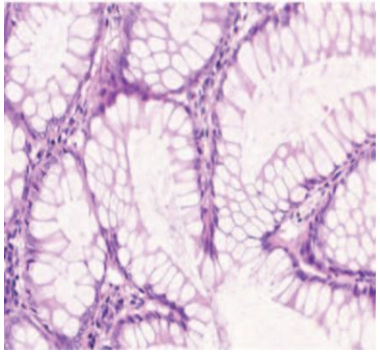
Lecture 5:
Medical Images:
3D and Video

Announcements

- A1 due Tue 10/6
- Project proposal due Fri 10/9

Last Time:
Richer visual recognition tasks: segmentation and detection

Classification

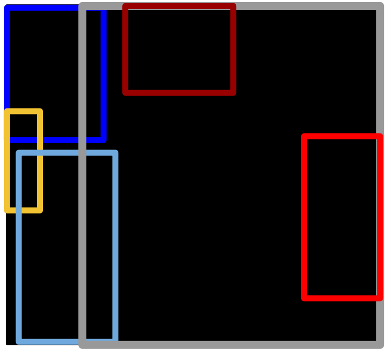


Output:
one category label for image (e.g., colorectal glands)

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

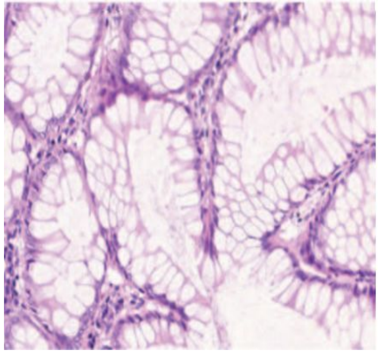
Output:
Category label and instance label for each pixel in the image

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

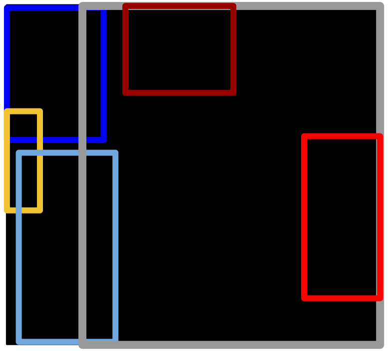


Output:
one category label for image (e.g., colorectal glands)

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

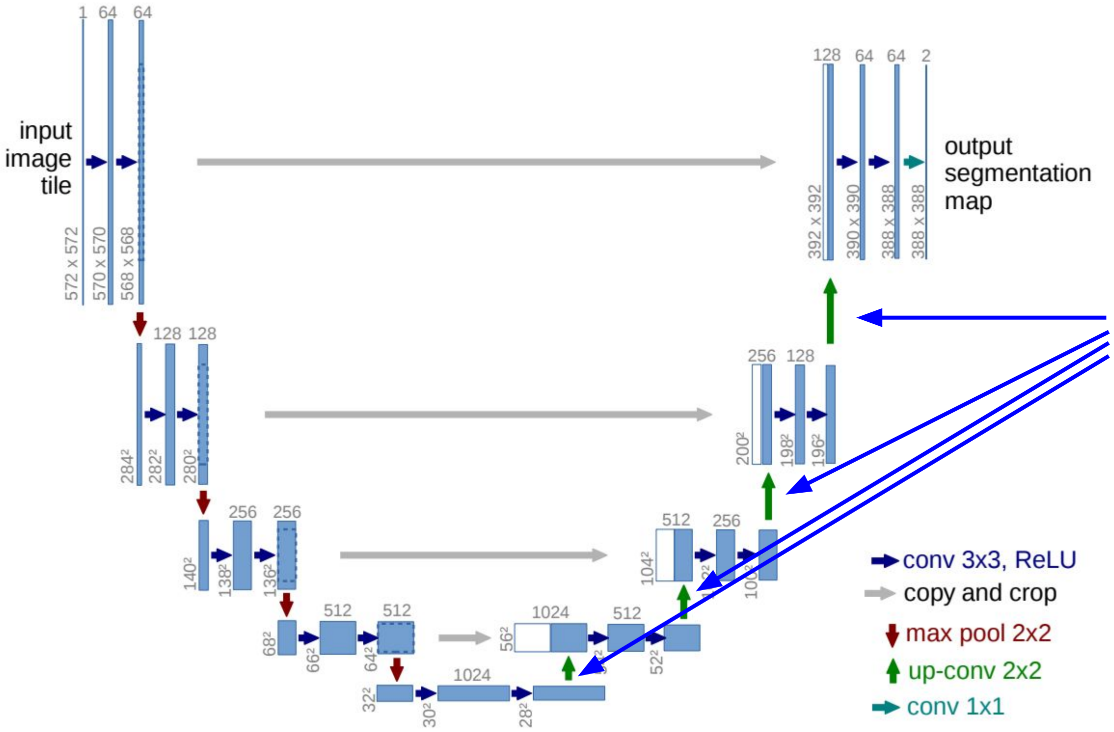
Instance Segmentation

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

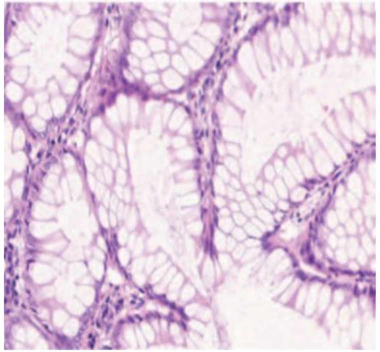


Up-convolutions to go from the global information encoded in highest-level features, back to individual pixel predictions

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

Last Time:
Richer visual recognition tasks: segmentation and detection

Classification

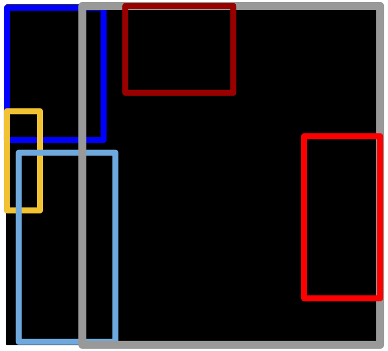


Output:
one category label for image (e.g., colorectal glands)

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

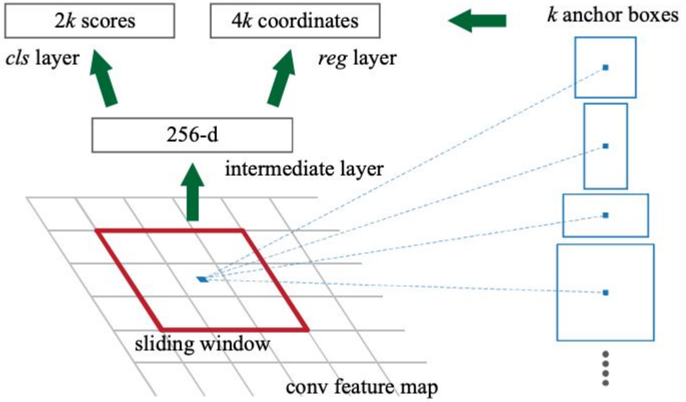
Output:
Category label and instance label for each pixel in the image

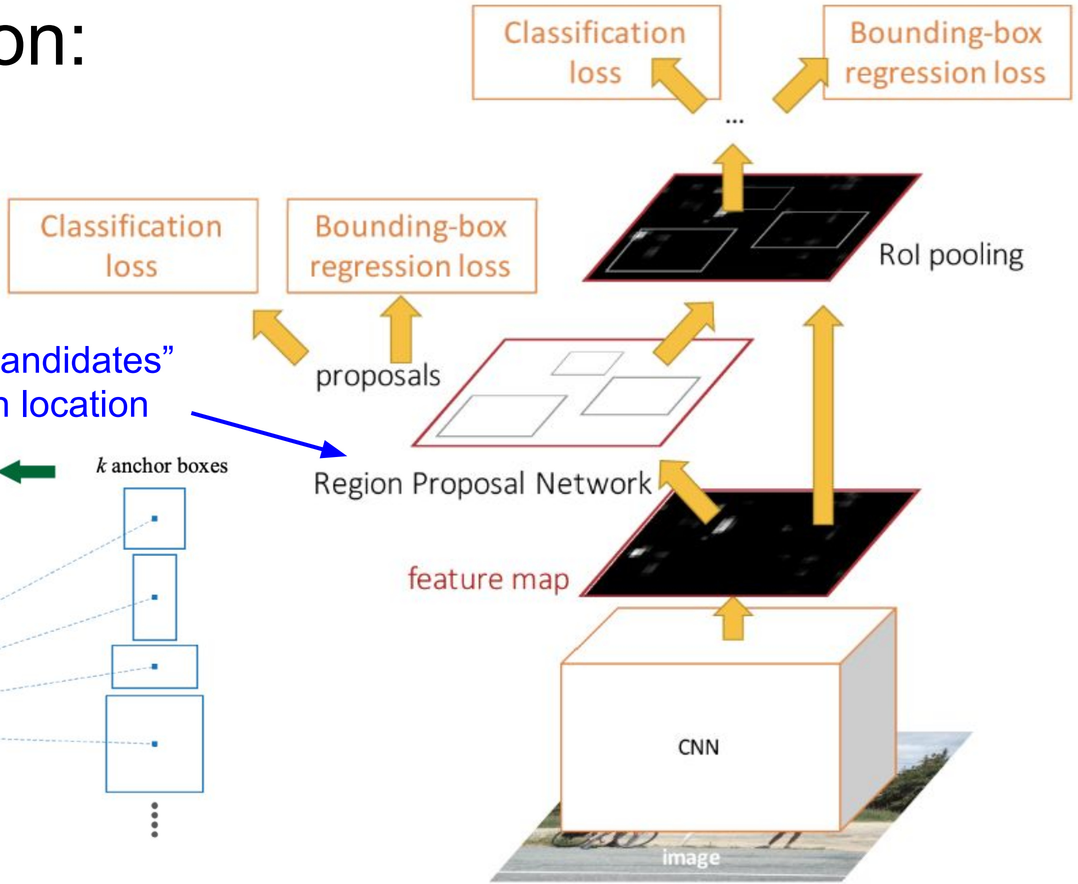
Distinguishes between different instances of an object

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

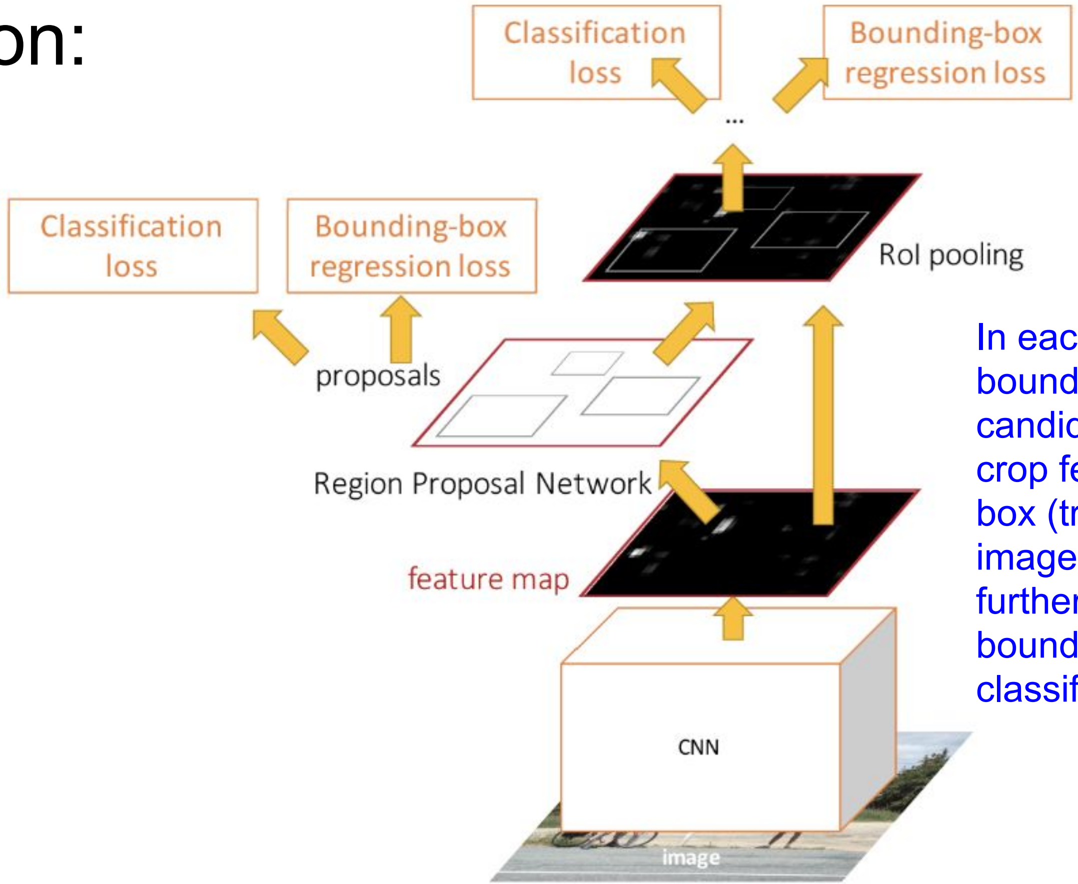
Object detection: Faster R-CNN

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



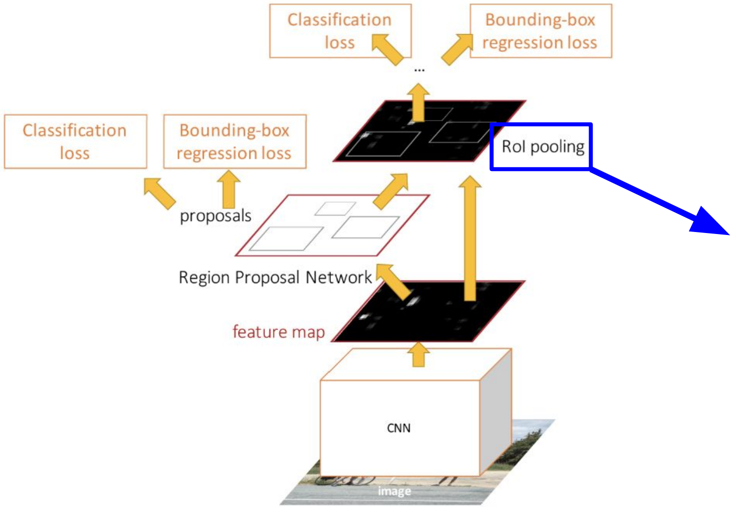


Object detection: Faster R-CNN



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



"Snap" to grid cells

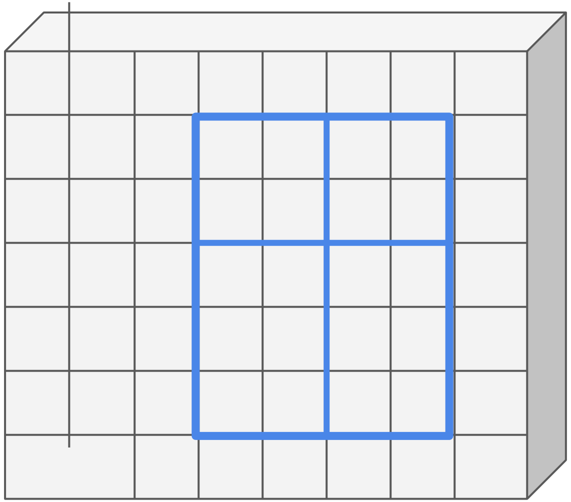
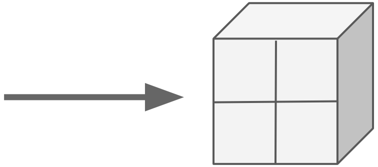


Image features

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

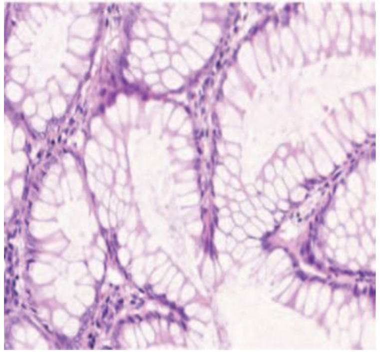
Max-pool within each subregion



Girshick, "Fast R-CNN", ICCV 2015.

Last Time:
Richer visual recognition tasks: segmentation and detection

Classification

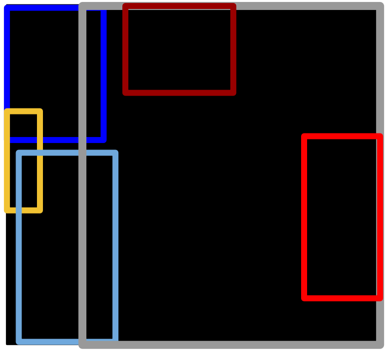


Output:
one category label for image (e.g., colorectal glands)

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

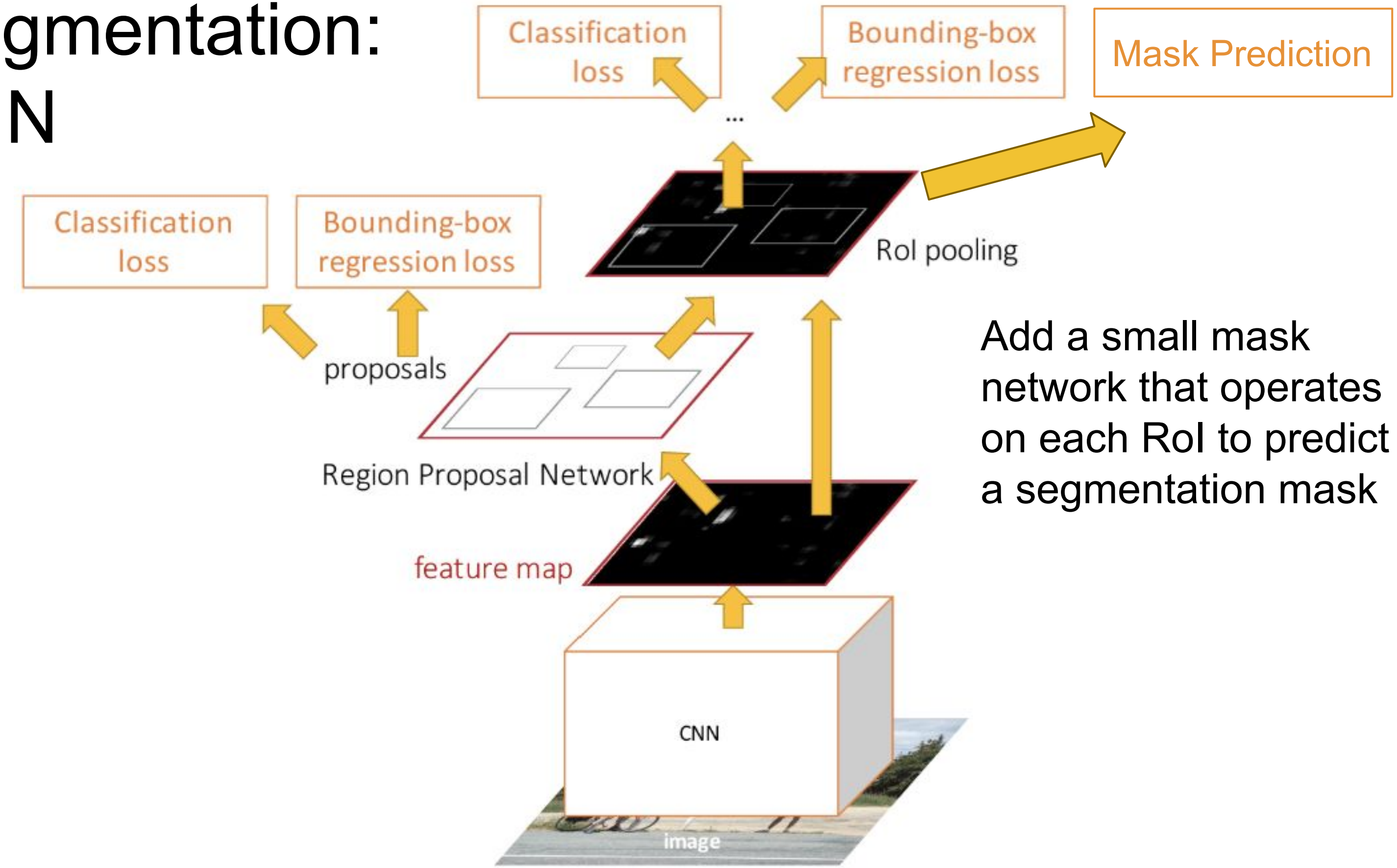
Instance Segmentation

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

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!

