plotting the PR curve. But how do we get TP, TN, FP, FN?

A: Choose an IOU threshold with ground truth boxes to determine if bounding box prediction is TP, TN, FP, or FN. Then can plot PR curve and obtain AP metric.

mAP (over all classes), with IOU threshold of 0.5. Often report mAP at multiple IOUs.

Average of mAP values at IOU thresholds regularly sampled in the interval between [.5, .95].



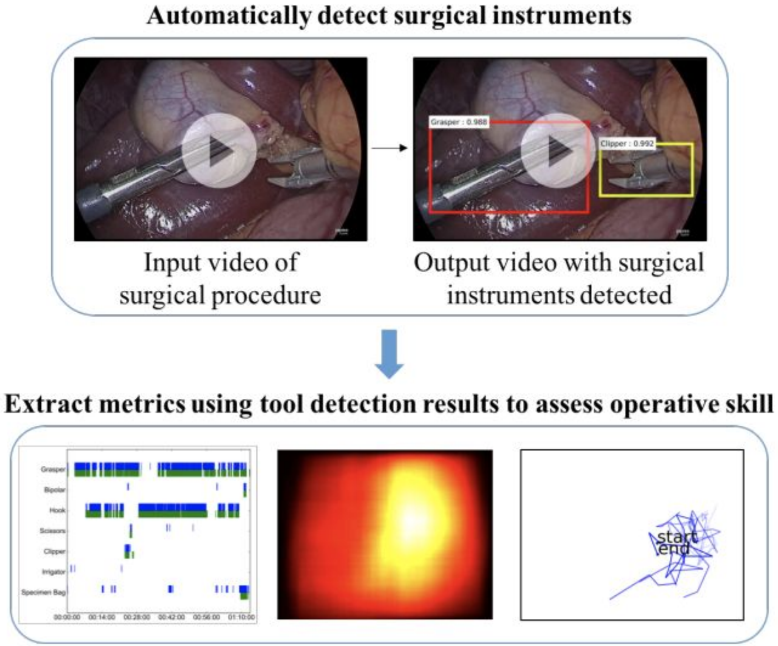
COCO test-dev

mAP@.5	mAP@[.5, .95]
35.9	19.7
39.3	19.3
42.1	21.5
42.7	**21.9**

If IOU threshold not specified in experiments description for a paper, may need to look in dataset evaluation documentation. Default is often 0.5.

Jin et al. 2018

- Detection of surgical instruments in surgery videos (in each video frame)
- Surgical instrument movement over the course of a video can be used to extract metrics such as tool switching, and spatial trajectories, that can be used to assess and provide feedback on operative skill.
- Used M2cai16-tool dataset of 15 surgical videos. Annotated 2532 frames with bounding boxes of 7 tools.



Automatically detect surgical instruments

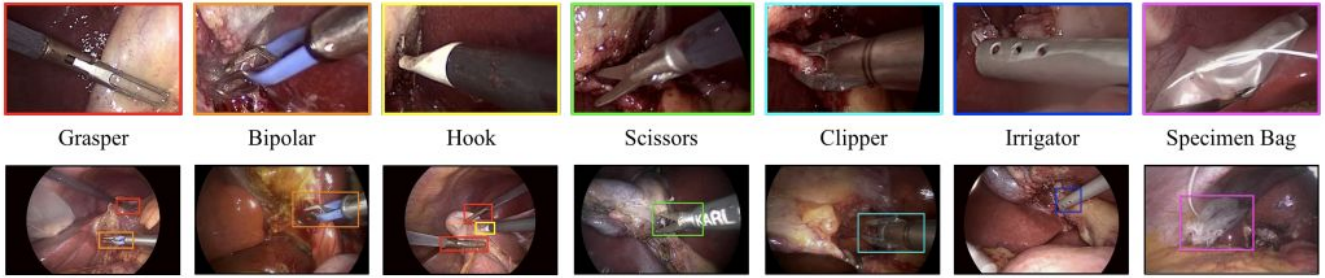
Input video of surgical procedure

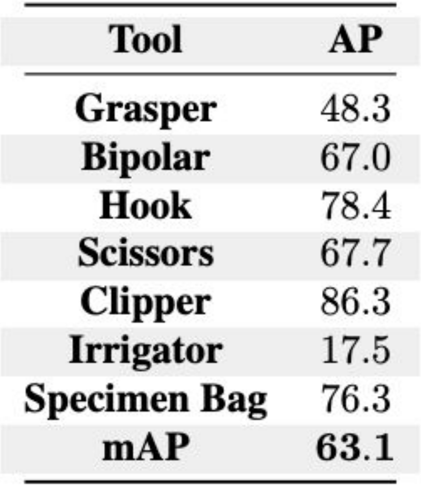
Output video with surgical instruments detected

Extract metrics using tool detection results to assess operative skill

Jin et al. Tool Detection and Operative Skill Assessment in Surgical Videos Using Region-Based Convolutional Neural Networks. WACV, 2018.

