Lecture 4:
Medical Images:
Segmentation and Detection (Part 2),
3D and Video

Announcements

- A1 released, due Tue 10/18
- Project proposal due Fri 10/21 – Project suggestions list on Ed (#35)
- Tensorflow review sesion this Friday, 10/7, Alway M106 at 1:30pm

Note on “Deep Learning Fundamentals” review session

What you are expected to know for the class:

- Definition and conceptual understanding of how the main components of different types of neural networks work
- Framework of training a deep learning model
- Conceptual understanding and trade-offs among design choices
- Good practices and techniques for effectively developing deep learning models for different biomedical tasks

What is not expected:

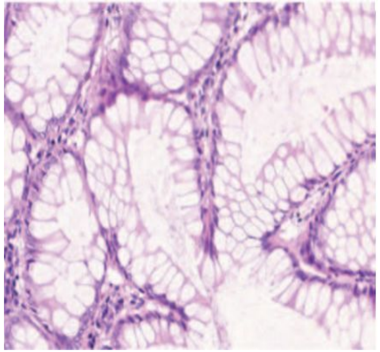
- Remembering / deriving complicated mathematical derivations of gradients, backpropagation, specific optimization methods (Adam, etc.), learning rate schedulers, etc.
- Mathematical details of design choices such as batch normalization, dropout (scaling), etc. Instead you are expected to understand them conceptually, understand trade-offs, and understand how to make good choices about using them

Note on course lectures

- Objective is to establish strong conceptual foundation for developing AI models in healthcare
- Assignments represent what you should be able to implement and know “in detail” from this class
- Lectures teach what you need to know for assignments, but may sometimes go a bit deeper. Goal is to give conceptual grounding such that you can refer back and have the foundation to explore independently in areas that you choose to dive further (e.g. for your class project or other future projects!)

Last Time:
Richer visual recognition tasks: segmentation and detection

Classification

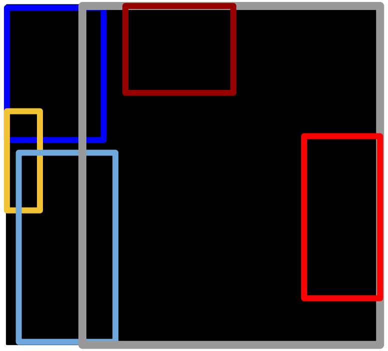


Output:
one category label for image (e.g., colorectal glands)

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

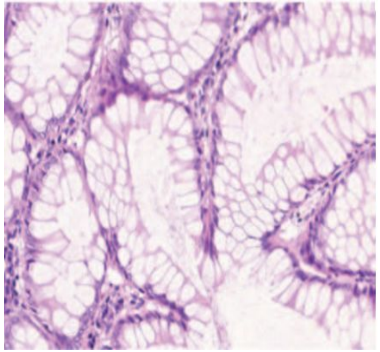
Output:
Category label and instance label for each pixel in the image

Distinguishes between different instances of an object

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

Last Time:
Richer visual recognition tasks: segmentation and detection

Classification

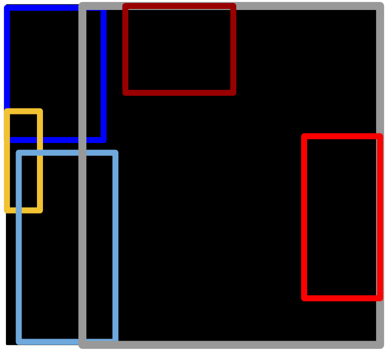


Output:
one category label for image (e.g., colorectal glands)

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

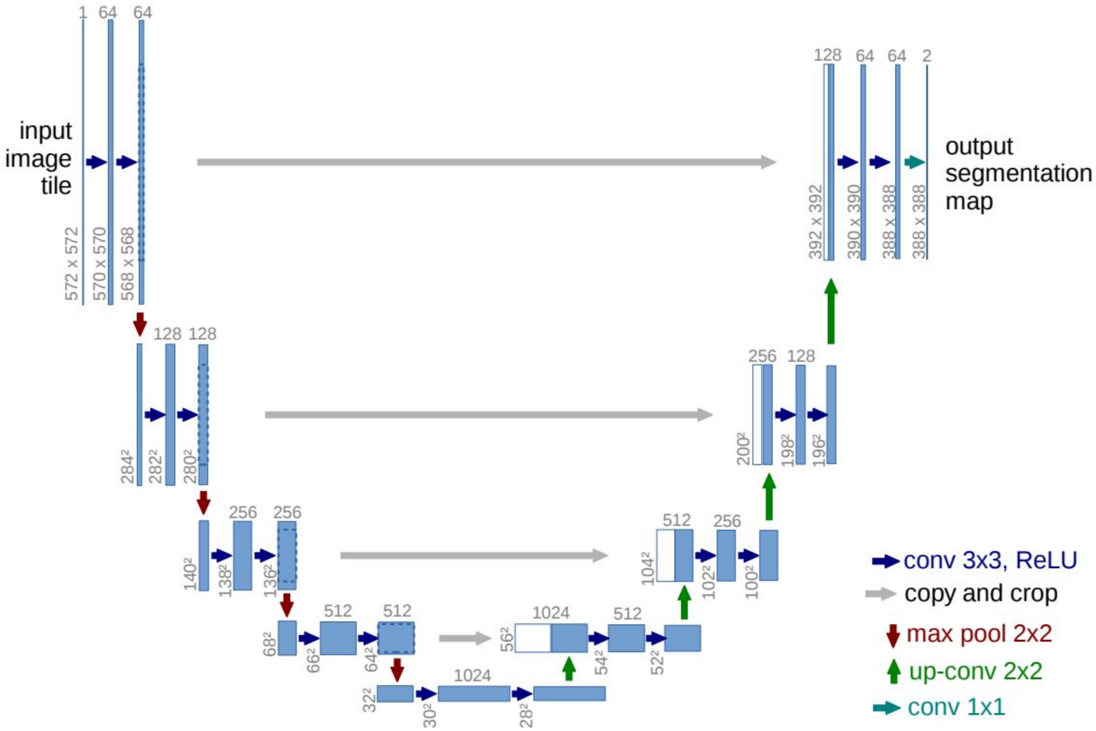
Instance Segmentation

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

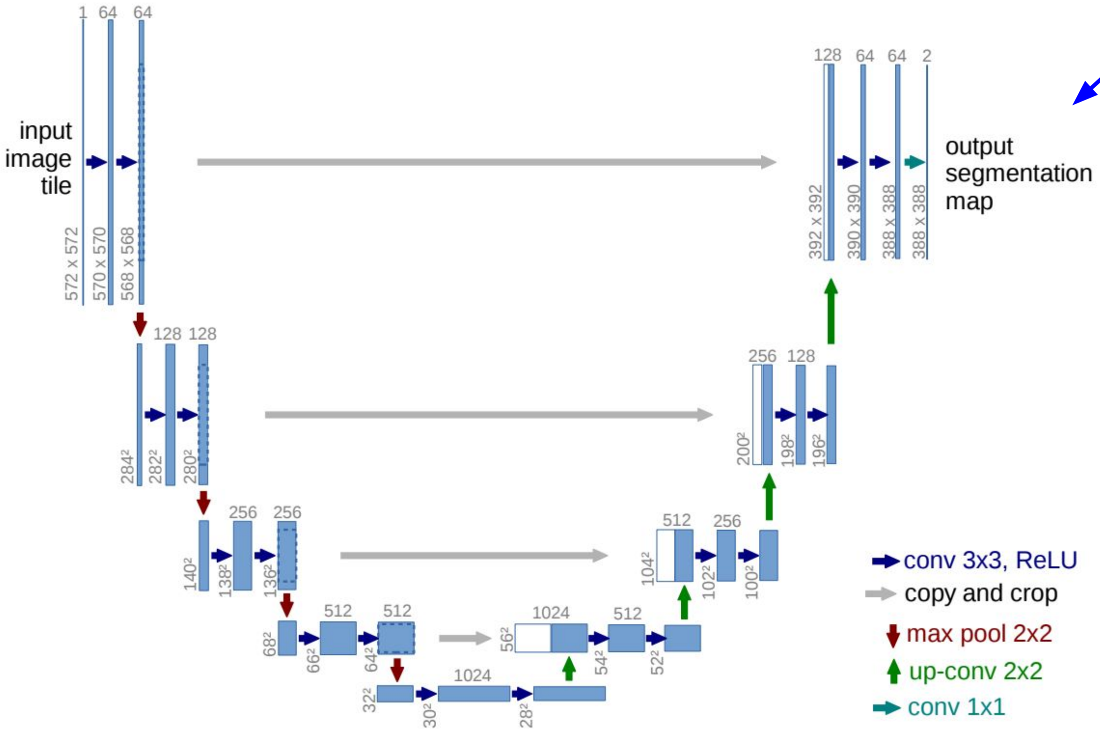
Semantic segmentation: U-Net



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

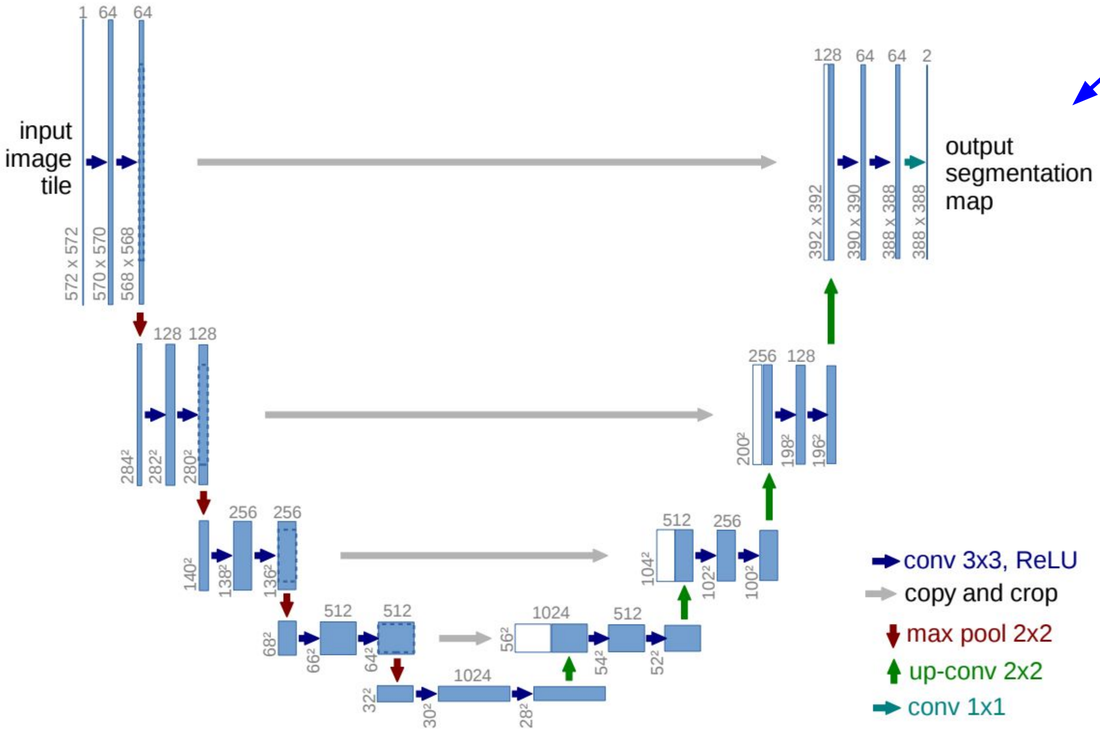
Semantic segmentation: U-Net

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



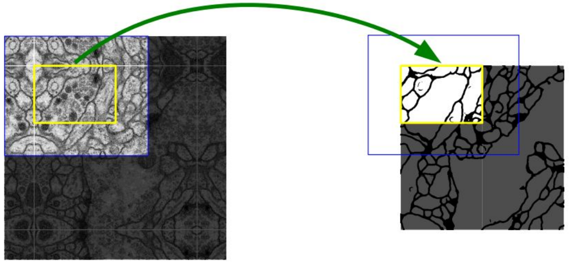
Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

Semantic segmentation: U-Net



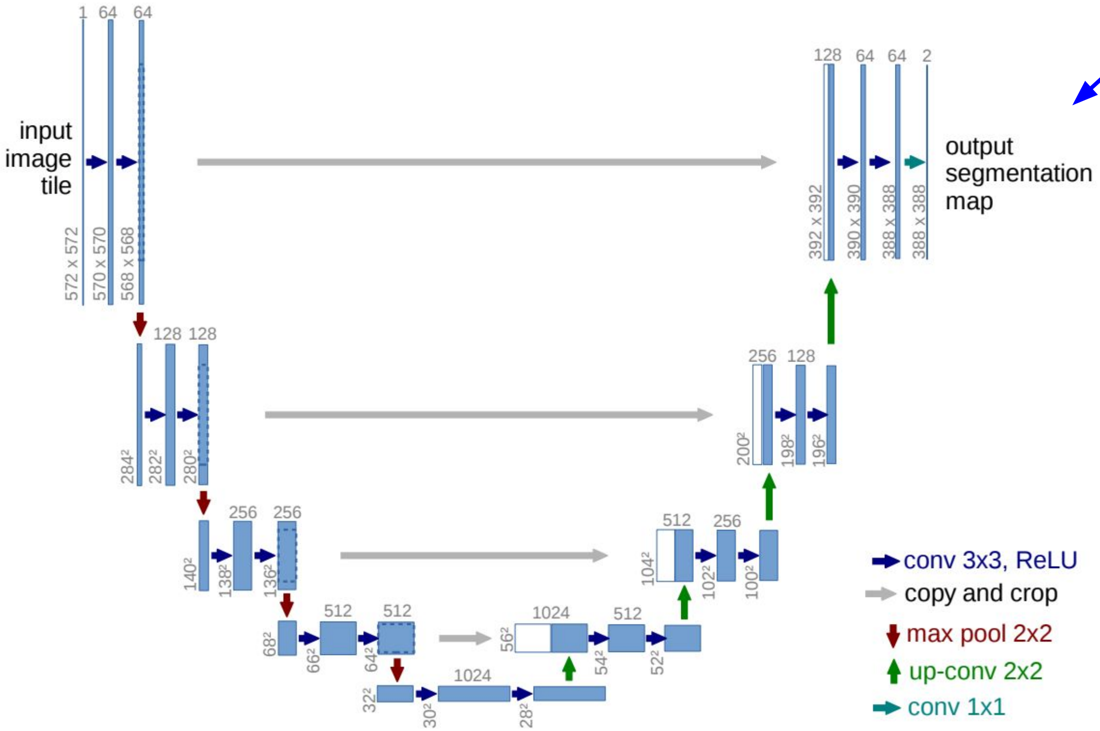
Output is an image mask: width x height x # classes

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



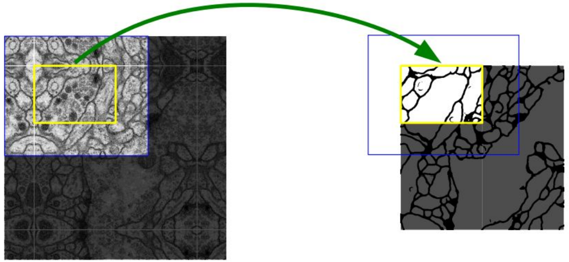
Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

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

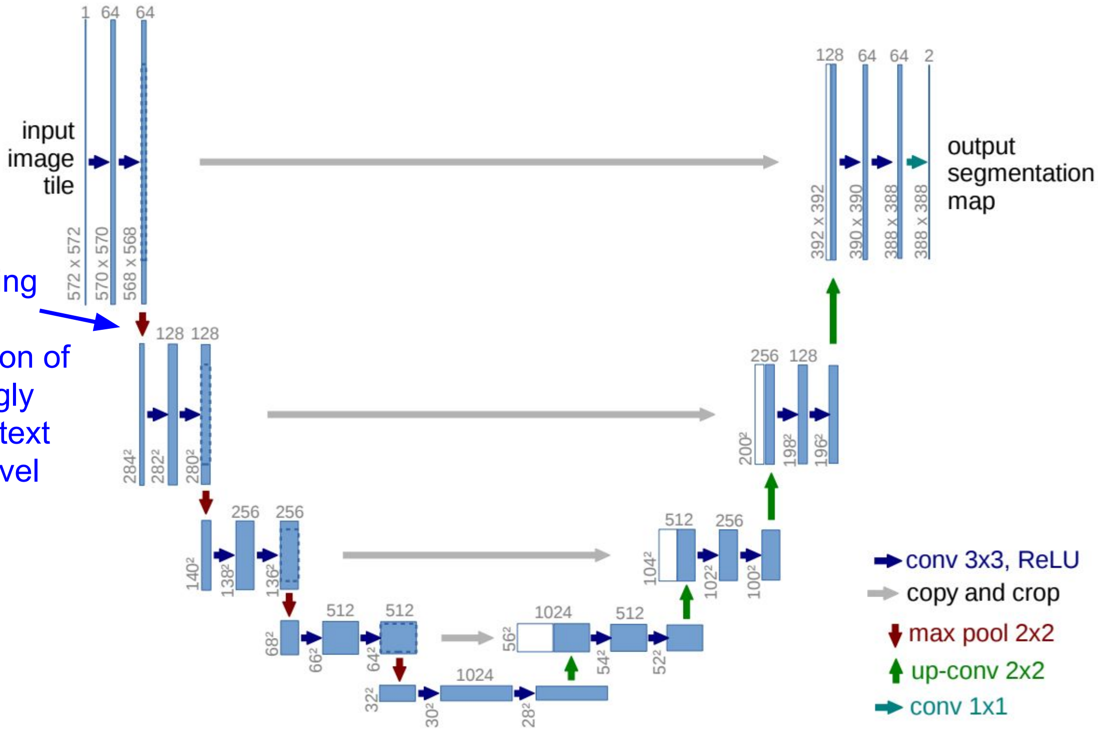


Gives more “true” context for reasoning over each image area. Can tile to make predictions for arbitrarily large images

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

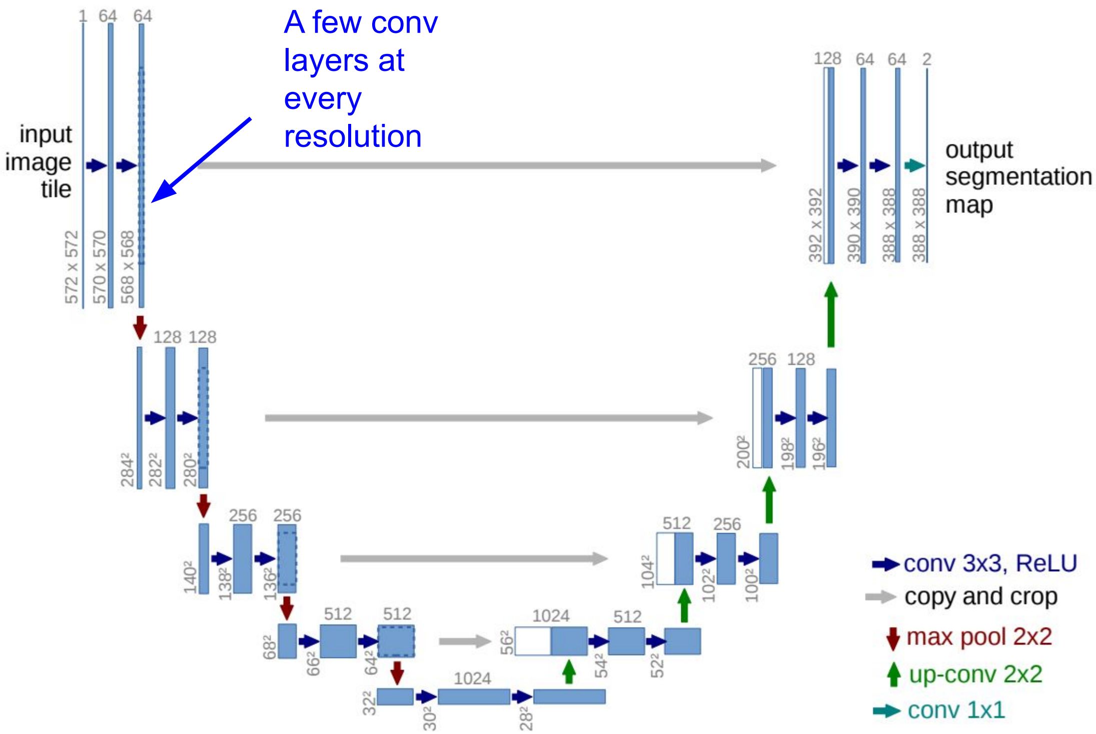
Semantic segmentation: U-Net

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



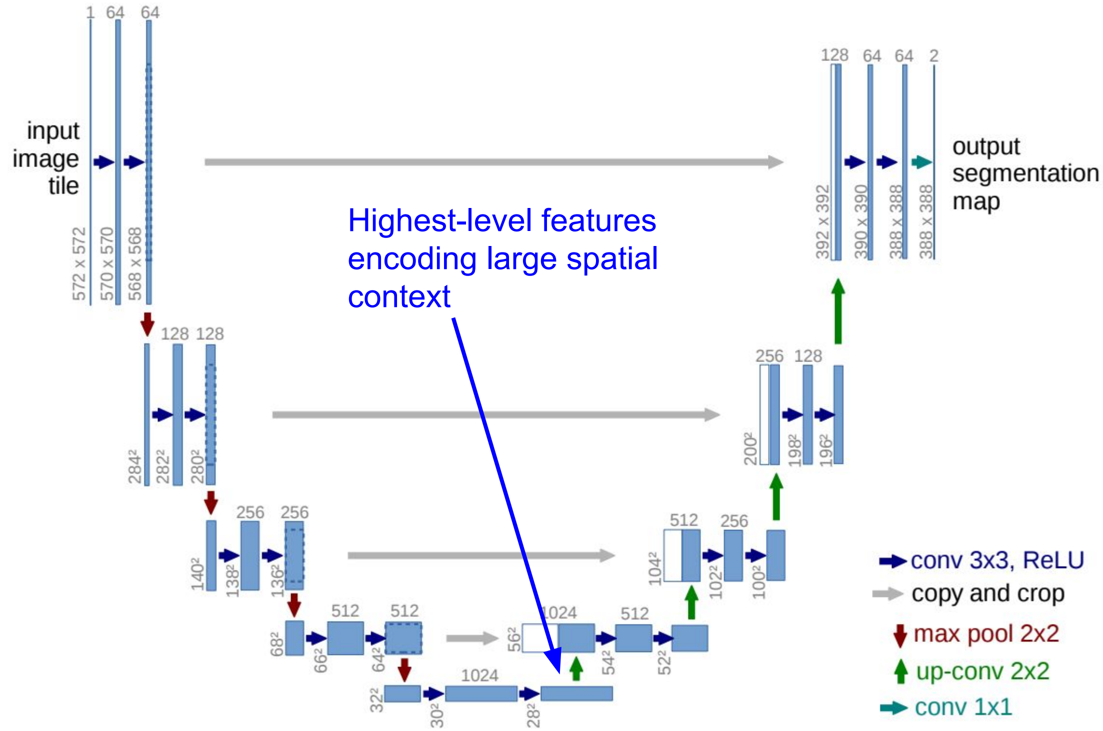
Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

Semantic segmentation: U-Net



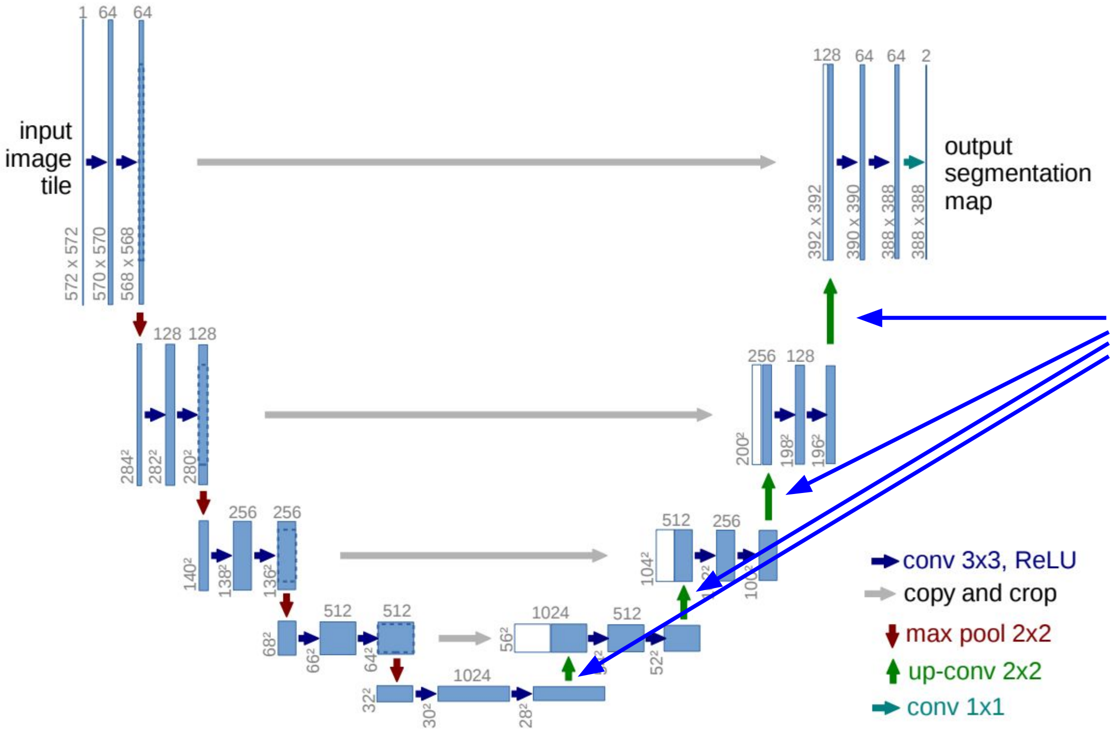
Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

Semantic segmentation: U-Net



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

Semantic segmentation: U-Net

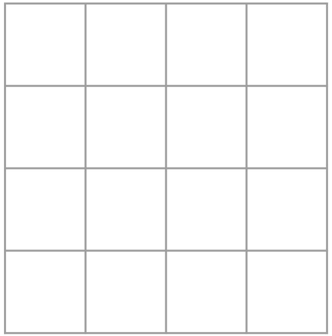


Up-convolutions to go from the global information encoded in highest-level features, back to individual pixel predictions

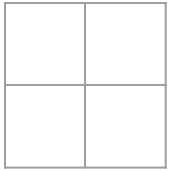
Ronneberger et al. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.

Up-convolutions

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



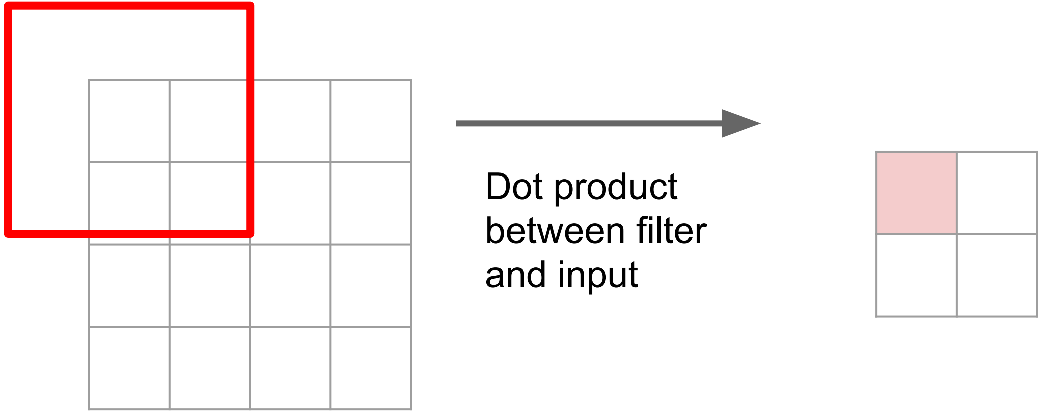
Input: 4 x 4



Output: 2 x 2

Up-convolutions

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

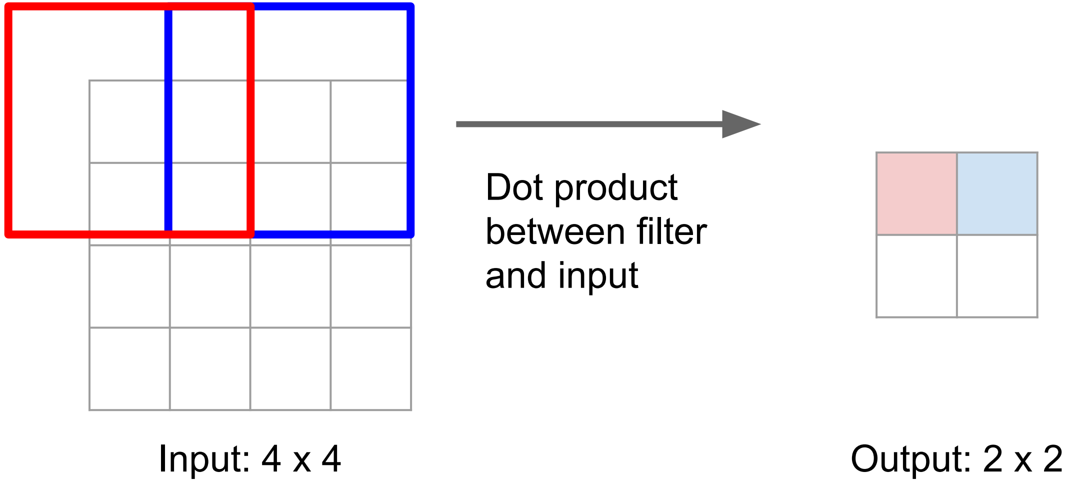


Input: 4 x 4

Output: 2 x 2

Up-convolutions

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

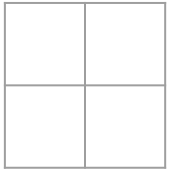


Filter moves 2 pixels in the input for every one pixel in the output

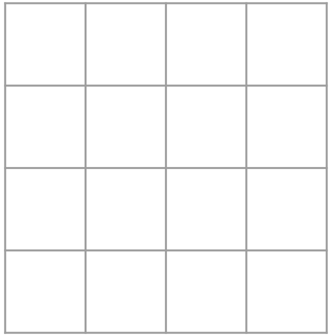
Stride gives ratio between movement in input and output