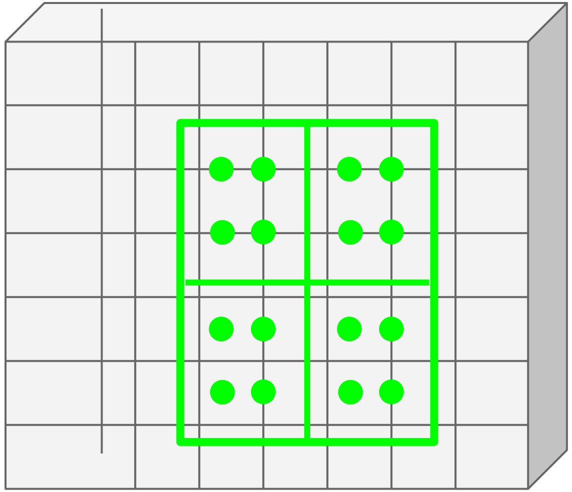


Image features
(e.g. 512 x 20 x 15)

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

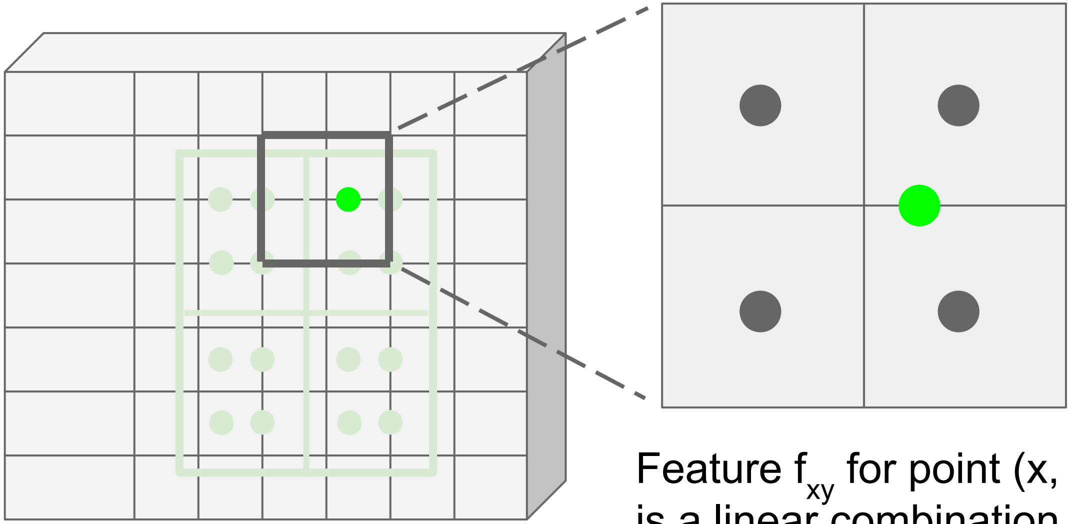


Image features

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)

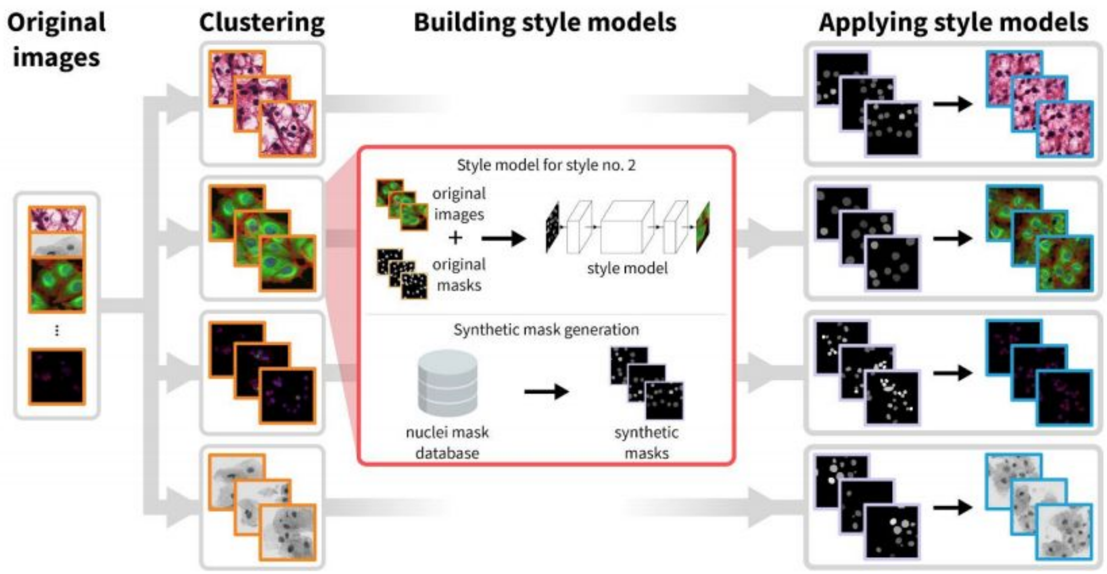
AP for small, medium, large objects

	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

Many interesting extensions

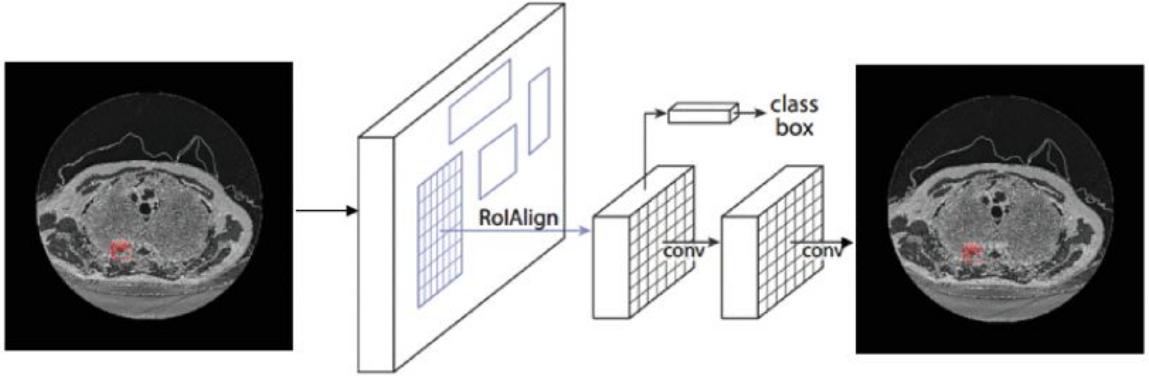
- E.g. Hollandi et al. 2019
  - Used “style transfer” approaches for rich data augmentation
  - Refined Mask-RCNN instance segmentation results with further U-Net-based boundary refinement



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 coming up!

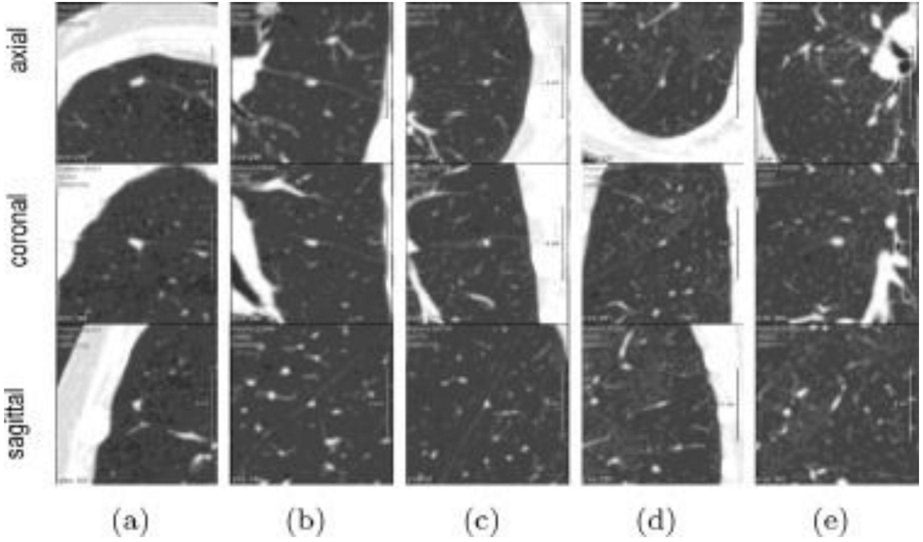
Liu et al. Segmentation of Lung Nodule in CT Images Based on Mask R-CNN. 2018.

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

How do we handle 3D data?

Recall: Ciompi et al. 2015

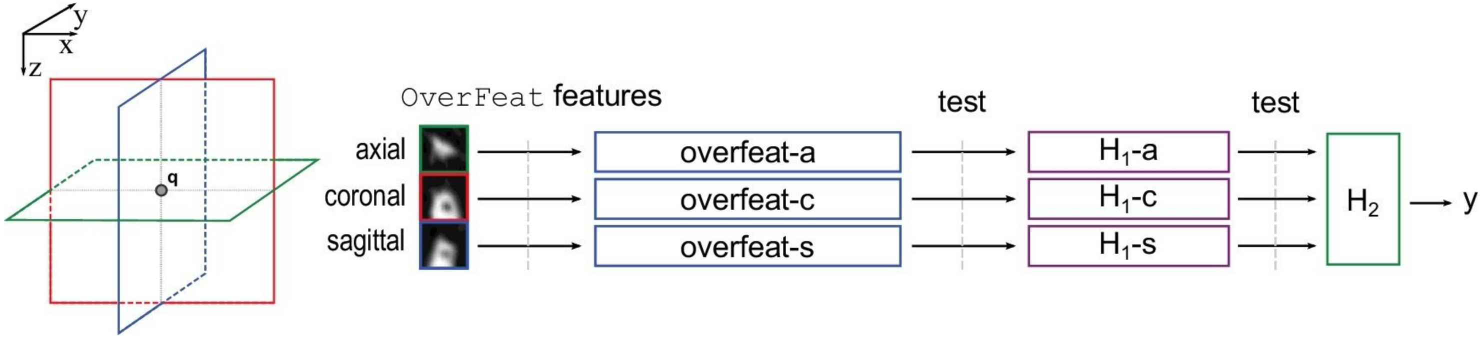
- Task: classification of lung nodules in 3D CT scans as peri-fissural nodules (PFN, likely to be benign) or not
- Dataset: 568 nodules from 1729 scans at a single institution. (65 typical PFNs, 19 atypical PFNs, 484 non-PFNs).
- Data pre-processing: prescaling from CT hounsfield units (HU) into [0,255].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

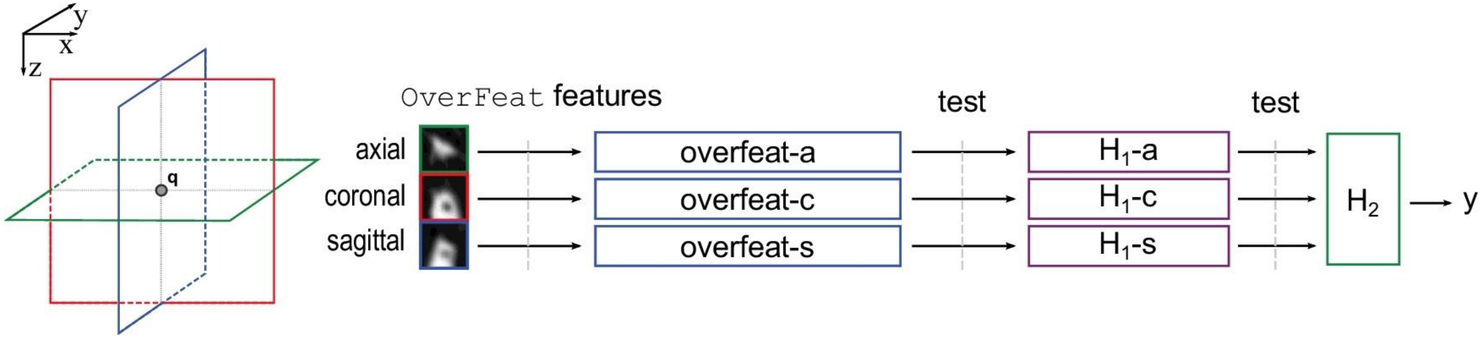


Ciompi et al. Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. Medical Image Analysis, 2015.

Ciompi et al. 2015

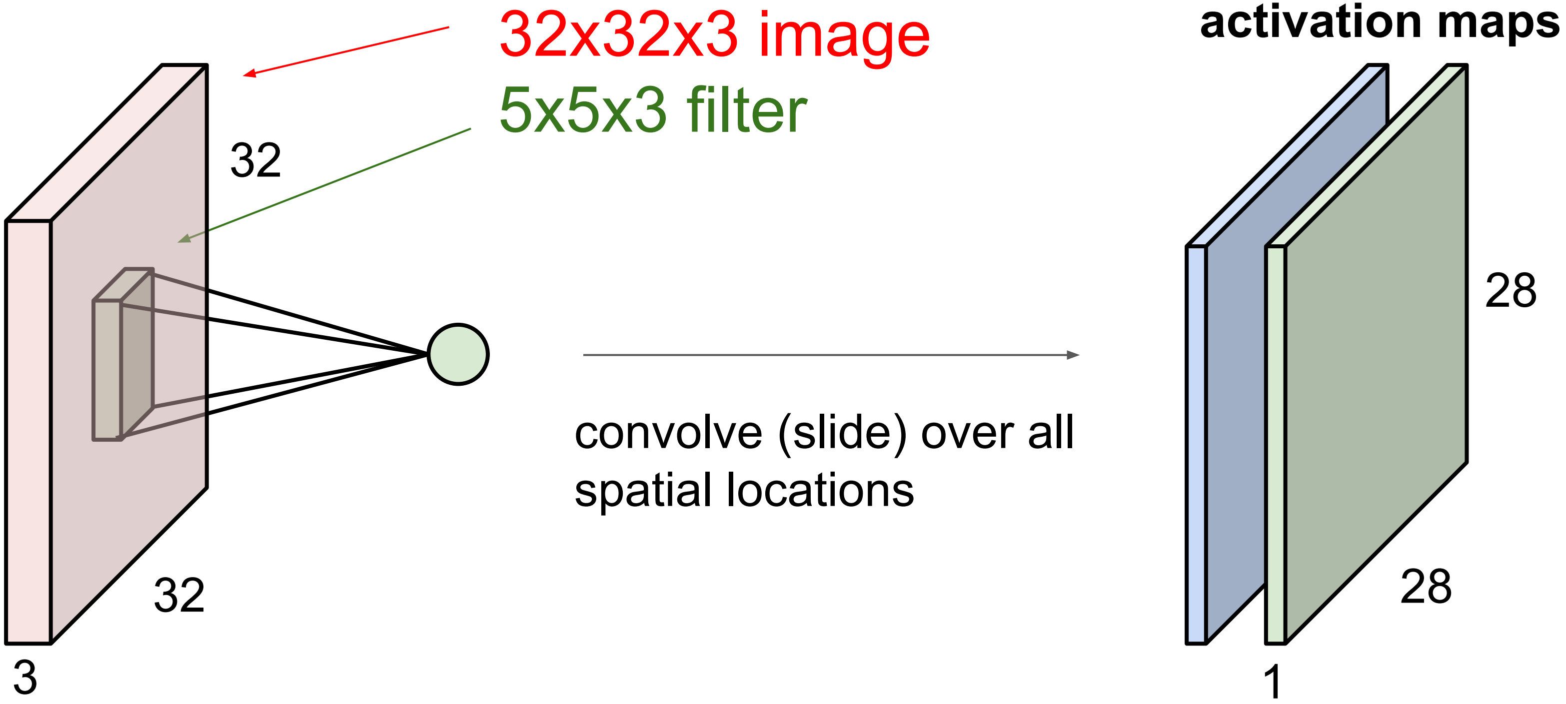
Another approach: 3D CNNs!

- 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).



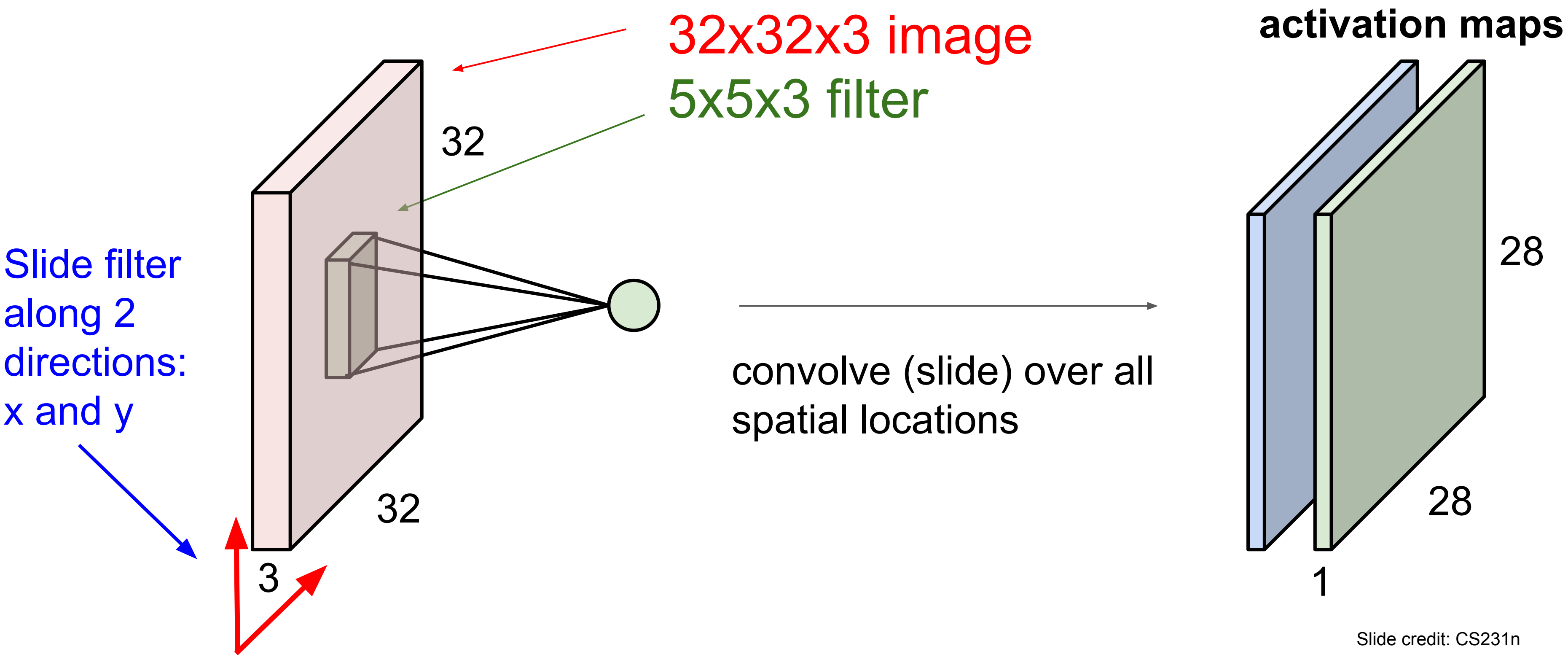
Ciompi et al. Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. Medical Image Analysis, 2015.