Jin et al. 2018

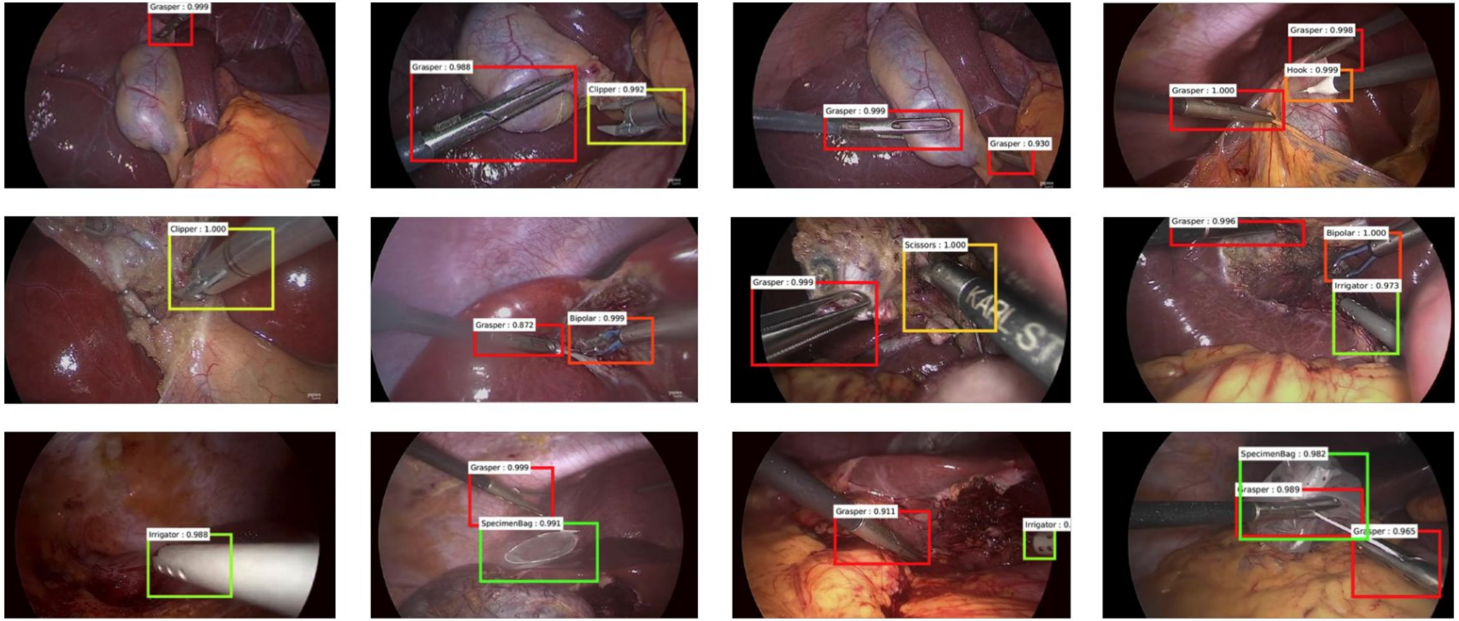




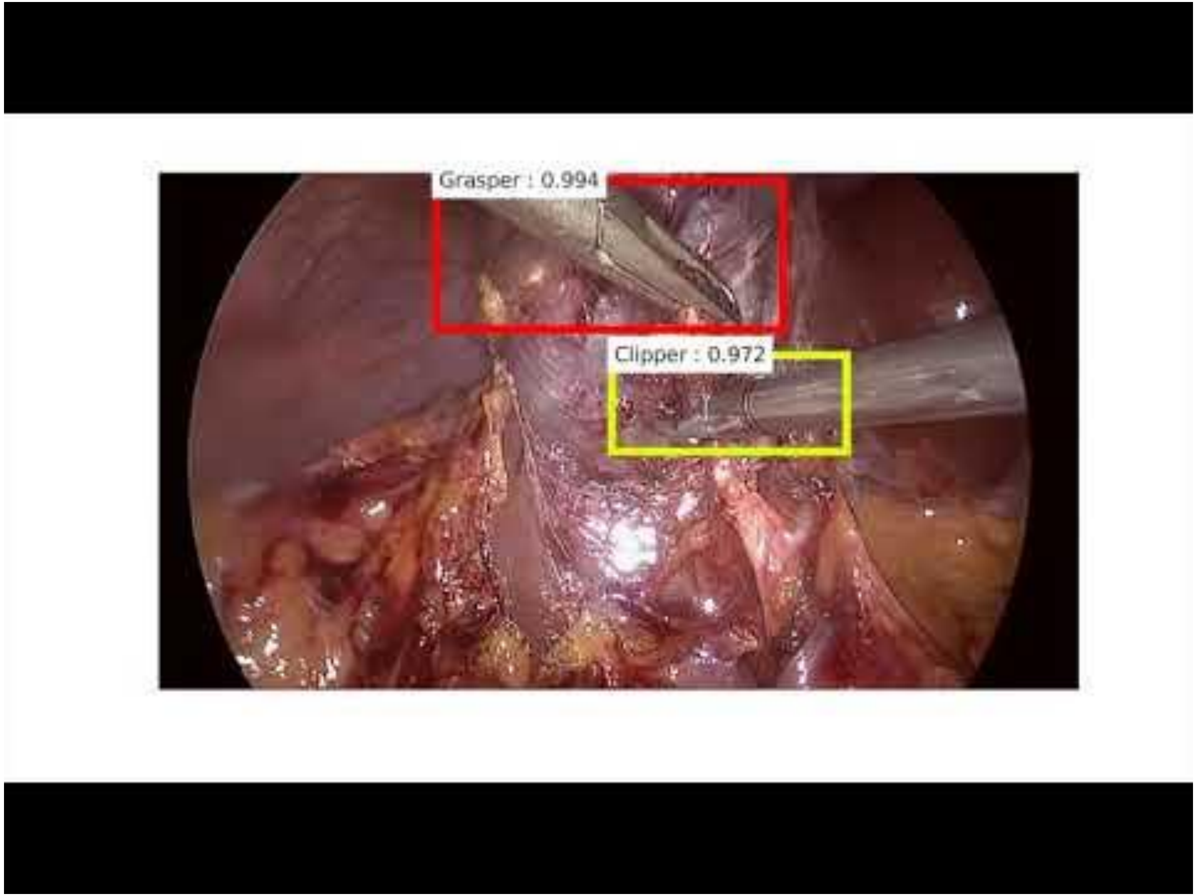
Tool	AP
Grasper	48.3
Bipolar	67.0
Hook	78.4
Scissors	67.7
Clipper	86.3
Irrigator	17.5
Specimen Bag	76.3
mAP	63.1

Jin et al. Tool Detection and Operative Skill Assessment in Surgical Videos Using Region-Based Convolutional Neural Networks. WACV, 2018.

Jin et al. 2018



Jin et al. Tool Detection and Operative Skill Assessment in Surgical Videos Using Region-Based Convolutional Neural Networks. WACV, 2018.



Jin et al. Tool Detection and Operative Skill Assessment in Surgical Videos Using Region-Based Convolutional Neural Networks. WACV, 2018.

Other object detection architectures

- RCNN, Fast RCNN: older and slower predecessors to Faster-RCNN
- YOLO, SSD: single-stage detectors that change region proposal generation -> region classification two-stage pipeline into a single stage.
    - Faster, but lower performance. Struggles more with class imbalance relative to two-stage networks that filter only top object candidate boxes for the second stage.
- RetinaNet: single-stage detector that uses a “focal loss” to adaptively weight harder examples over easy background examples. Able to outperform Faster R-CNN on some benchmark tasks, while being more efficient.

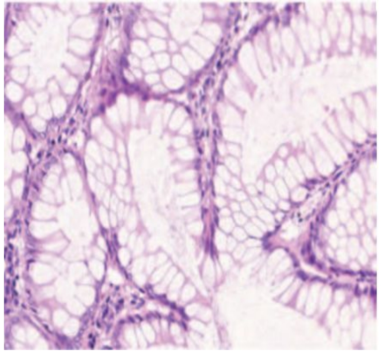
Other object detection architectures

- RCNN, Fast RCNN: older and slower predecessors to Faster-RCNN
- YOLO, SSD: single-stage detectors that change region proposal generation -> region classification two-stage pipeline into a single stage.
    - Faster, but lower performance. Struggles more with class imbalance relative to two-stage networks that filter only top object candidate boxes for the second stage.
- RetinaNet: single-stage detector that uses a “focal loss” to adaptively weight harder examples over easy background examples. Able to outperform Faster R-CNN on some benchmark tasks, while being more efficient.

RetinaNet also worth trying
for object detection projects!