Up-convolutions

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



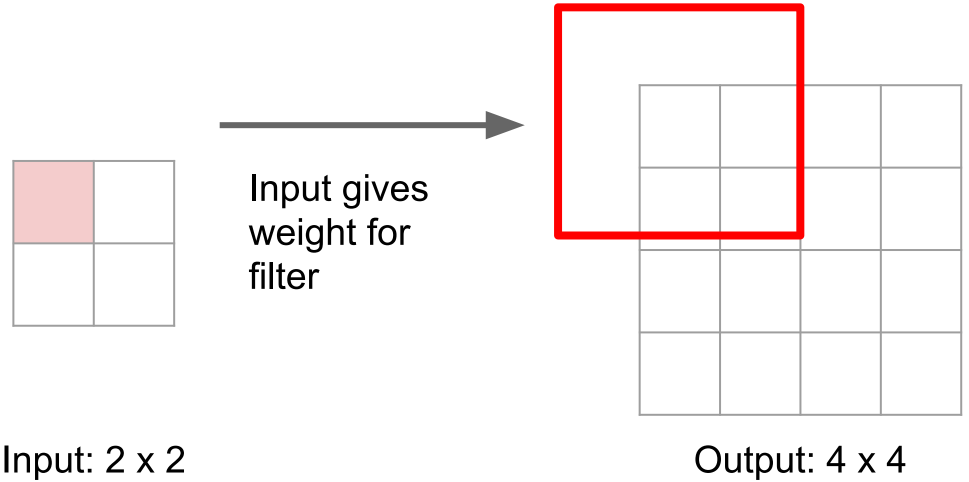
Input: 2 x 2



Output: 4 x 4

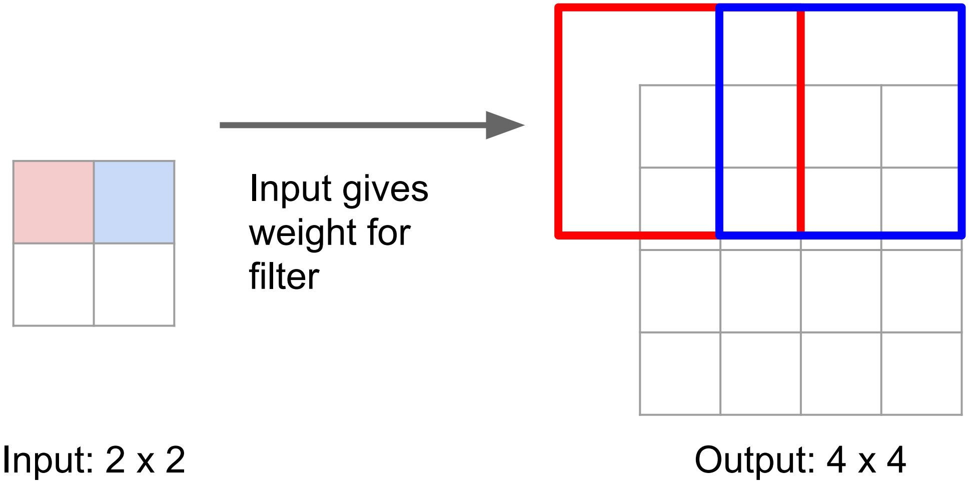
Up-convolutions

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



Up-convolutions

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

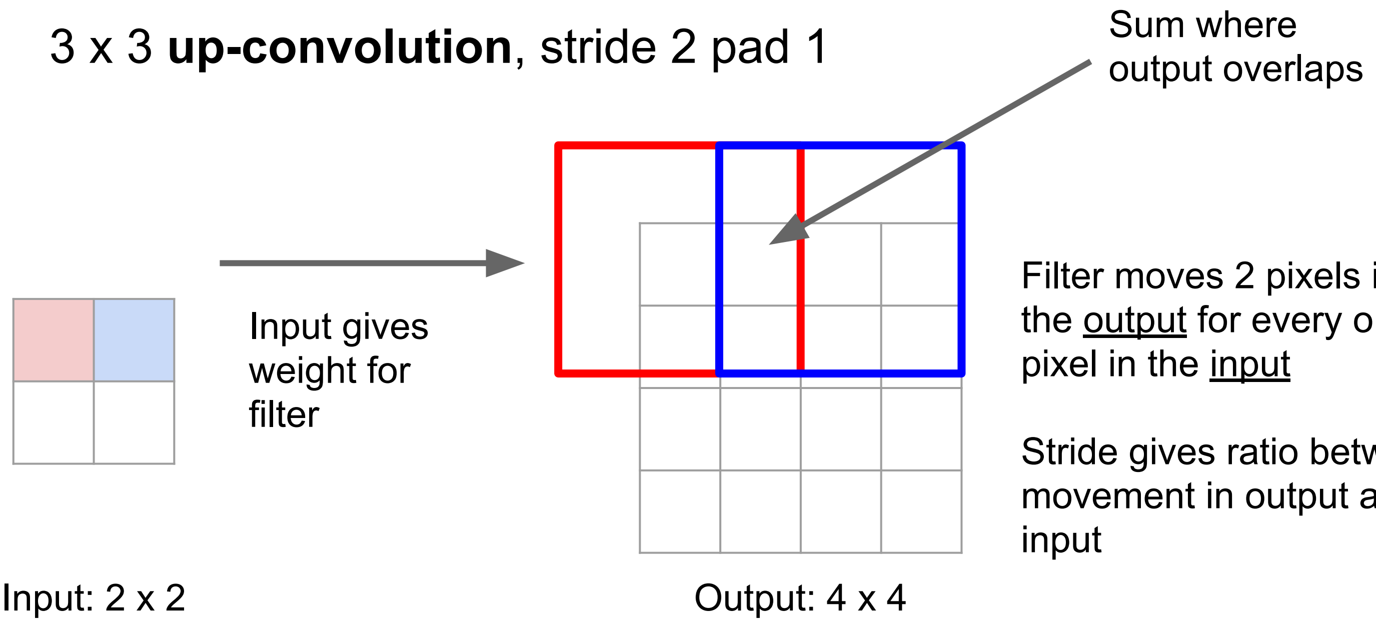


Filter moves 2 pixels in the <u>output</u> for every one pixel in the <u>input</u>

Stride gives ratio between movement in output and input

Up-convolutions

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



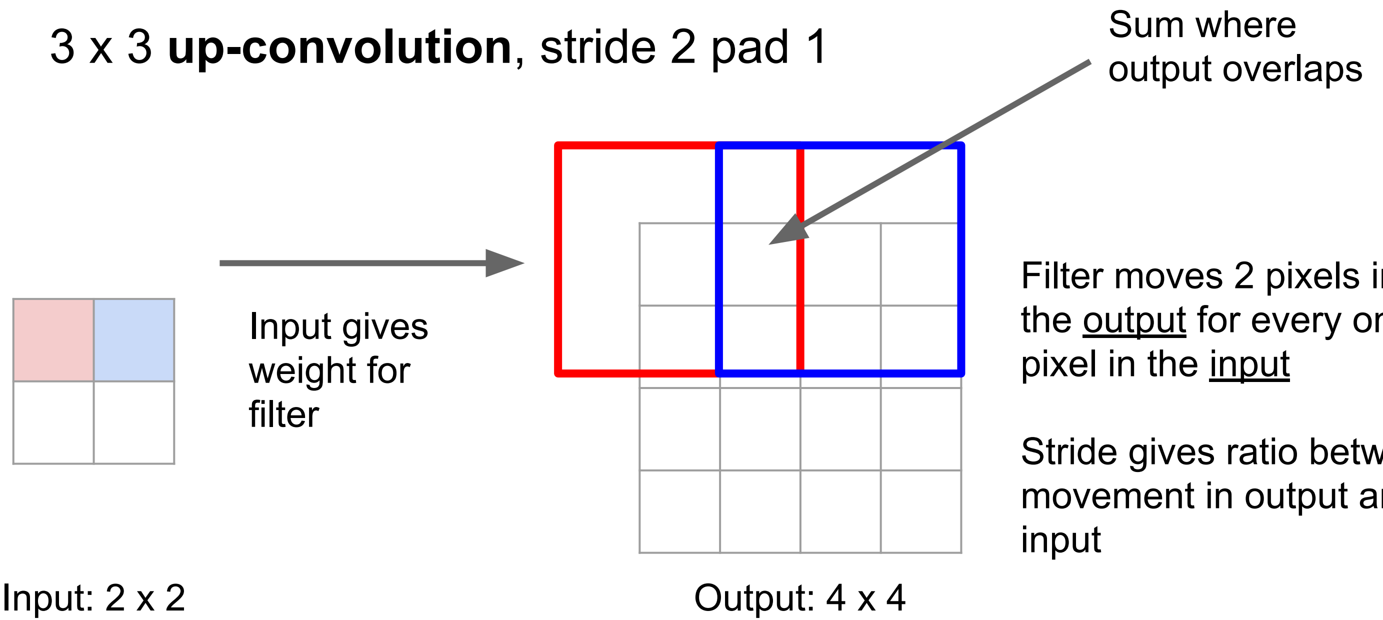
Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

Up-convolutions

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

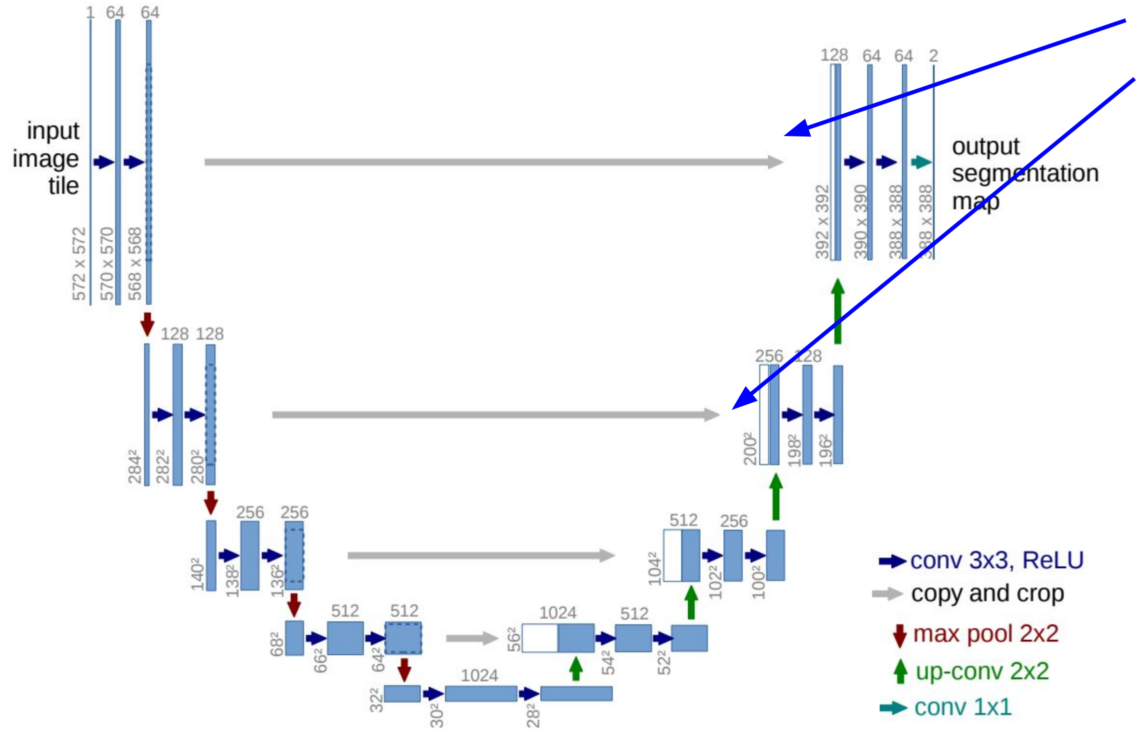
3 x 3 **up-convolution**, stride 2 pad 1



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

Stride gives ratio between movement in output and input

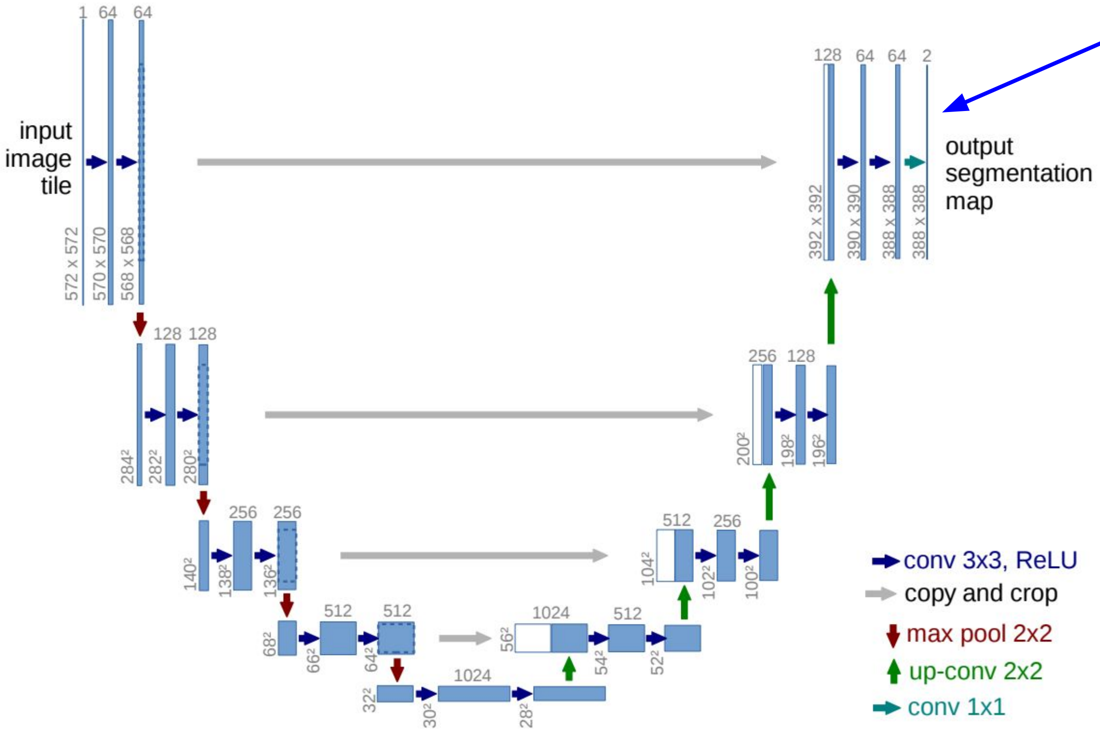
Semantic segmentation: U-Net



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

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

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

2 * intersection

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

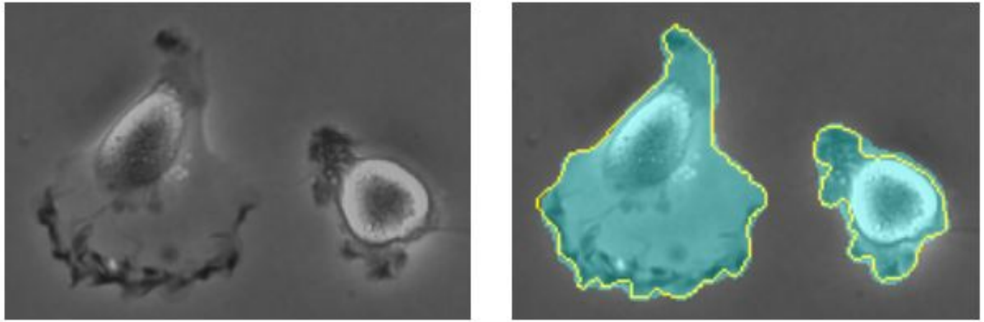
Sum of target mask size
+ prediction mask size

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**

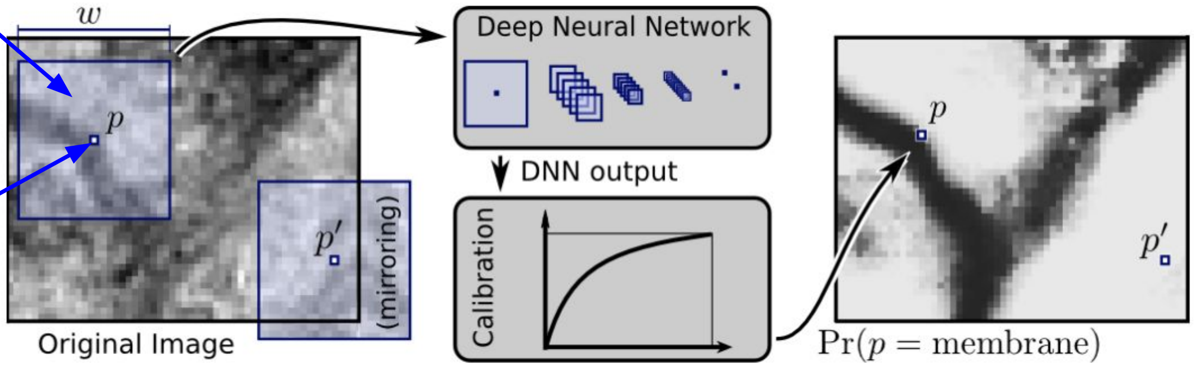
Very small dataset: 30 training images of size 512x512, in the ISBI 2012 Electron Microscopy (EM) segmentation challenge. Used excessive data augmentation to compensate.

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

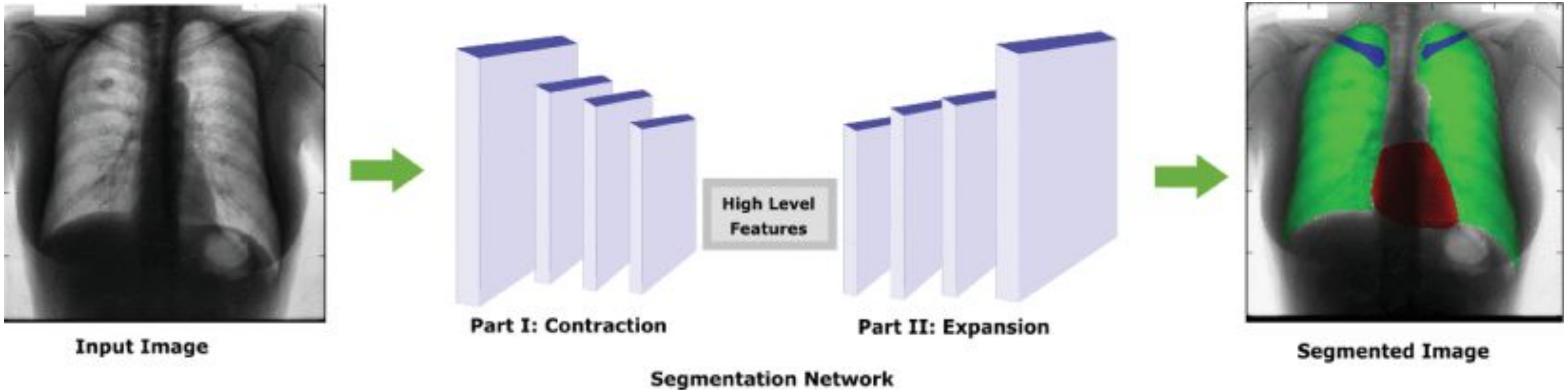


Note: a simple approach to segmentation can also be applying a classification CNN on image patches in a dense, sliding-window fashion (e.g. Ciresan et al.). But fully convolutional approaches such as U-Net generally achieve better performance.

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

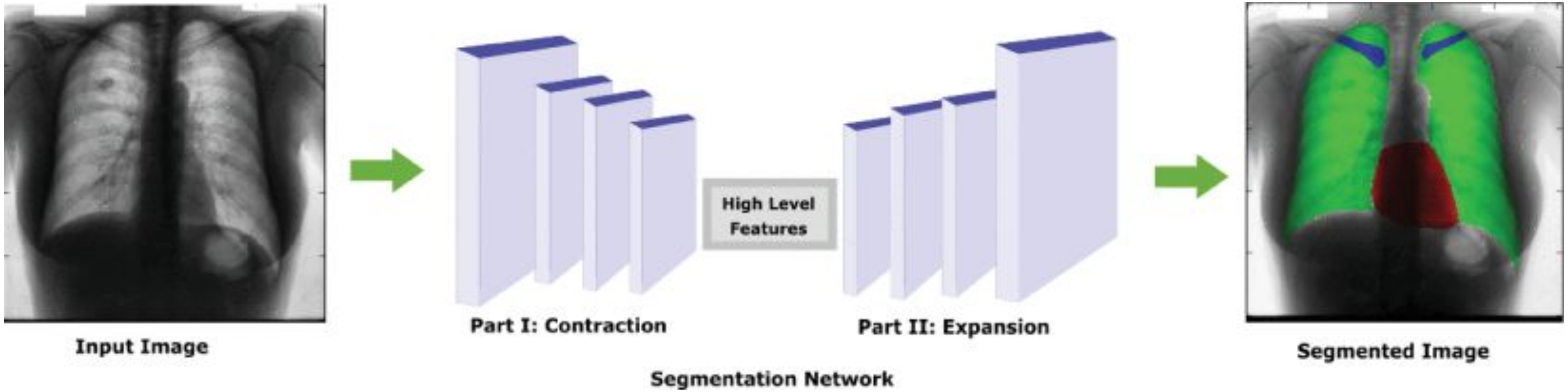


Novikov et al. Fully Convolutional Architectures for Multiclass Segmentation in Chest Radiographs. IEEE Trans. on Medical Imaging, 2018.

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)

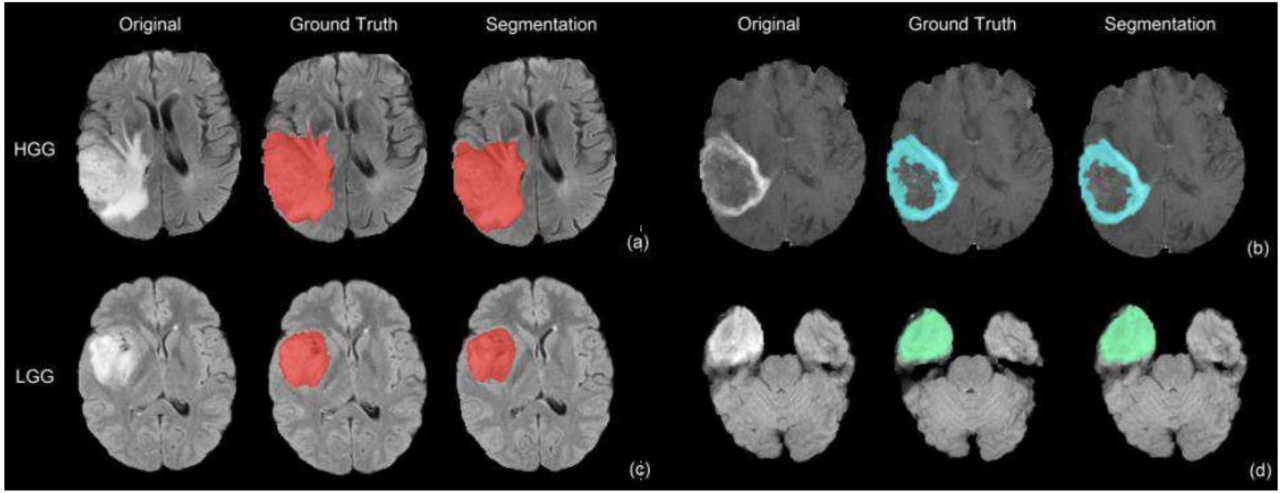
Body Part	Lungs		Clavicles		Heart	
Evaluation Metric	D	J	D	J	D	J
InvertedNet	0.972	0.946	**0.902**	**0.821**	0.935	0.879
All-Dropout	**0.973**	**0.948**	0.896	0.812	**0.941**	**0.888**
All-Convolutional	0.971	0.944	0.876	0.780	0.938	0.883
Original U-Net	0.971	0.944	0.880	0.785	0.938	0.883

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

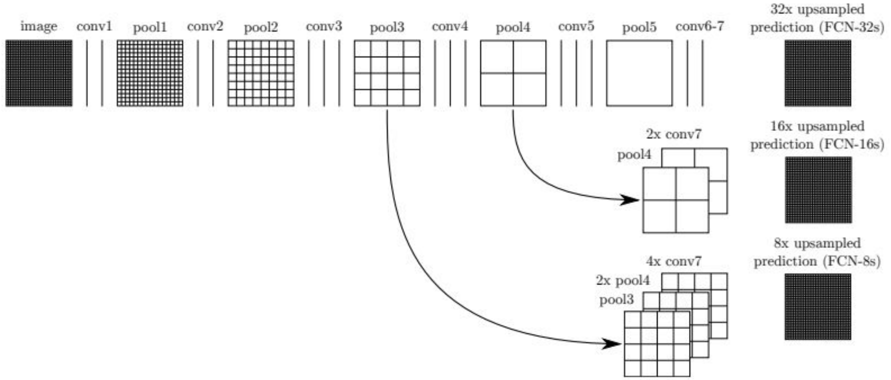


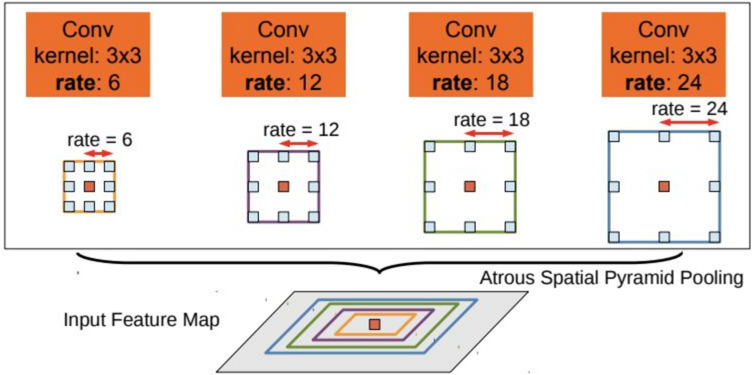
Dong et al. Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks. MIUA, 2017.

Other segmentation architectures

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

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





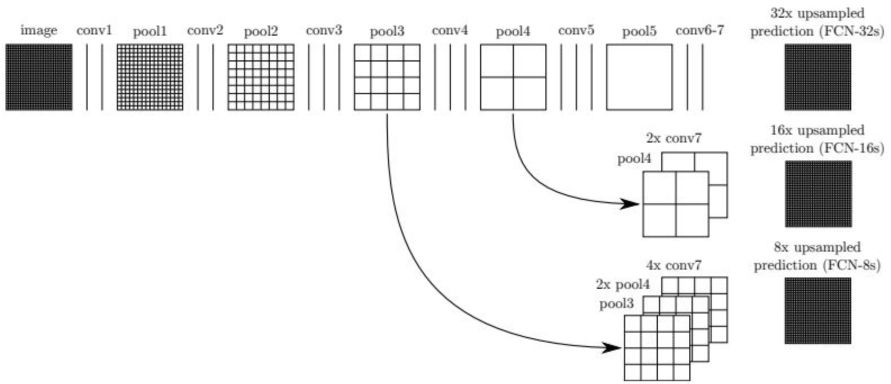
Shelhamer*, Long*, et al. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

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.

Other segmentation architectures

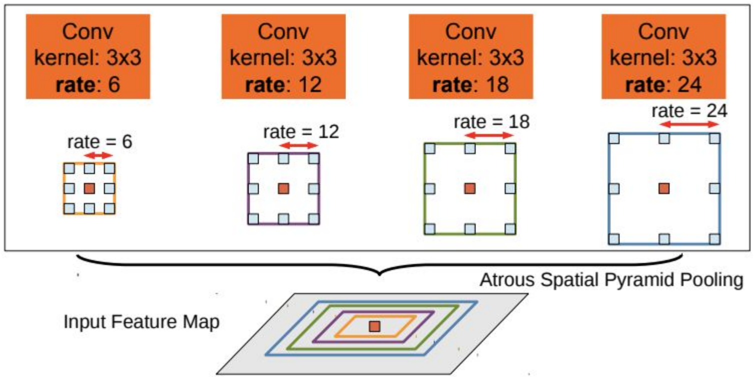
Can try DeepLab v3+ for segmentation projects!

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



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

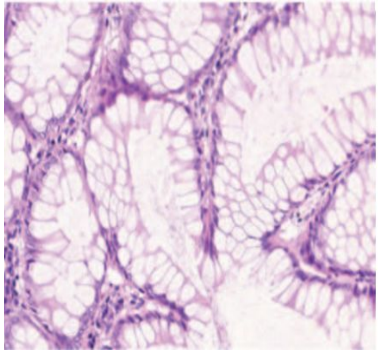
- **DeepLab (v1-v3+)**
- Uses "atrous convolutions" to control a filter's field of view
- Parallel atrous convolutions with different rates for multi-scale features



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.

Continuing today:
Richer visual recognition tasks: segmentation and detection

Classification

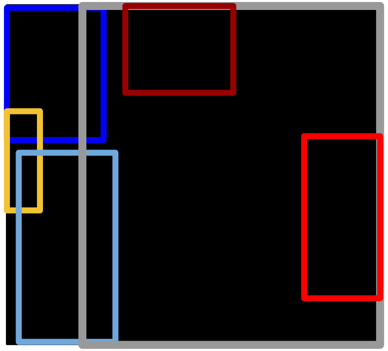


Output:
one category label for image (e.g., colorectal glands)

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

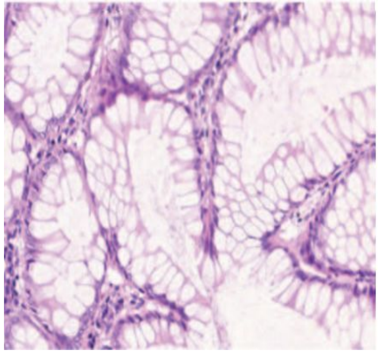
Output:
Category label and instance label for each pixel in the image

Distinguishes between different instances of an object

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

Continuing today:
Richer visual recognition tasks: segmentation and detection

Classification

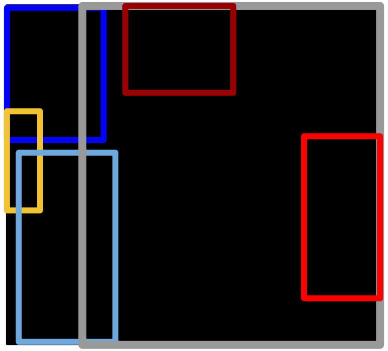


Output:
one category label for image (e.g., colorectal glands)