Remember 2D convolutions



Slide credit: CS231n

Remember 2D convolutions



Slide credit: CS231n

3D convolutions

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

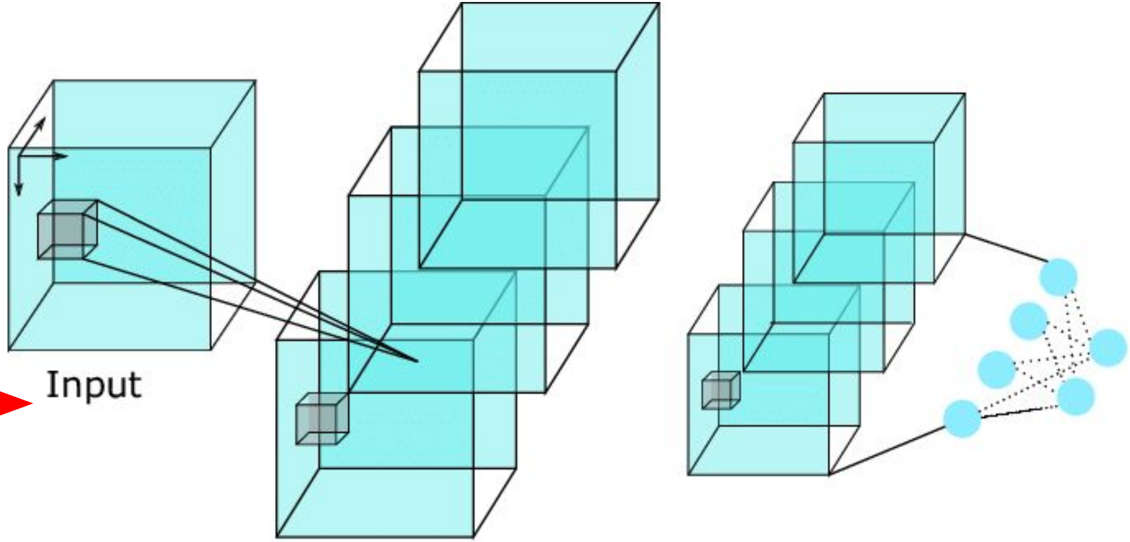


Figure credit:
https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

3D convolutions

When might you use 3D convolutions?

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

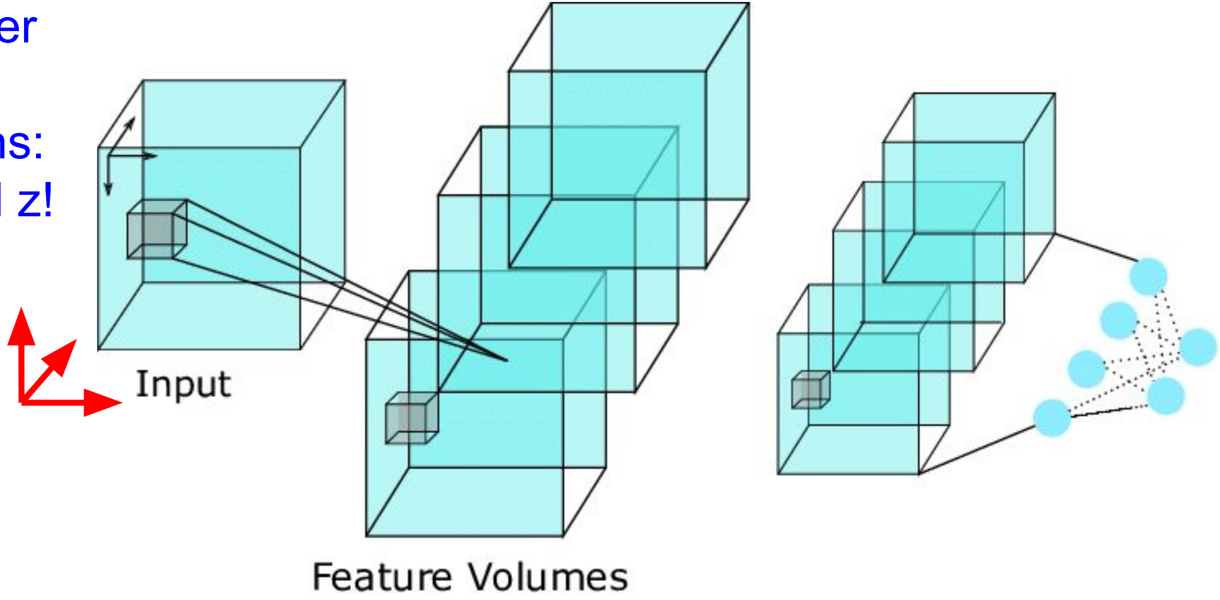


Figure credit: https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

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!

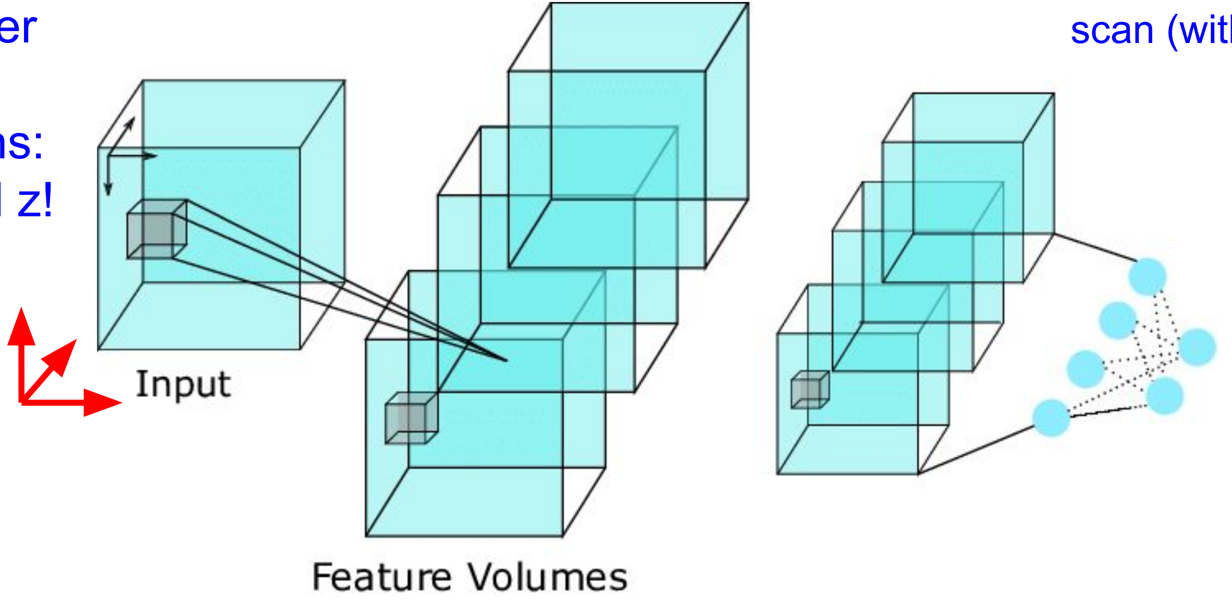
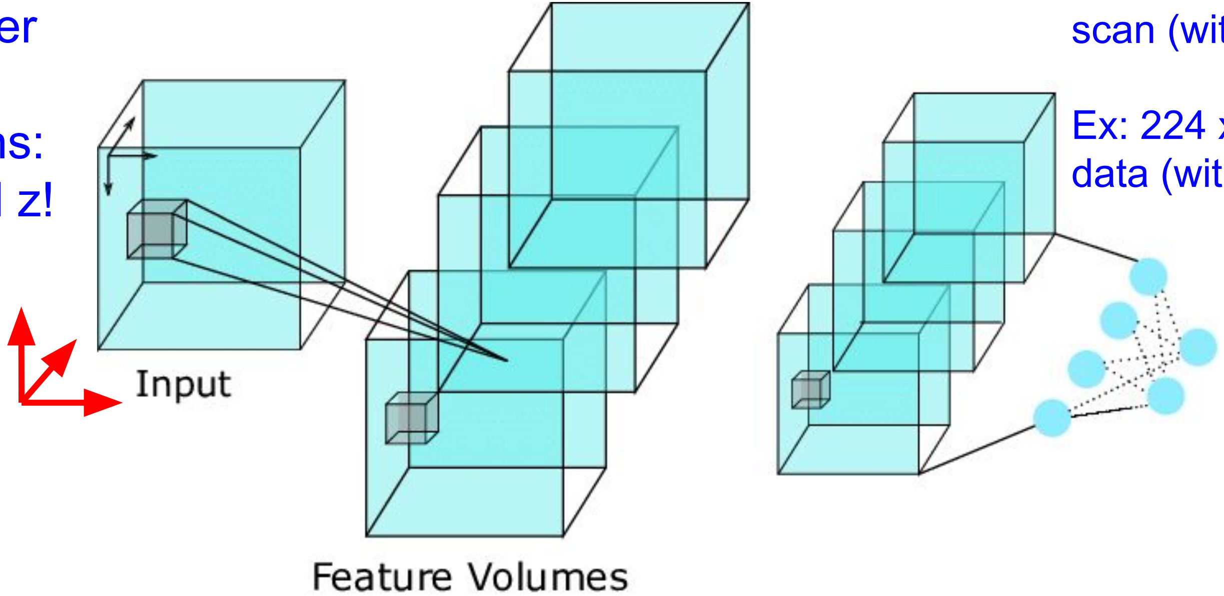


Figure credit:
https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

3D convolutions

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



When might you use 3D convolutions?

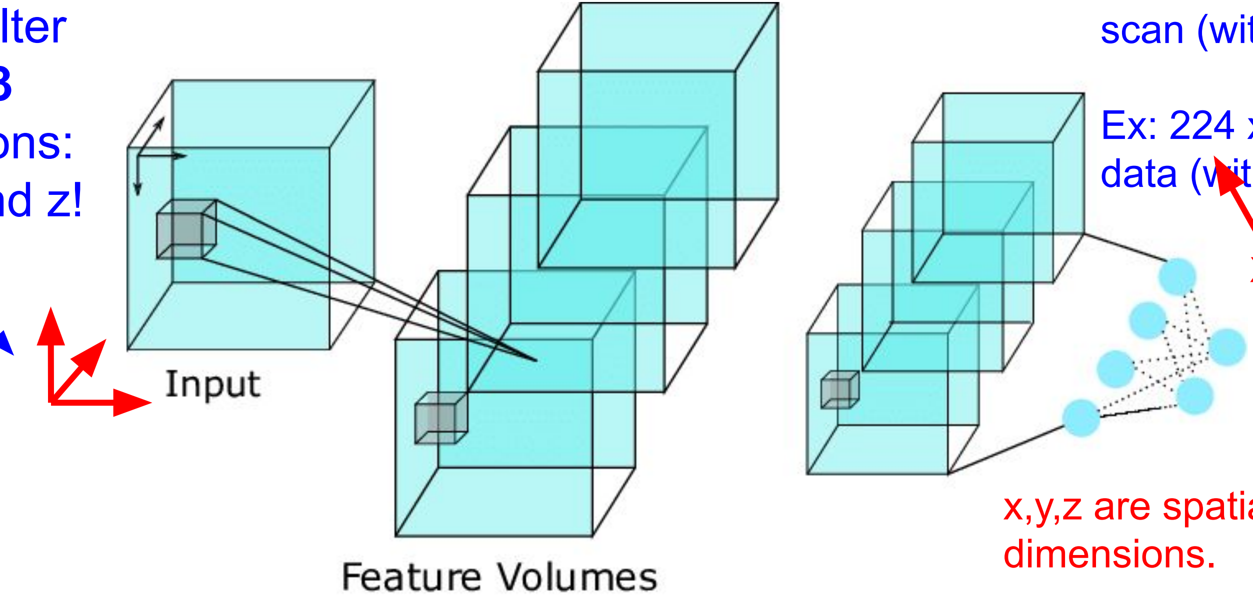
Ex: 224 x 224 x 1 x 256 3D CT scan (with 256 slices)

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

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)

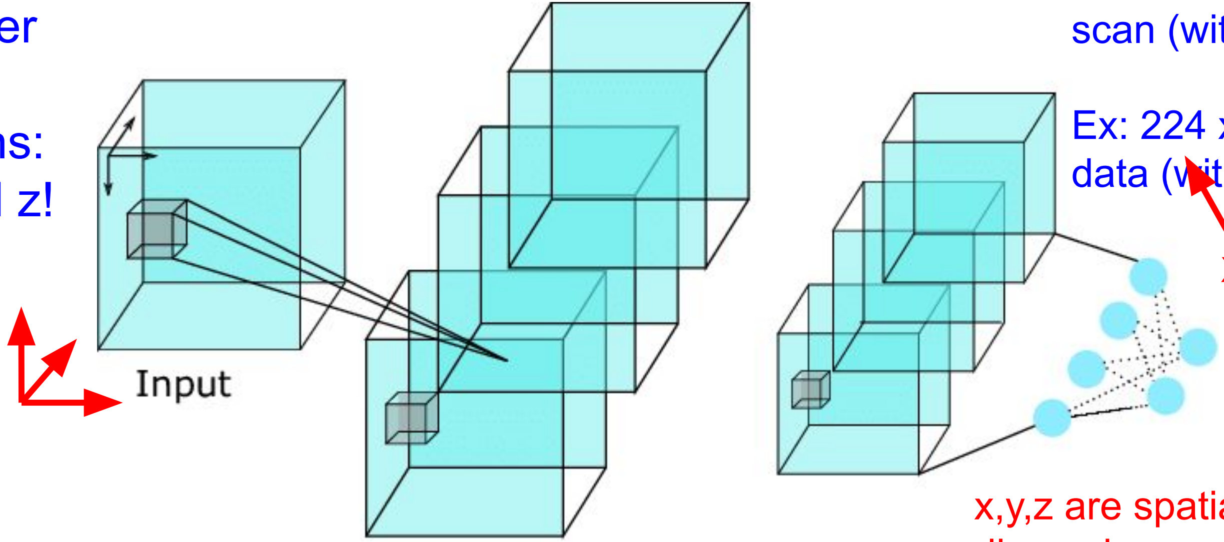
x y z

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

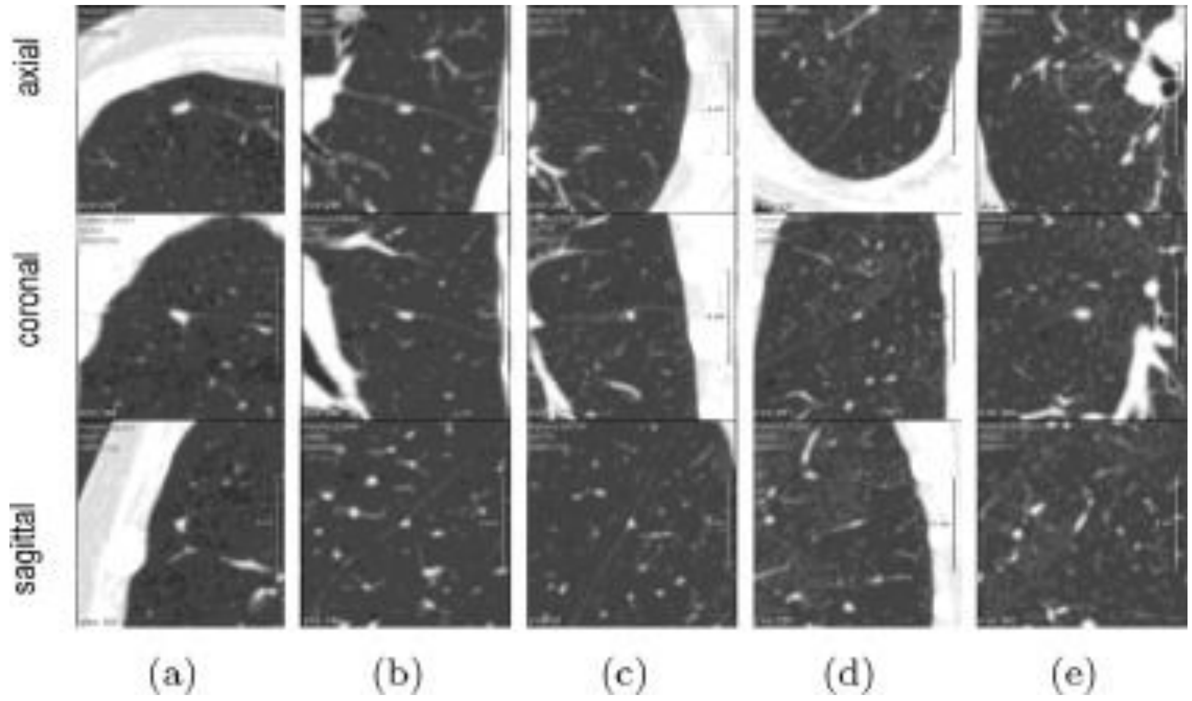
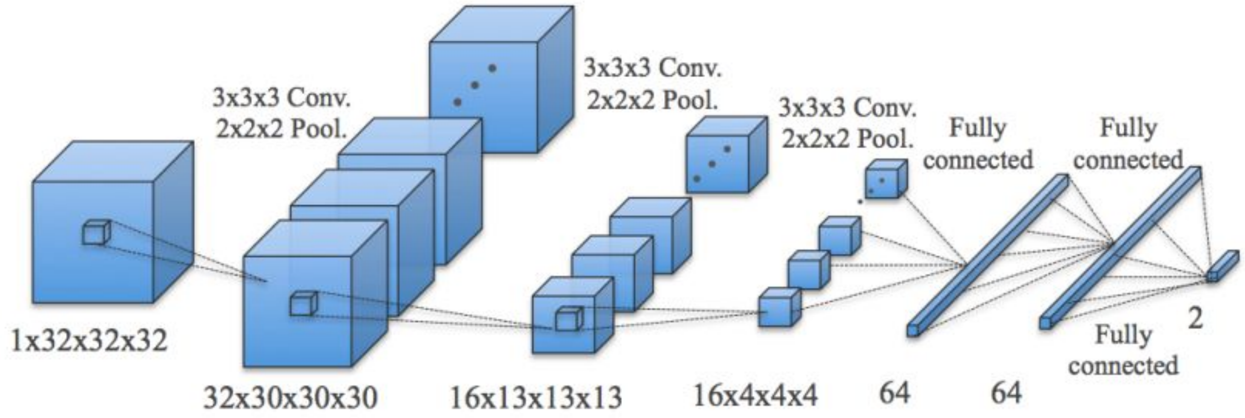


Figure credit: Ciompi et al. Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. Medical Image Analysis, 2015.