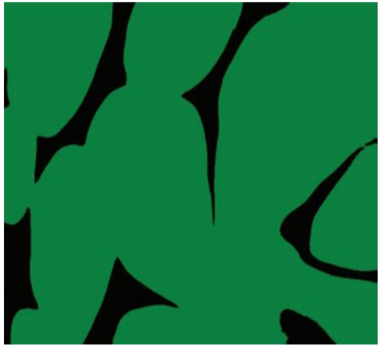
Richer visual recognition tasks: segmentation and detection

Classification



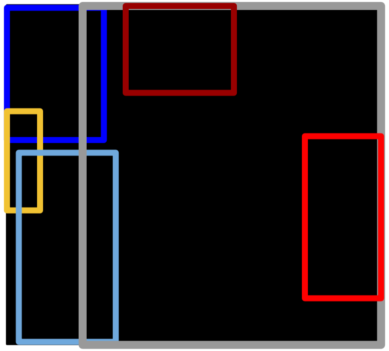
Output:
one category label for image (e.g., colorectal glands)

Semantic Segmentation



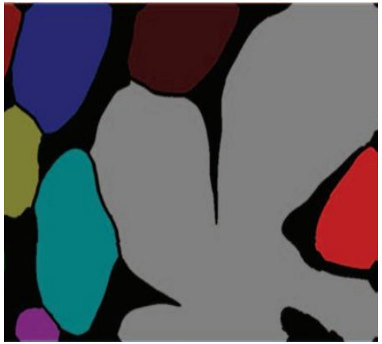
Output:
category label for each pixel in the image

Detection



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

Instance Segmentation

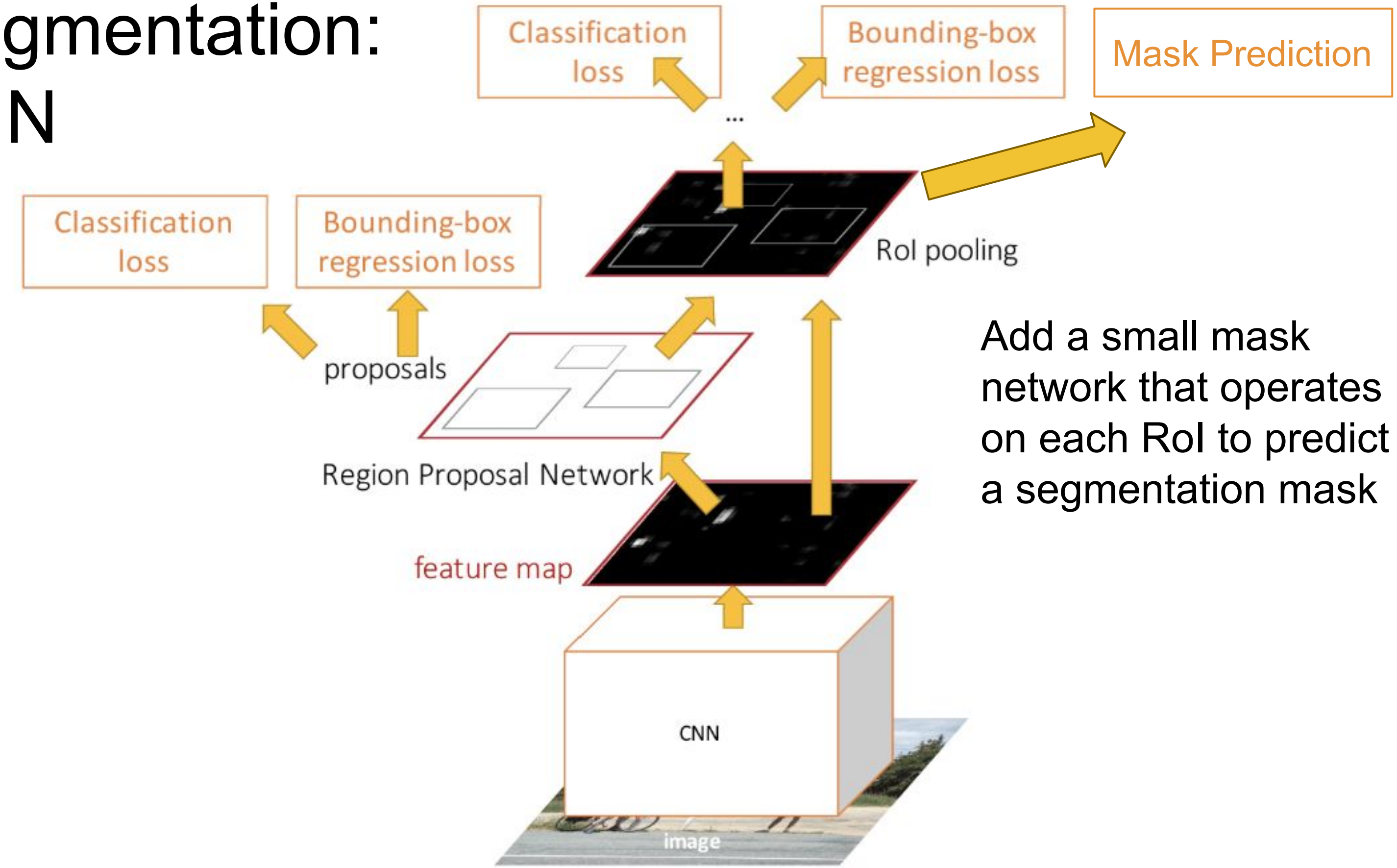


Output:
Category label and instance label for each pixel in the image

Distinguishes between different instances of an object

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

Instance segmentation: Mask R-CNN



Add a small mask network that operates on each RoI to predict a segmentation mask

Cropping Features: RoI *Align*

Sample at regular points in each subregion using bilinear interpolation

No “snapping”!