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

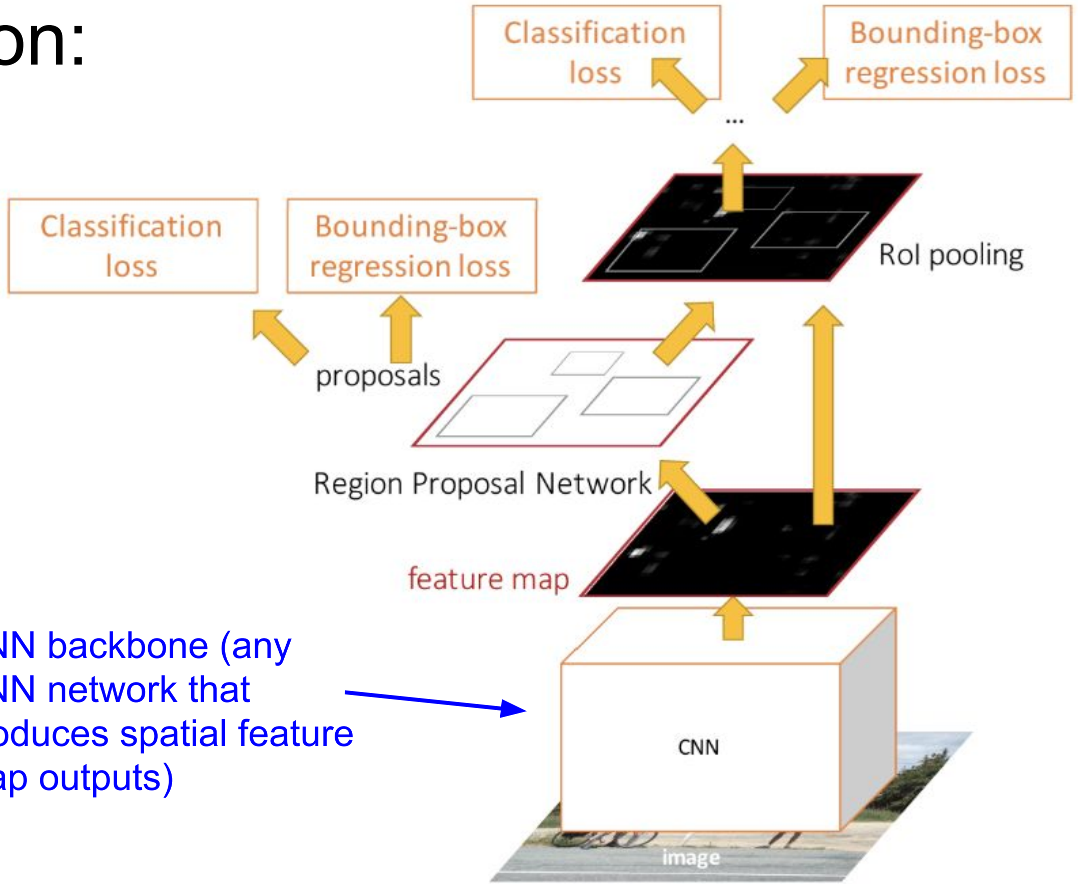
Instance Segmentation

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

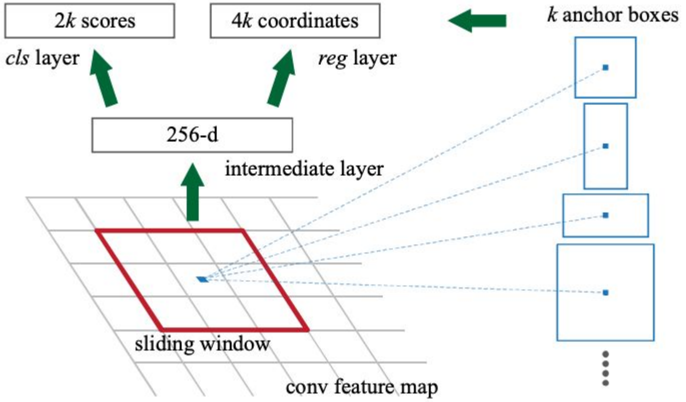
Object detection: Faster R-CNN

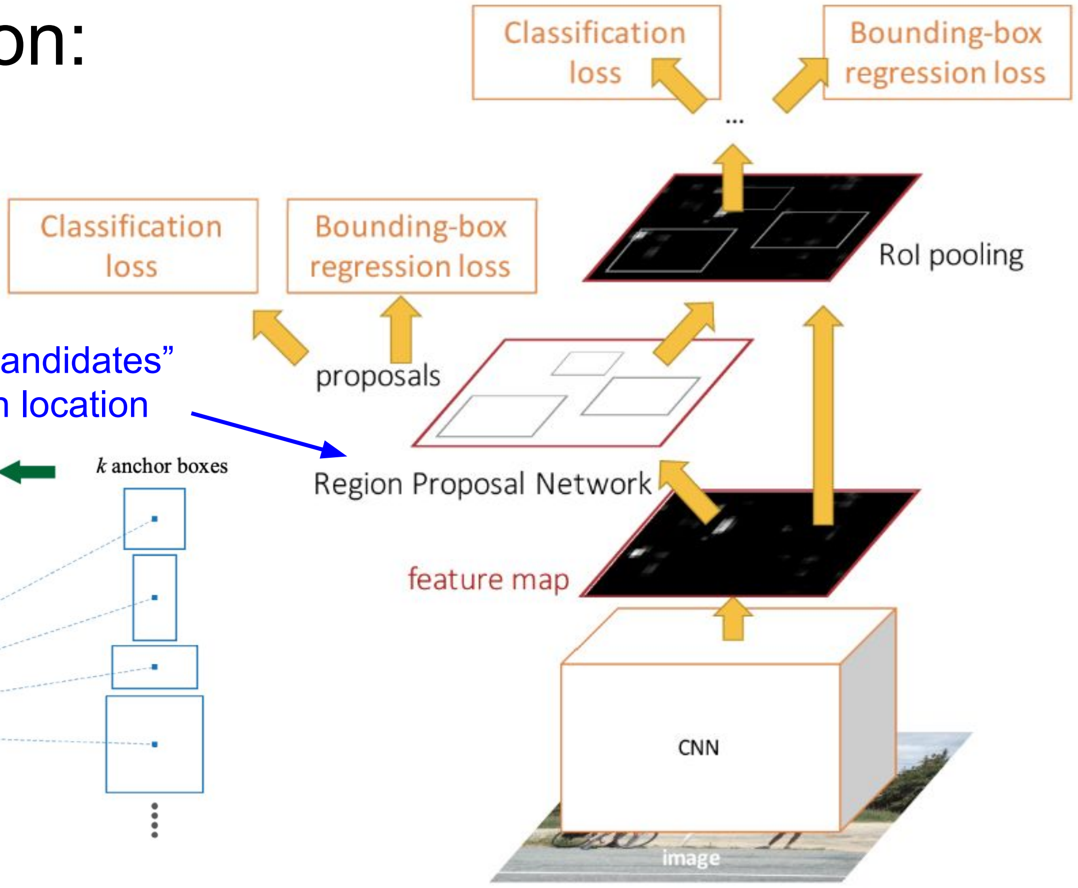


CNN backbone (any CNN network that produces spatial feature map outputs)

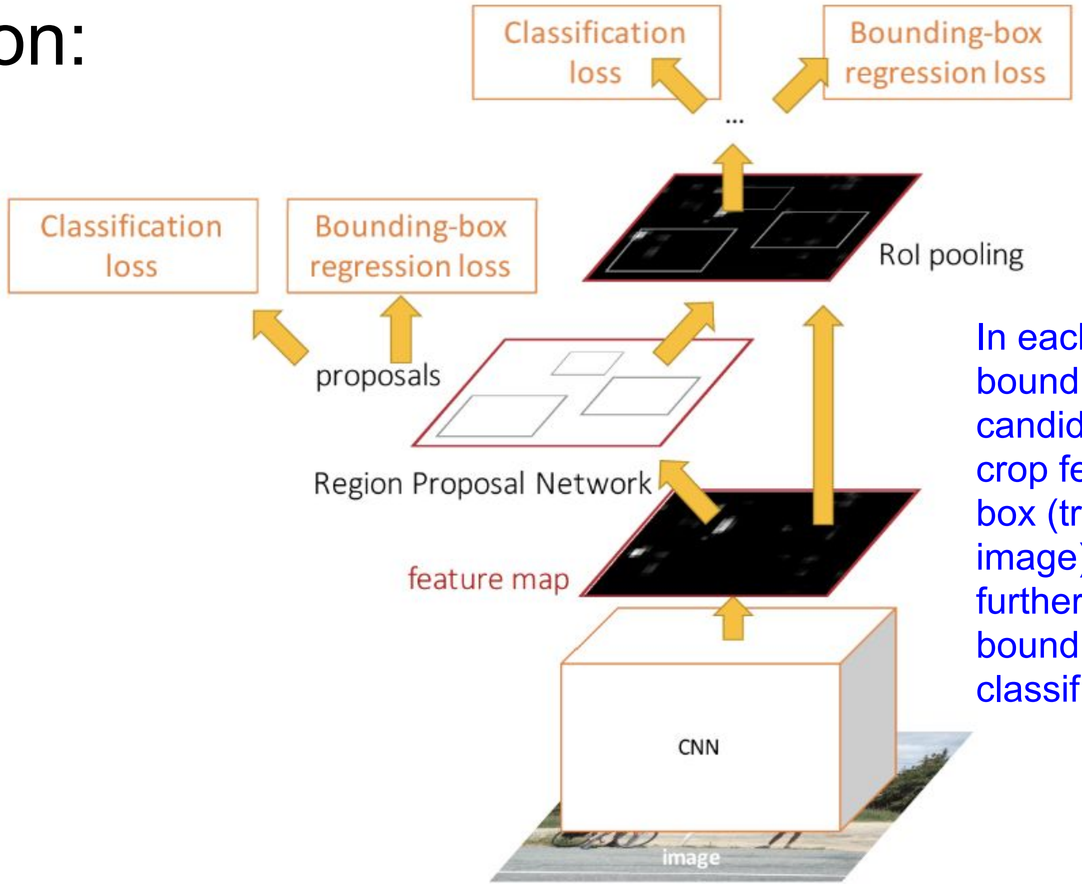
Object detection: Faster R-CNN

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





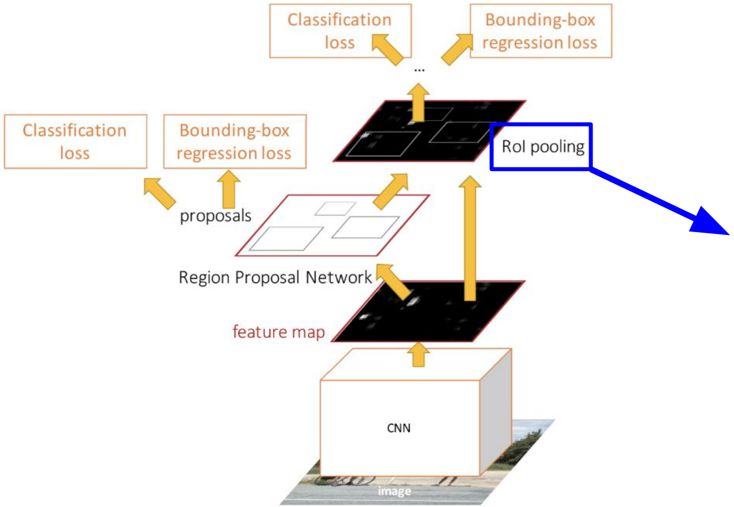
Object detection: Faster R-CNN



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

Max-pool within each subregion

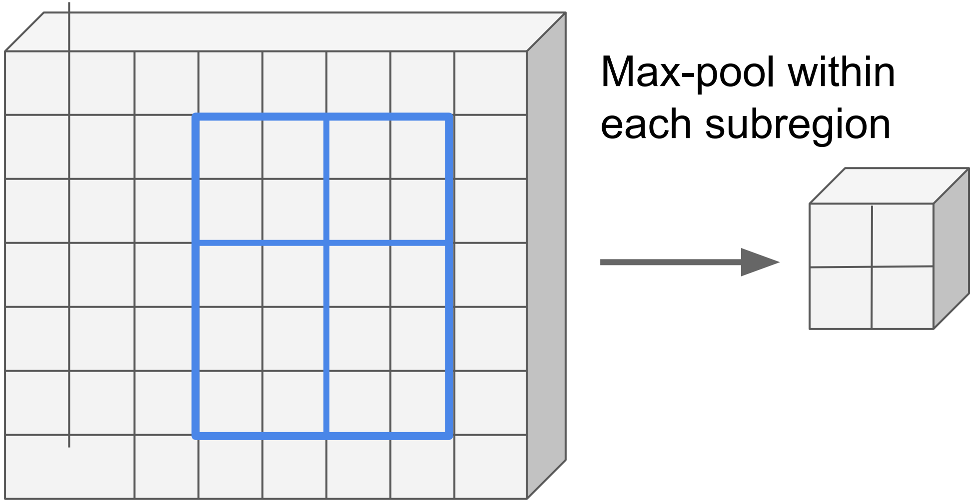


Image features

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

Obtain from model outputs

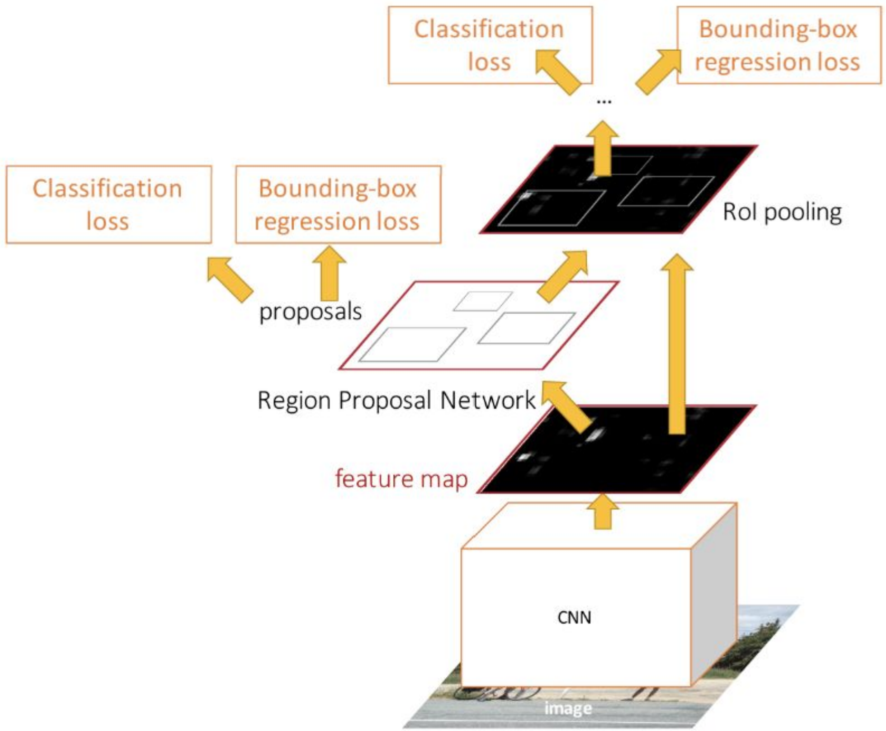
Standard output of object detection

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

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

Bounding box

Class confidence



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)

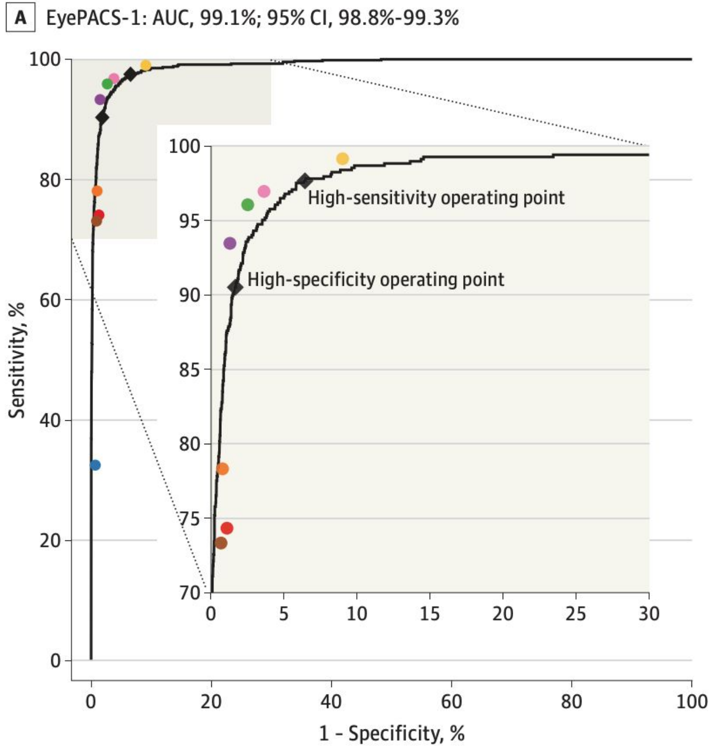
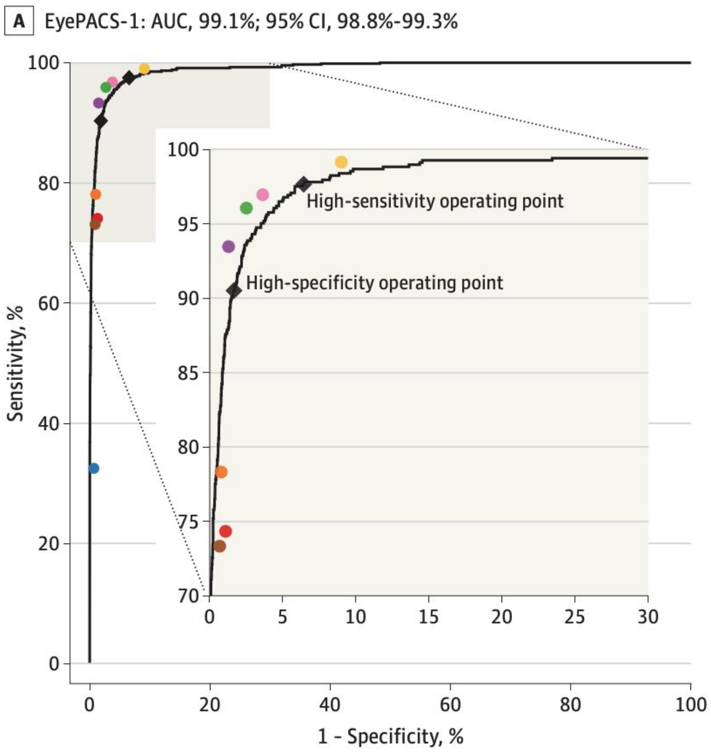


Figure credit: Gulshan et al. 2016

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	Prediction 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)

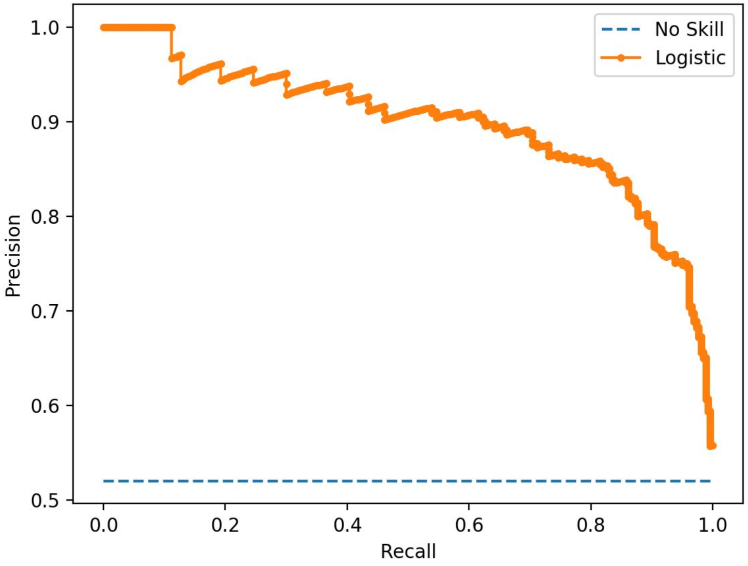


Figure credit: https://3qeqpr26caki16dnhd19sv6by6v-wpengine.netdna-ssl.com/wp-content/uploads/2018/08/Precision-Recall-Plot-for-a-No-Skill-Classifier-and-a-Logistic-Regression-Model4.png

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

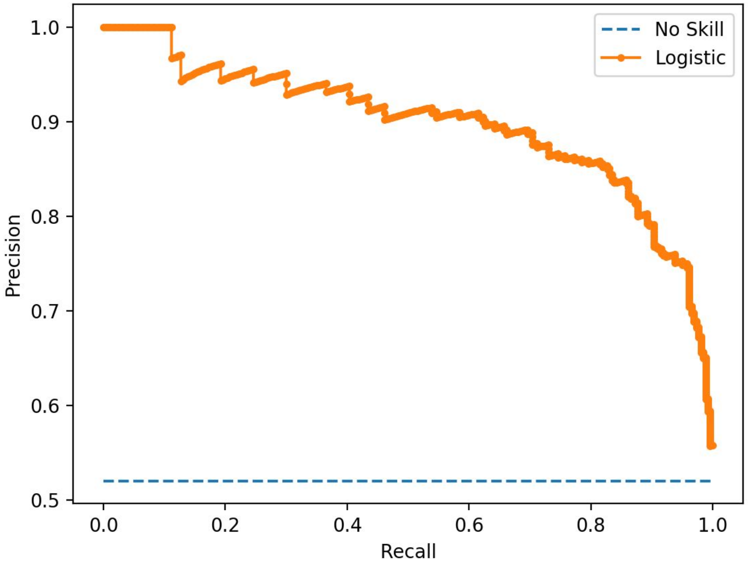
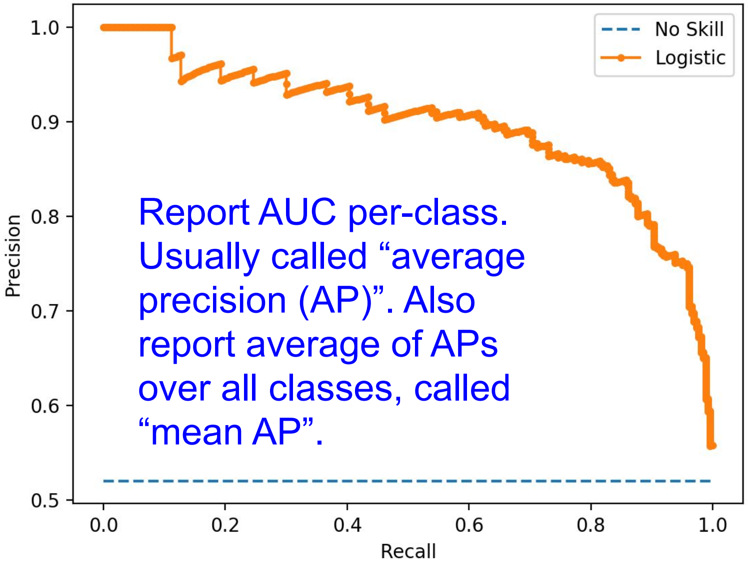


Figure credit: https://3qeqpr26caki16dnhd19sv6by6v-wpengine.netdna-ssl.com/wp-content/uploads/2018/08/Precision-Recall-Plot-for-a-No-Skill-Classifier-and-a-Logistic-Regression-Model4.png

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

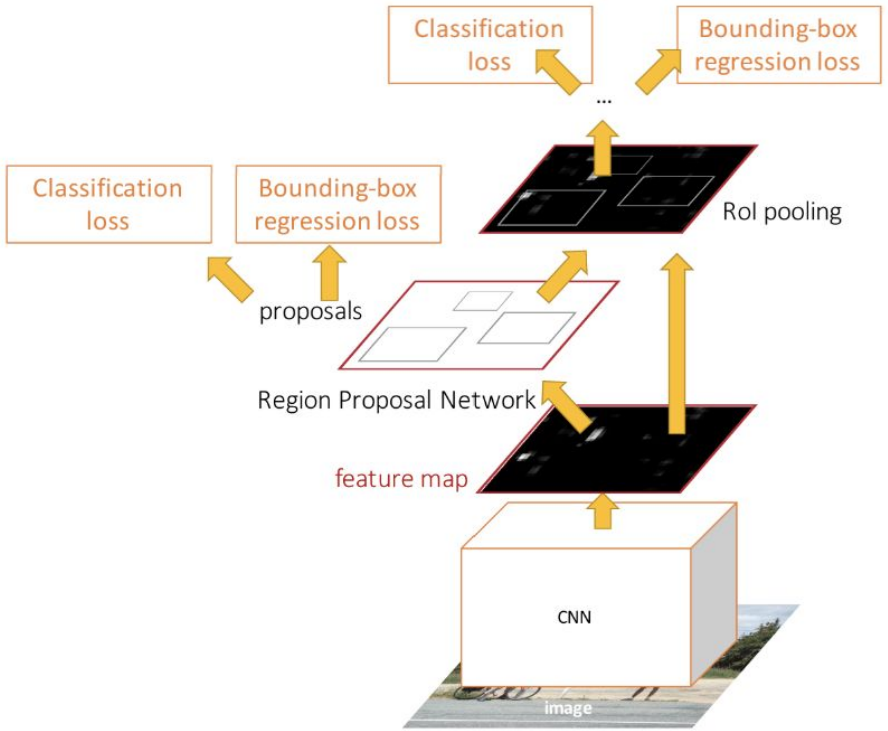
Standard output of object detection

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



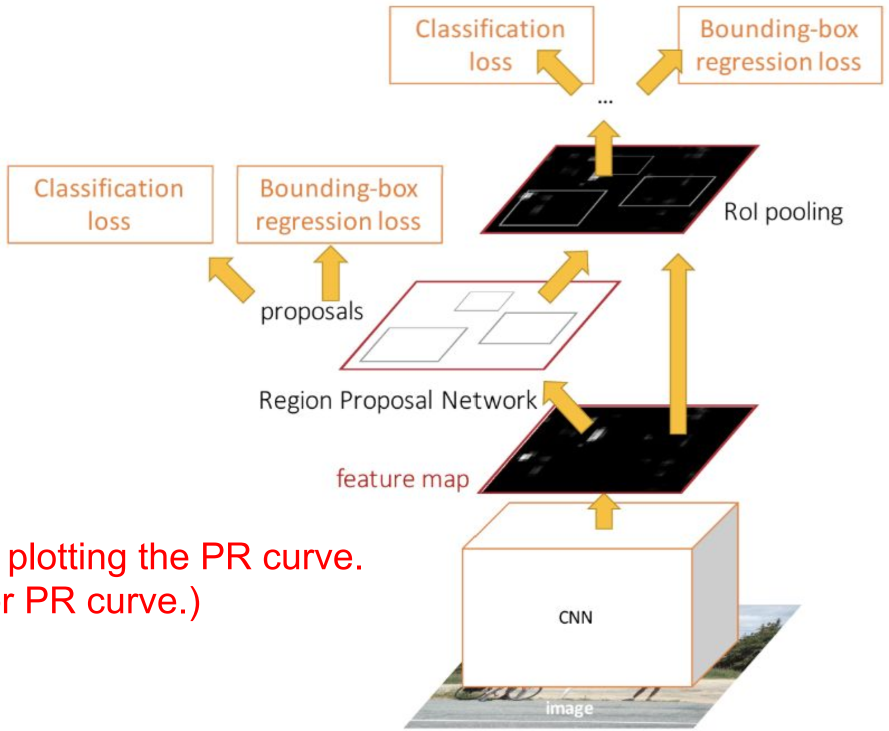
Evaluation of object detection

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 or FP? (note TN,FN, not used for PR curve.)

Evaluation of object detection

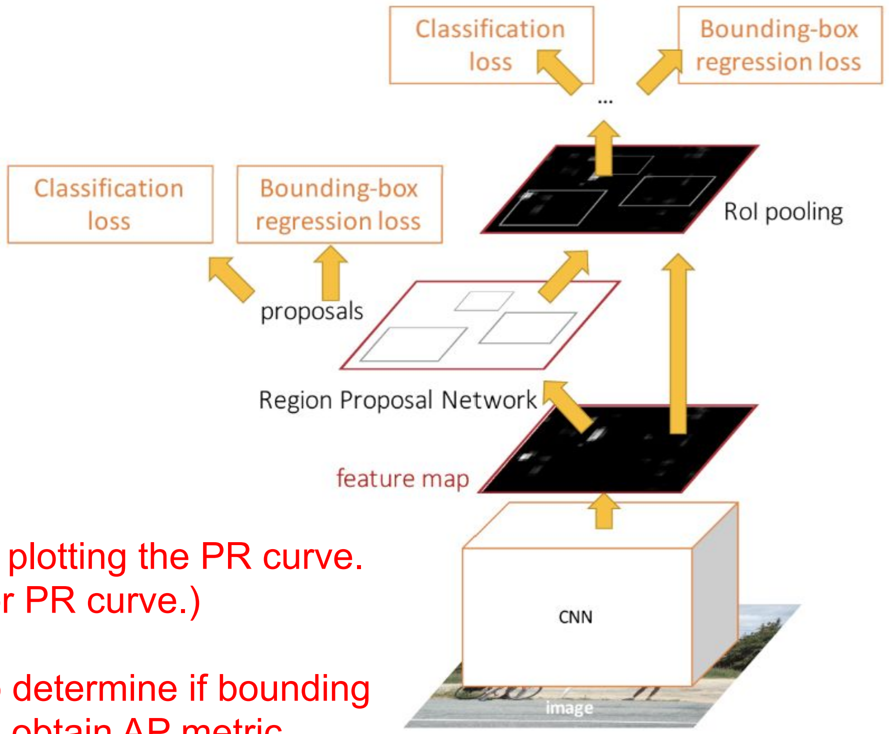
Standard output of object detection

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



We have the class confidences to vary the threshold in plotting the PR curve. But how do we get TP or FP? (note TN,FN, not used for PR curve.)

A: Choose an IOU threshold with ground truth boxes to determine if bounding box prediction is TP or FP. 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., $(\underbrace{x, y, h, w}_{\text{Bounding box}}, c)$

Bounding box

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 or FP? (note TN,FN, not used for PR curve.)

A: Choose an IOU threshold with ground truth boxes to determine if bounding box prediction is TP or FP. 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

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**

We have the class confidences to vary the threshold in plotting the PR curve.
But how do we get TP or FP? (note TN,FN, not used for PR curve.)

A: Choose an IOU threshold with ground truth boxes to determine if bounding box prediction is TP or FP. 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

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**

We have the class confidences to vary the threshold in plotting the PR curve. But how do we get TP or FP? (note TN,FN, not used for PR curve.)

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

If IOU threshold not specified in experiments description for a paper, may need to look in dataset evaluation documentation. Default is often 0.5 or [.5,.95].

Evaluation of object detection

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].

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

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

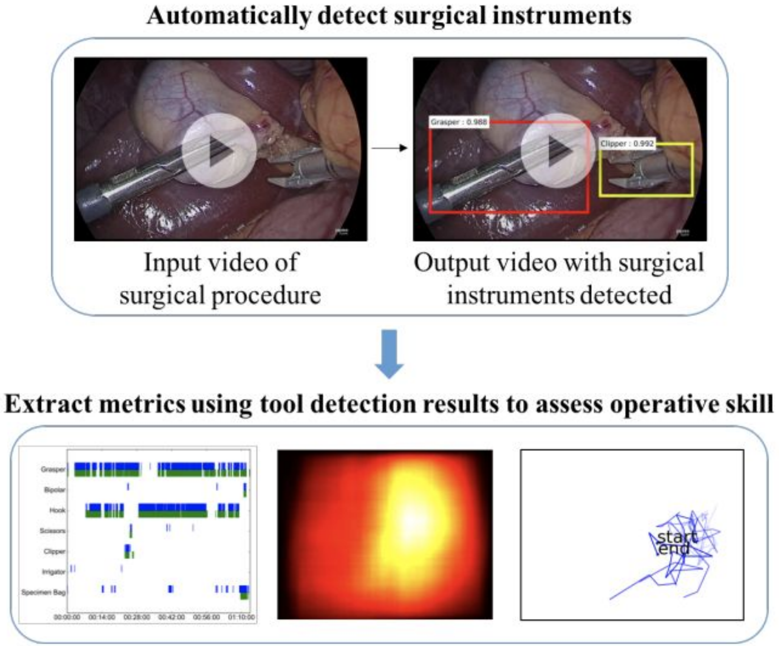
We have the class confidences to vary the threshold in plotting the PR curve. But how do we get TP or FP? (note TN,FN, not used for PR curve.)

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

If IOU threshold not specified in experiments description for a paper, may need to look in dataset evaluation documentation. Default is often 0.5 or [.5,.95].

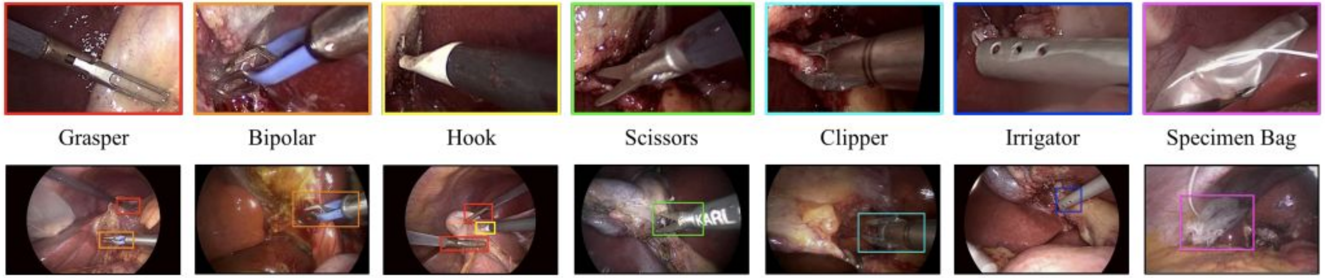
Jin et al. 2018

- 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.