Huang et al. 2017

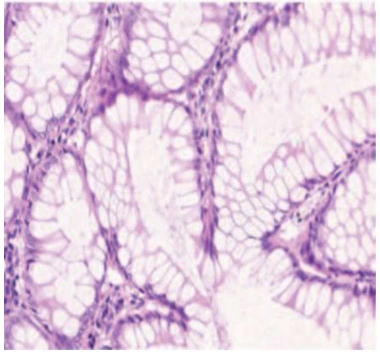
- Simple 3D CNN for lung nodule classification
- Used image processing approaches to extract candidate nodules, then 3D CNN to classify the surrounding volume
- Used the Lung Image Database Consortium (LIDC) Dataset, with 99 3D CT scans



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

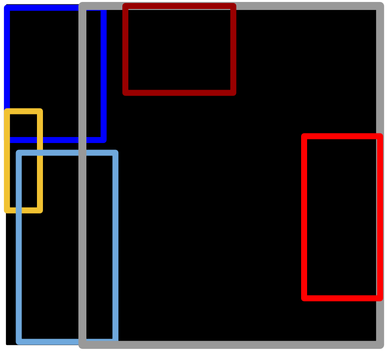


Output:
one category label for image (e.g., colorectal glands)

Semantic Segmentation

Output:
category label for each pixel in the image

Detection



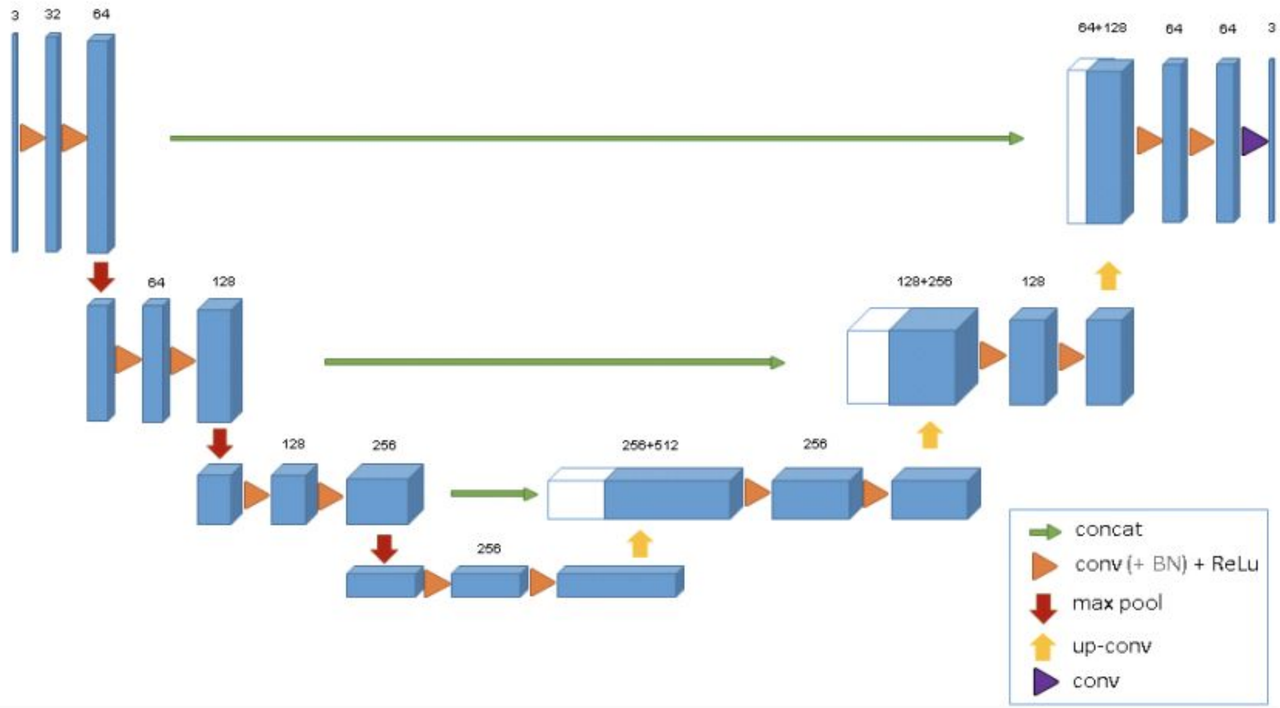
Output:
Spatial bounding box for each **instance** of a category object in the image

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

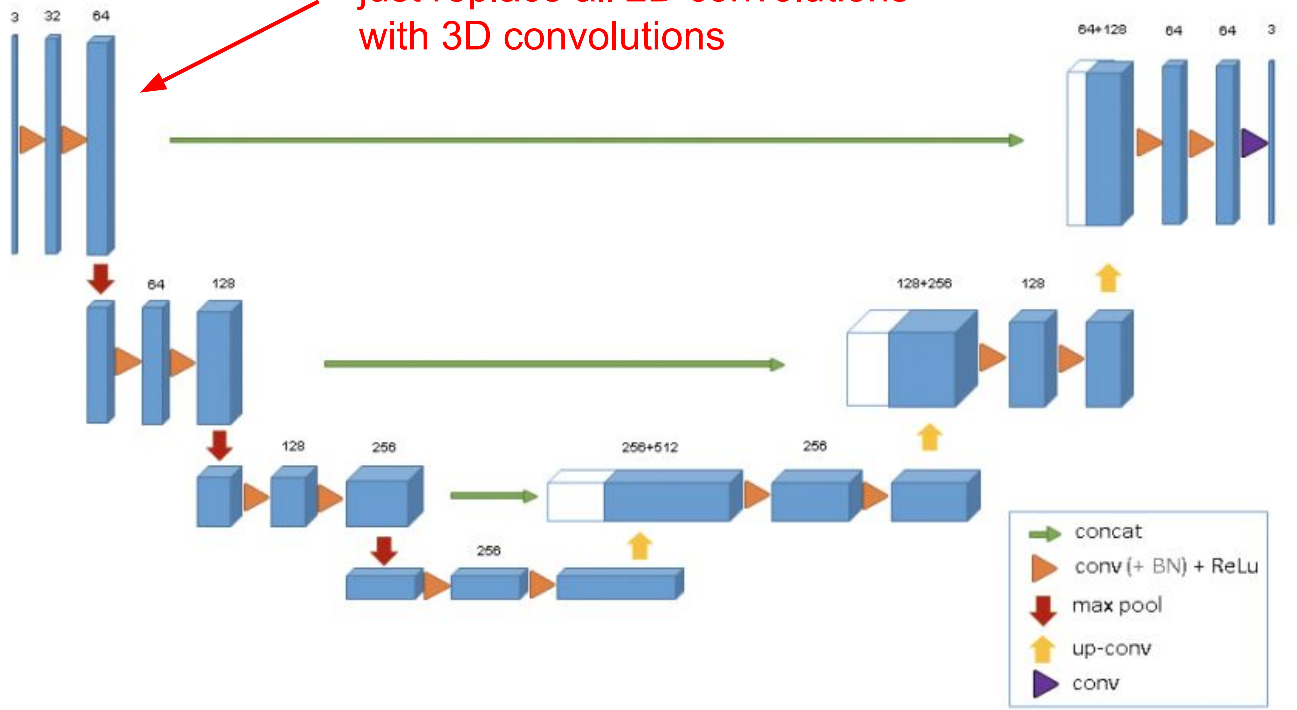
E.g. 3D U-Net



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

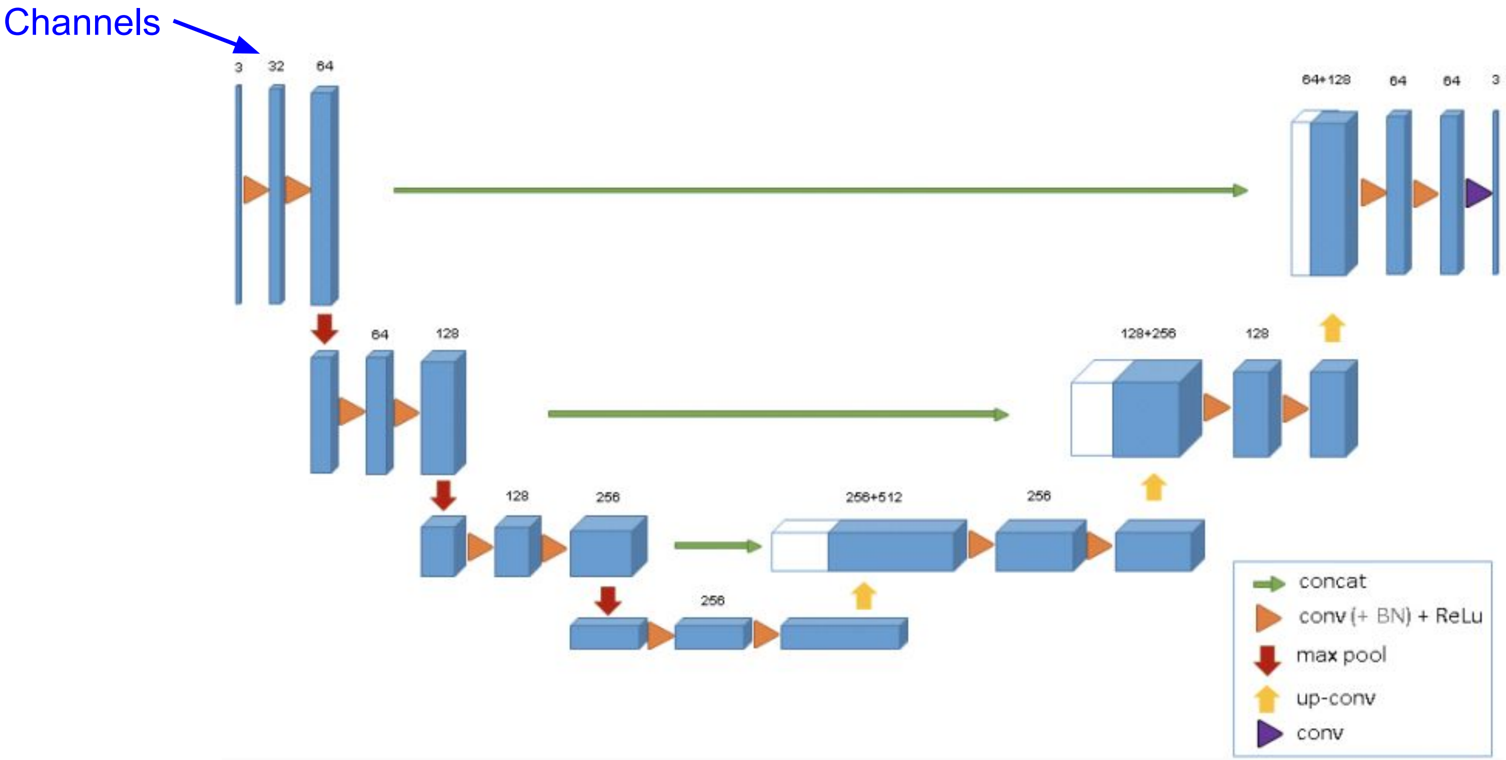
E.g. 3D U-Net

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.

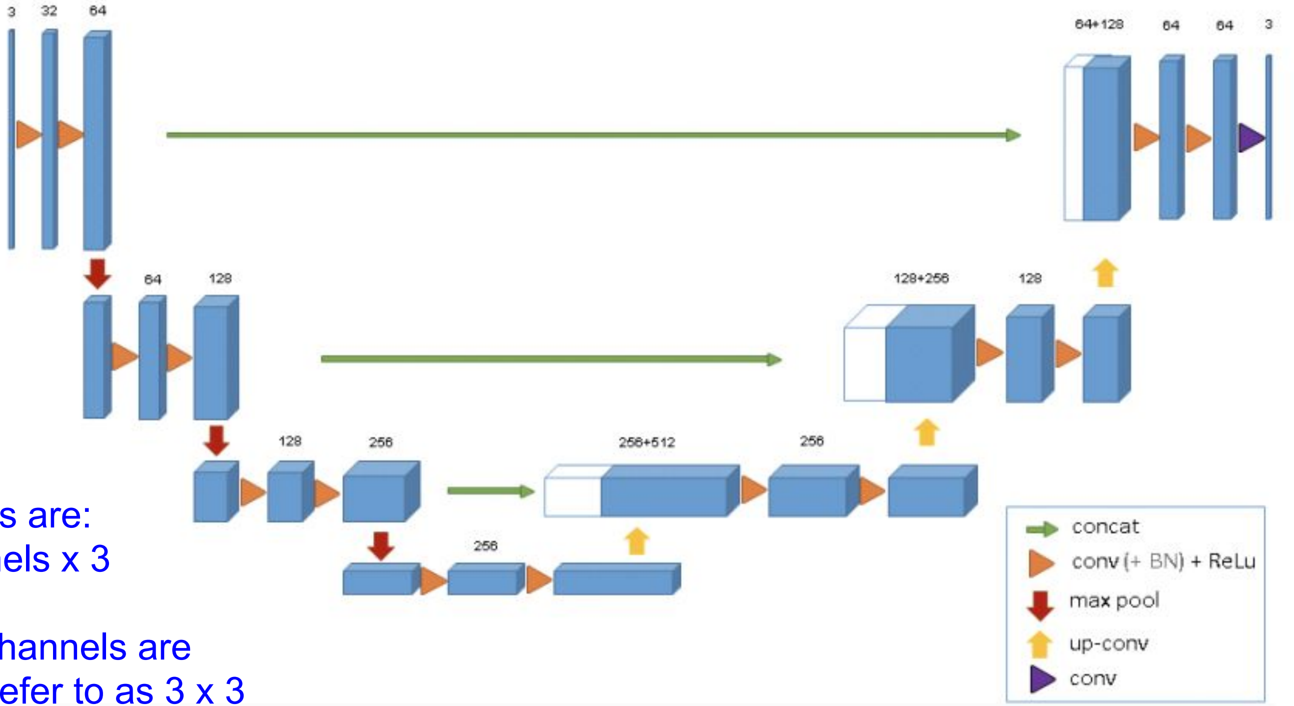
E.g. 3D U-Net



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



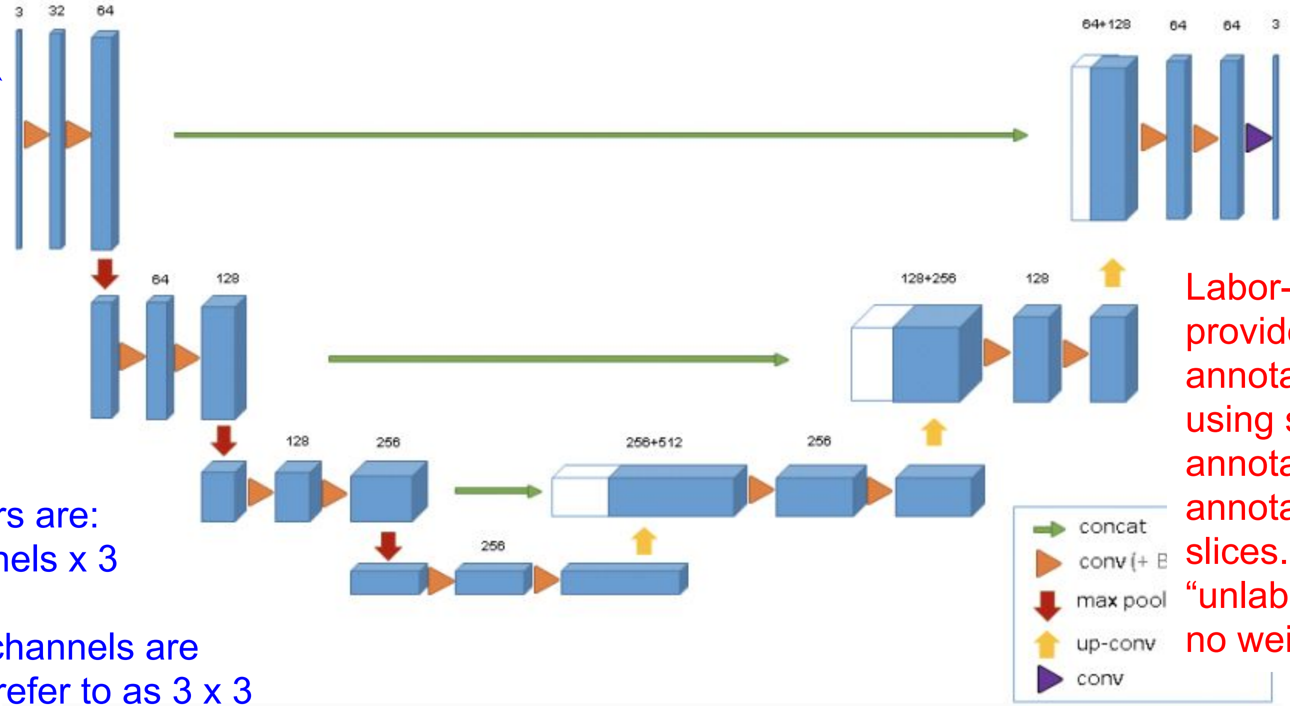
3D conv filters are:
3 x 3 x channels x 3

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



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

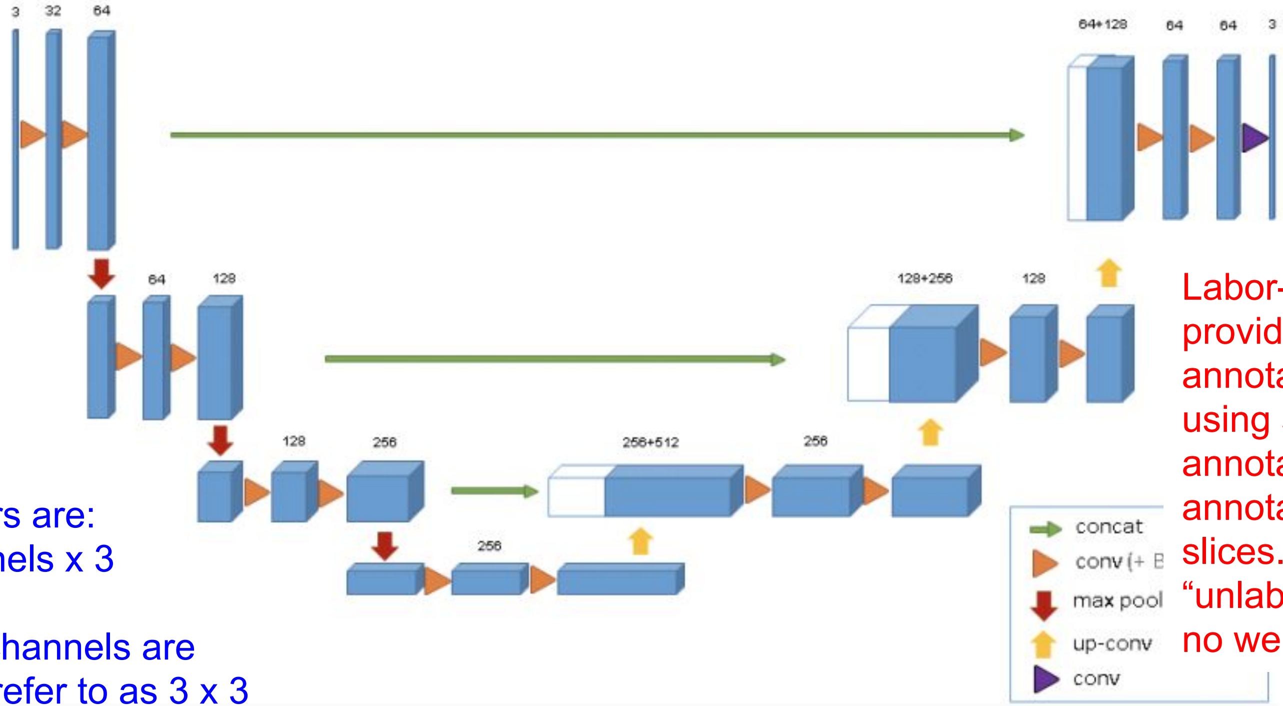
Sometimes channels are implicit, can refer to as 3 x 3 x 3 conv filter

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

E.g. 3D U-Net

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.