Improved version of RoI Pool since we now care about pixel-level segmentation accuracy!

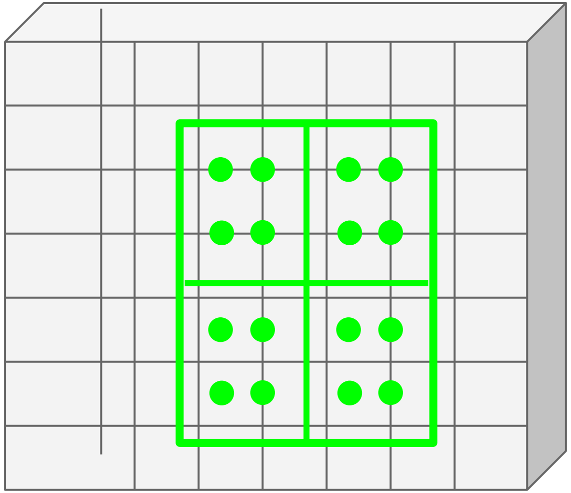


Image features
(e.g. 512 x 20 x 15)

Cropping Features: RoI *Align*

Improved version of RoI Pool since we now care about pixel-level segmentation accuracy!

No "snapping"!

Sample at regular points in each subregion using bilinear interpolation

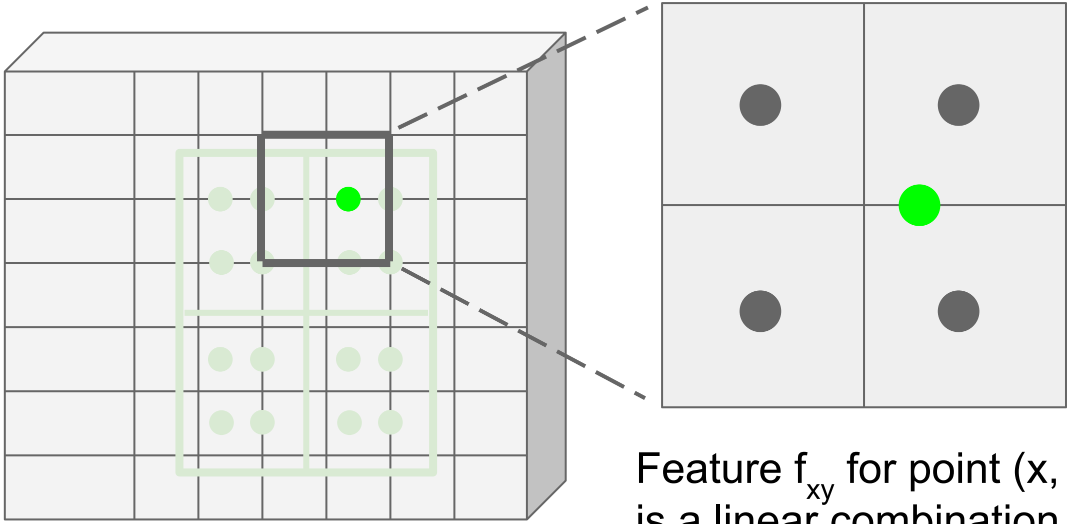


Image features

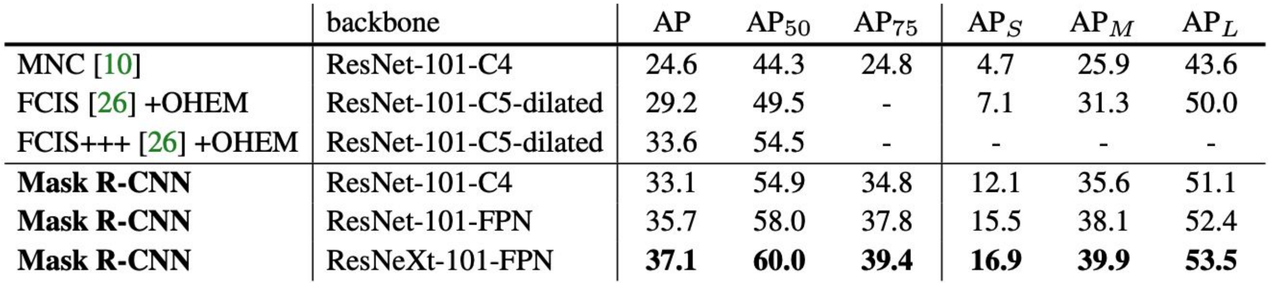
Feature f_{xy} for point (x, y) is a linear combination of features at its four neighboring grid cells