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

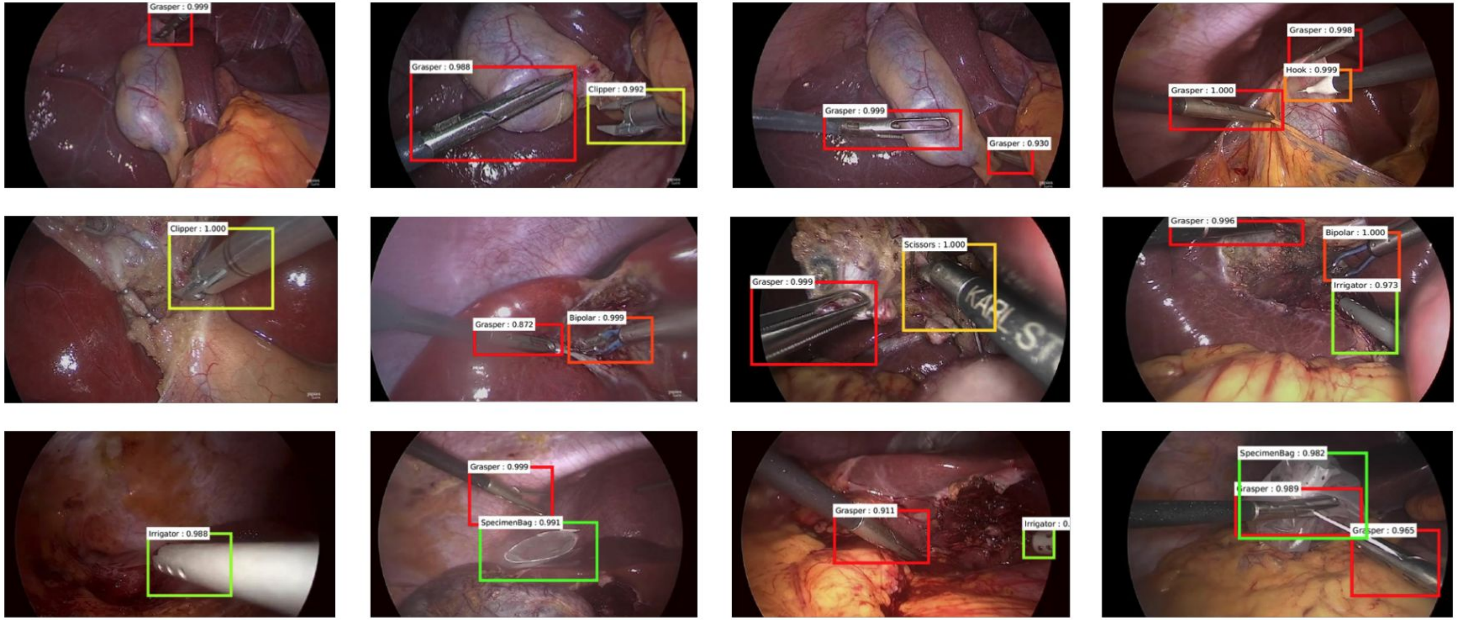
Jin et al. 2018



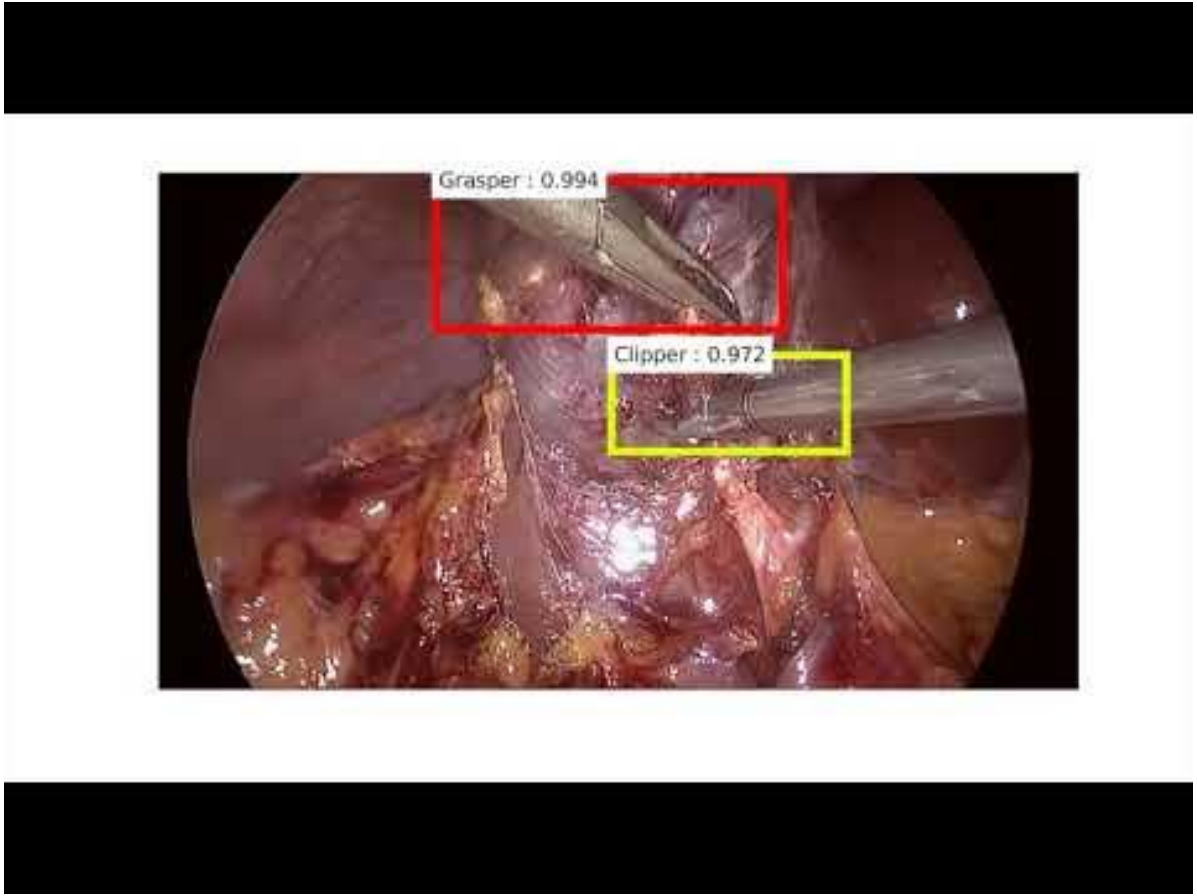
Tool	AP
Grasper	48.3
Bipolar	67.0
Hook	78.4
Scissors	67.7
Clipper	86.3
Irrigator	17.5
Specimen Bag	76.3
mAP	63.1

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.

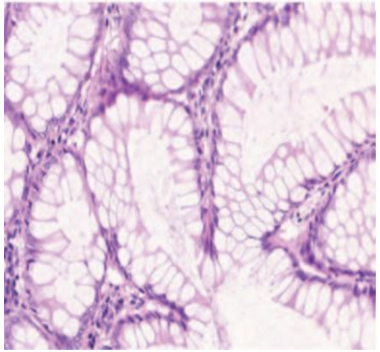
Other object detection architectures

- 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

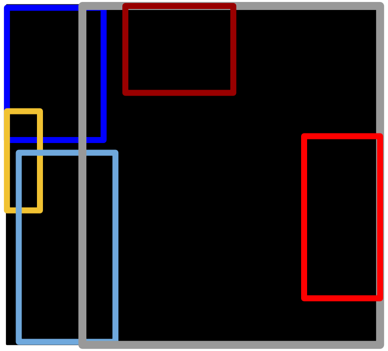


Output:
one category label for image (e.g., colorectal glands)

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

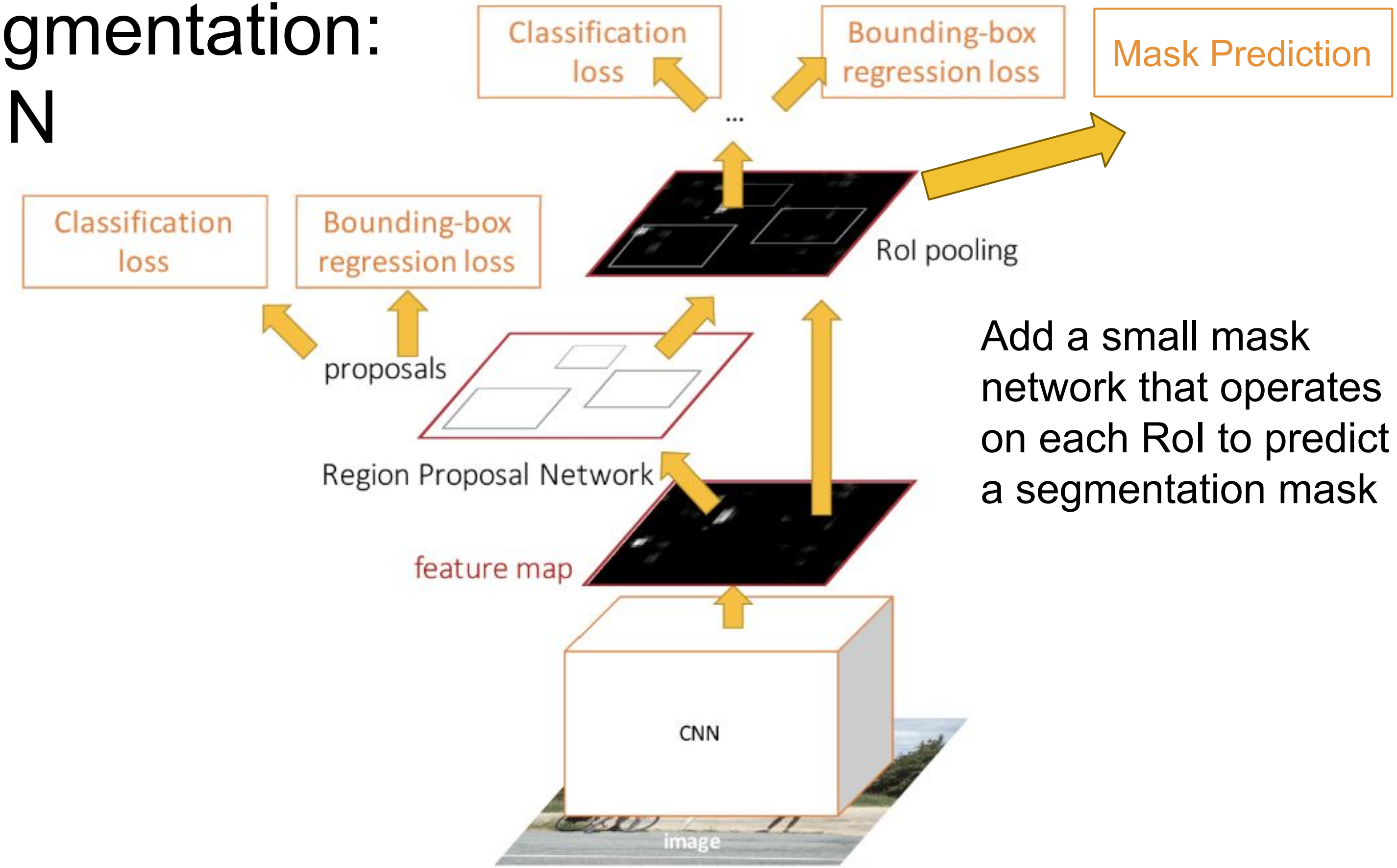
Instance Segmentation

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!

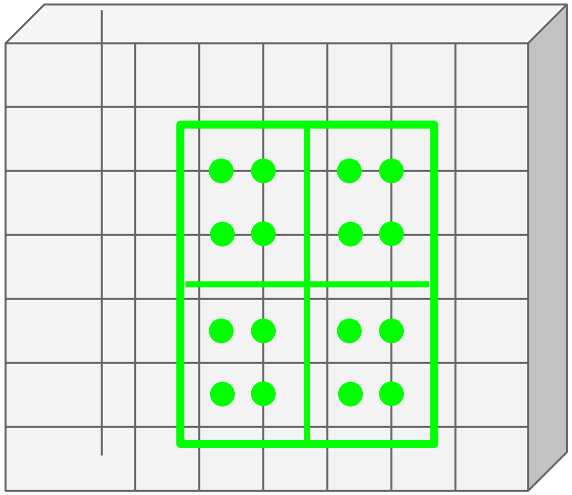


Image features
(e.g. 512 x 20 x 15)

Cropping Features: RoI *Align*

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!

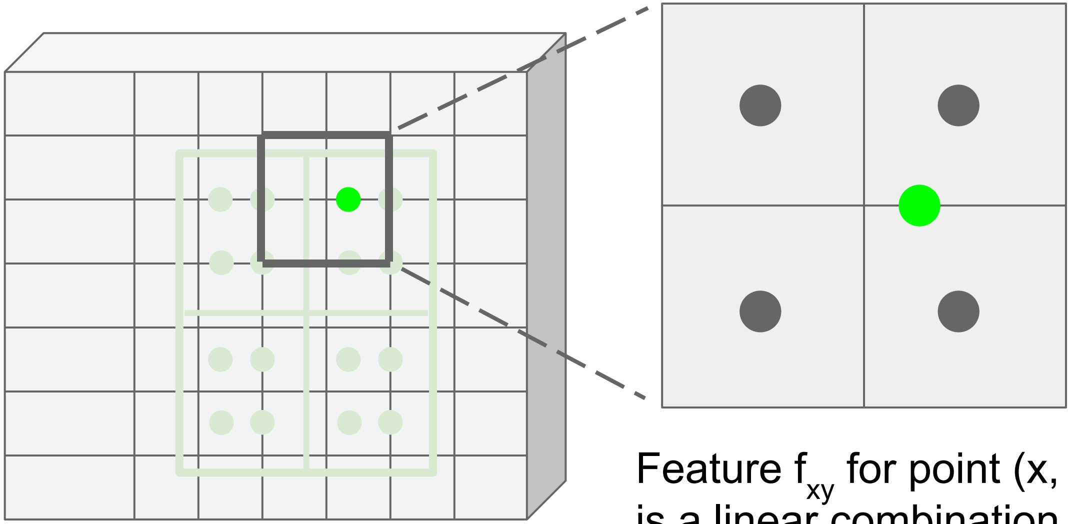


Image features

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

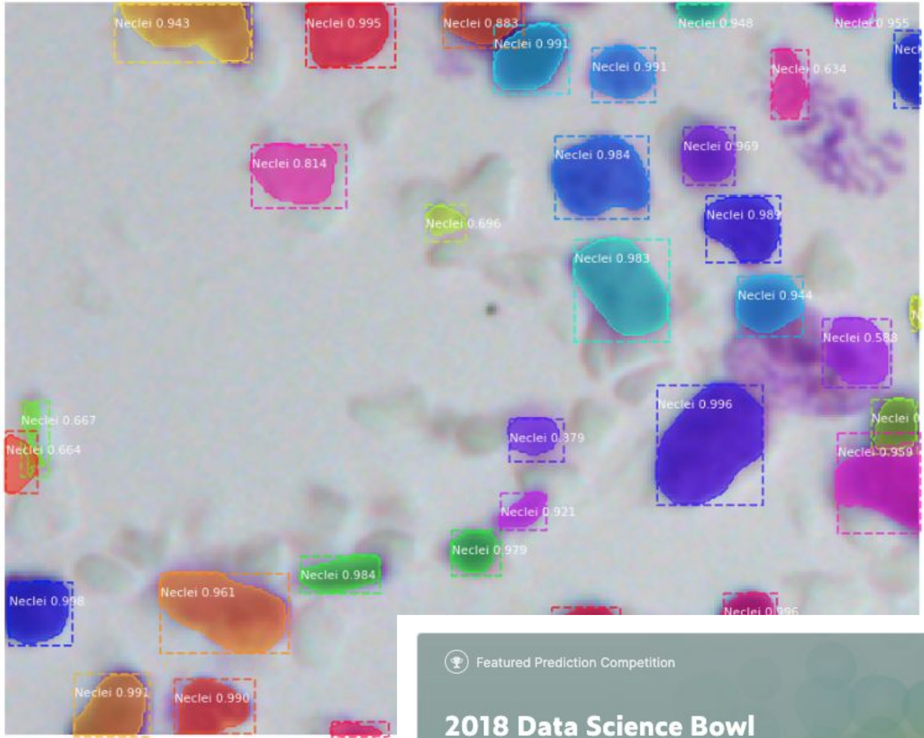
Average AP over different IOU thresholds

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

AP for small, medium, large objects

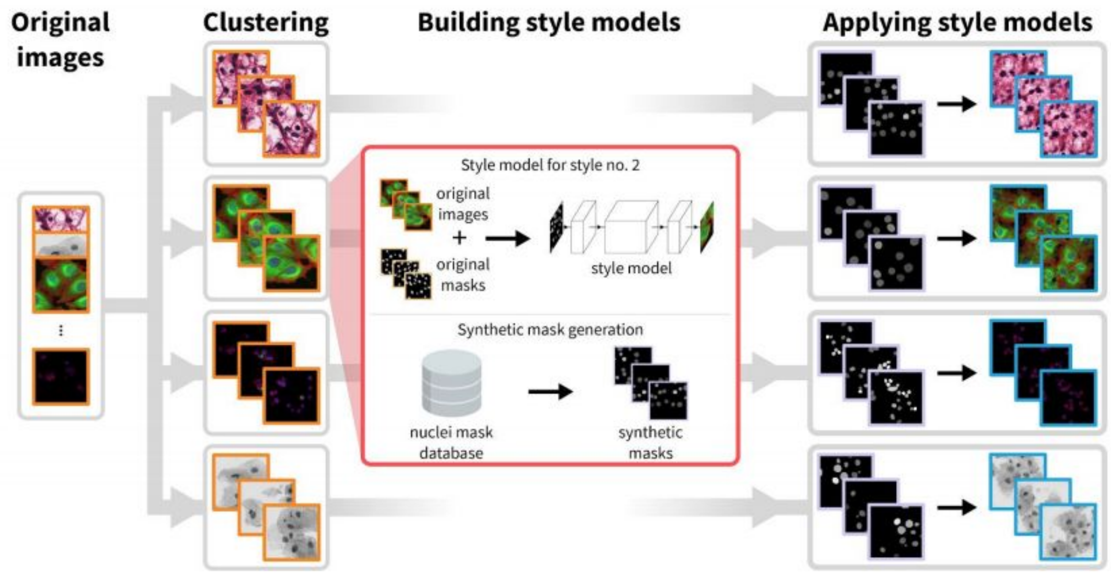
	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	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



Many interesting extensions

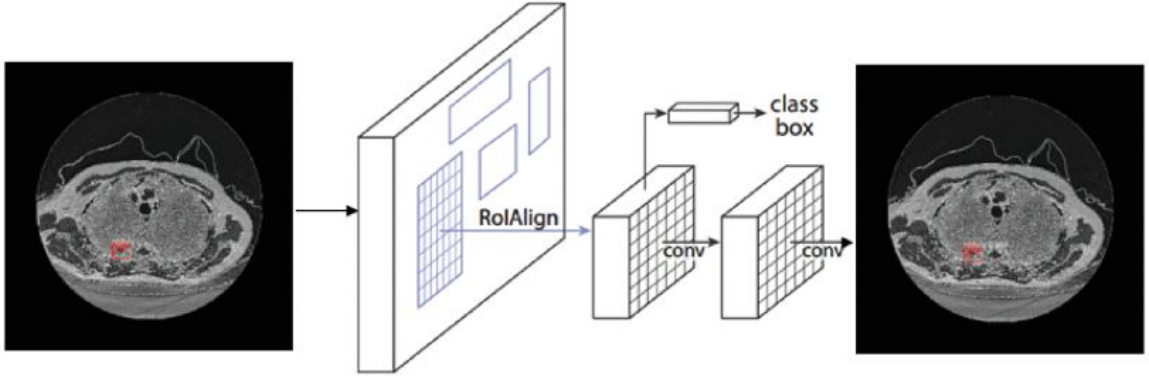
- E.g. Hollandi et al. 2019
    - Used “style transfer” approaches for rich data augmentation
    - Refined Mask-RCNN instance segmentation results with further U-Net-based boundary refinement



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 next

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

Where we are

First topic: medical image classification

Then: Beyond classification to richer visual recognition tasks

- Semantic segmentation (last lecture)
- Object detection (today)
- Instance segmentation (today)

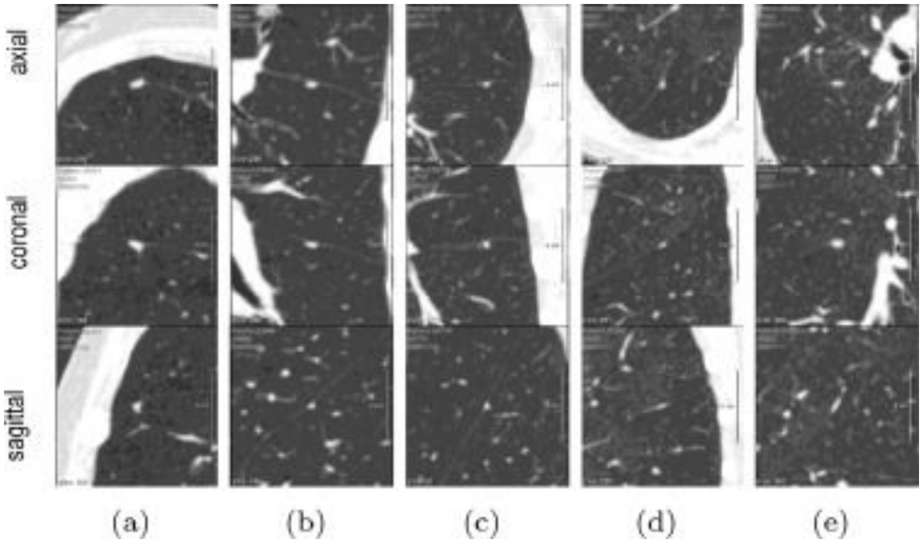
Next topic: Advanced vision models (3D and video)

Next Topic:
Advanced Vision Models for Higher-Dimensional (3D and Video) Data

How do we handle 3D data?

Recall: Ciompi et al. 2015

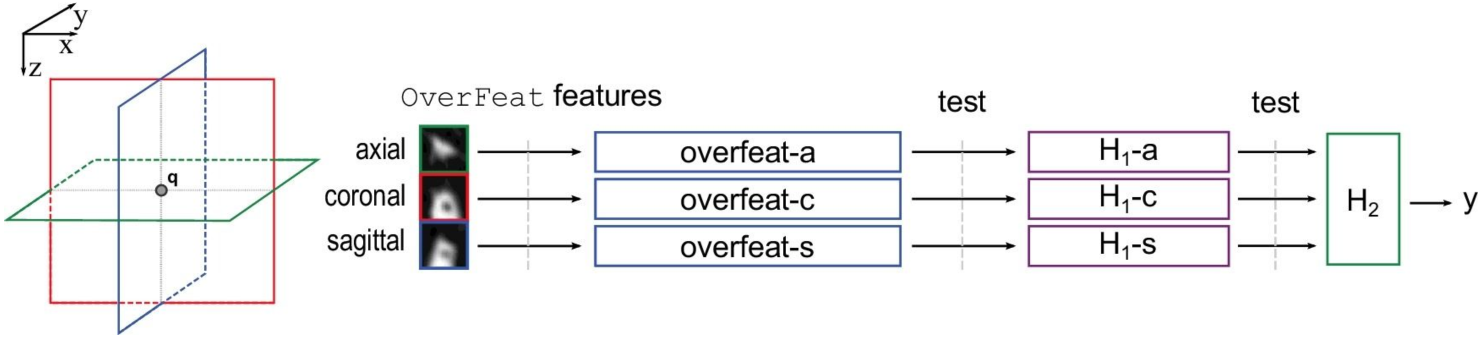
- Task: classification of lung nodules in 3D CT scans as peri-fissural nodules (PFN, likely to be benign) or not
- Dataset: 568 nodules from 1729 scans at a single institution. (65 typical PFNs, 19 atypical PFNs, 484 non-PFNs).
- Data pre-processing: prescaling from CT hounsfield units (HU) into [0,255]. 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).

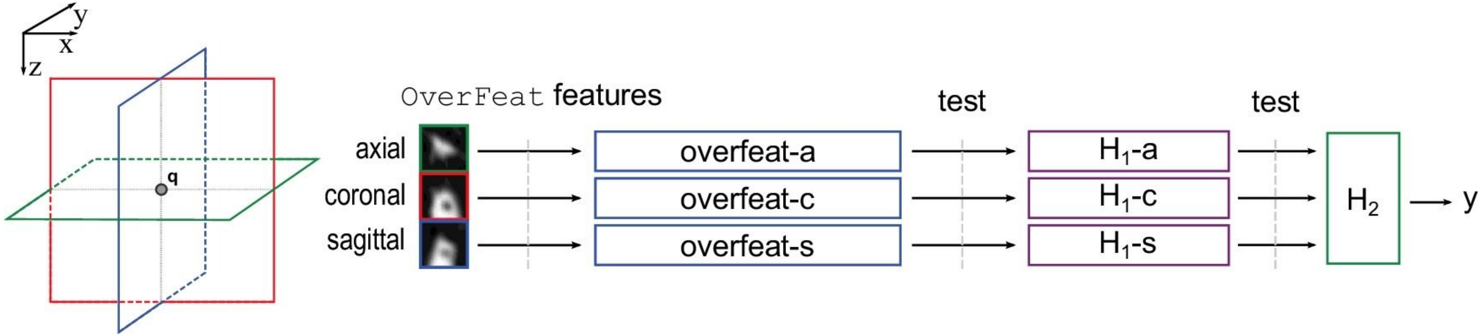


Ciompi et al. Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. Medical Image Analysis, 2015.

Ciompi et al. 2015

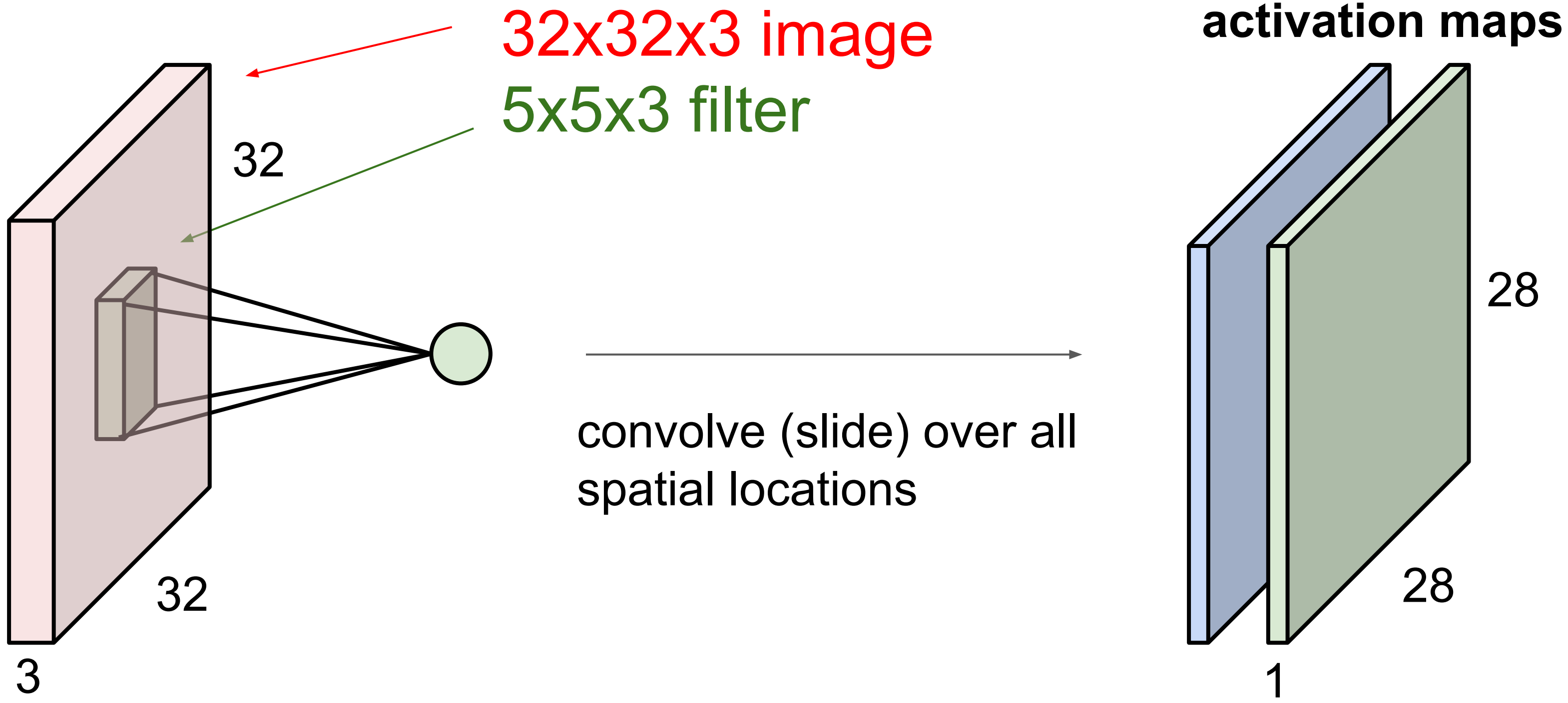
Another approach: 3D CNNs!

- 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).



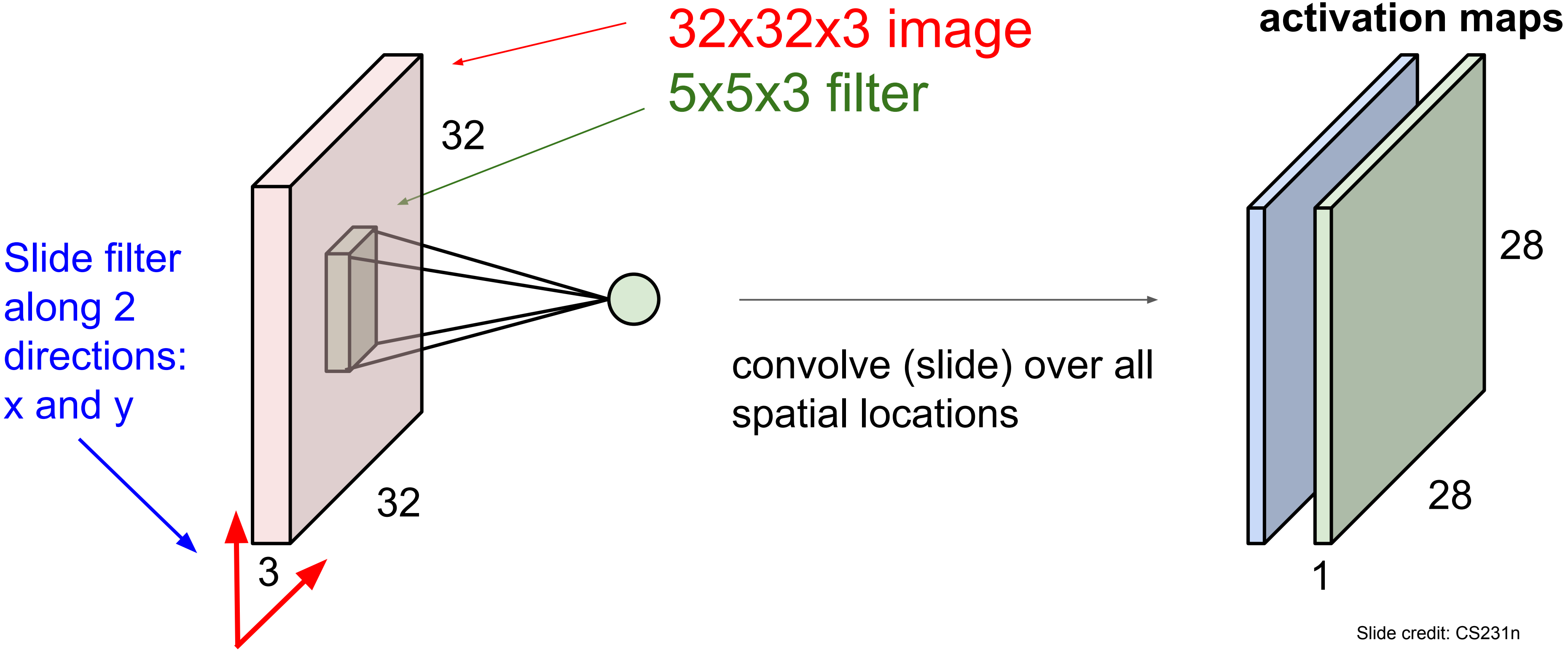
Ciompi et al. Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. Medical Image Analysis, 2015.

Remember 2D convolutions



Slide credit: CS231n

Remember 2D convolutions



Slide credit: CS231n

3D convolutions

Slide filter along **3** directions: x, y, and z!