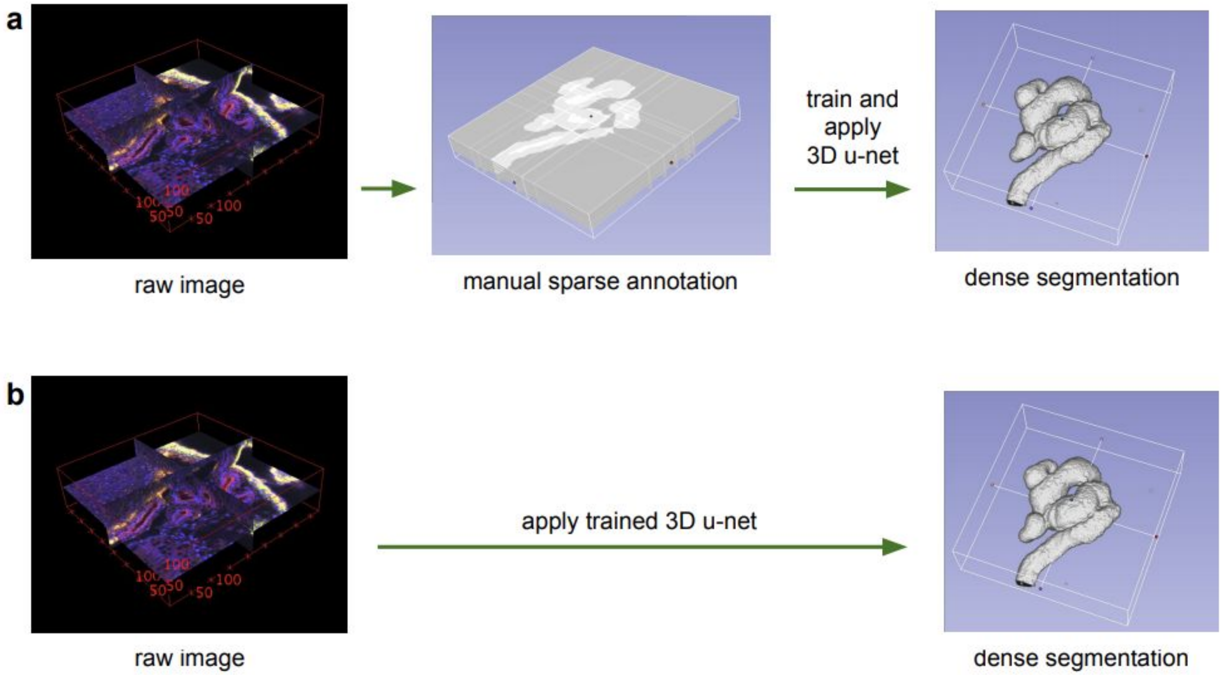
3D conv filters are:
3 x 3 x channels x 3

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

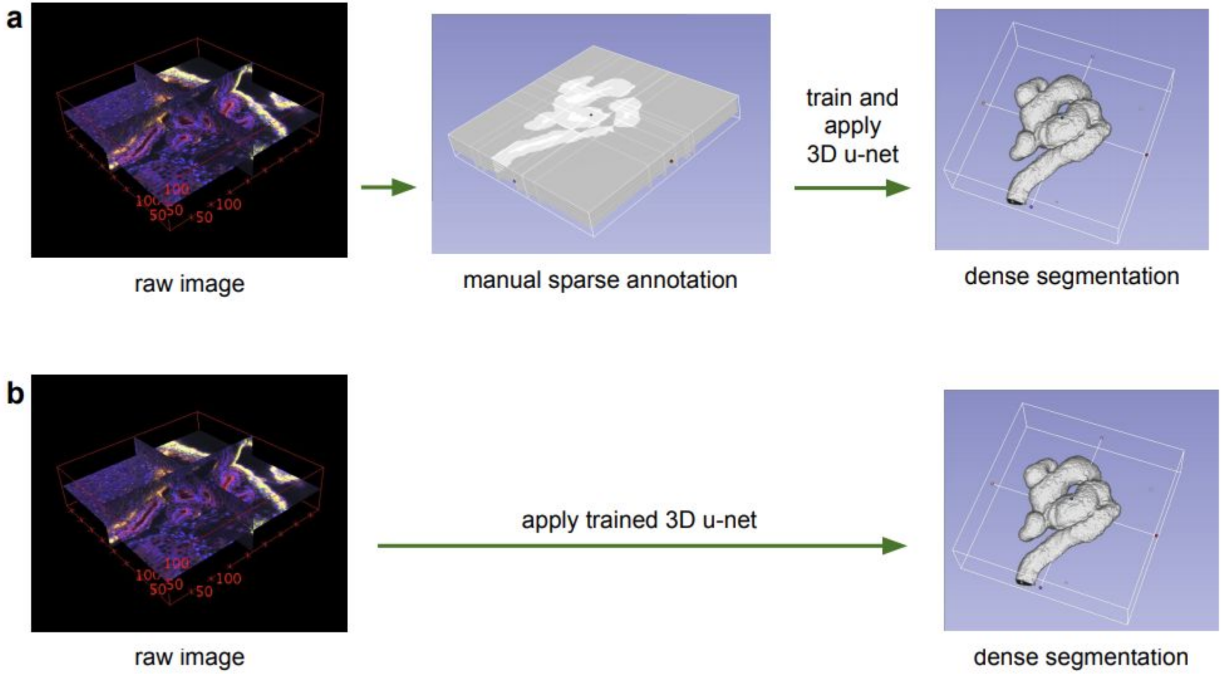


Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

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



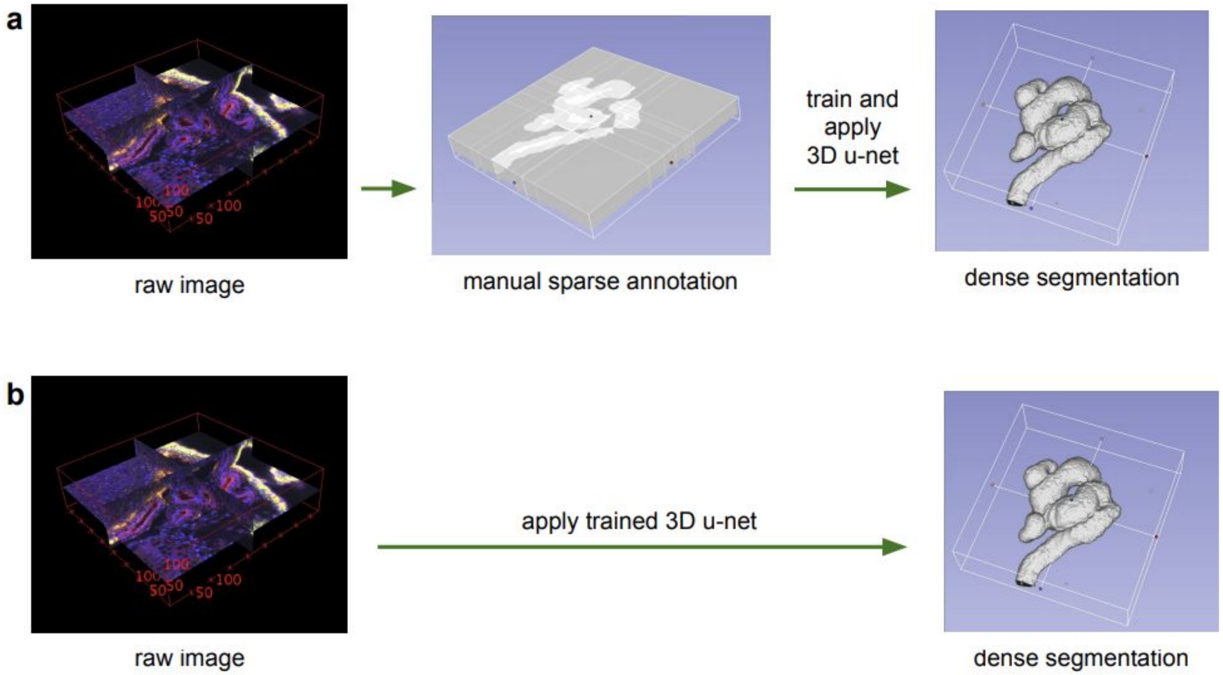
Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

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.



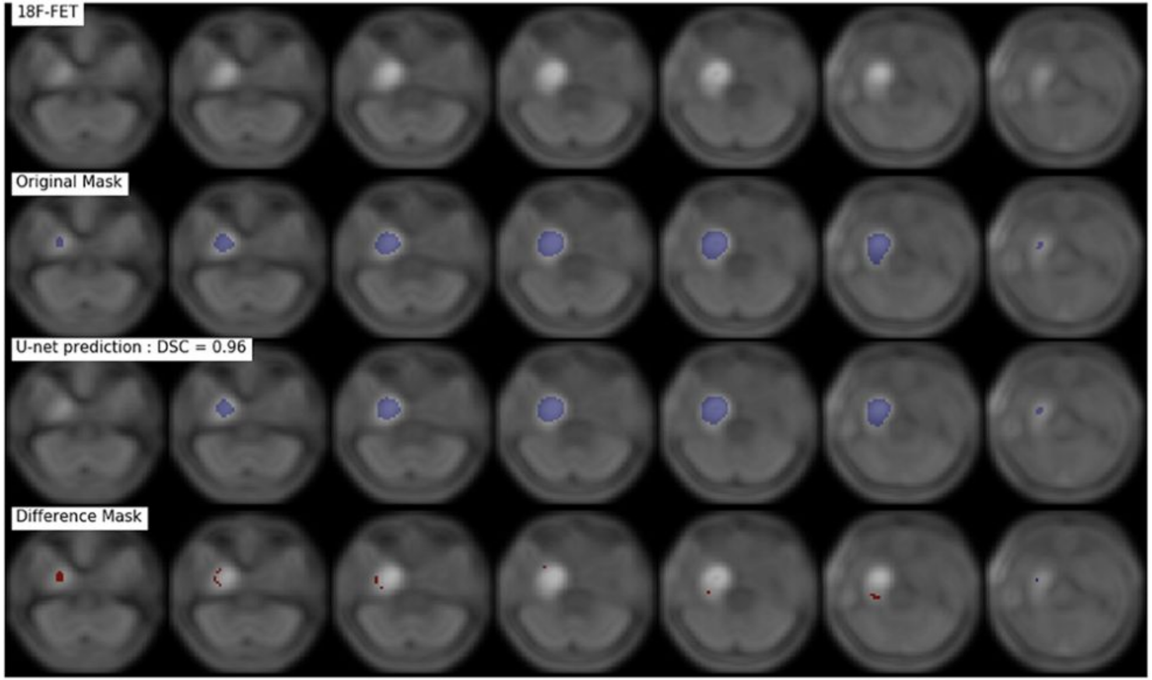
Cicek et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. MICCAI 2016.

Ex: Brain lesion segmentation

Training set: 37 PET scans (3D volumes)

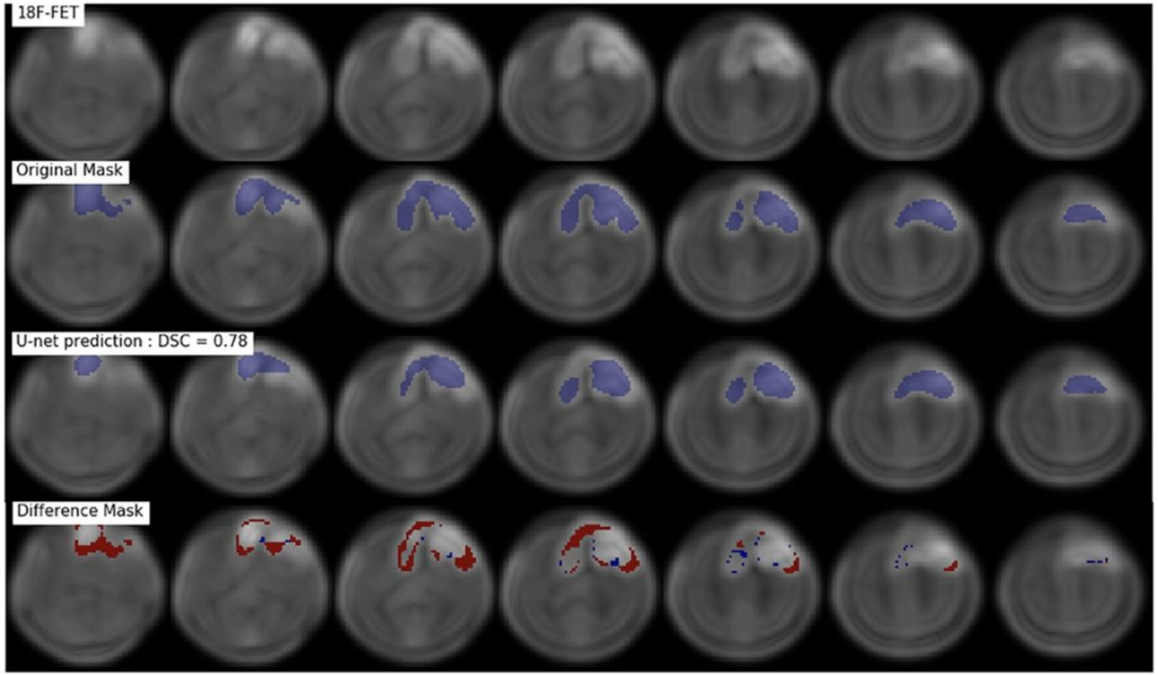
Evaluation set: 11 PET scans

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

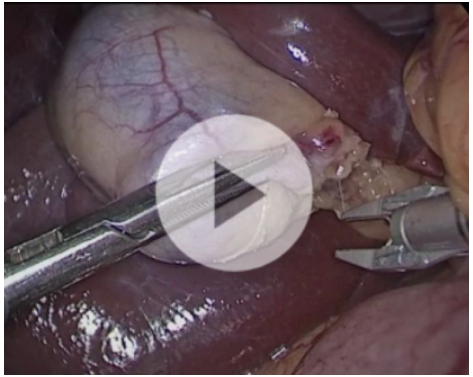


Blanc-Durand et al. Automatic lesion detection and segmentation of 18F-FET PET in gliomas: A full 3D U-Net convolutional neural network study. PLoS One, 2018.

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

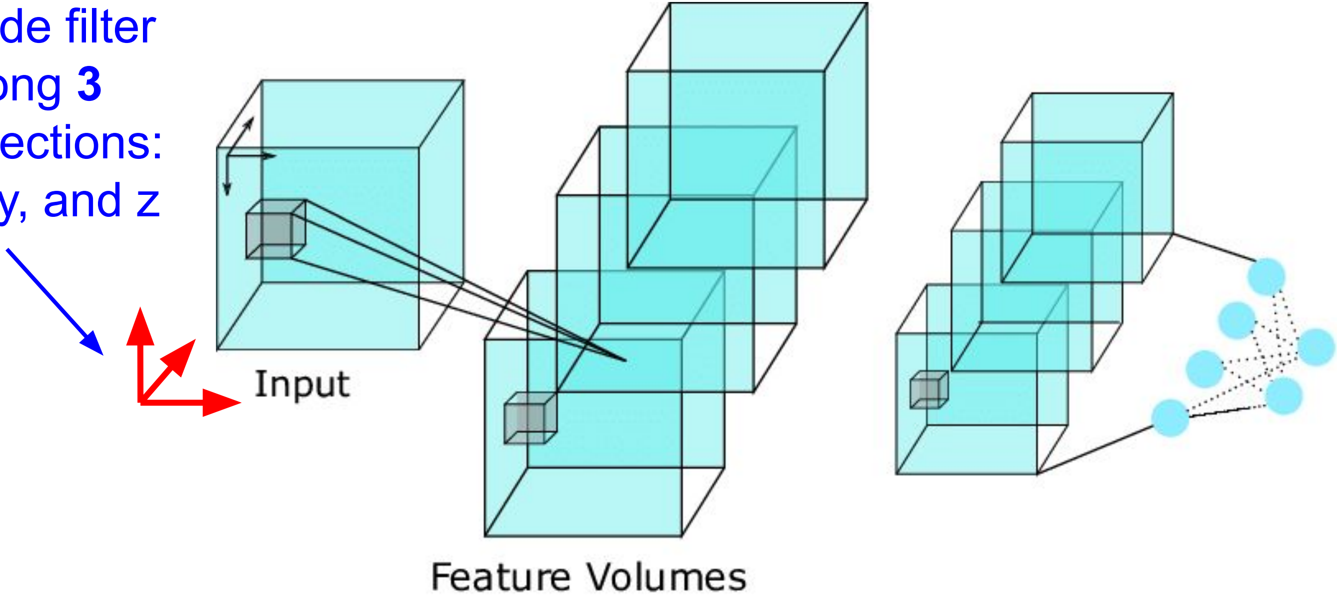
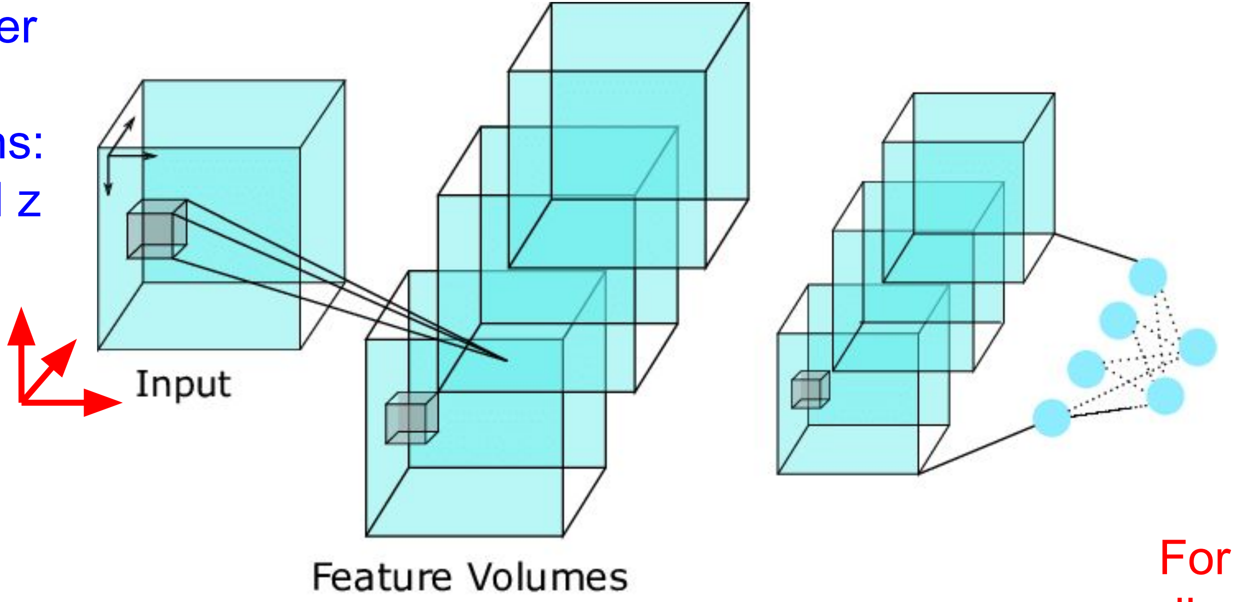


Figure credit:
https://www.researchgate.net/profile/Deepak_Mishra19/publication/330912338/figure/fig1/AS:723363244810254@1549474645742/Basic-3D-CNN-architecture-the-3D-filter-is-convolved-with-the-video-in-three-dimensions.png