Instance segmentation evaluation

- Instance-based task, like object detection
- Also use same precision-recall curve and AP evaluation metrics
- Only difference is that IOU is now a mask IOU
  - Same as the IOU for semantic segmentation, but now per-instance

Instance segmentation evaluation

- Instance-based task, like object detection
- Also use same precision-recall curve and AP evaluation metrics
- Only difference is that IOU is now a mask IOU
    - Same as the IOU for semantic segmentation, but now per-instance

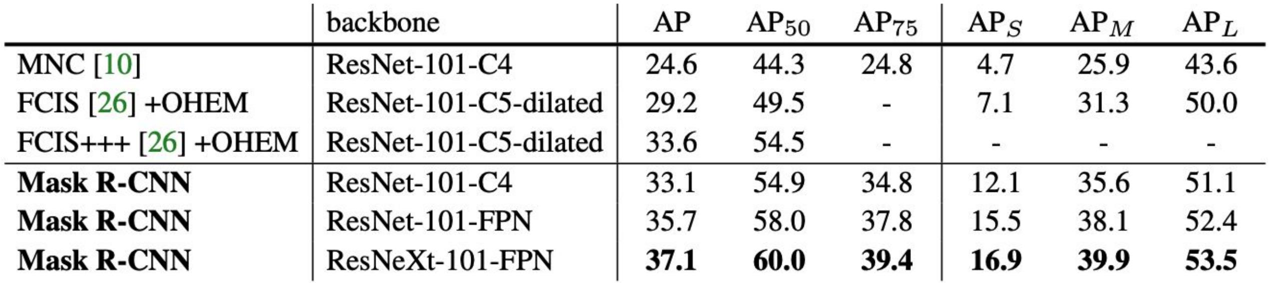


	backbone	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

Instance segmentation evaluation

- Instance-based task, like object detection
- Also use same precision-recall curve and AP evaluation metrics
- Only difference is that IOU is now a mask IOU
  - Same as the IOU for semantic segmentation, but now per-instance

Average AP over different IOU thresholds



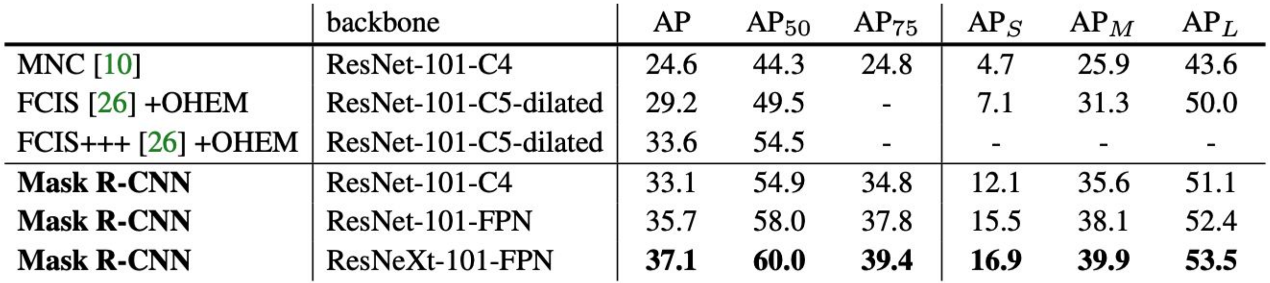
	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

Instance segmentation evaluation

- Instance-based task, like object detection
- Also use same precision-recall curve and AP evaluation metrics
- Only difference is that IOU is now a mask IOU
  - Same as the IOU for semantic segmentation, but now per-instance

Average AP over different IOU thresholds

AP at specific thresholds ("mean AP" is implicit here)