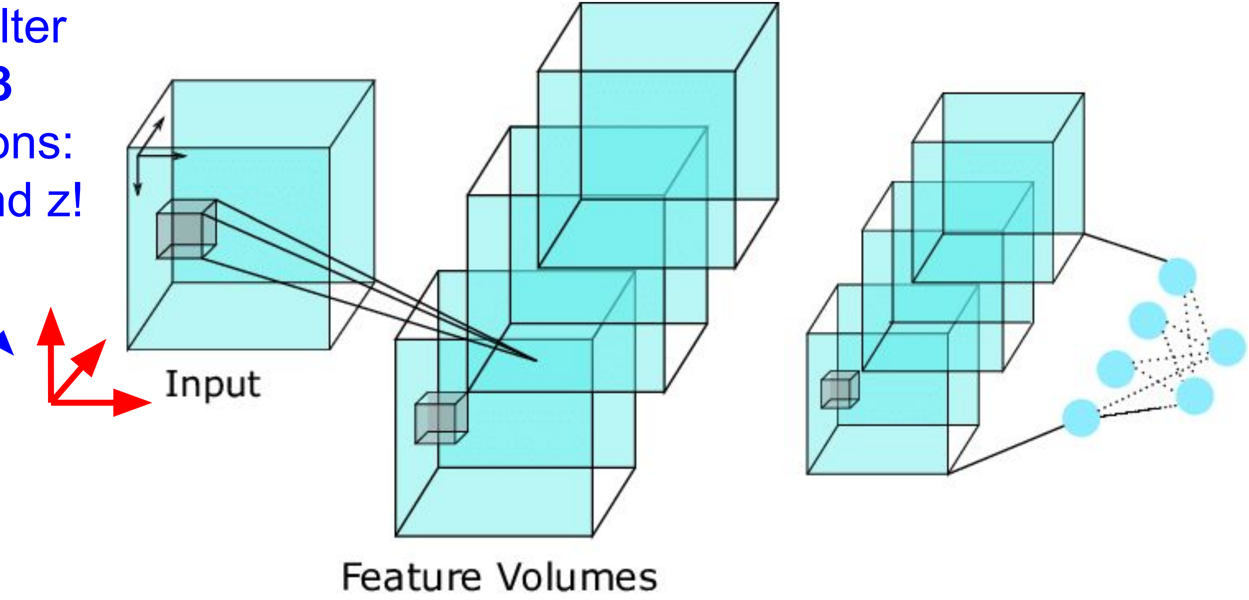


Figure credit:
https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@15494 74645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

3D convolutions

When might you use 3D convolutions?

Slide filter along **3** directions: x, y, and z!

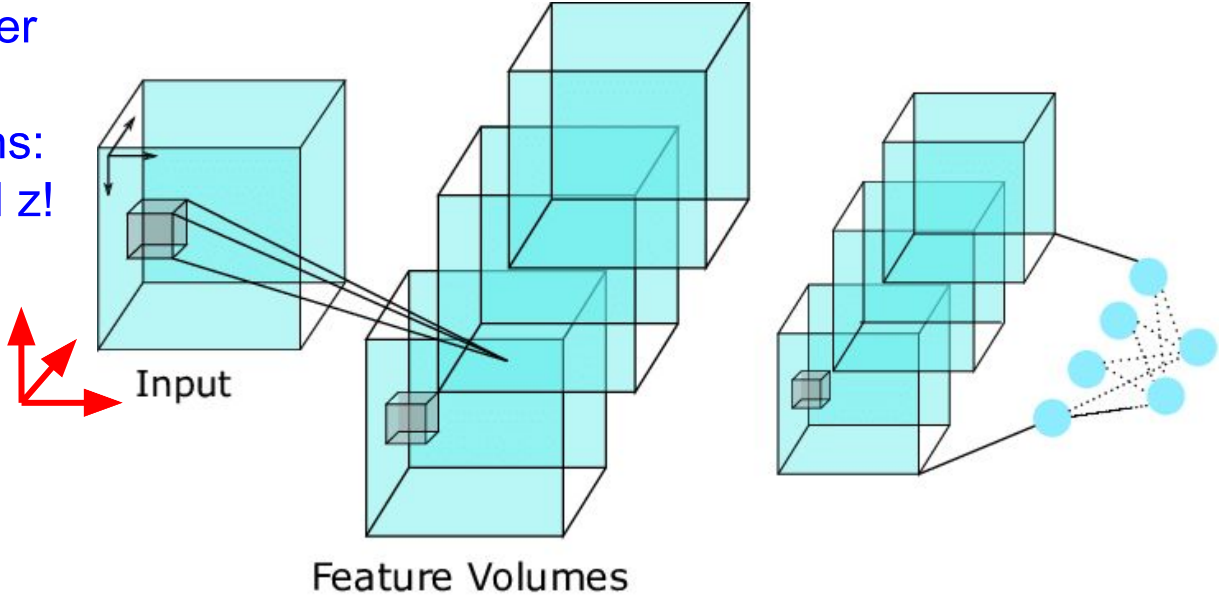


Figure credit:
https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

3D convolutions

When might you use 3D convolutions?

Ex: 224 x 224 x 1 x 256 3D CT scan (with 256 slices)

Slide filter along **3** directions: x, y, and z!

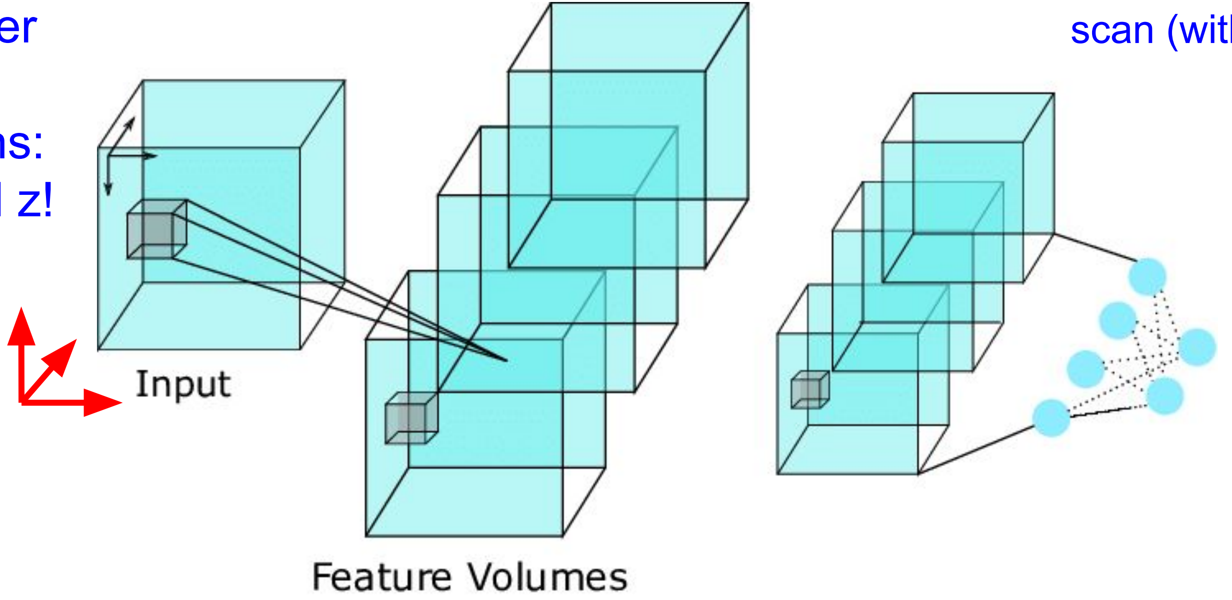
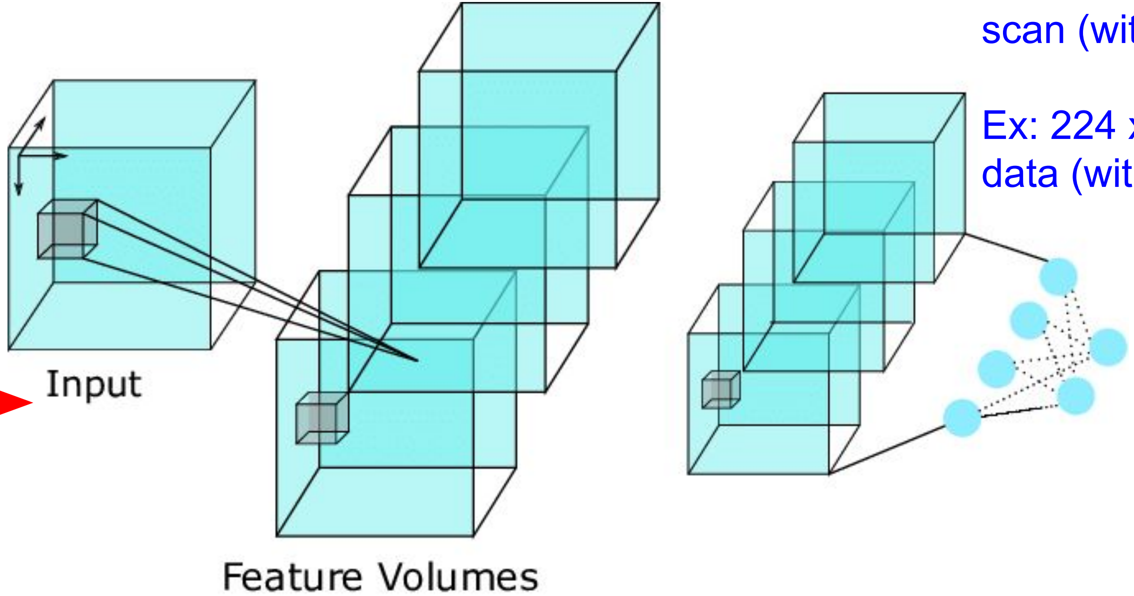


Figure credit:
https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

3D convolutions

Slide filter along **3** directions: x, y, and z!



When might you use 3D convolutions?

Ex: 224 x 224 x 1 x 256 3D CT scan (with 256 slices)

Ex: 224 x 224 x 3 x 500 video data (with 500 temporal frames)

Figure credit:
https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

3D convolutions

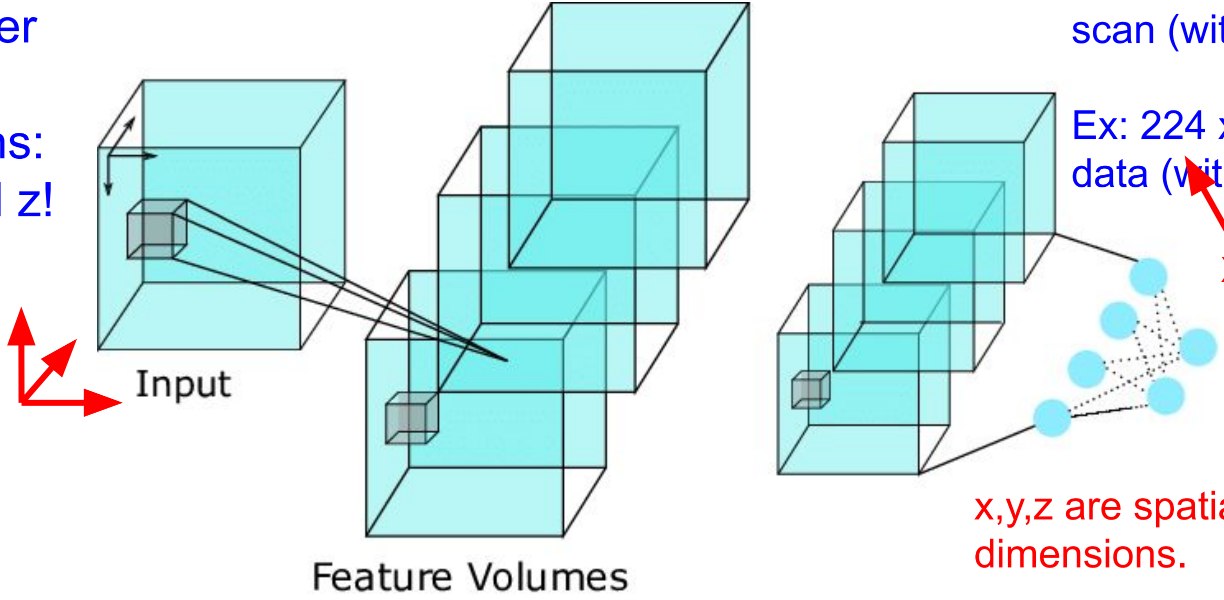
When might you use 3D convolutions?

Ex: 224 x 224 x 1 x 256 3D CT scan (with 256 slices)

Ex: 224 x 224 x 3 x 500 video data (with 500 temporal frames)

x y z

Slide filter along **3** directions: x, y, and z!

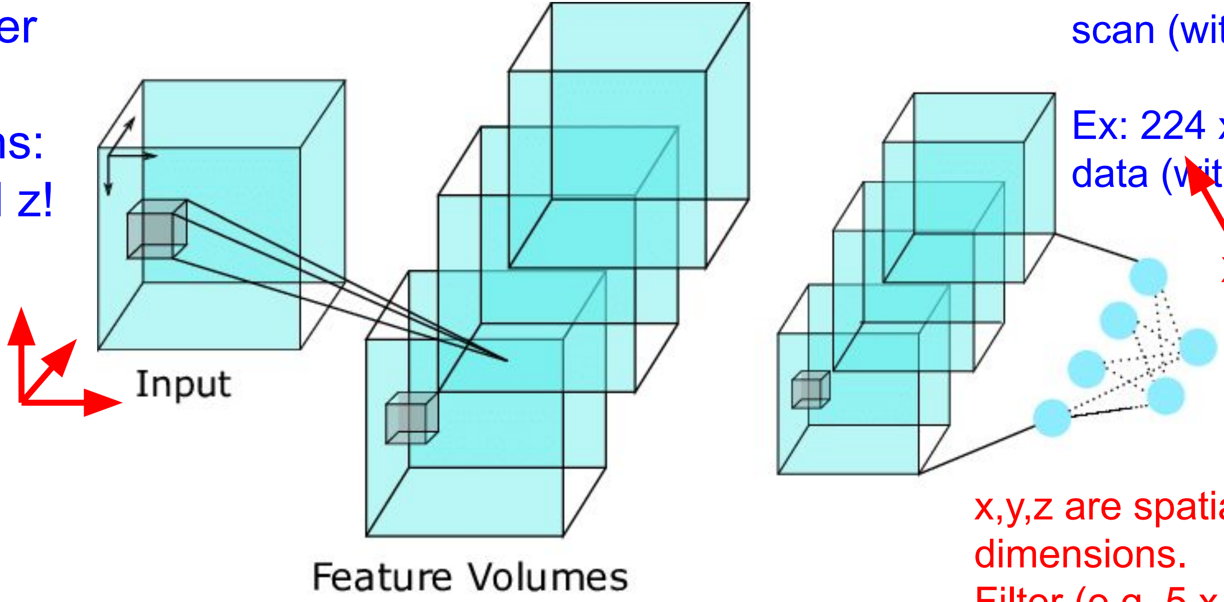


x,y,z are spatial and/or temporal dimensions.

Figure credit:
https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

3D convolutions

Slide filter along **3** directions: x, y, and z!



When might you use 3D convolutions?

Ex: 224 x 224 x 1 x 256 3D CT scan (with 256 slices)

Ex: 224 x 224 x 3 x 500 video data (with 500 temporal frames)

x y z

channels (e.g. R,G,B)

x,y,z are spatial and/or temporal dimensions.
Filter (e.g. 5 x 5 x 3 x 10 filter) goes all the way through the "channels" dimension as before.

Figure credit: https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

Now: 3D CNNs for lung nodule classification

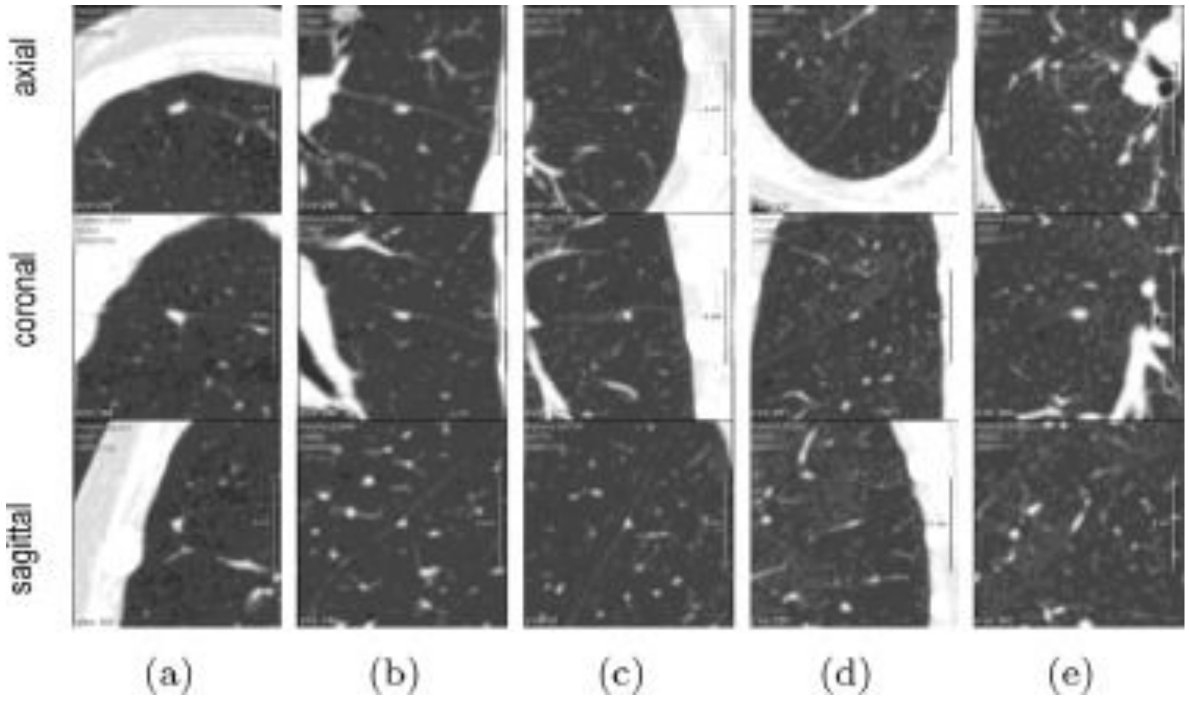
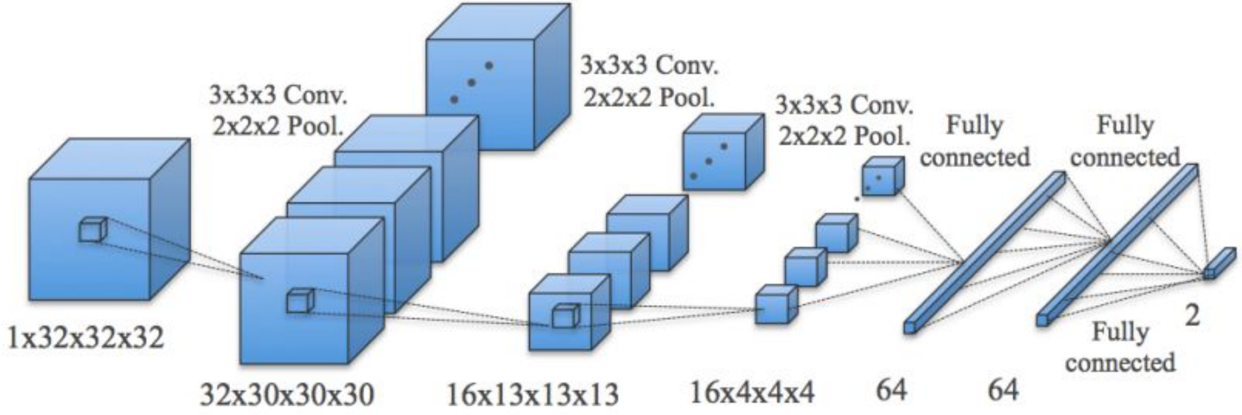


Figure credit: Ciompi et al. Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. Medical Image Analysis, 2015.

Huang et al. 2017

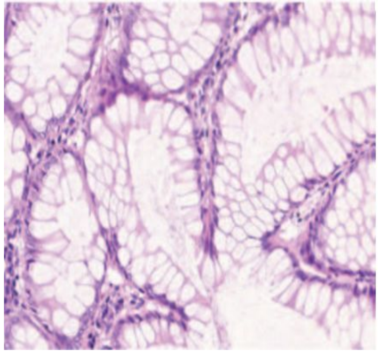
- Simple 3D CNN for lung nodule classification
- Used image processing approaches to extract candidate nodules, then 3D CNN to classify the surrounding volume
- Used the Lung Image Database Consortium (LIDC) Dataset, with 99 3D CT scans



Huang et al. Lung Nodule Detection in CT Using 3D Convolutional Neural Networks. ISBI 2017.

For richer visual recognition tasks, can also extend respective CNN architectures to use 3D convolutions

Classification

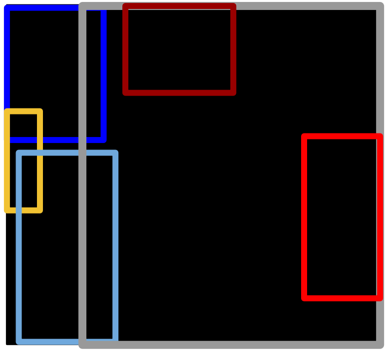


Output:
one category label for image (e.g., colorectal glands)

Semantic Segmentation

Output:
category label for each pixel in the image

Detection



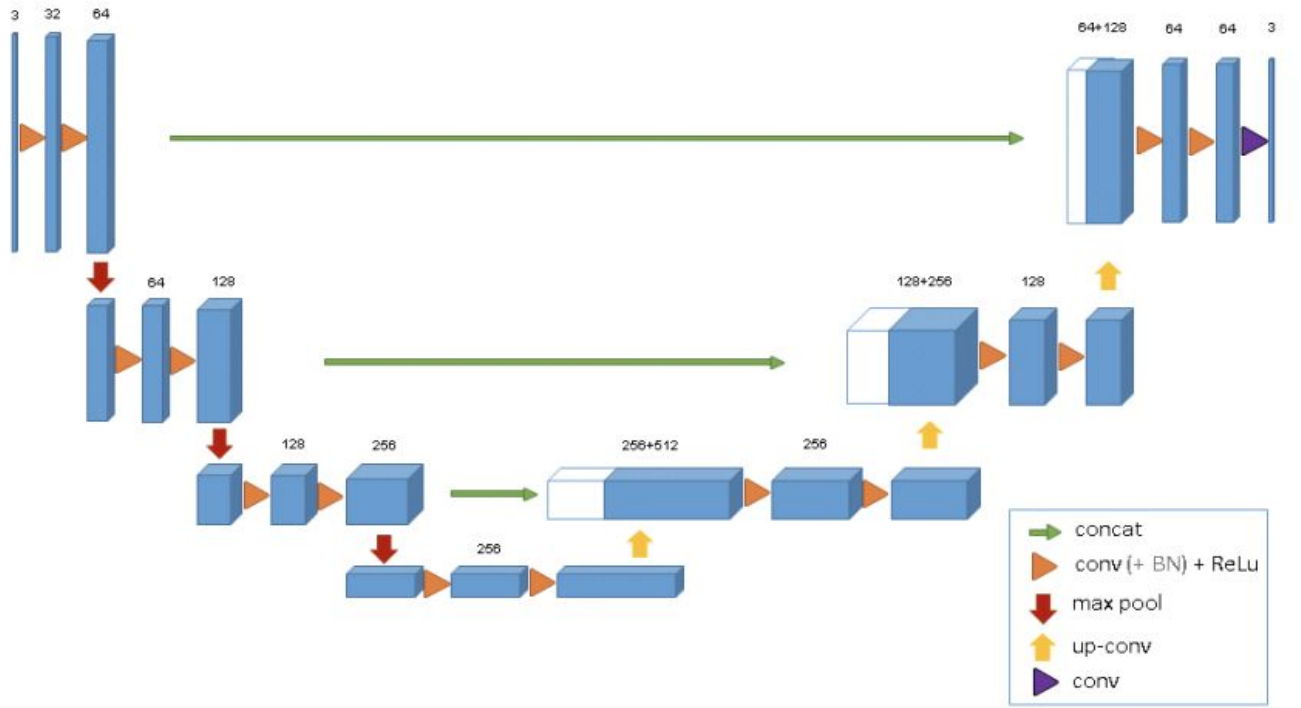
Output:
Spatial bounding box for each **instance** of a category object in the image

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

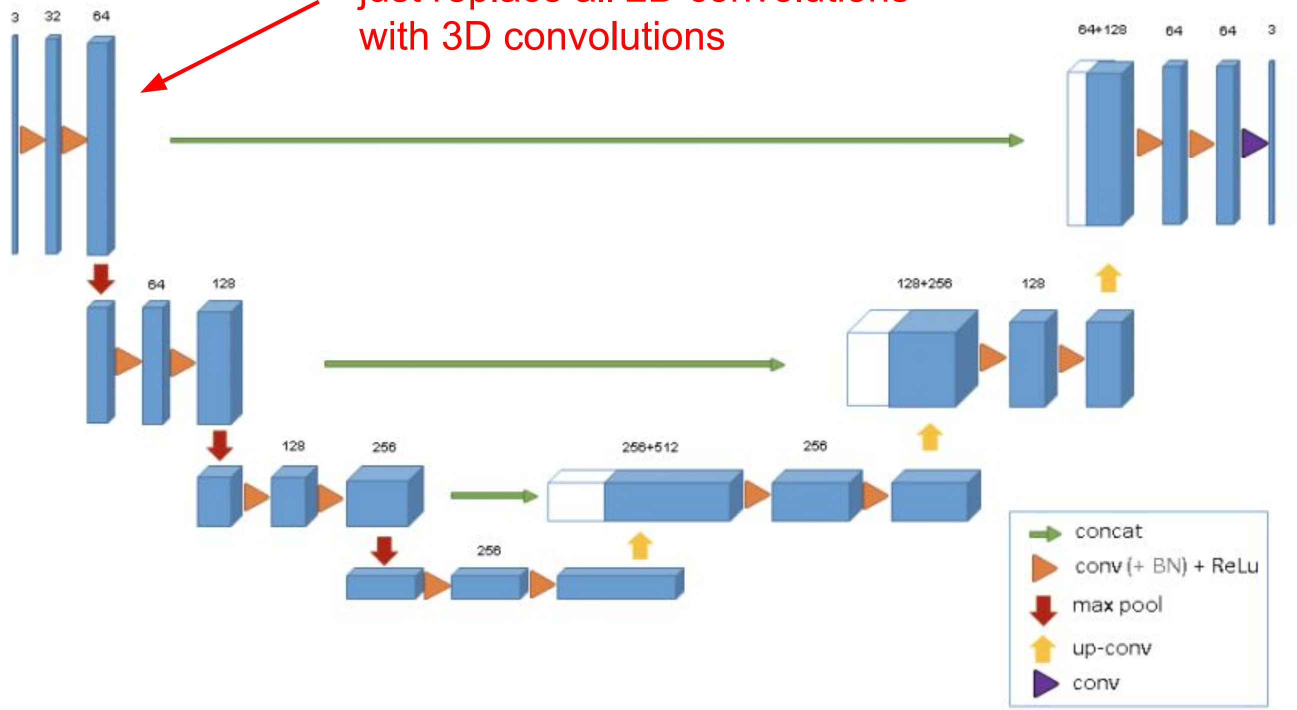
E.g. 3D U-Net



Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

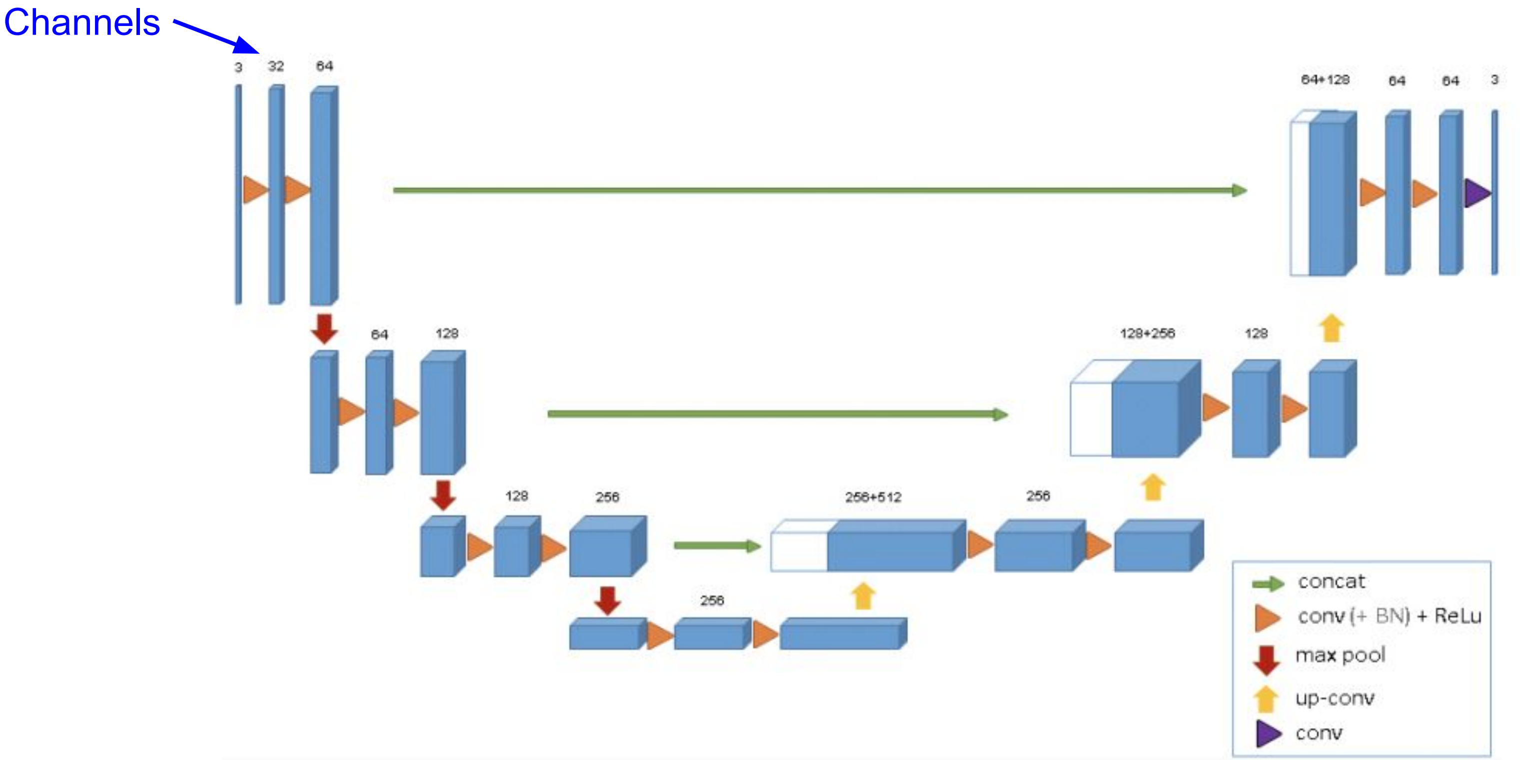
E.g. 3D U-Net

Same structure as 2D version, just replace all 2D convolutions with 3D convolutions



Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

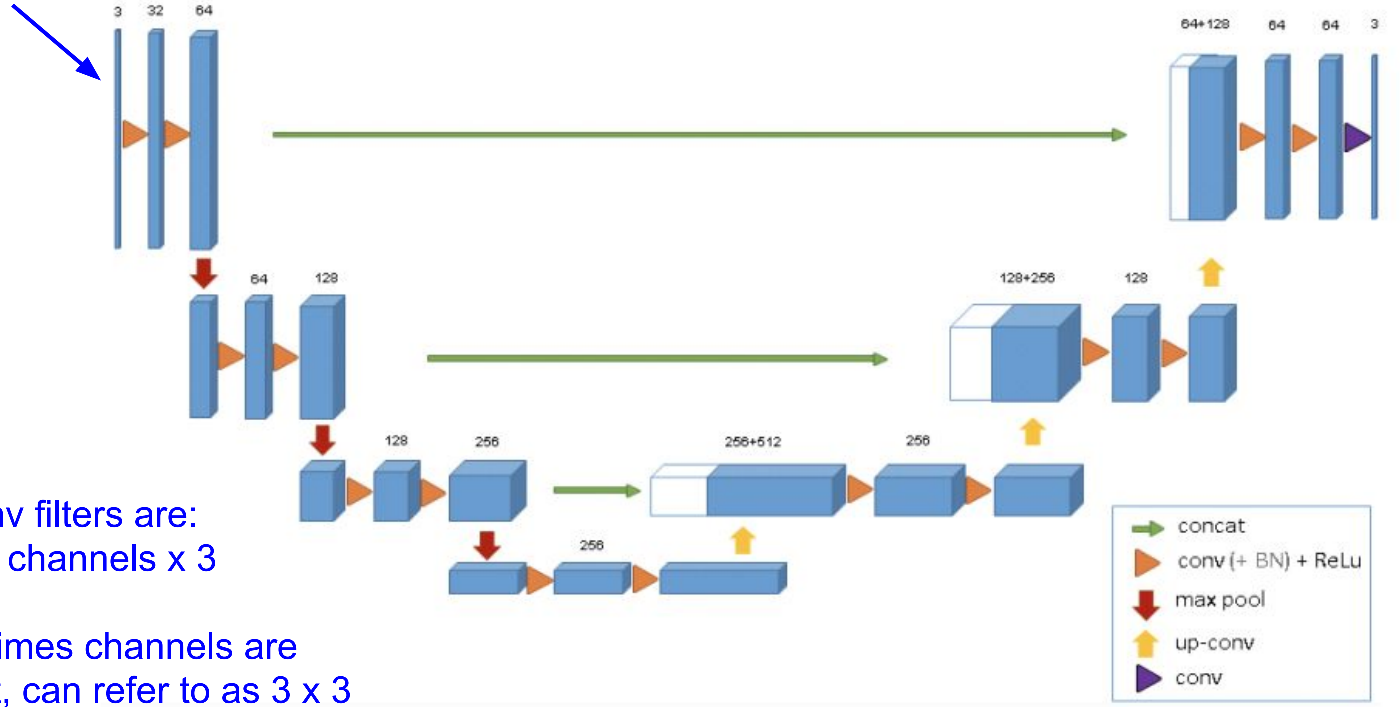
E.g. 3D U-Net



Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

E.g. 3D U-Net

Ex. input: 132 x 132 x 3 x 116



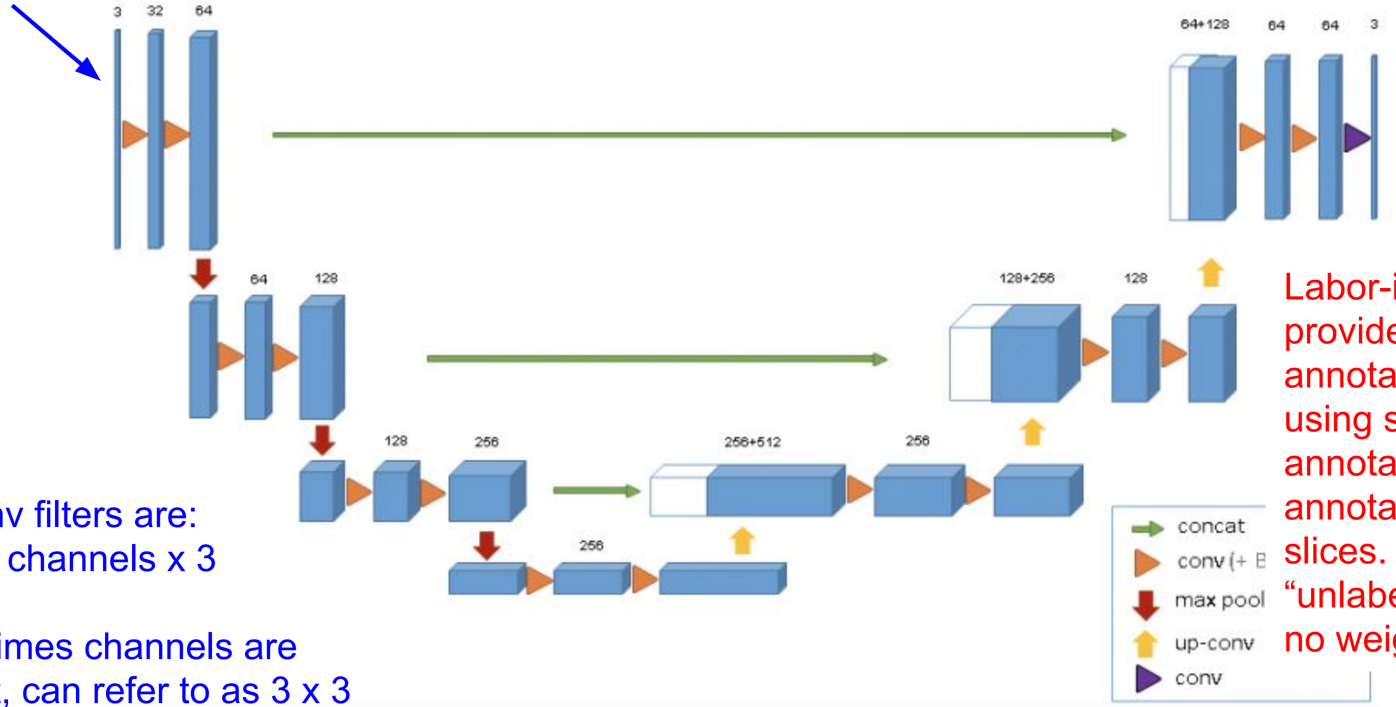
3D conv filters are:
3 x 3 x channels x 3

Sometimes channels are implicit, can refer to as 3 x 3 x 3 conv filter

Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

E.g. 3D U-Net

Ex. input: 132 x 132 x 3 x 116



3D conv filters are:
3 x 3 x channels x 3

Sometimes channels are implicit, can refer to as 3 x 3 x 3 conv filter

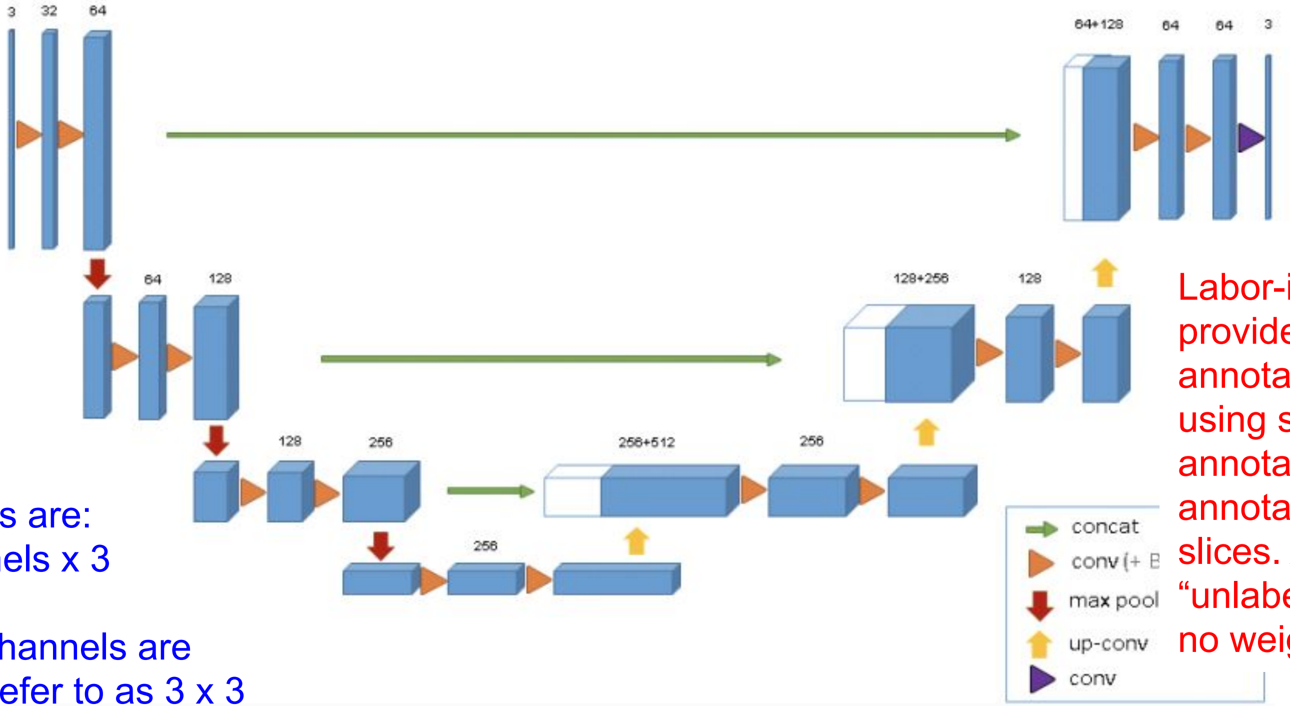
Labor-intensive to provide ground truth 3D annotation. Train instead using sparse annotations: a handful of annotated xy, xz, yz 2D slices. All others are “unlabeled” pixels with no weight in the loss.

Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

E.g. 3D U-Net

Semi-supervised learning: learning from datasets that are partially labeled (small amount of labeled data + larger amount of unlabelled data). Lots of active research on ways (e.g. loss functions which don't require manual labels) to simultaneously learn richer information from the unlabeled data.

Ex. input: 132 x 132 x 3 x 116



Labor-intensive to provide ground truth 3D annotation. Train instead using sparse annotations: a handful of annotated xy, xz, yz 2D slices. All others are "unlabeled" pixels with no weight in the loss.

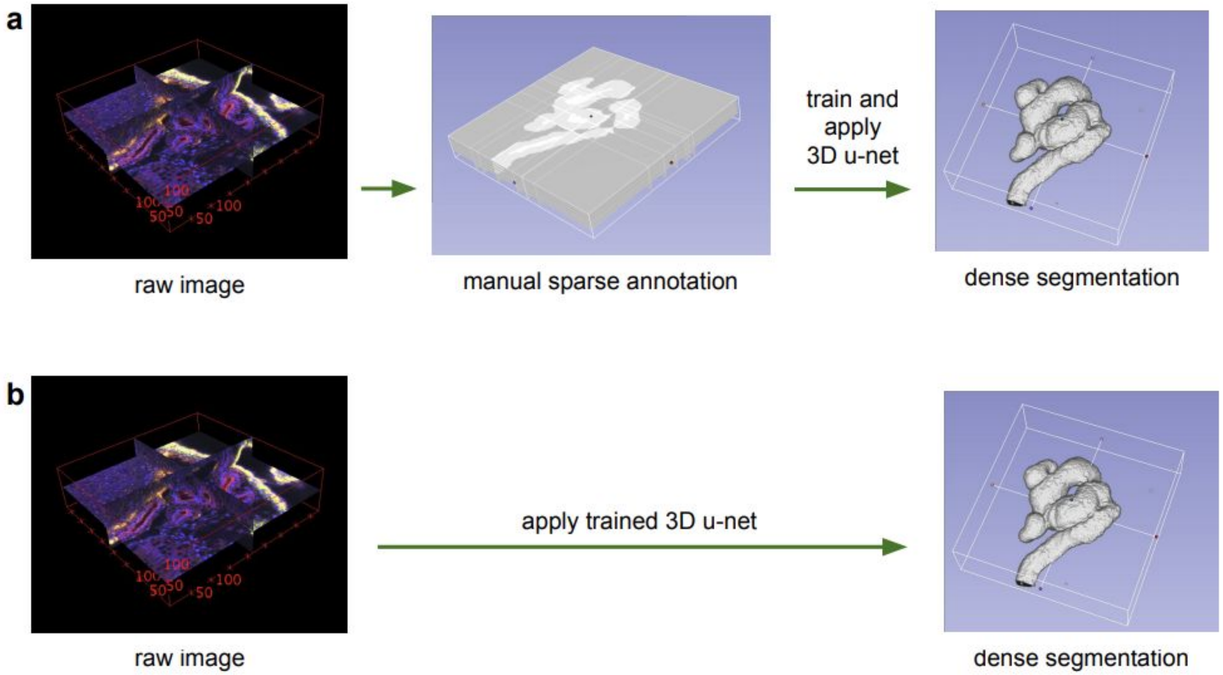
3D conv filters are:
3 x 3 x channels x 3

Sometimes channels are implicit, can refer to as 3 x 3 x 3 conv filter

Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

E.g. 3D U-Net

Ex: 3D segmentation of Xenopus kidney in confocal microscopic data

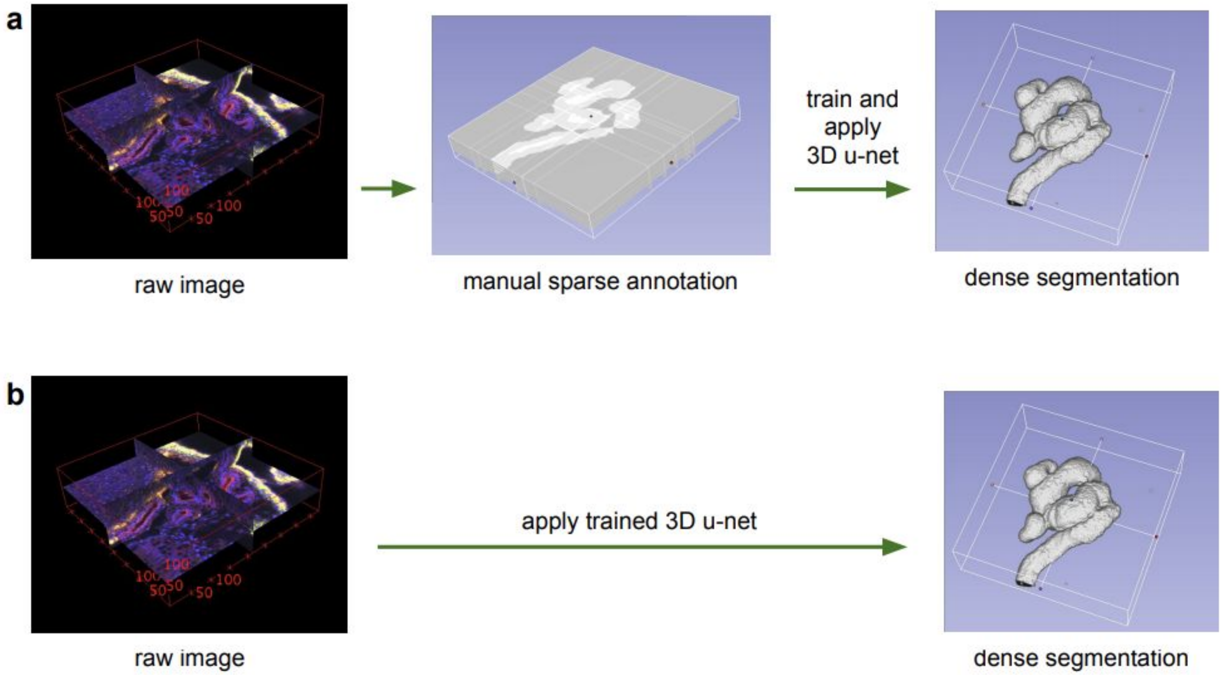


Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

E.g. 3D U-Net

Ex: 3D segmentation of Xenopus kidney in confocal microscopic data

Spatial dims: ~ 250 x 250 x 60.
3 channels: each channel corresponds to a different type of data capture



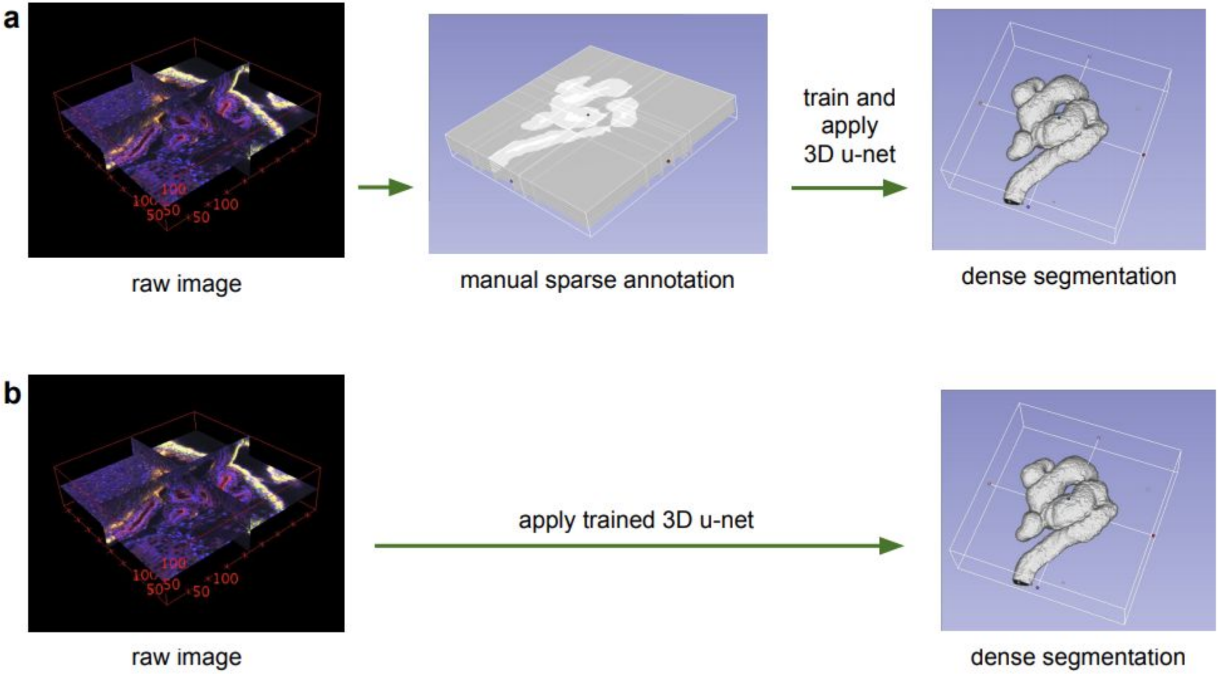
Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

E.g. 3D U-Net

Ex: 3D segmentation of Xenopus kidney in confocal microscopic data

Spatial dims: ~ 250 x 250 x 60.
3 channels: each channel corresponds to a different type of data capture

Used only 3 samples total! (with total of 77 annotated 2D slices). Leverages fact that each sample contains many instances of same repetitive structures w/ variation.



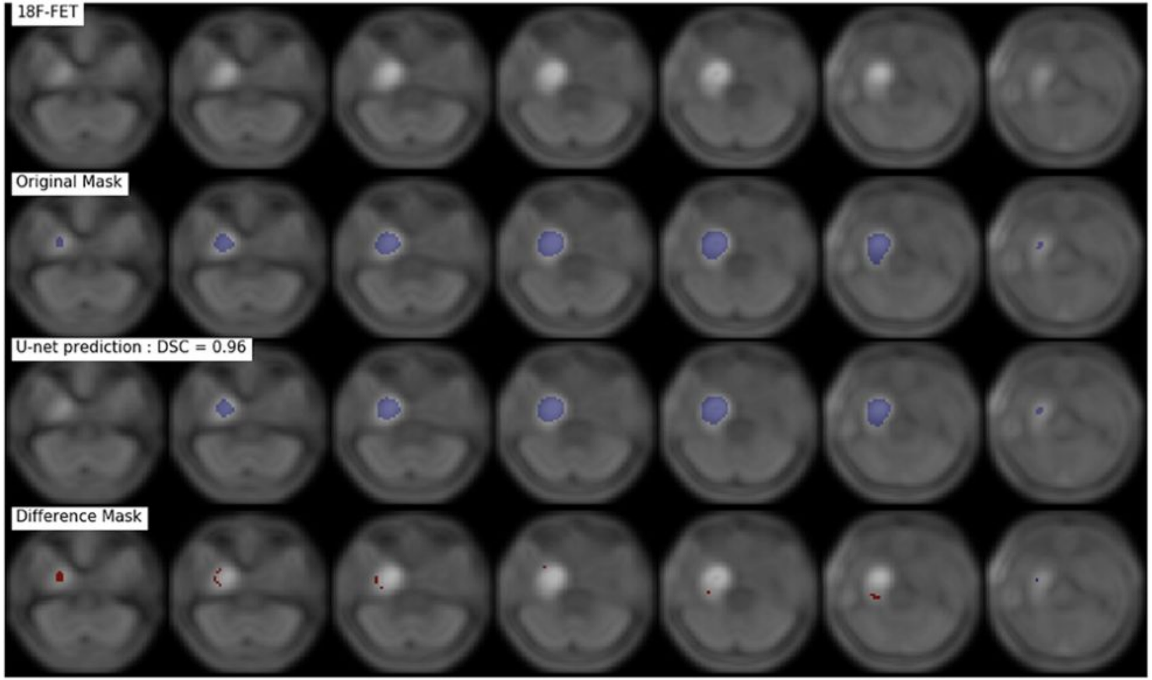
Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

Ex: Brain lesion segmentation

Training set: 37 PET scans (3D volumes)

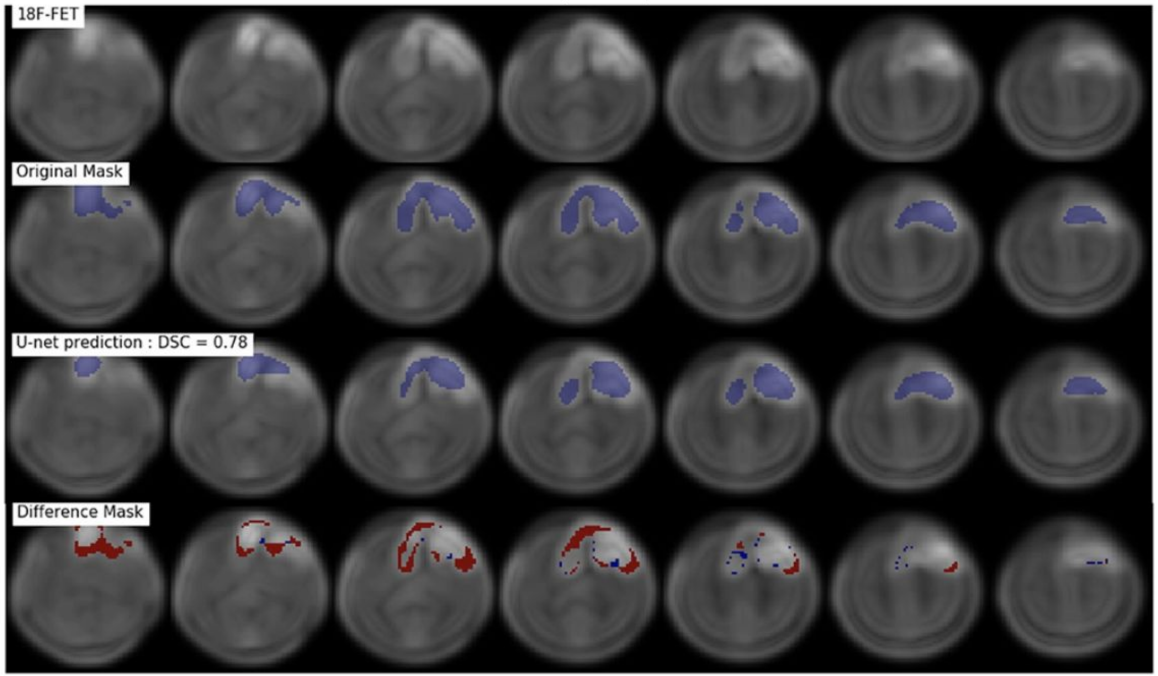
Evaluation set: 11 PET scans

Volumes resized to 64x64x40 for computational efficiency



Blanc-Durand et al. Automatic lesion detection and segmentation of 18F-FET PET in gliomas: A full 3D U-Net convolutional neural network study. PLoS One, 2018.

Ex: Brain lesion segmentation

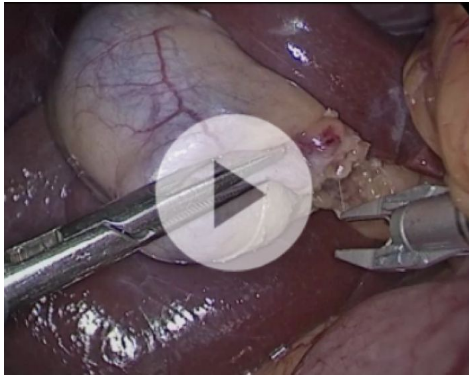


Blanc-Durand et al. Automatic lesion detection and segmentation of 18F-FET PET in gliomas: A full 3D U-Net convolutional neural network study. PLoS One, 2018.

Video data (high dimensional in time)

E.g. in:

Surgery



Hospital patient monitoring

Psychology

Another approach: 3D convolutions

Slide filter along **3** directions: x, y, and z

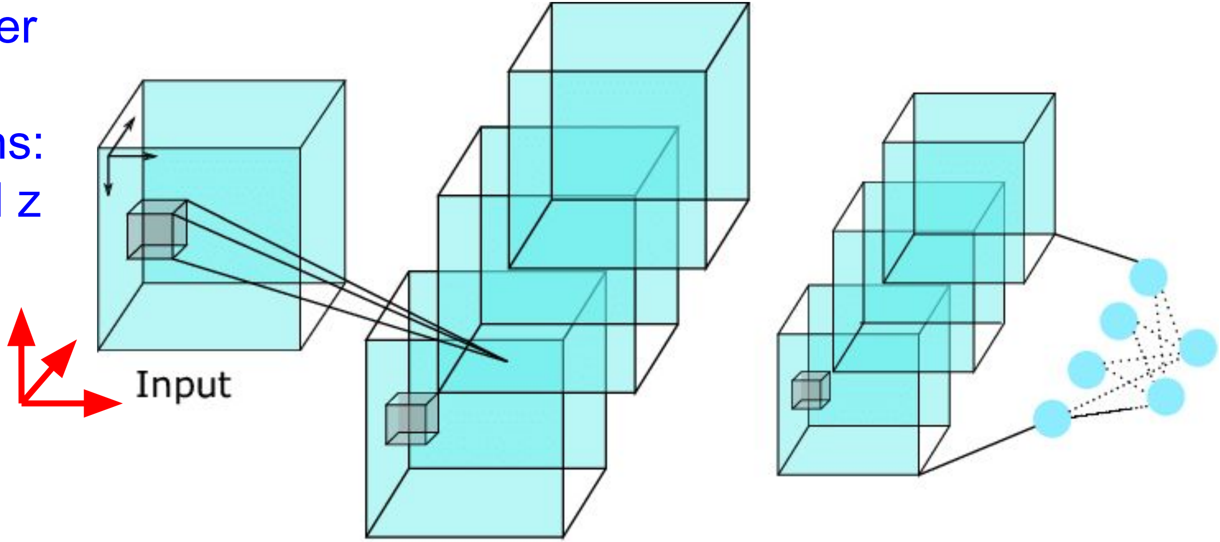
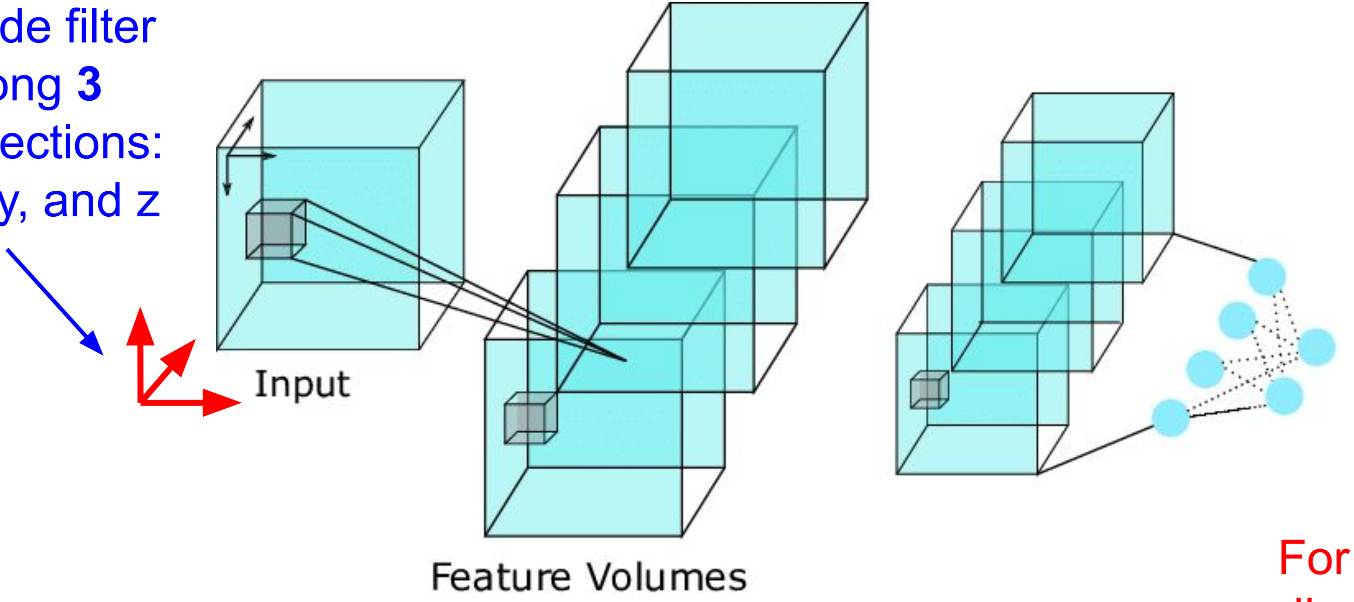


Figure credit:
https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

Another approach: 3D convolutions

Slide filter along **3** directions: x, y, and z

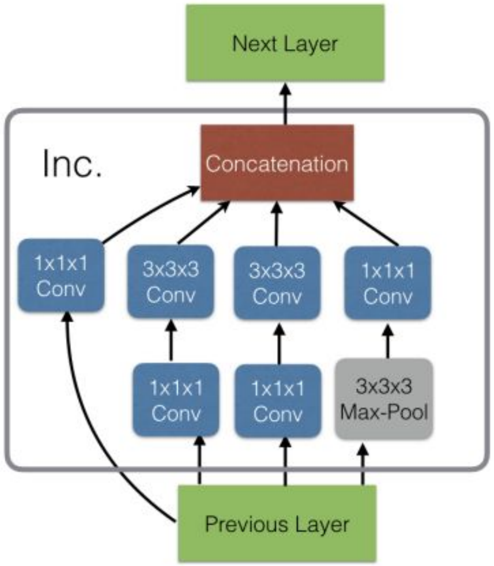


For video data, 3rd dimension is time

Figure credit:
https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

I3D: 3D convolutional network for video data

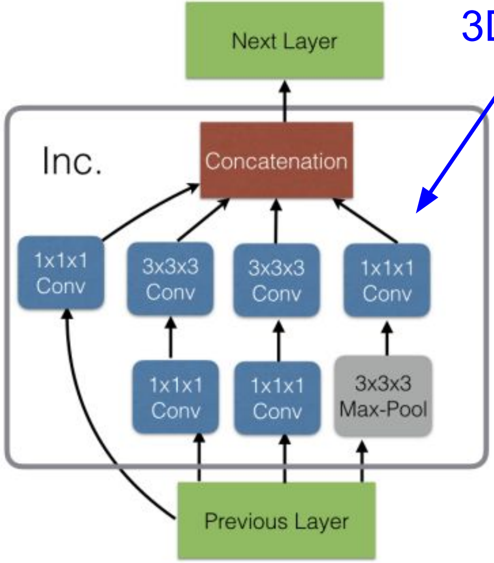
Inception Module (Inc.) w/
3D convolutions



Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

I3D: 3D convolutional network for video data

Inception Module (Inc.) w/
3D convolutions

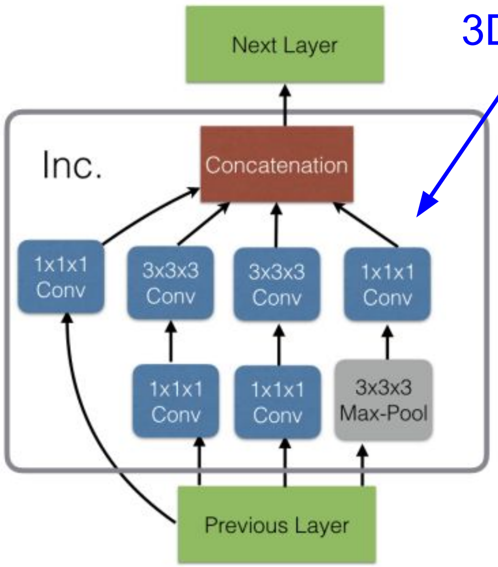


3D convs

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

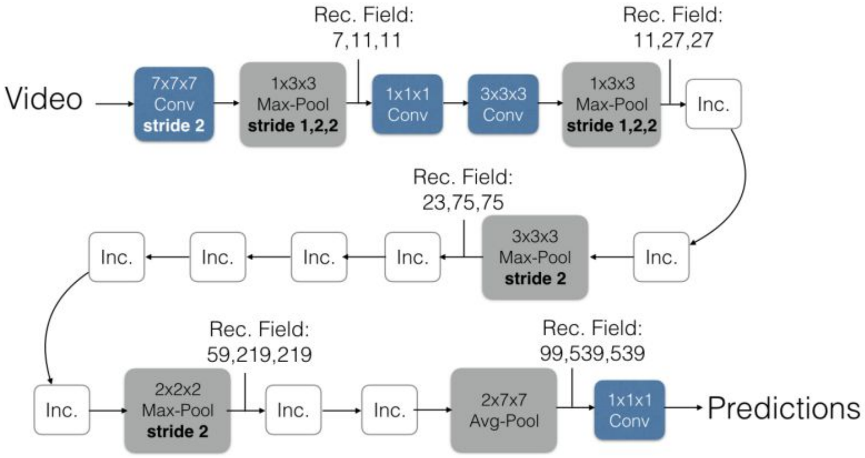
I3D: 3D convolutional network for video data

Inception Module (Inc.) w/ 3D convolutions



3D convs

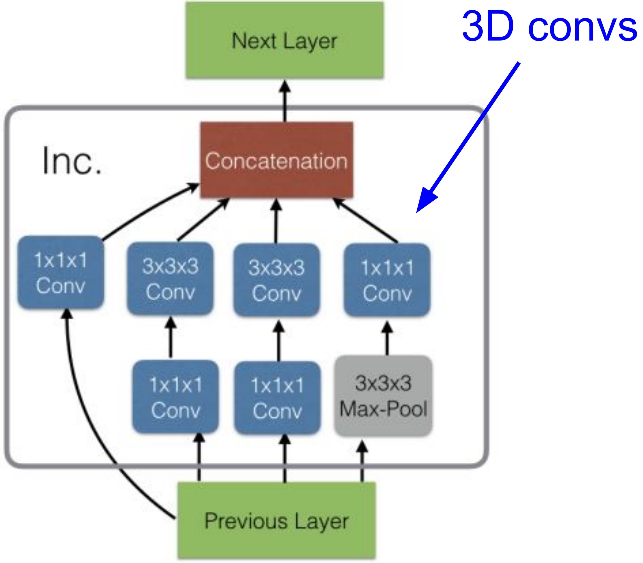
3D Inception Module used in Inception Network (also known as GoogLeNet)



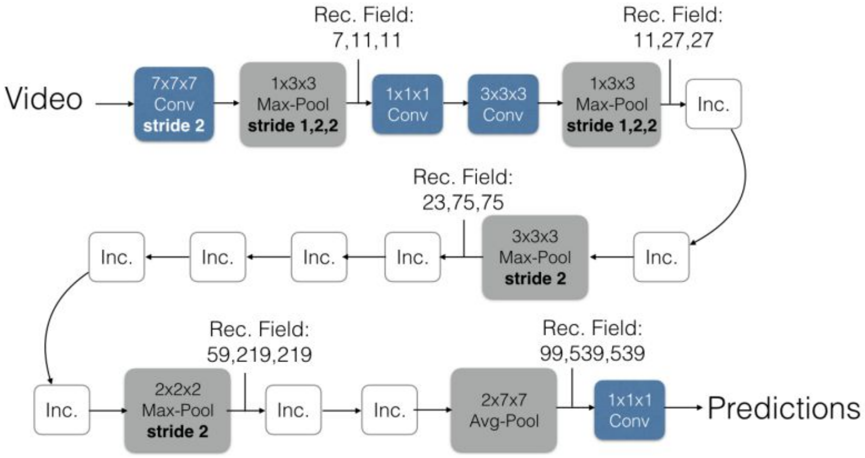
Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

I3D: 3D convolutional network for video data

Inception Module (Inc.) w/ 3D convolutions



3D Inception Module used in Inception Network (also known as GoogLeNet)



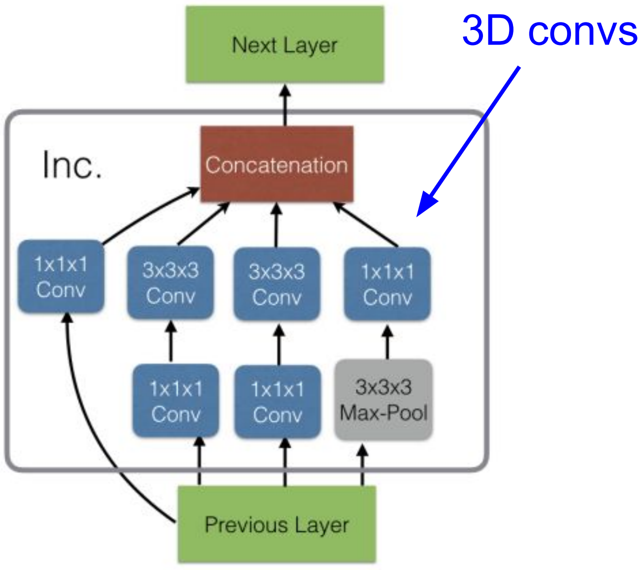
Can pre-train from 2D datasets e.g. ImageNet by replicating and normalizing 2D weights over additional dimension!

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

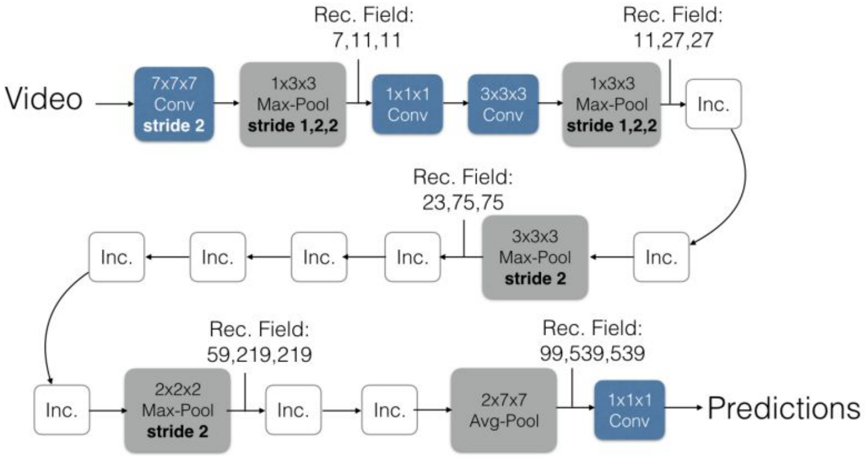
I3D: 3D convolutional network for video data

Note: in general, can 3D-ify many 2D architectures!

Inception Module (Inc.) w/ 3D convolutions



3D Inception Module used in Inception Network (also known as GoogLeNet)



Can pre-train from 2D datasets e.g. ImageNet by replicating and normalizing 2D weights over additional dimension!

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

Video classifiers (including I3D) can be enhanced with optical flow

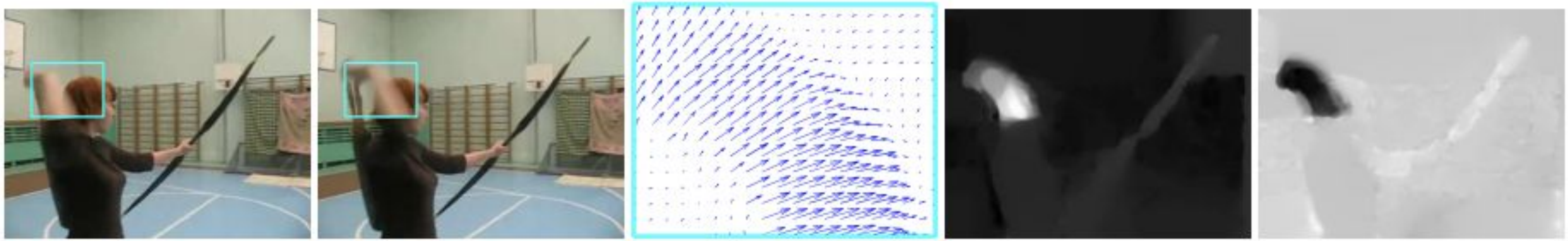


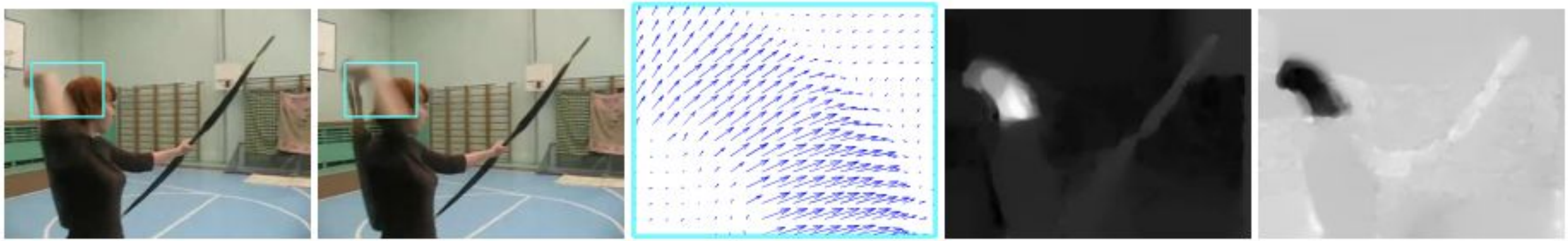
Figure credit: Simonyan and Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. NeurIPS 2014.

Video classifiers (including I3D) can be enhanced with optical flow

Two consecutive frames

Optical flow displacement vectors

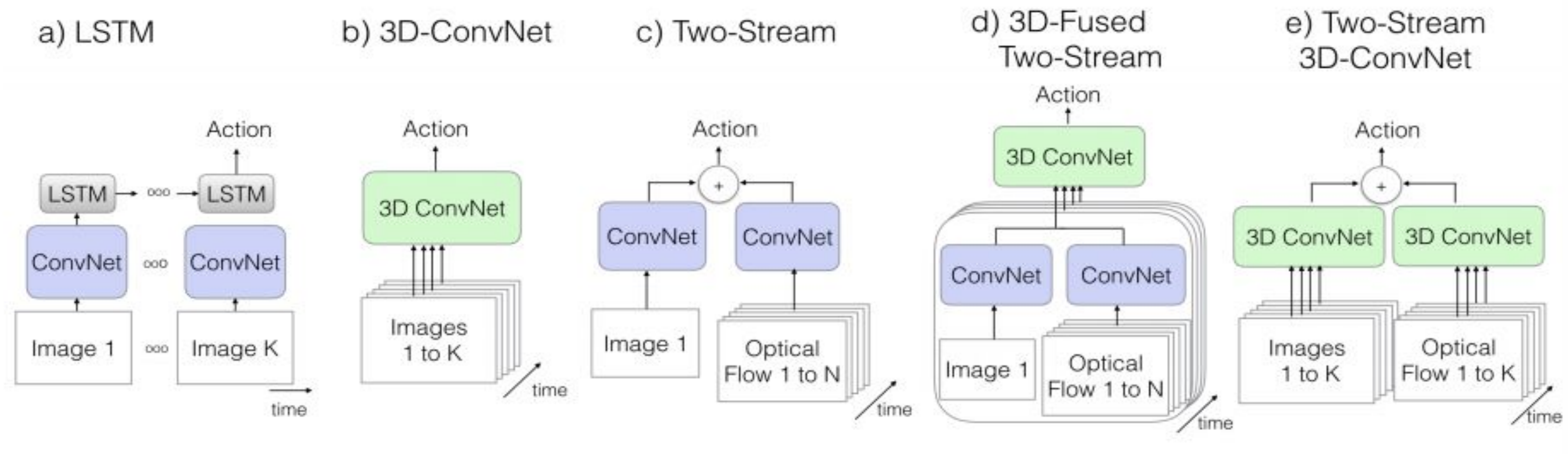
horizontal (L) and vertical (R) components of displacement



Directional components can be represented as images (or multiple channels of input volume!)

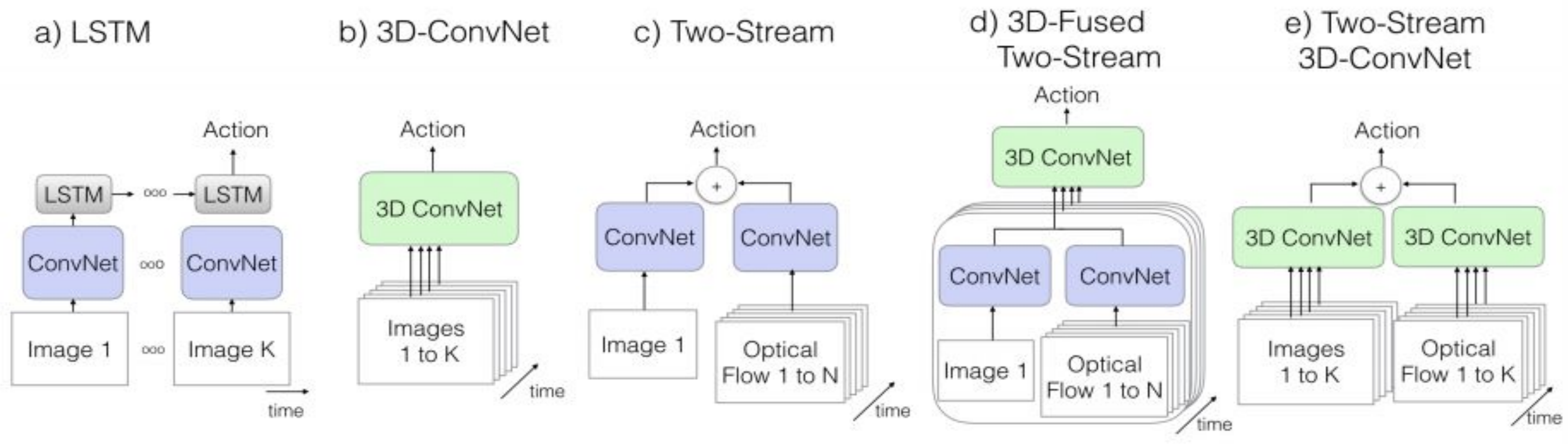
Figure credit: Simonyan and Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. NeurIPS 2014.

Video classifiers (including I3D) can be enhanced with optical flow



Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

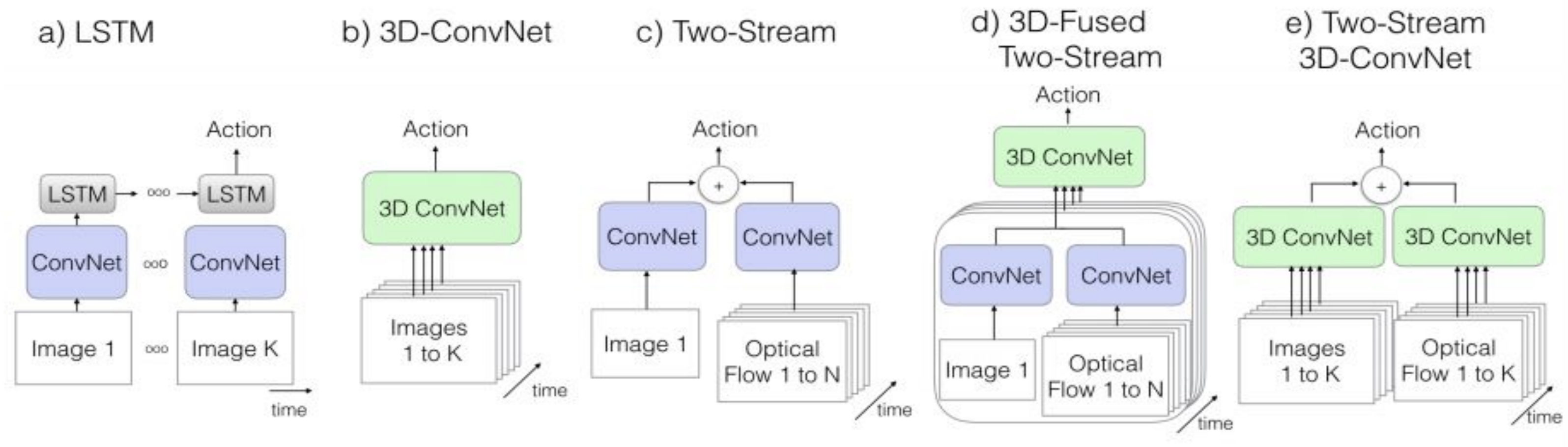
Video classifiers (including I3D) can be enhanced with optical flow



LSTM over RGB

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

Video classifiers (including I3D) can be enhanced with optical flow

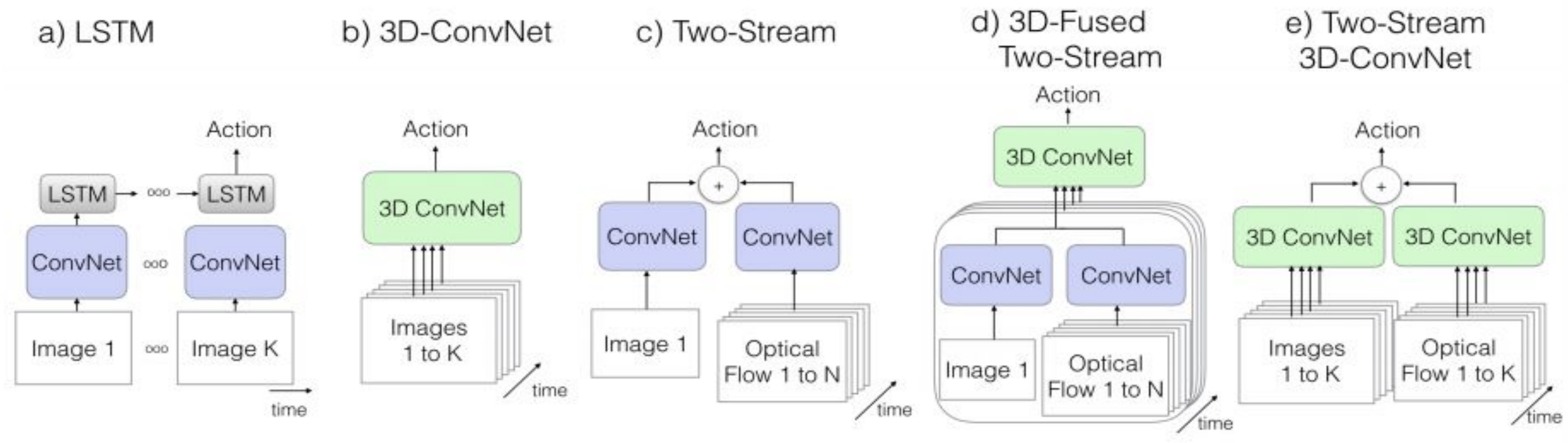


LSTM over RGB

(LSTM is a type of recurrent neural network.
We will talk more about these soon!)

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

Video classifiers (including I3D) can be enhanced with optical flow

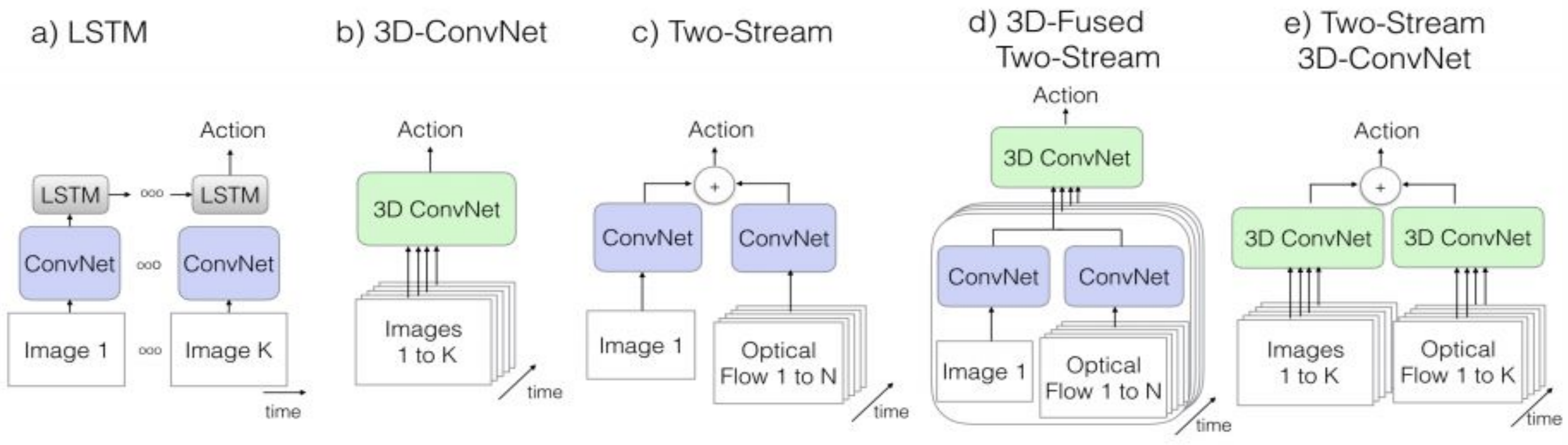


LSTM over RGB

I3D (3D convs) over RGB

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

Video classifiers (including I3D) can be enhanced with optical flow



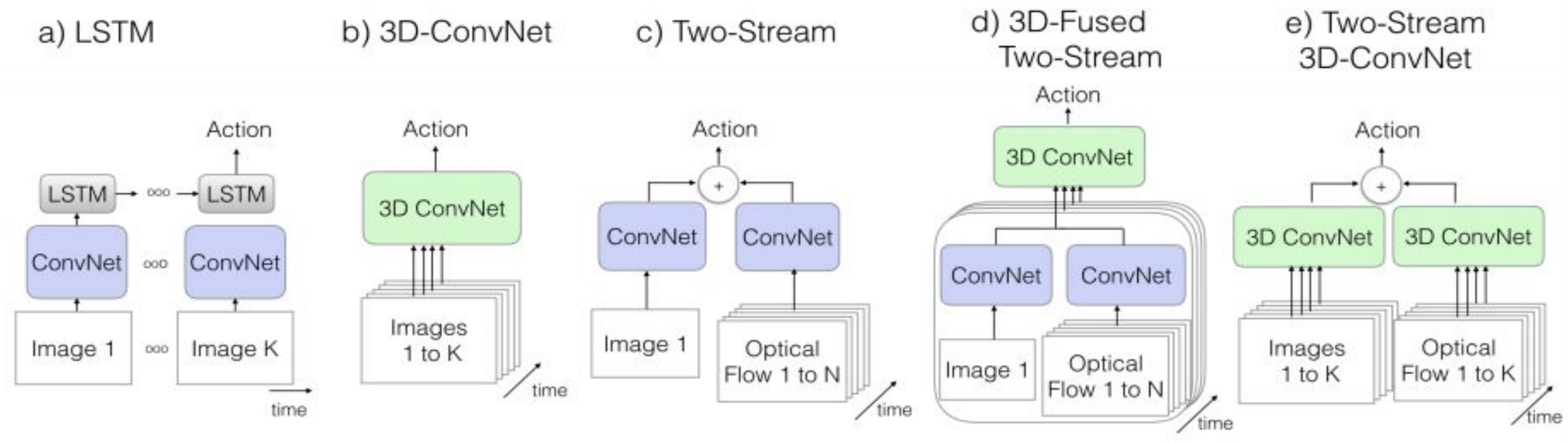
LSTM over RGB

I3D (3D convs) over RGB

2D convs over RGB + optical flow (OF)

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

Video classifiers (including I3D) can be enhanced with optical flow



LSTM over RGB

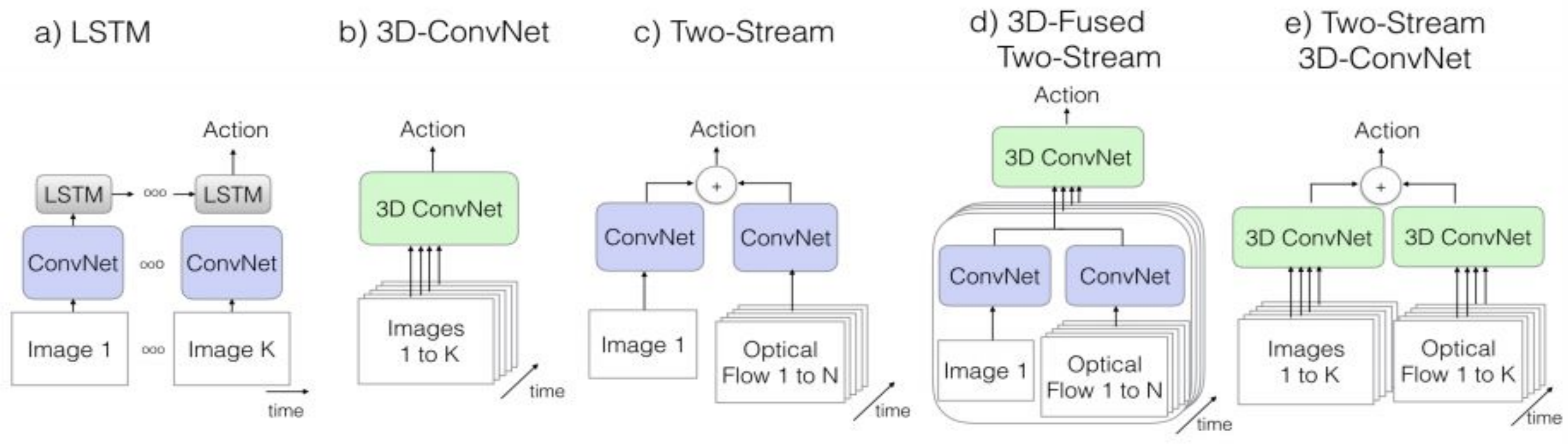
I3D (3D convs) over RGB

2D convs over RGB + optical flow (OF)

Late 3D fusion of RGB + OF

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

Video classifiers (including I3D) can be enhanced with optical flow



LSTM over RGB

I3D (3D convs) over RGB

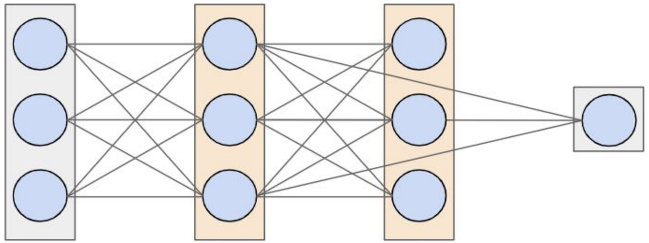
2D convs over RGB + optical flow (OF)

Late 3D fusion of RGB + OF

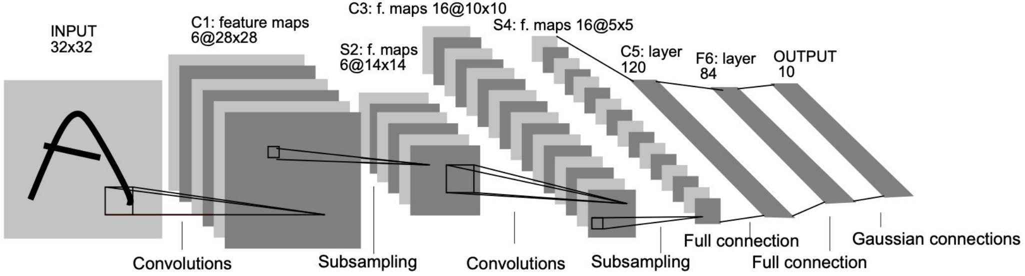
Two I3D streams over RGB + OF

Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

Preview: Recurrent neural networks

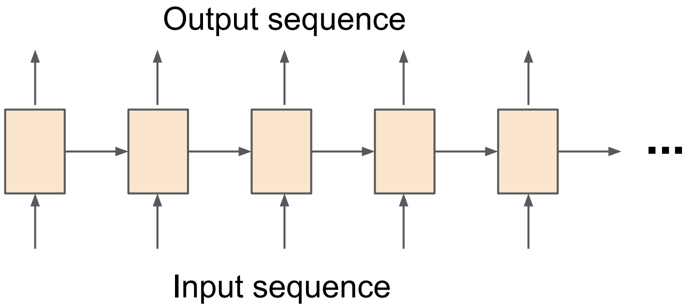


Fully connected neural networks
(linear layers, good for “feature vector” inputs)



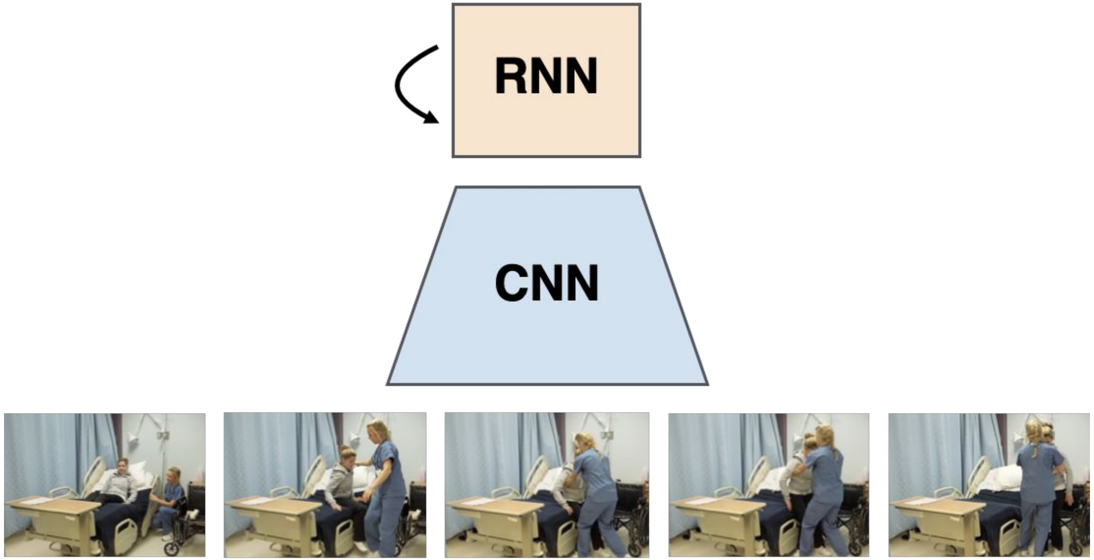
Convolutional neural networks
(convolutional layers, good for image inputs)

Recurrent neural networks
(linear layers modeling recurrence relation across sequence, good for sequence inputs)