Another approach: 3D convolutions

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

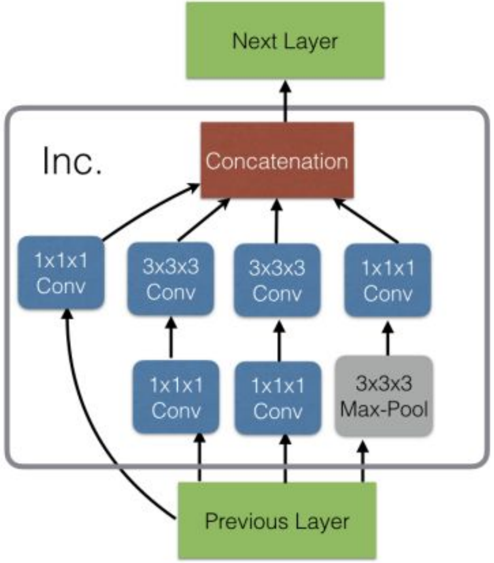


For video data, 3rd dimension is time

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

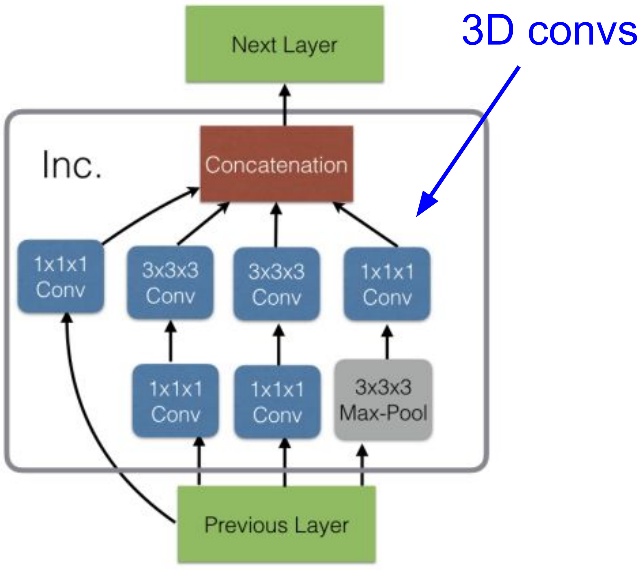
Inception Module (Inc.) w/
3D convolutions



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

I3D: 3D convolutional network for video data

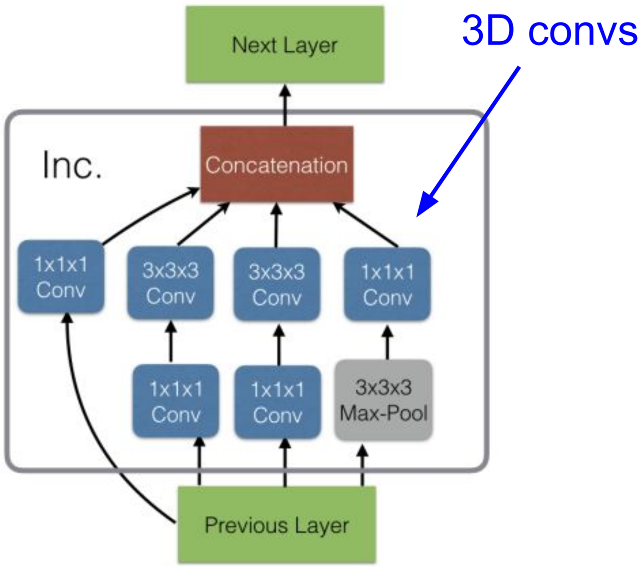
Inception Module (Inc.) w/
3D convolutions



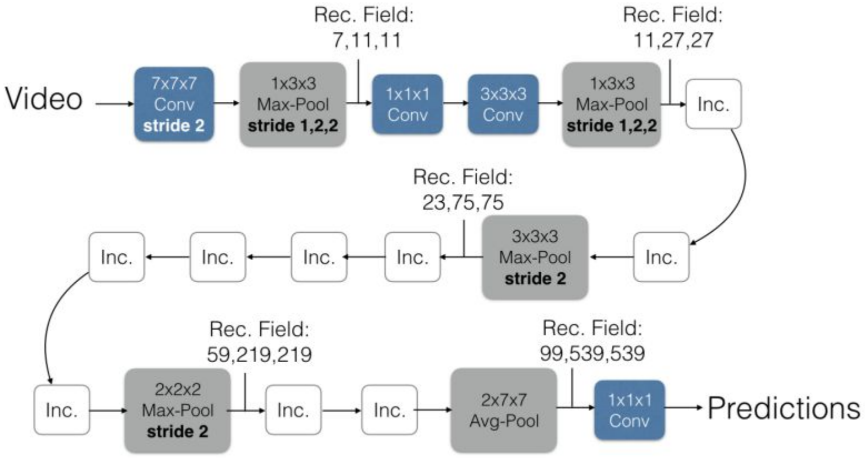
Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

I3D: 3D convolutional network for video data

Inception Module (Inc.) w/
3D convolutions



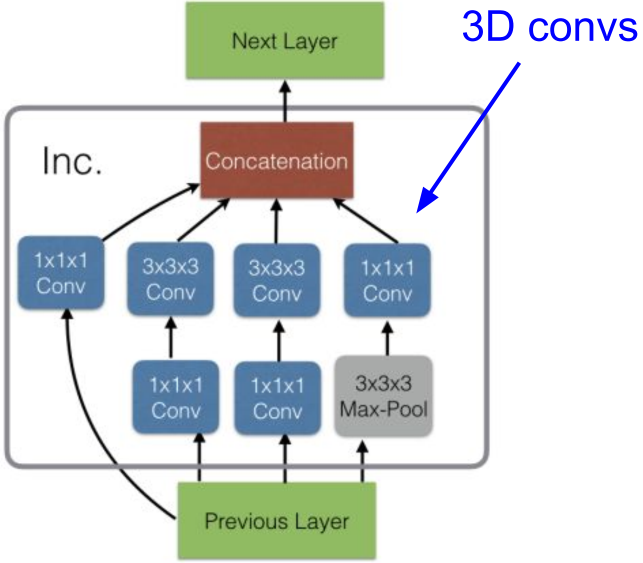
3D Inception Module used in Inception
Network (also known as GoogLeNet)



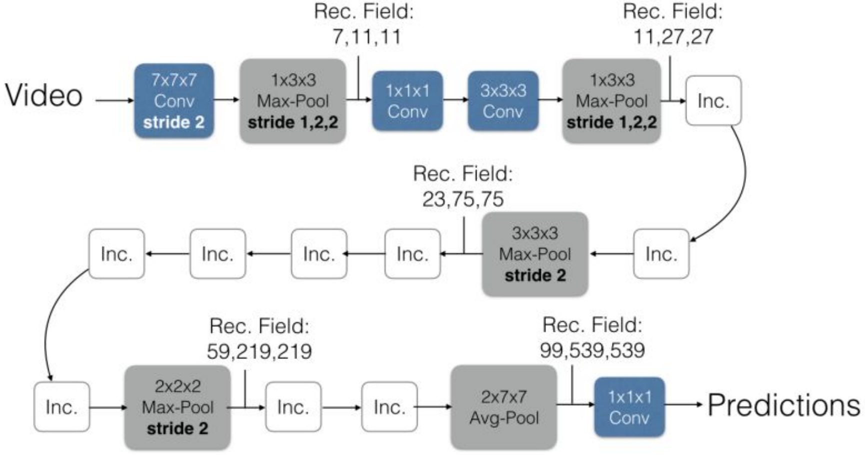
Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

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!

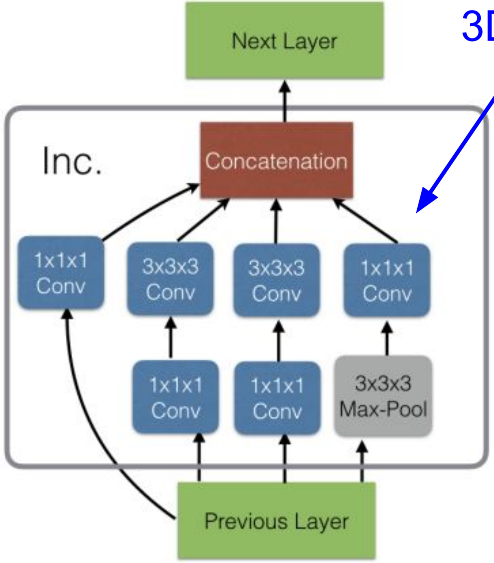
Carreira and Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR 2017.

I3D: 3D convolutional network for video data

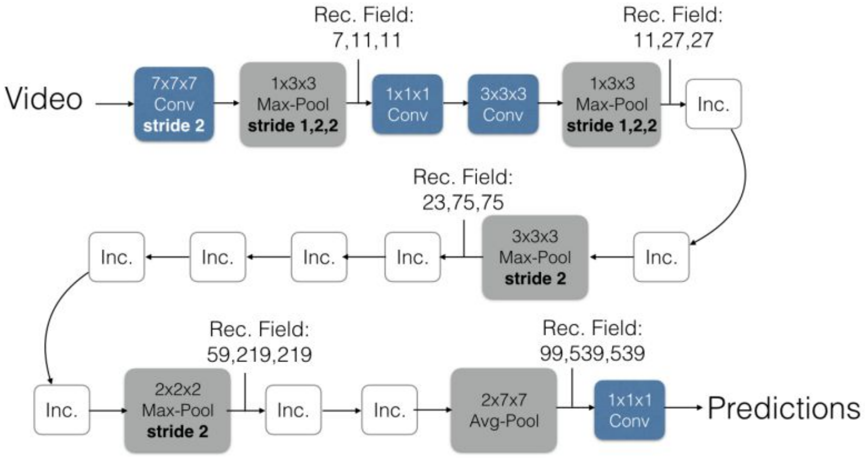
Note: in general, can 3D-ify many 2D architectures!

Inception Module (Inc.) w/ 3D convolutions

3D convs



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) often enhanced with optical flow

Two consecutive frames

Optical flow displacement vectors

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

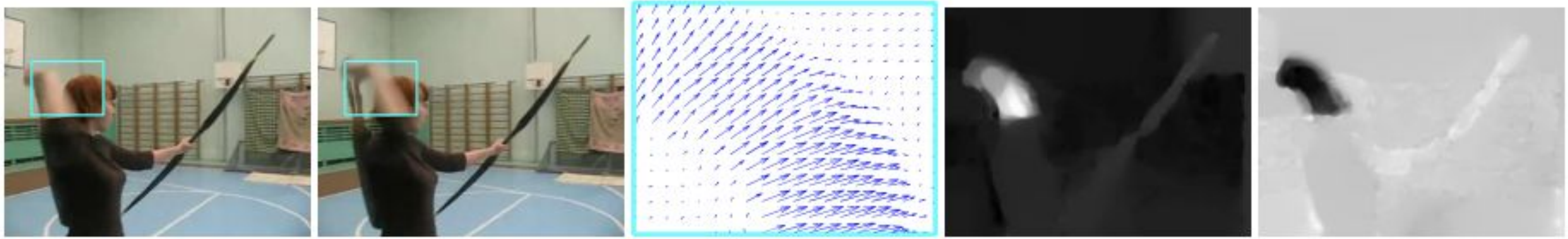


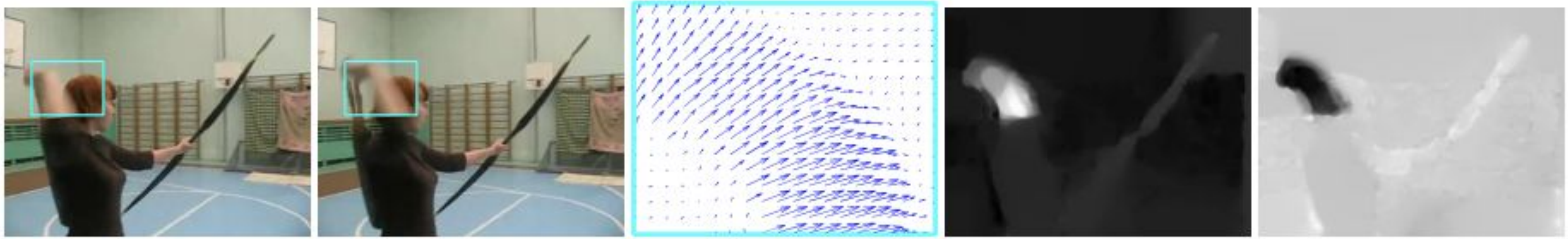
Figure credit: Simonyan and Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. NeurIPS 2014.

Video classifiers (including I3D) often enhanced with optical flow

Two consecutive frames

Optical flow displacement vectors

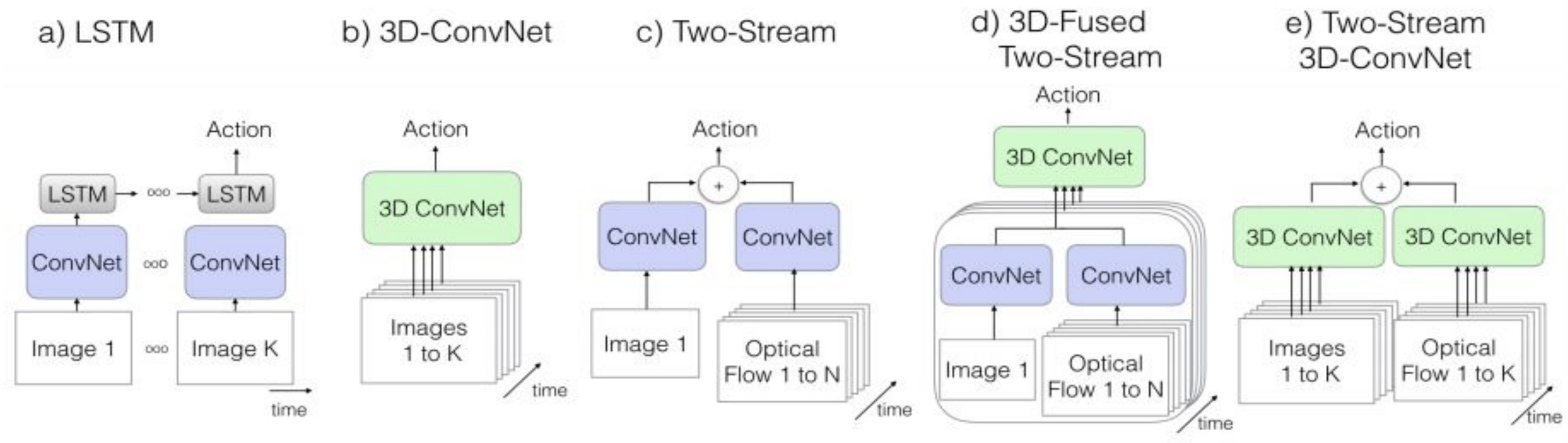
horizontal (L) and vertical (R) components of displacement



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

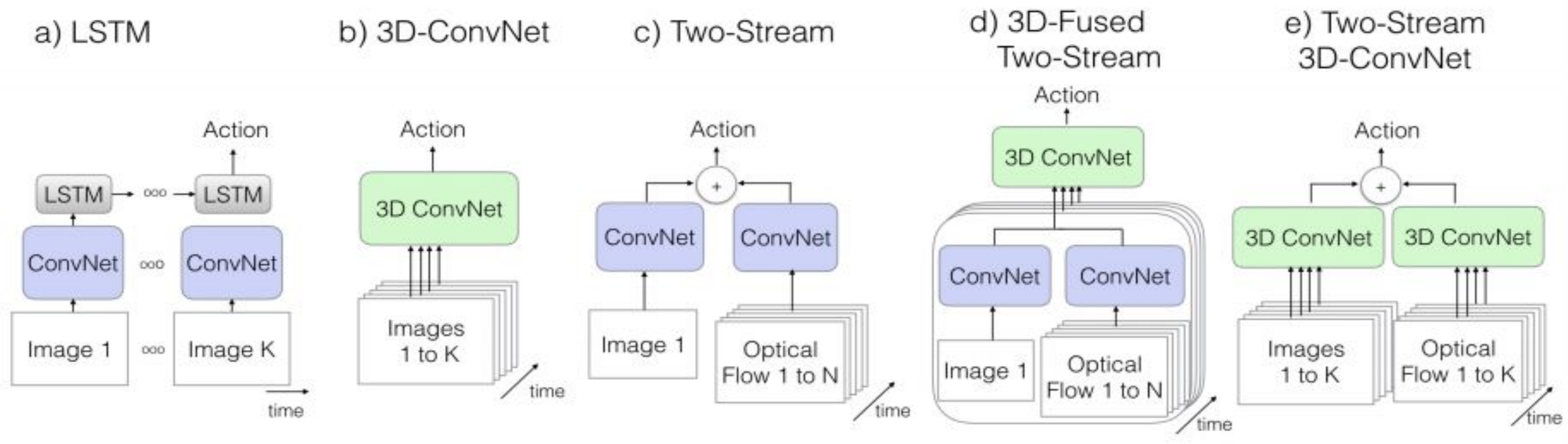
Figure credit: Simonyan and Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. NeurIPS 2014.

Video classifiers (including I3D) often enhanced with optical flow



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

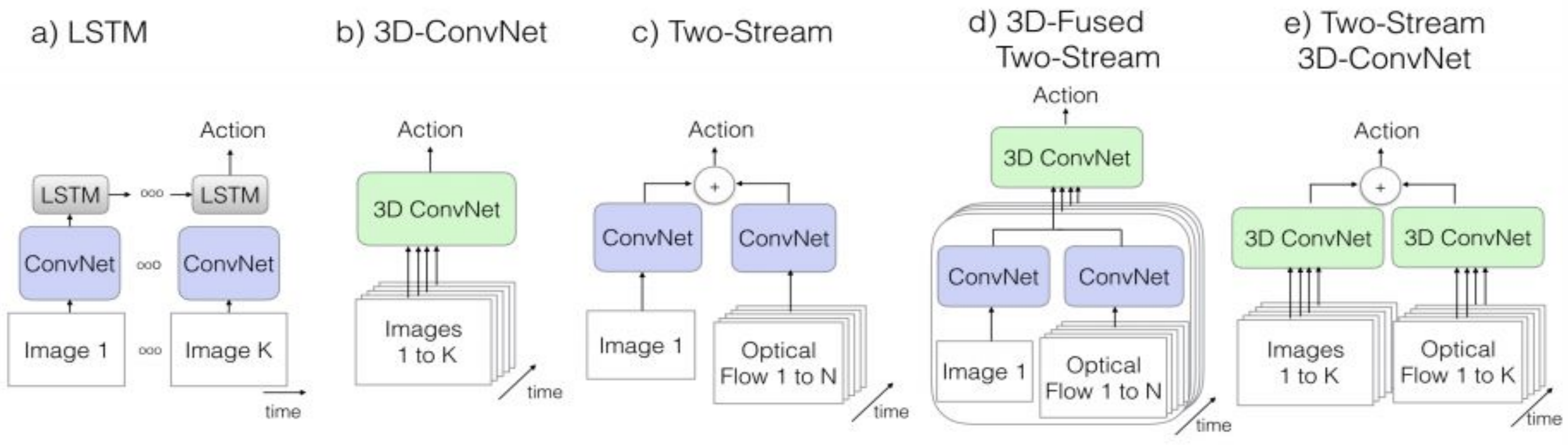
Video classifiers (including I3D) often enhanced with optical flow



LSTM over RGB

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

Video classifiers (including I3D) often enhanced with optical flow

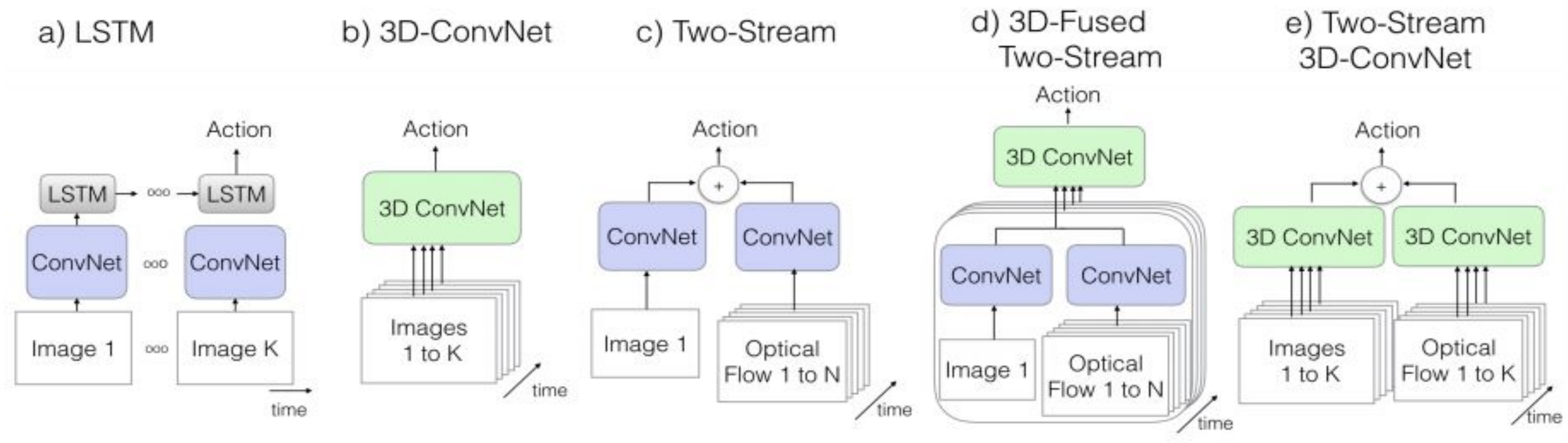


LSTM over RGB

(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) often enhanced with optical flow