	backbone	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

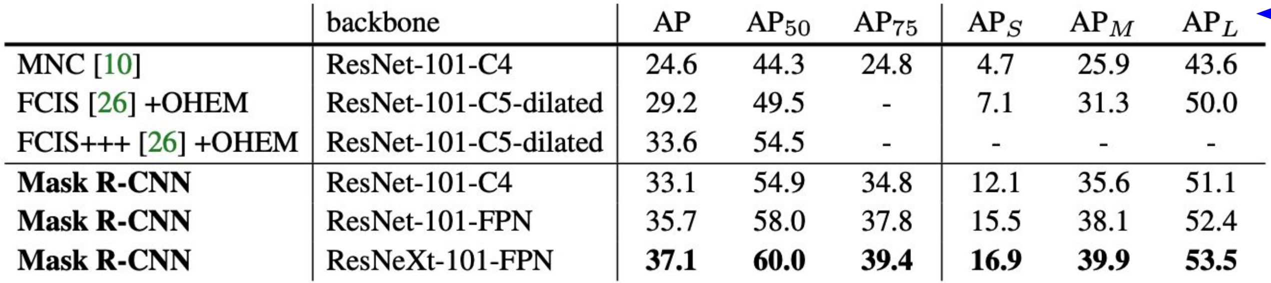
Instance segmentation evaluation

- Instance-based task, like object detection
- Also use same precision-recall curve and AP evaluation metrics
- Only difference is that IOU is now a mask IOU
    - Same as the IOU for semantic segmentation, but now per-instance

Average AP over different IOU thresholds

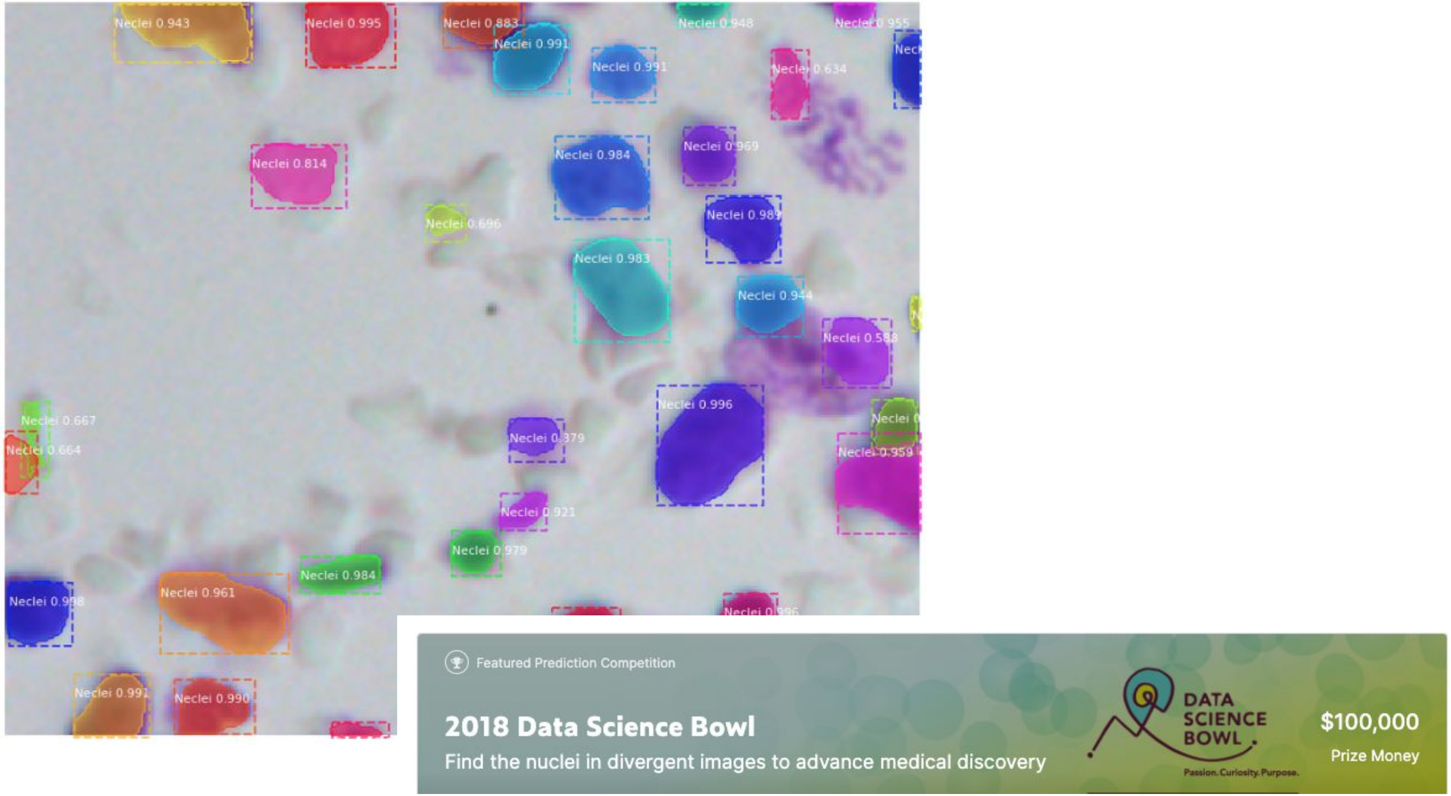
AP at specific thresholds (“mean AP” is implicit here)