Videos are sequences: natural fit for recurrent networks

$$\mathbf{y} = \{y_0, y_1, ..., y_T\}$$

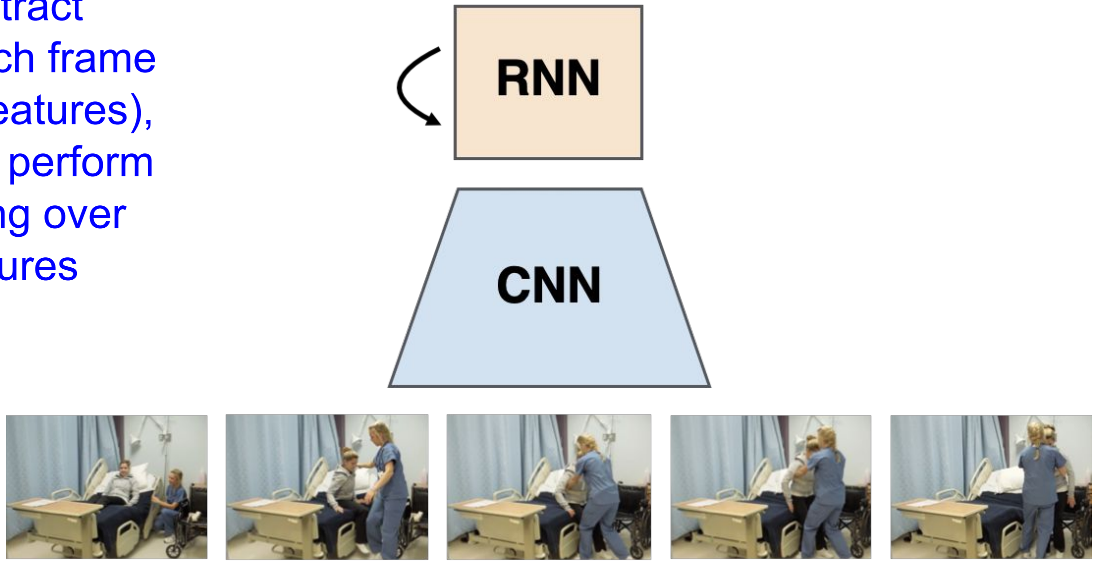


$$\mathbf{x} = \{x_0, x_1, ..., x_T\}$$

Videos are sequences: natural fit for recurrent networks

Abstracted overview:
Use a CNN to extract features from each frame (e.g. final-layer features), then use RNN to perform temporal modeling over sequence of features

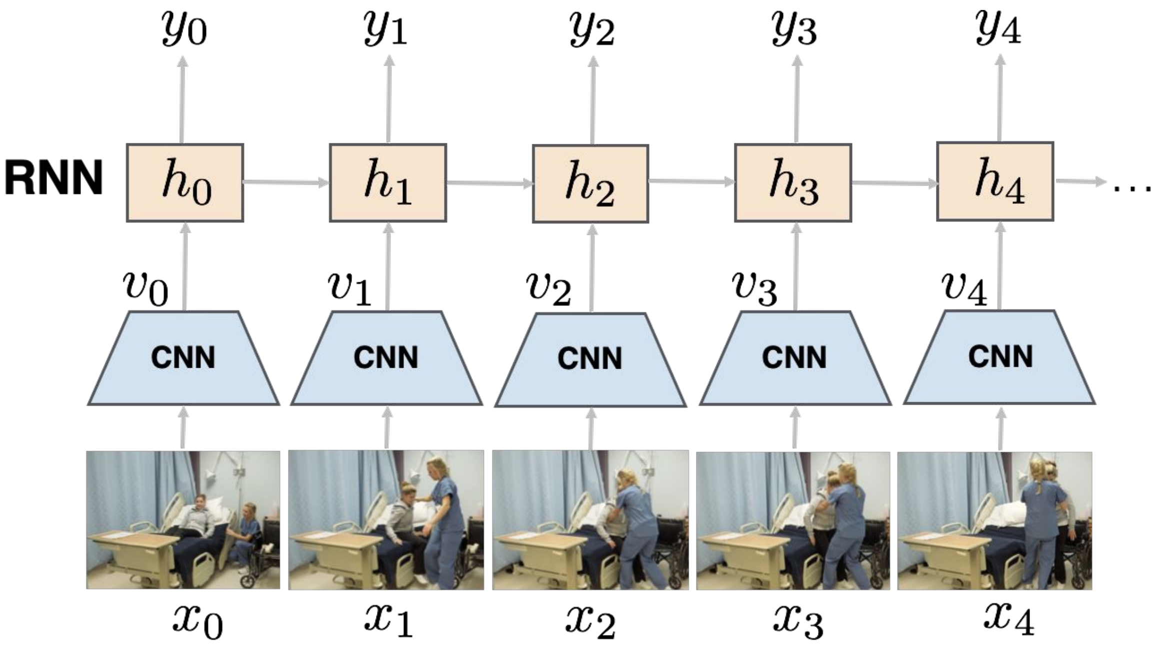
$$\mathbf{y} = \{y_0, y_1, ..., y_T\}$$



$$\mathbf{x} = \{x_0, x_1, ..., x_T\}$$

Videos are sequences: natural fit for recurrent networks

Diagram of a CNN + RNN “rolled out” over time

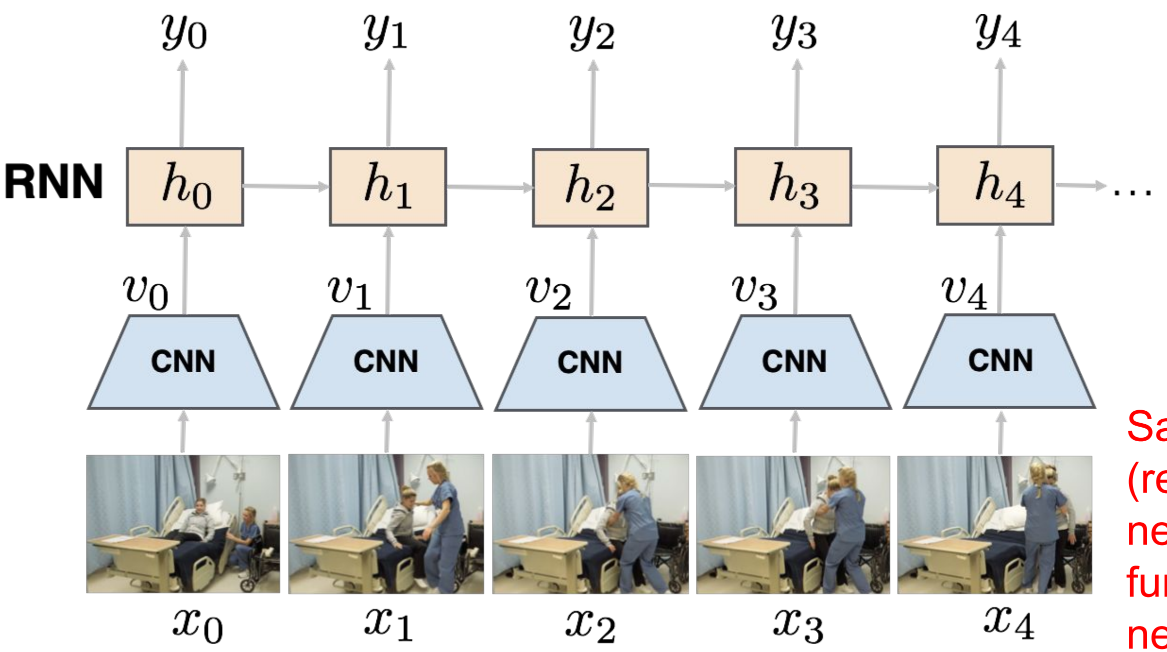


$$
\begin{aligned}
h_t &= f_W(h_{t-1}, v_t) \\
&= \tanh(W_{hh}h_{t-1} + W_{vh}v_t)
\end{aligned}
$$

$$y_t = W_{hy}h_t$$

Videos are sequences: natural fit for recurrent networks

Diagram of a CNN + RNN "rolled out" over time



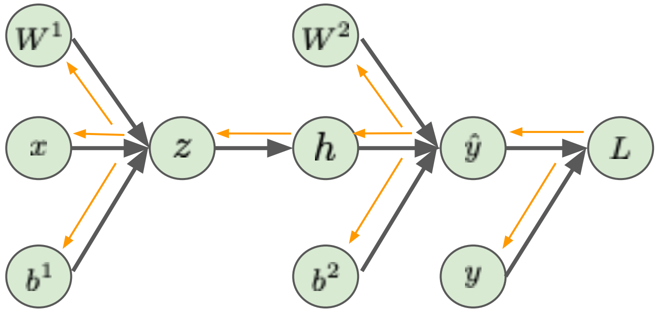
$$
\begin{aligned}
h_t &= f_W(h_{t-1}, v_t) \\
&= \boxed{\tanh(W_{hh}h_{t-1} + W_{vh}v_t)} \\
y_t &= \boxed{W_{hy}h_t}
\end{aligned}
$$

Same idea of weight matrices (remember fully-connected networks) and nonlinear activation functions! Just applied to a neural network with a different connectivity structure

Aside: how do we compute gradient updates? Remember backpropagation.

Network output: $\hat{y} = W^2(\sigma(W^1 x + b^1)) + b^2$

Think of computing loss function as staged computation of intermediate variables:



"Forward pass":
$$z = W^1 x + b^1$$
$$h = \sigma(z)$$
$$\hat{y} = W^2 h + b^2$$
$$L = (\hat{y} - y)^2$$

Now, can use a repeated application of the chain rule, going backwards through the computational graph, to obtain the gradient of the loss with respect to each node of the computation graph.

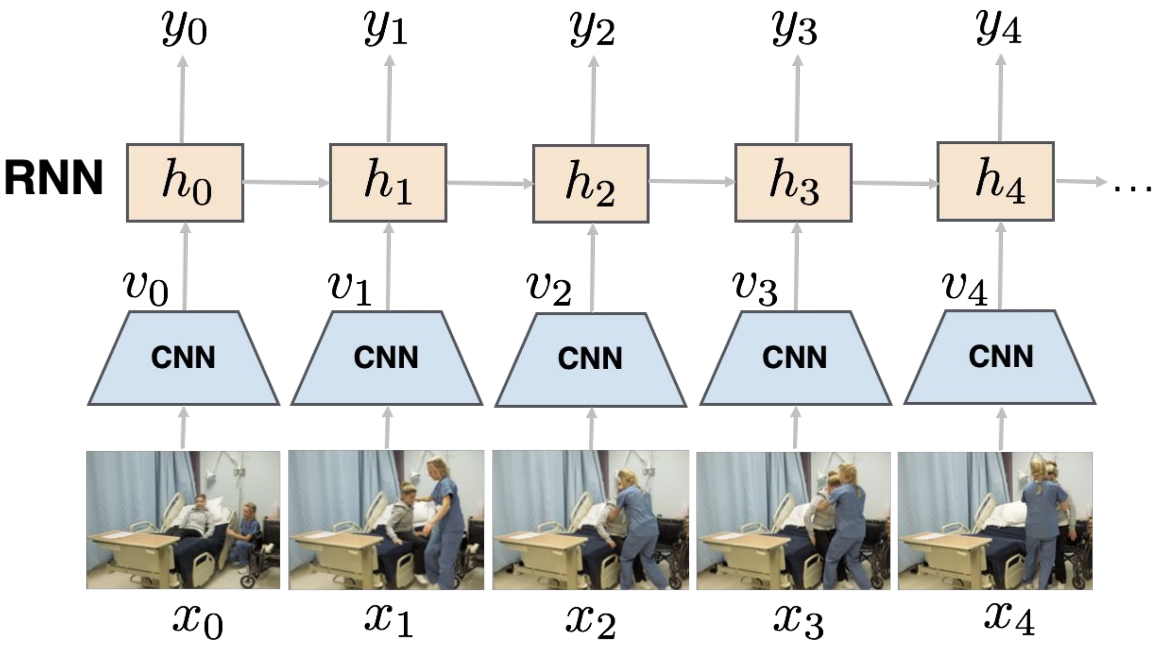
"Backward pass": $\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y)$ (not all gradients shown)

$$\frac{\partial L}{\partial W^2} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial W^2}$$
$$\frac{\partial L}{\partial H} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial H}$$
$$\frac{\partial L}{\partial Z} = \frac{\partial L}{\partial H} \frac{\partial H}{\partial Z}$$
$$\frac{\partial L}{\partial W^1} = \frac{\partial L}{\partial Z} \frac{\partial z}{\partial W^1}$$

Plug in from earlier computations via chain rule

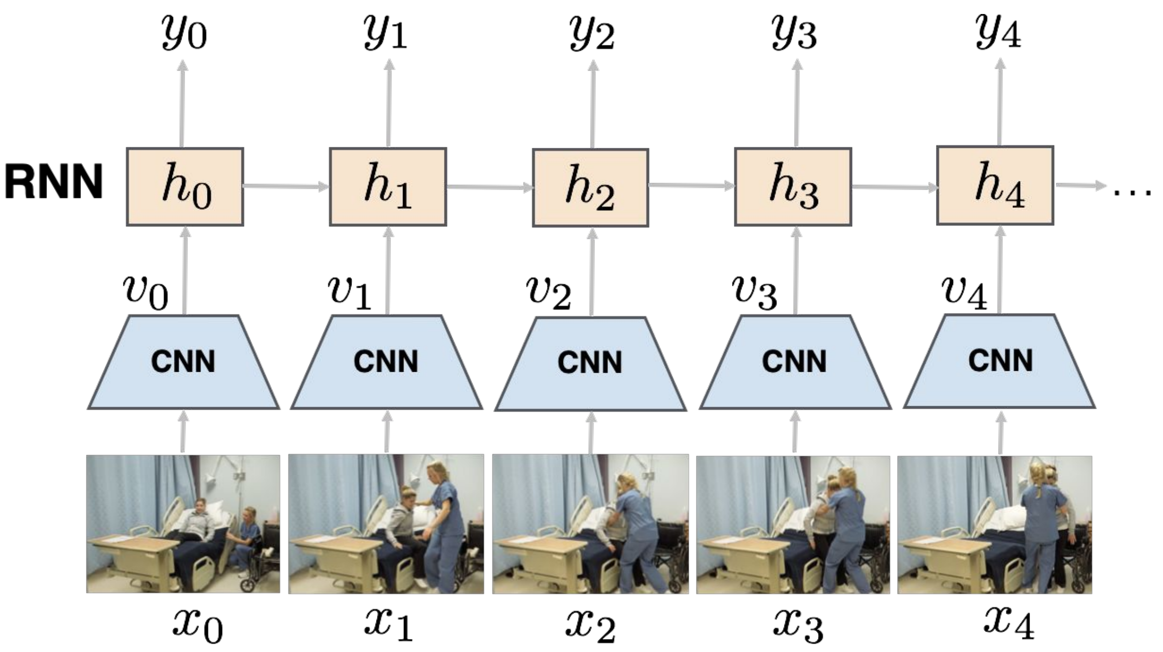
Local gradients to derive

Videos are sequences: natural fit for recurrent networks



This is a computational graph
-> can backprop and train
RNN and CNN jointly

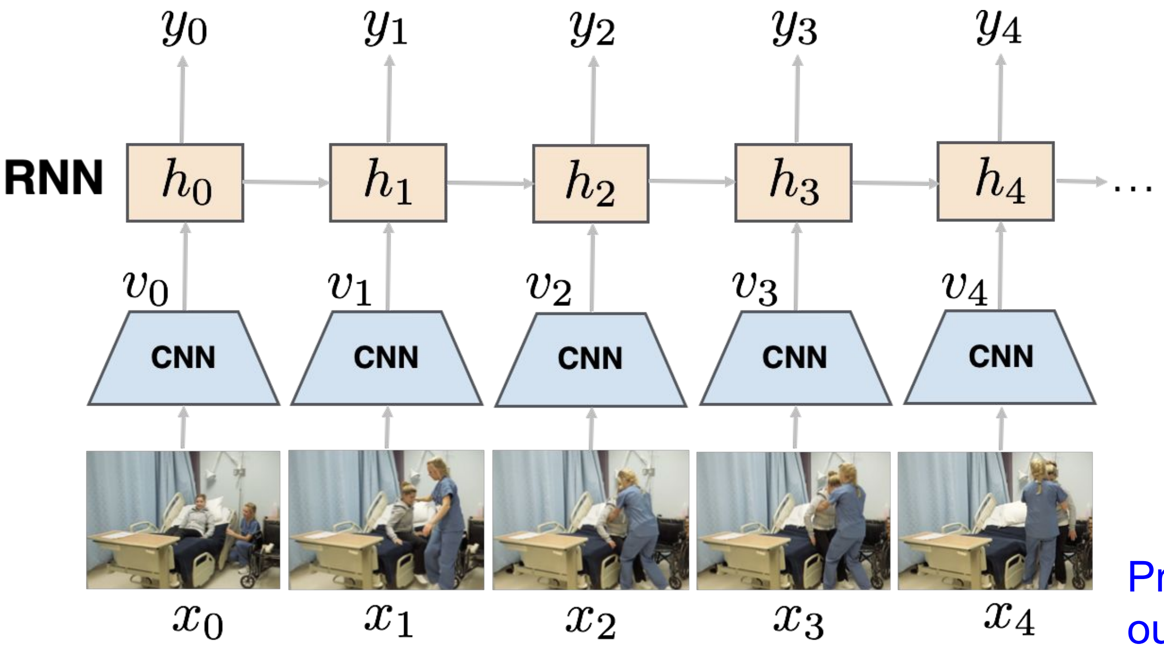
Videos are sequences: natural fit for recurrent networks



This is a computational graph -> can backprop and train RNN and CNN jointly

But a very large number of parameters to train simultaneously… more common to fine-tune a single-frame CNN over the data first (or use pre-trained CNN), then extract features and train the RNN separately

Videos are sequences: natural fit for recurrent networks

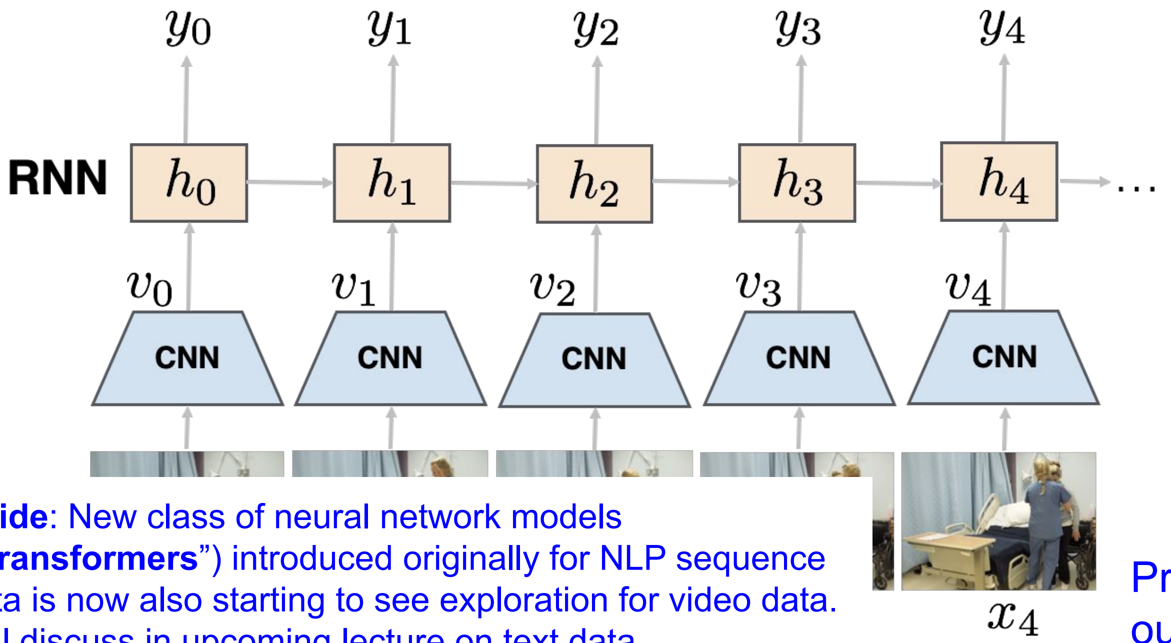


This is a computational graph -> can backprop and train RNN and CNN jointly

But a very large number of parameters to train simultaneously… more common to fine-tune a single-frame CNN over the data first (or use pre-trained CNN), then extract features and train the RNN separately

Preview of RNNs. Will see again in our discussion of sequence EHR data.

Videos are sequences: natural fit for recurrent networks



Aside: New class of neural network models ("**Transformers**") introduced originally for NLP sequence data is now also starting to see exploration for video data. Will discuss in upcoming lecture on text data.

This is a computational graph -> can backprop and train RNN and CNN jointly

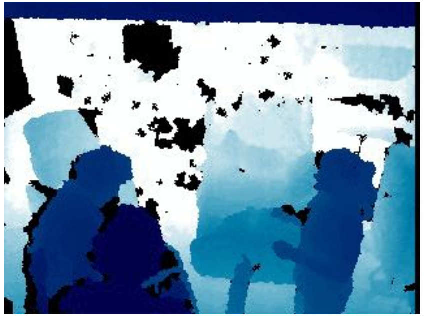
But a very large number of parameters to train simultaneously… more common to fine-tune a single-frame CNN over the data first (or use pre-trained CNN), then extract features and train the RNN separately

Preview of RNNs. Will see again in our discussion of sequence EHR data.

Detecting patient mobilization activities in the ICU

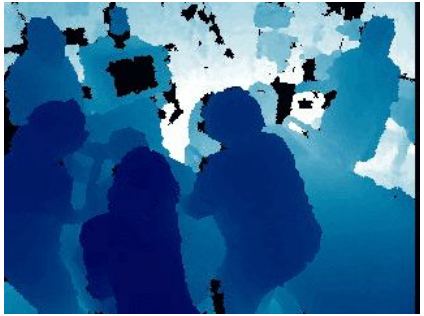
Get patient out of bed

Sit patient in chair

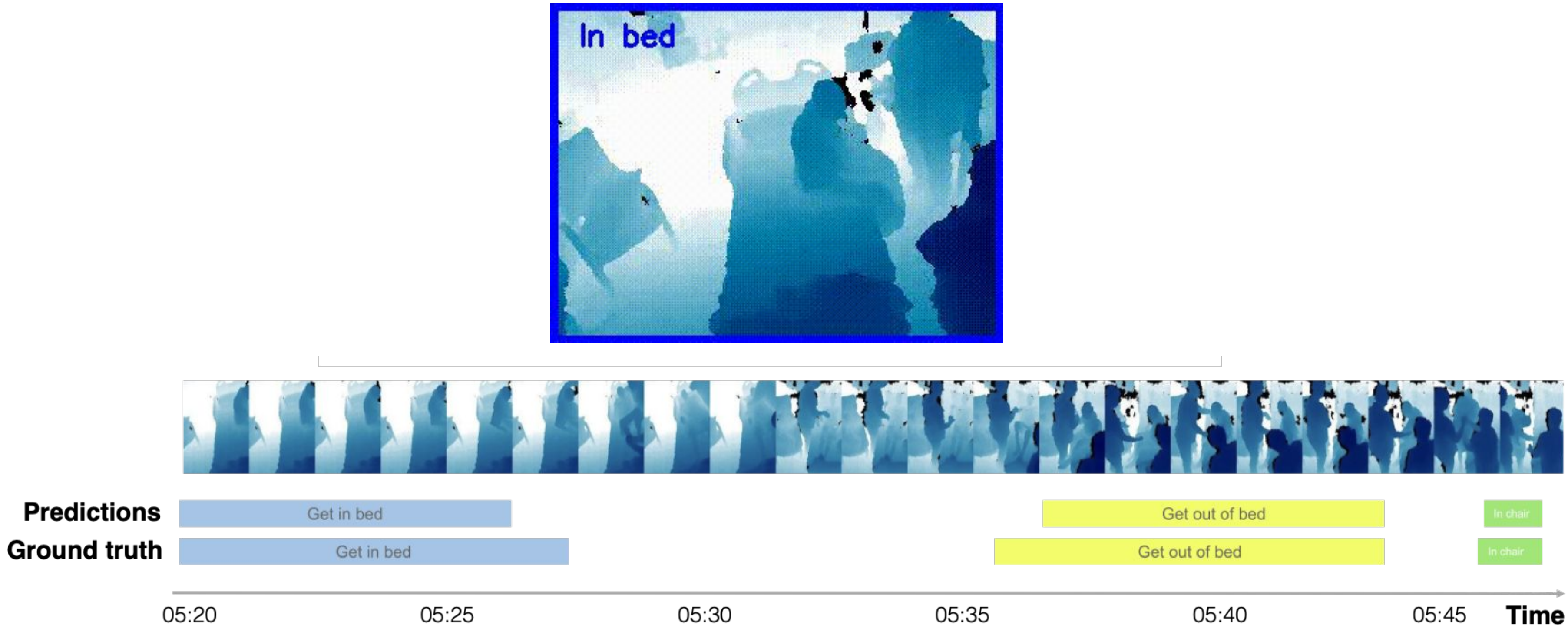


Get patient in bed

Get patient out of chair

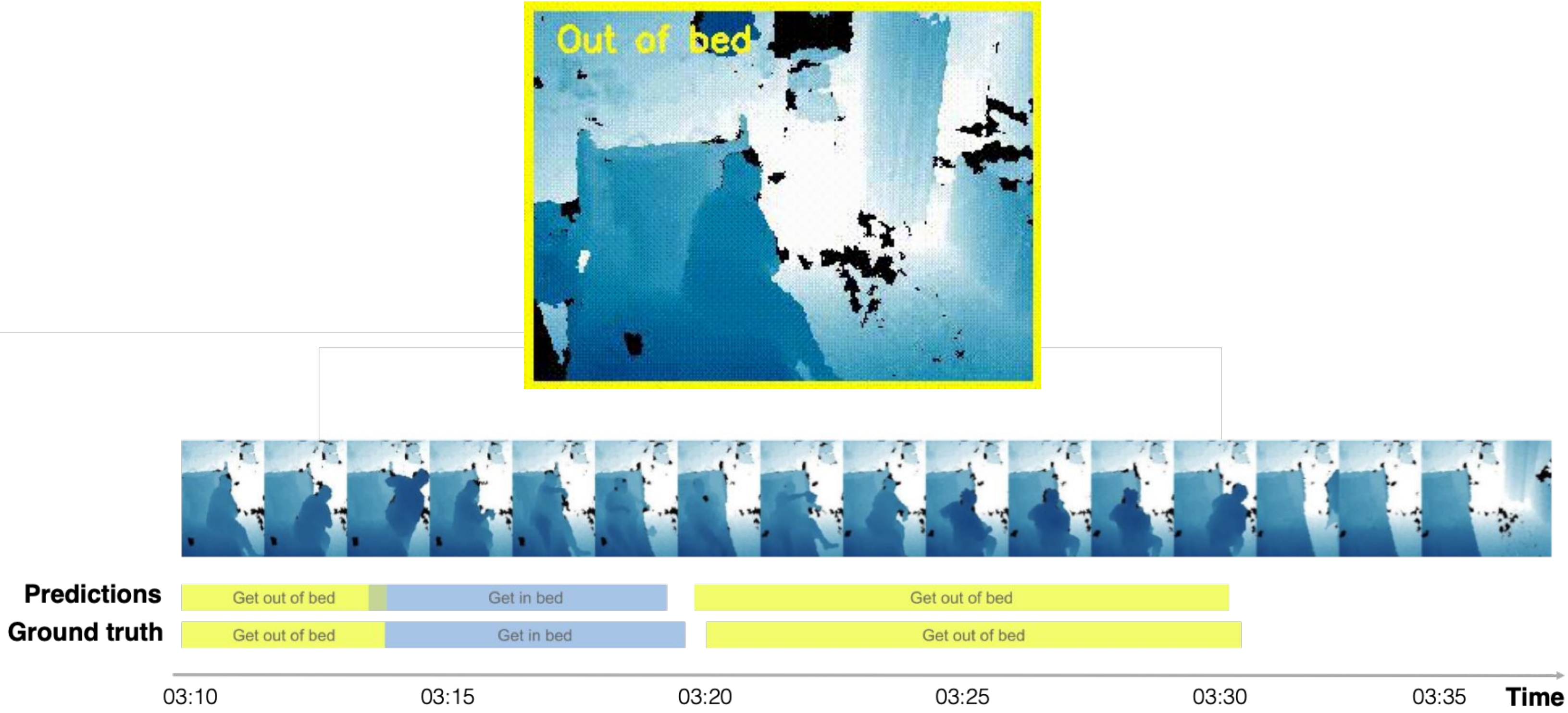


Detecting patient mobilization activities in the ICU



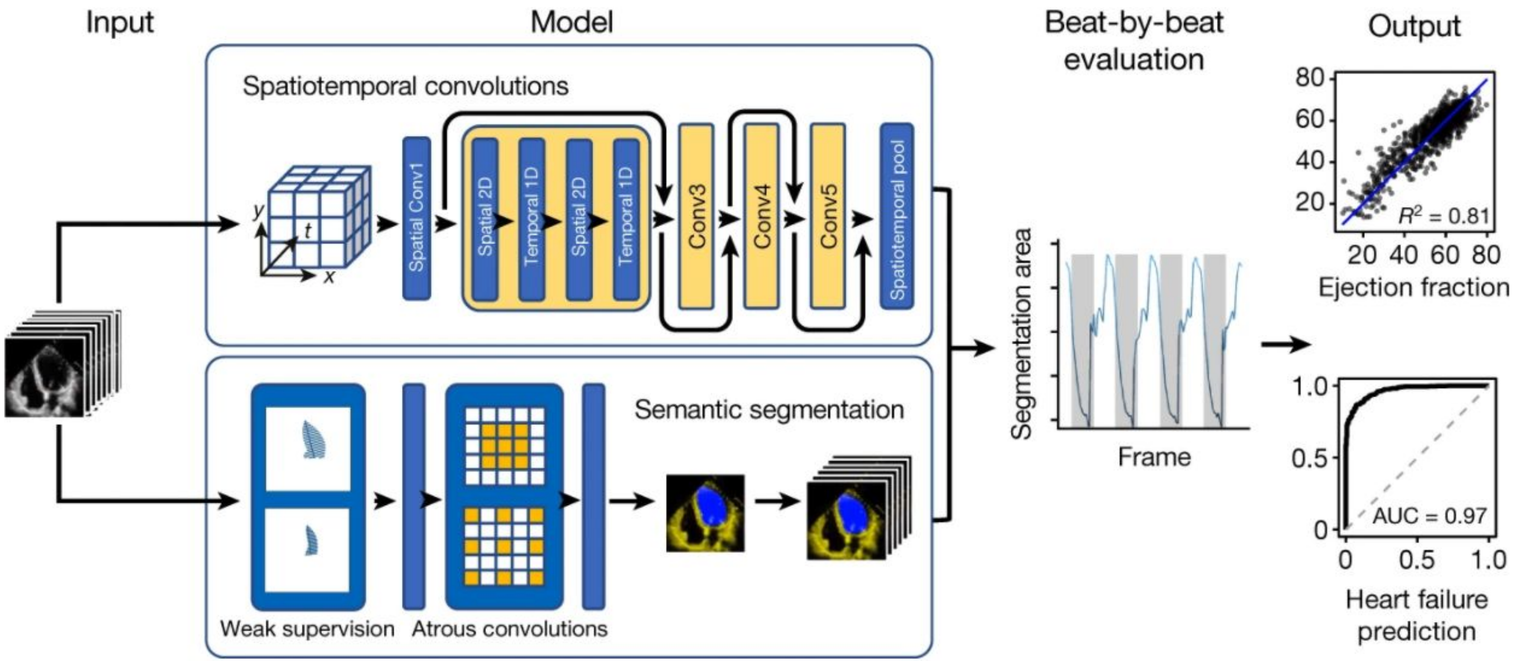
Yeung*,Salipur*, et al. A Computer Vision System for Deep Learning-Based Detection of Patient Mobilization Activities in the ICU. npj Digital Medicine, 2019.

Detecting patient mobilization activities in the ICU



Yeung*,Salipur*, et al. A Computer Vision System for Deep Learning-Based Detection of Patient Mobilization Activities in the ICU. npj Digital Medicine, 2019.

Predicting ejection fraction in echocardiograms



Ouyang et al. Video-based AI for beat-to-beat assessment of cardiac function. Nature, 2020.

Summary

Finished up advanced deep learning models for visual recognition tasks

- Classification
- Semantic segmentation
- **Object detection**
- **Instance segmentation**
- **3D and Video**

Will revisit some of these later with multimodal models and weakly / self- / un-supervised paradigms

Next time: Introduction to Electronic Health Records