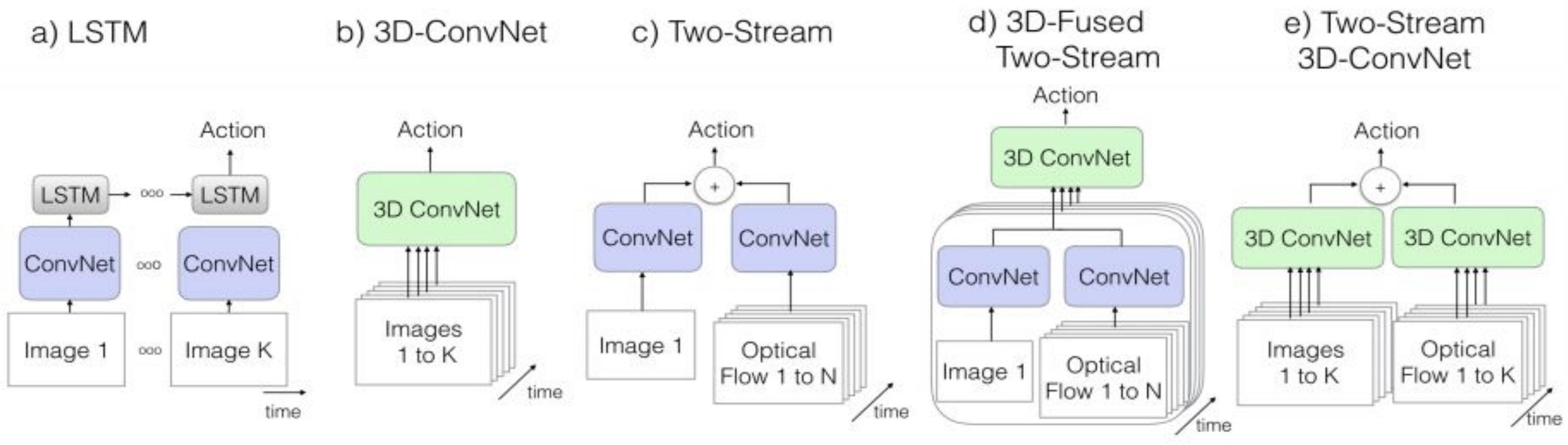


LSTM over RGB

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) often enhanced with optical flow



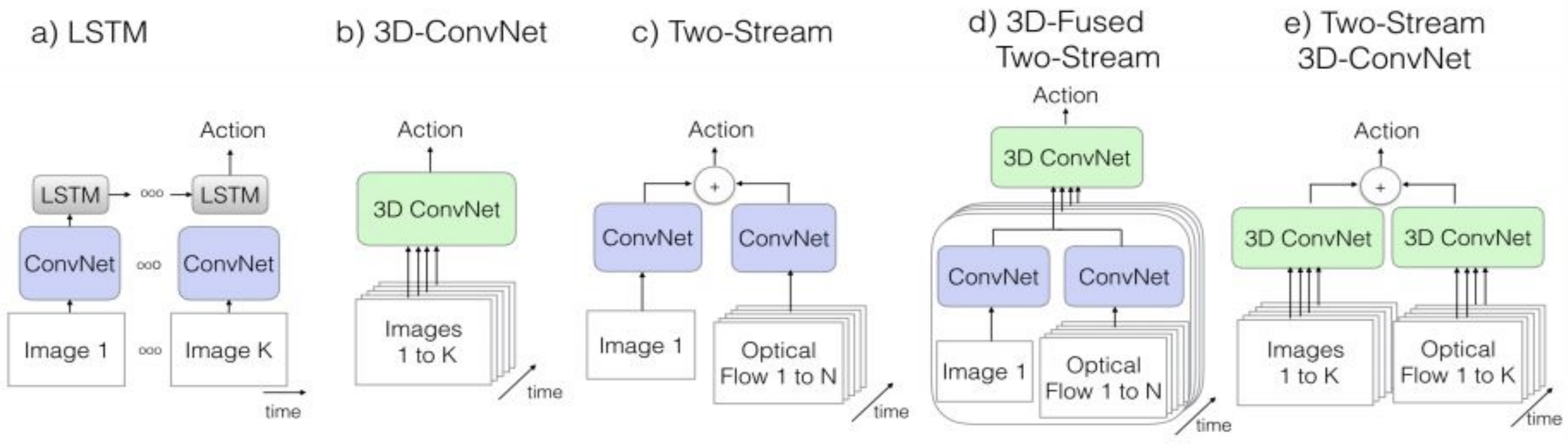
LSTM over RGB

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) often enhanced with optical flow



LSTM over RGB

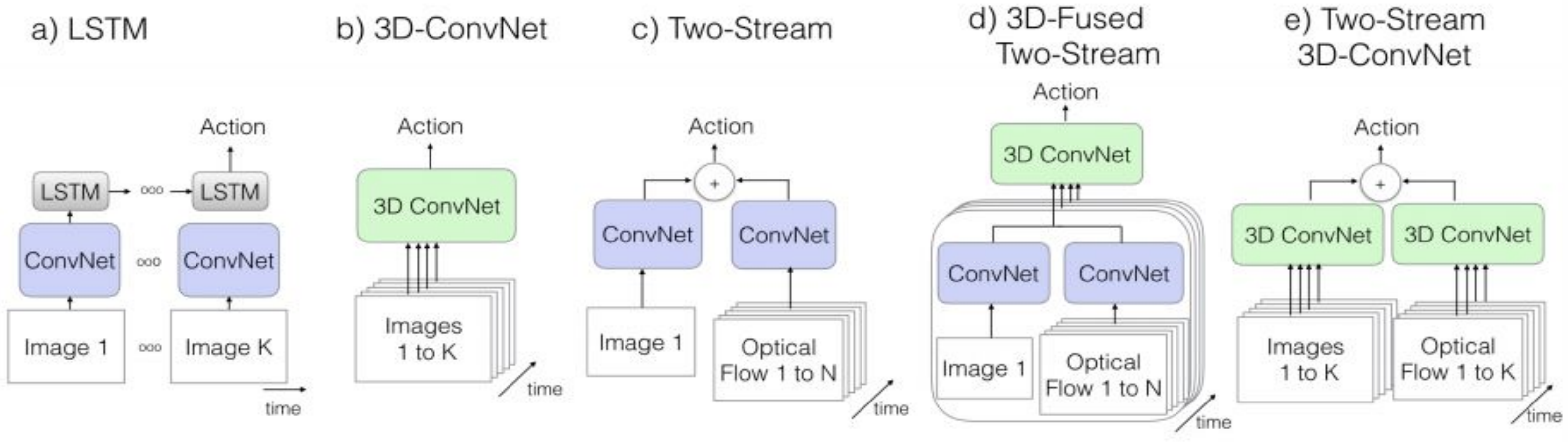
I3D (3D convs) over RGB

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.

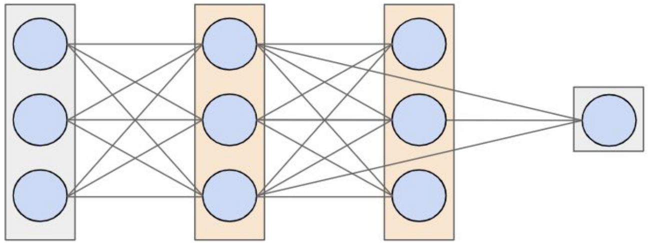
Video classifiers (including I3D) often enhanced with optical flow



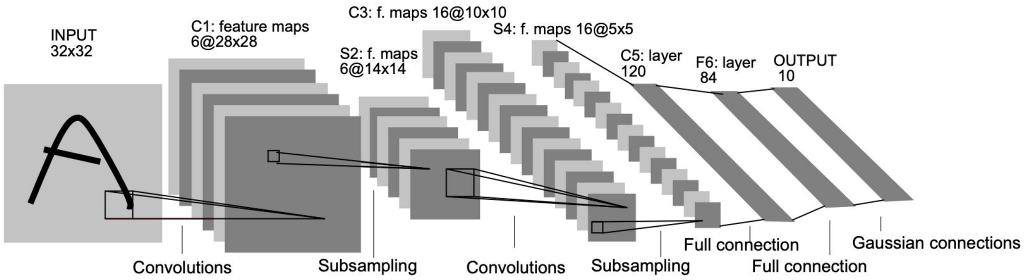
LSTM over RGB	I3D (3D convs) over RGB	2D convs over RGB + optical flow (OF)	Late 3D fusion of RGB + OF	Two I3D streams over RGB + OF

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

Preview: Recurrent neural networks

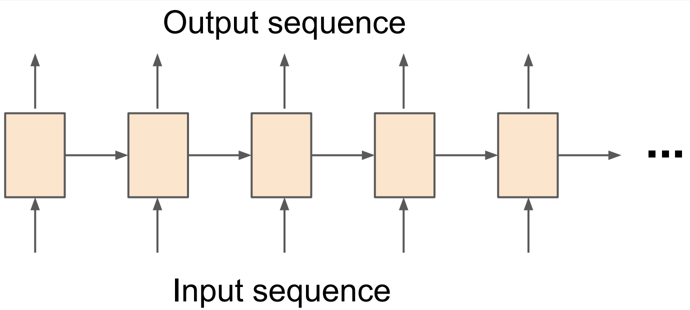


Fully connected neural networks
(linear layers, good for “feature vector” inputs)



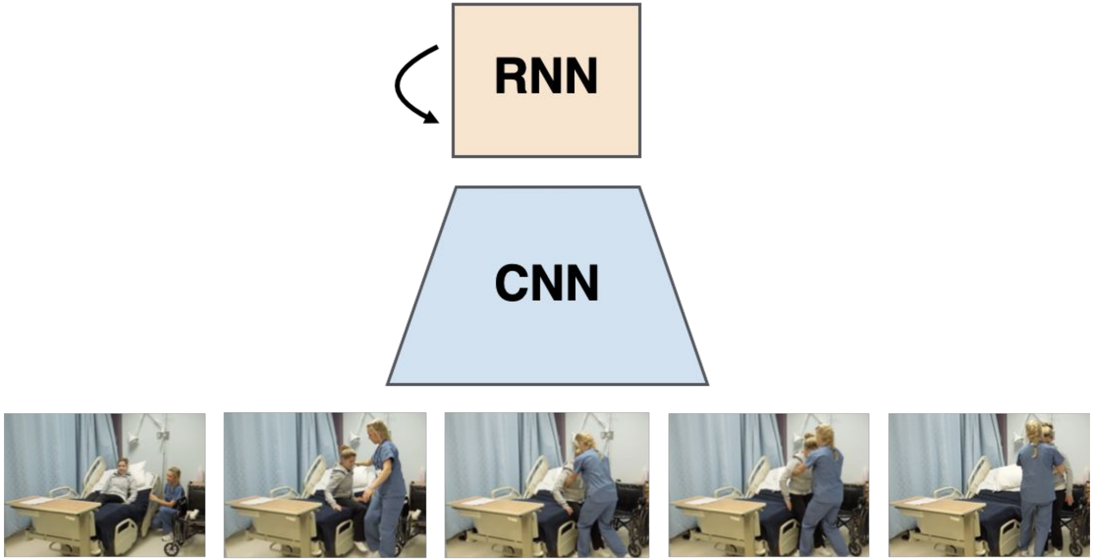
Convolutional neural networks
(convolutional layers, good for image inputs)

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



Videos are sequences: natural fit for recurrent networks

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

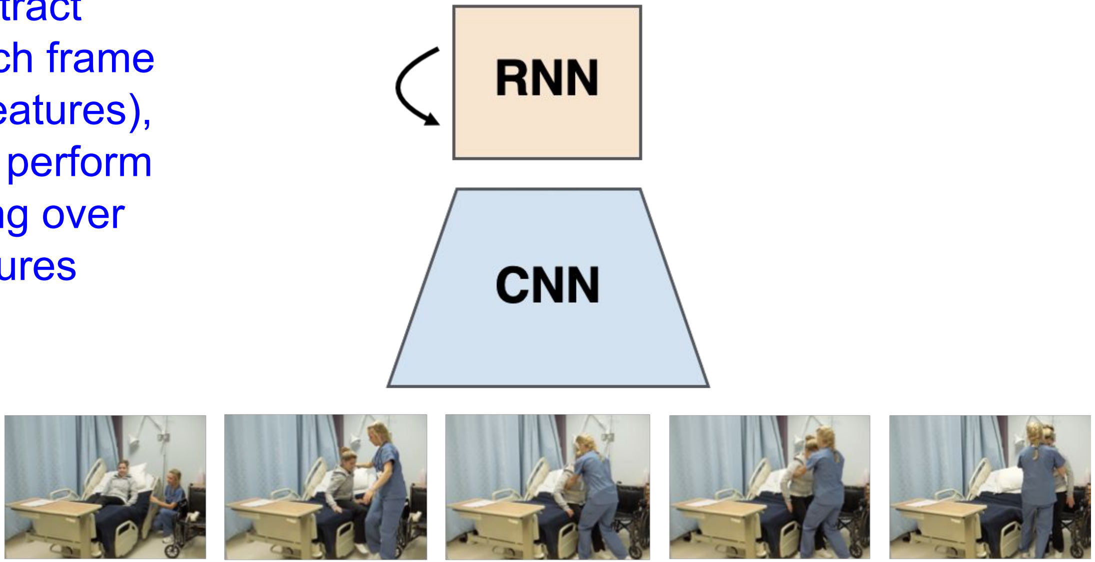


$$\mathbf{x} = \{x_0, x_1, ..., x_T\}$$

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

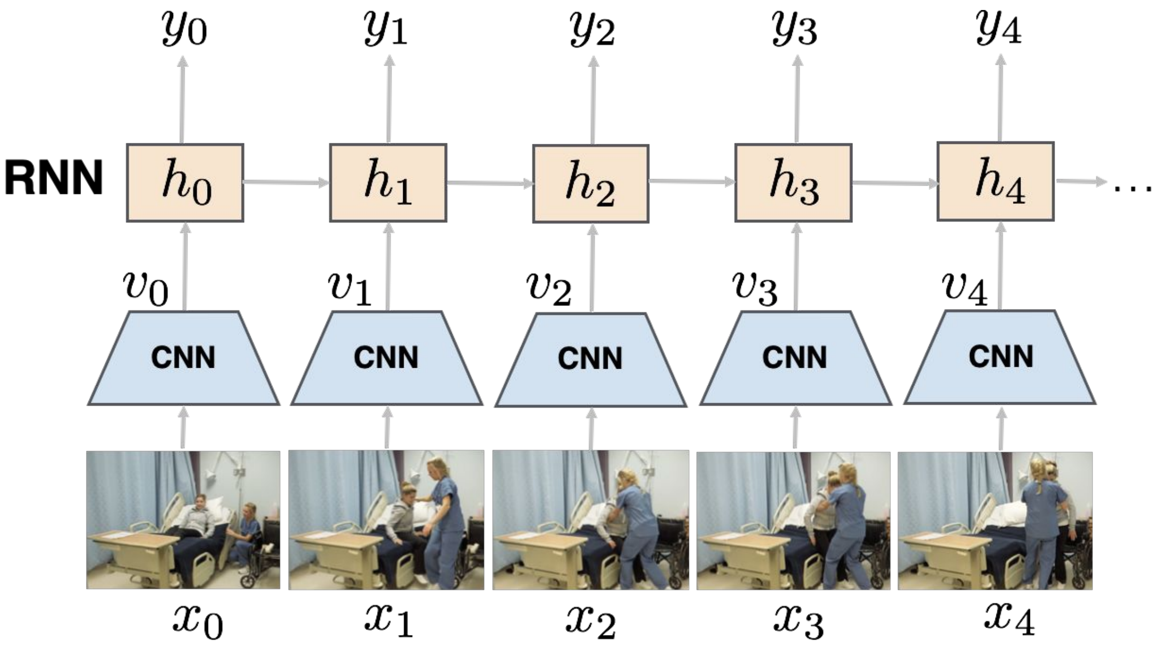
$$\mathbf{y} = \{y_0, y_1, ..., y_T\}$$



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

Videos are sequences: natural fit for recurrent networks

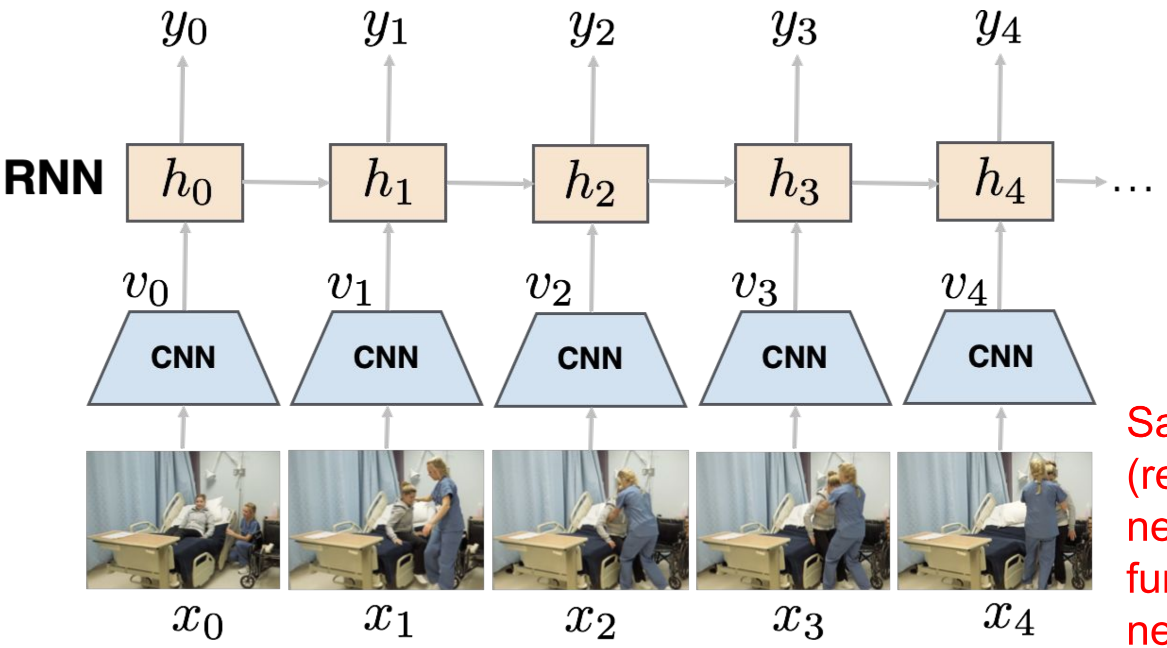
Diagram of a CNN + RNN "rolled out" over time