AP for small, medium, large objects



	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

Example: instance segmentation of cell nuclei



Featured Prediction Competition

2018 Data Science Bowl

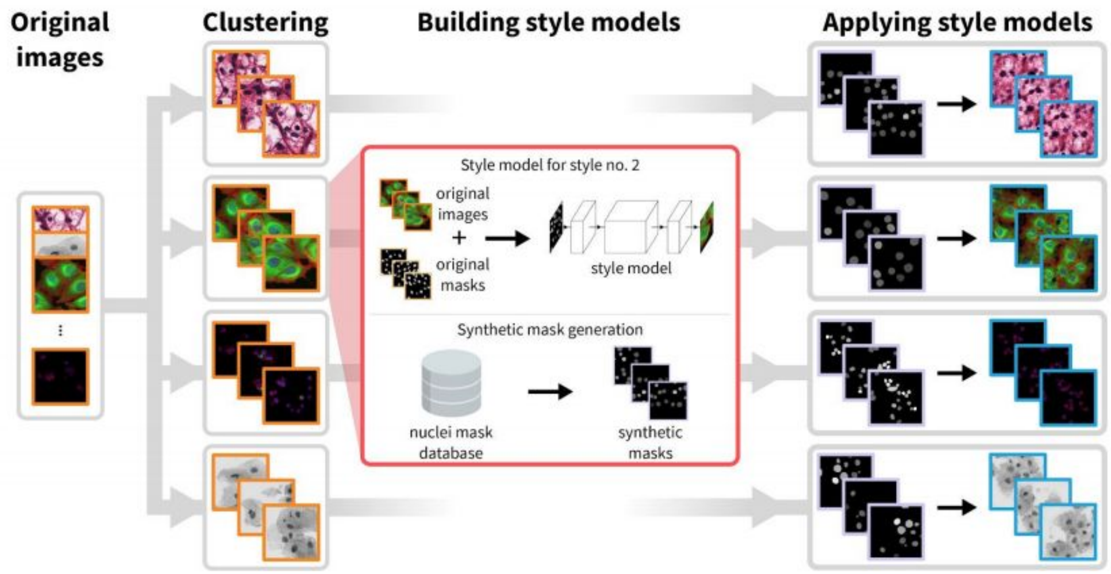
Find the nuclei in divergent images to advance medical discovery

$100,000

Prize Money

Many interesting extensions

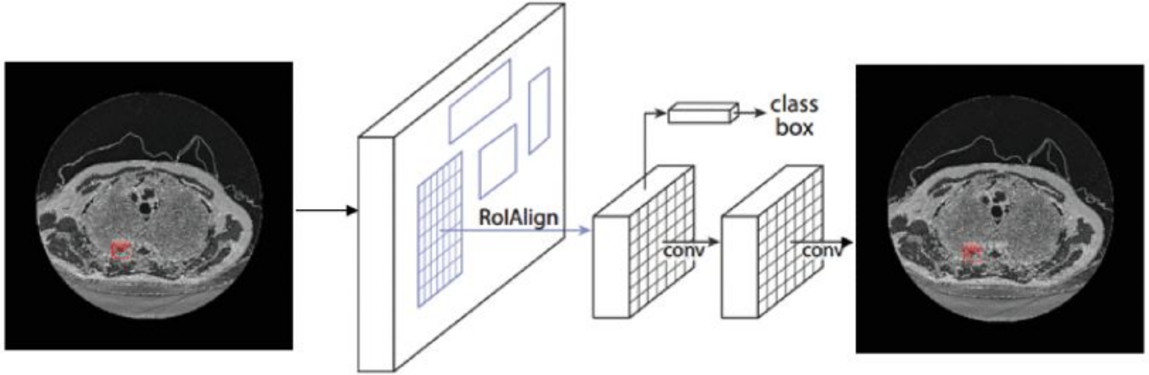
- E.g. Hollandi et al. 2019
    - Used “style transfer” approaches for rich data augmentation
    - Refined Mask-RCNN instance segmentation results with further U-Net-based boundary refinement



Hollandi et al. A deep learning framework for nucleus segmentation using image style transfer. 2019.

Lung nodule segmentation

- E.g. Liu et al. 2018
  - Dataset: Lung Nodule Analysis (LUNA) challenge, 888 512x512 CT scans from the Lung Image Data Consortium database (LIDC-IDRI).
  - Performed 2D instance segmentation in 2D CT slices



We will see other ways to handle 3D medical data types in the next lecture

Liu et al. Segmentation of Lung Nodule in CT Images Based on Mask R-CNN. 2018.

Summary

Finished up medical image classification

Beyond classification to richer visual recognition tasks

- Semantic segmentation
- Object detection
- Instance segmentation

Next time: Advanced vision models (3D and video)