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

Videos are sequences: natural fit for recurrent networks

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



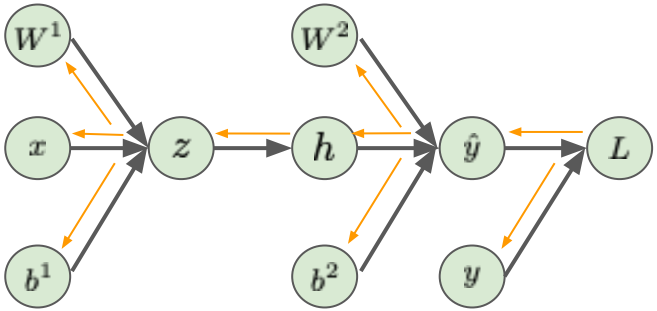
$$h_t = f_W(h_{t-1}, v_t)$$
$$= \boxed{\tanh(W_{hh}h_{t-1} + W_{vh}v_t)}$$
$$y_t = \boxed{W_{hy}h_t}$$

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":

$$\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y)$$

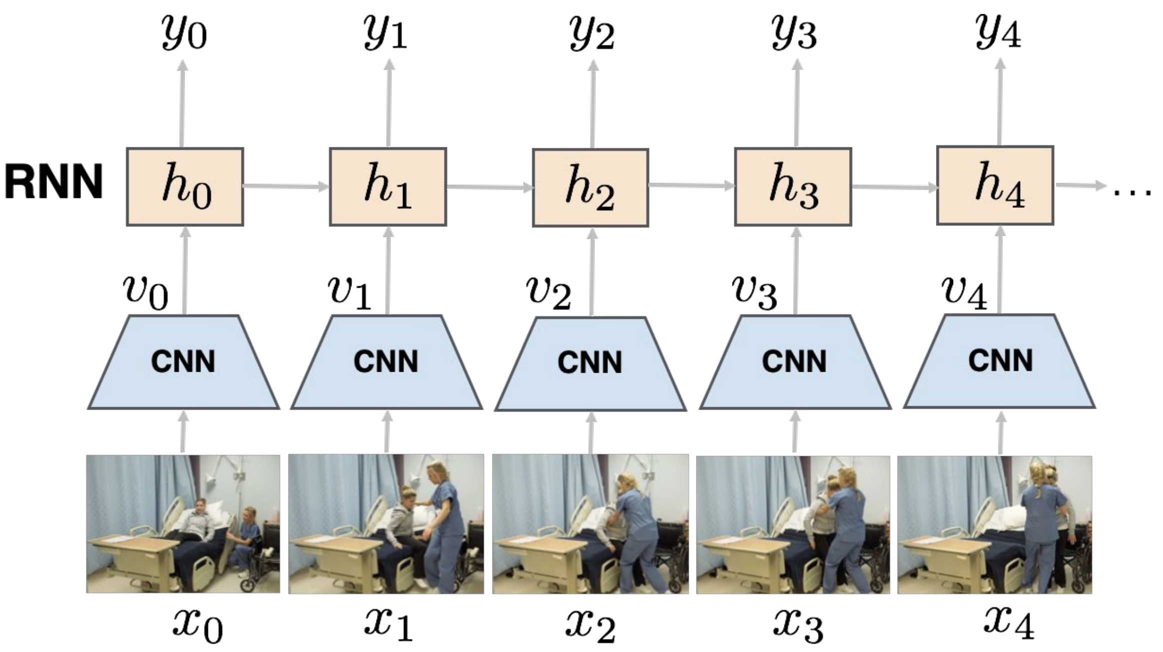
(not all gradients shown)

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

Plug in from earlier computations via chain rule

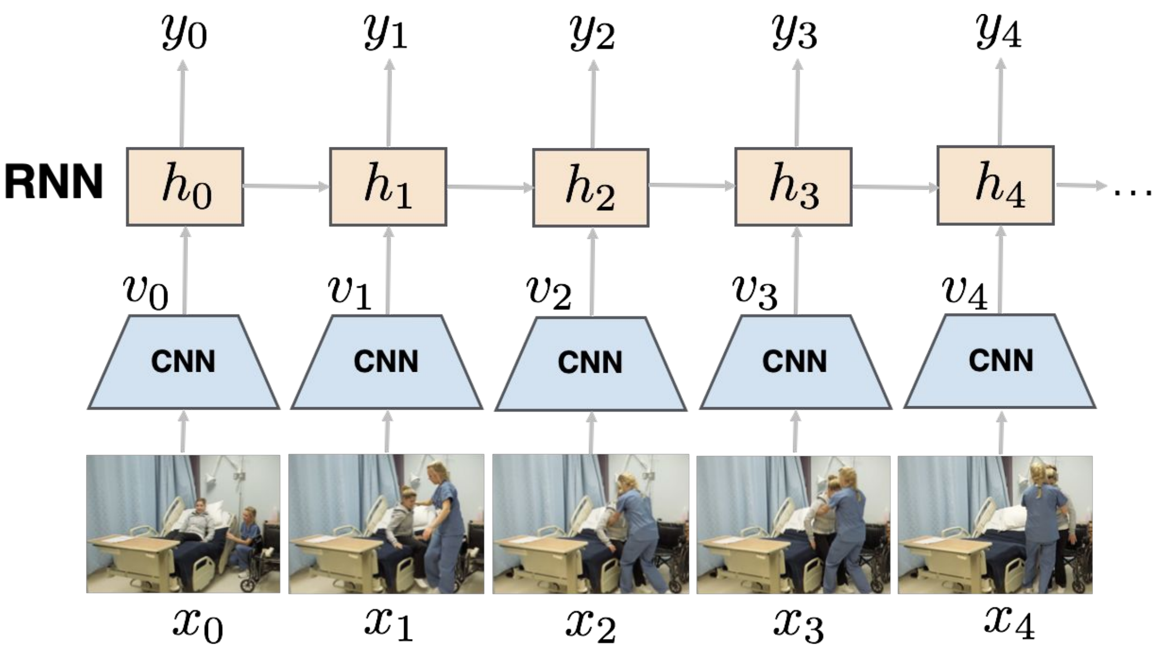
Local gradients to derive

Videos are sequences: natural fit for recurrent networks



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

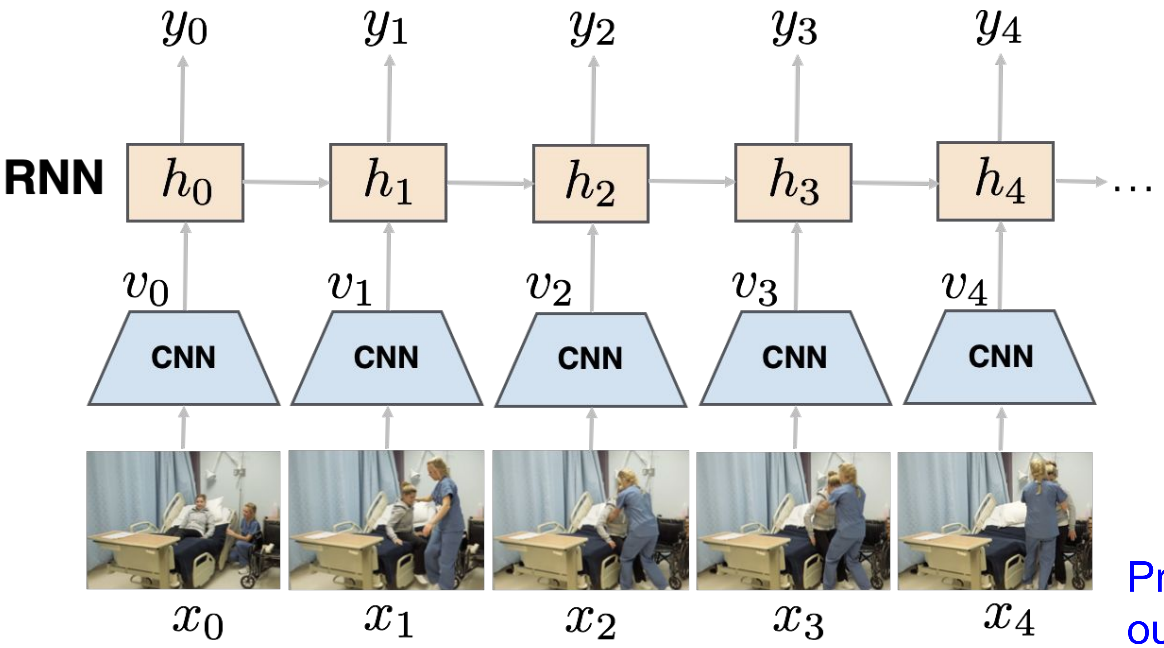
Videos are sequences: natural fit for recurrent networks



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

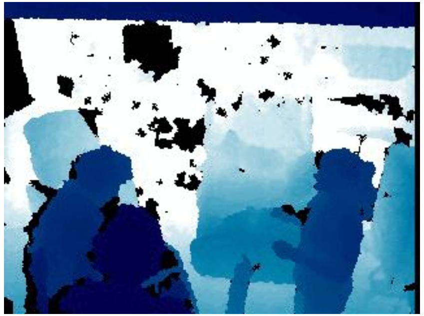
But a very large number of parameters to train simultaneously… more common to fine-tune a single-frame CNN over the data first (or use pre-trained CNN), then extract features and train the RNN separately

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

Detecting patient mobilization activities in the ICU≈

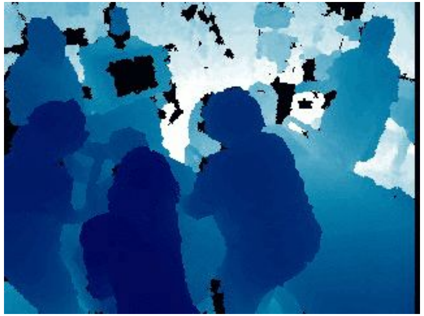
Get patient out of bed

Sit patient in chair

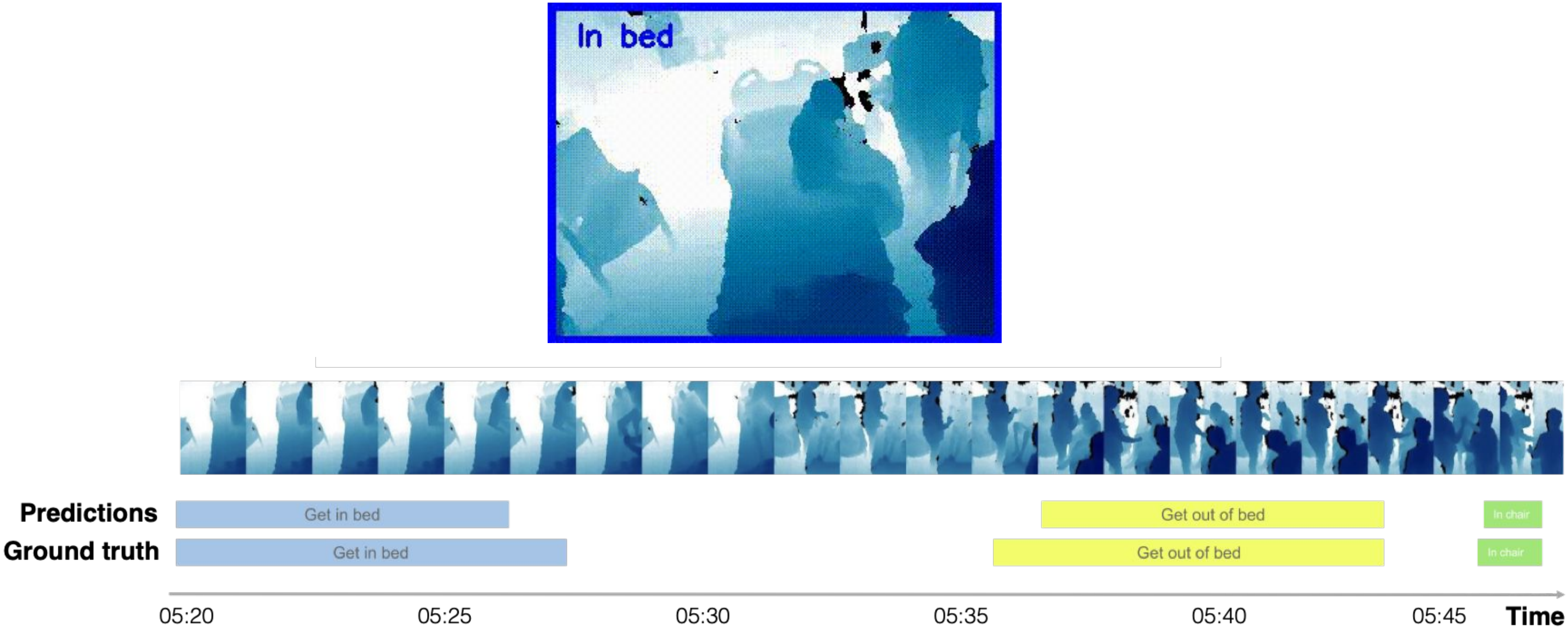


Get patient in bed

Get patient out of chair

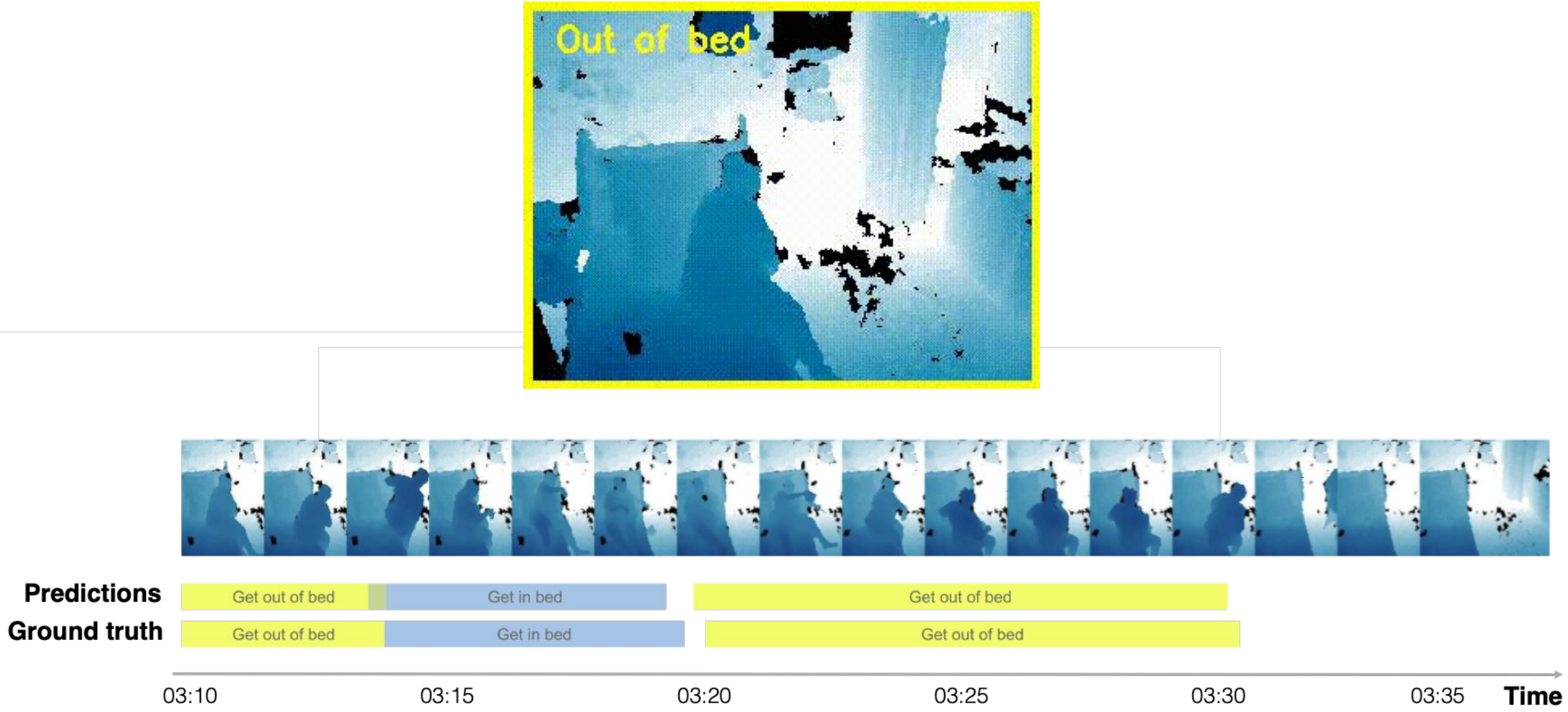


Detecting patient mobilization activities in the ICU≈



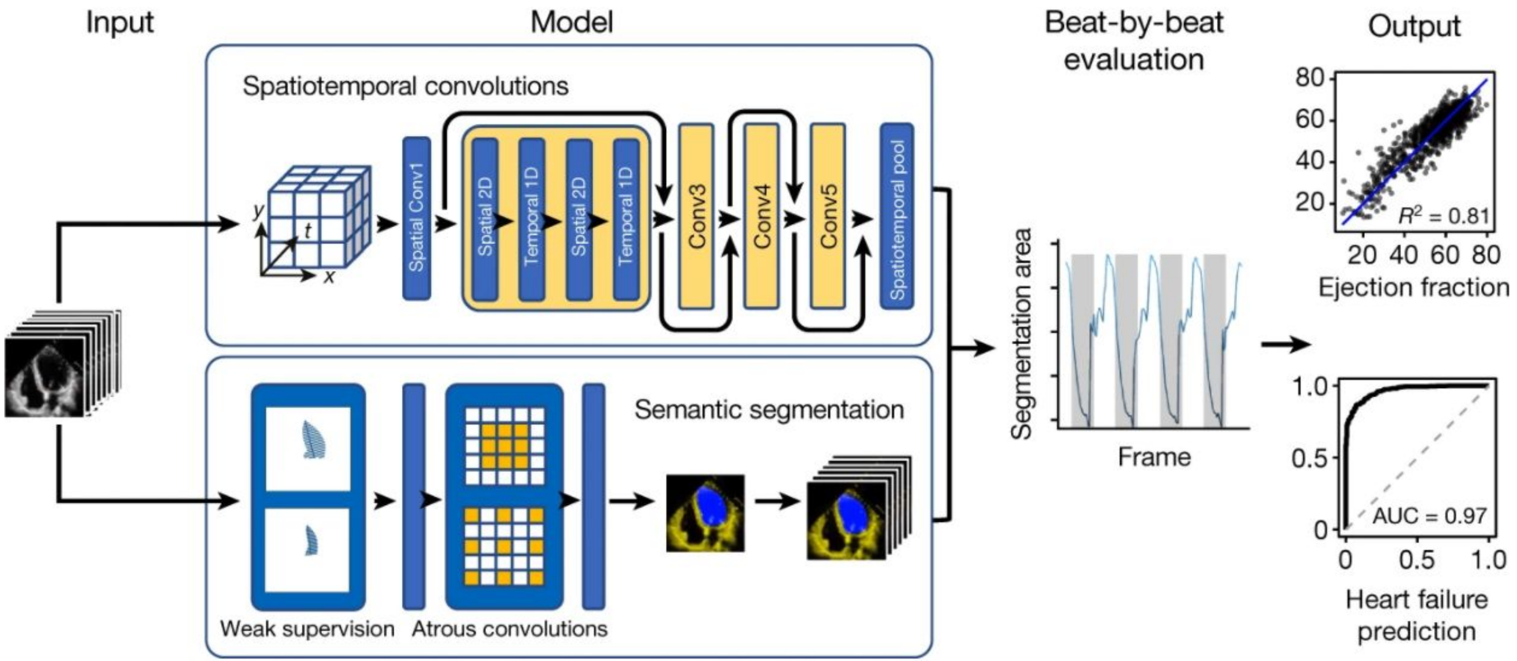
Yeung*,Salipur*, et al. A Computer Vision System for Deep Learning-Based Detection of Patient Mobilization Activities in the ICU. npj Digital Medicine, 2019.

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