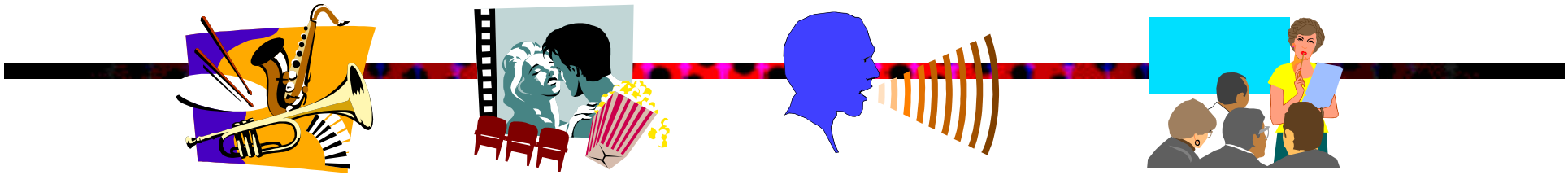


Image and Video Processing

Convolutional Networks for Image Processing (Part III)

Yao Wang
Tandon School of Engineering, New York University

Outline (Part I)

- Supervised learning: General concepts
- Neural network architecture
  - From single perceptron to multi-layer perceptrons
- Convolutional network architecture
  - Why using convolution and many layers
  - Multichannel convolution
  - Pooling
- Deep networks
- Model training
  - Loss functions
  - Stochastic gradient descent: general concept
  - Data Preprocessing and Regularization
- Training, validation and testing and cross validation
- Demo: training a ConvNet classifier

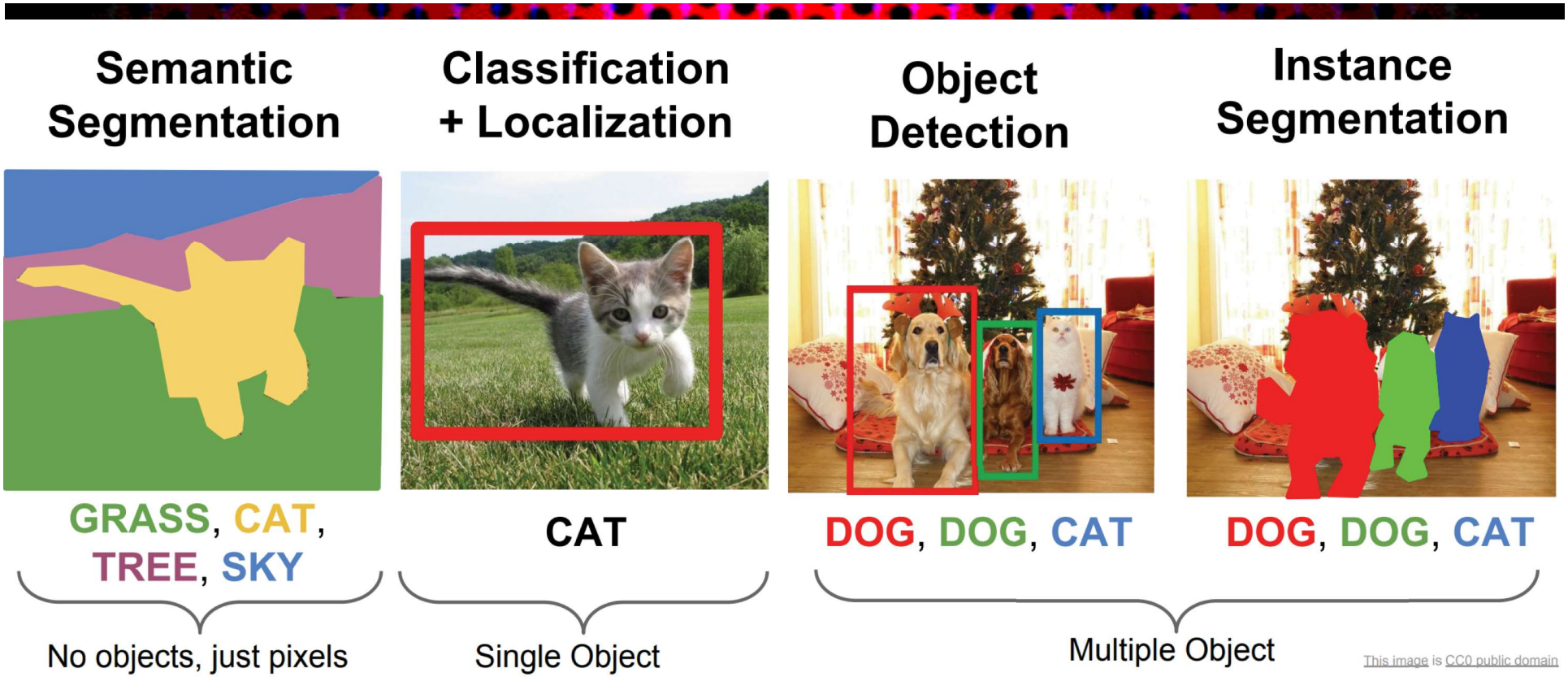
Outline (Part II)

- Neural Nets and Conv Nets and Model Training (Review)
- Gradient calculation
- Some important extensions of conv. layers
- Popular classification models and transfer learning

Outline (Part III): Beyond Classification

- Image to image autoencoder
  - Denoising
- Semantic Segmentation using Multiresolution Autoencoder
- Object detection and classification
- Instance segmentation

Beyond Image Classification …



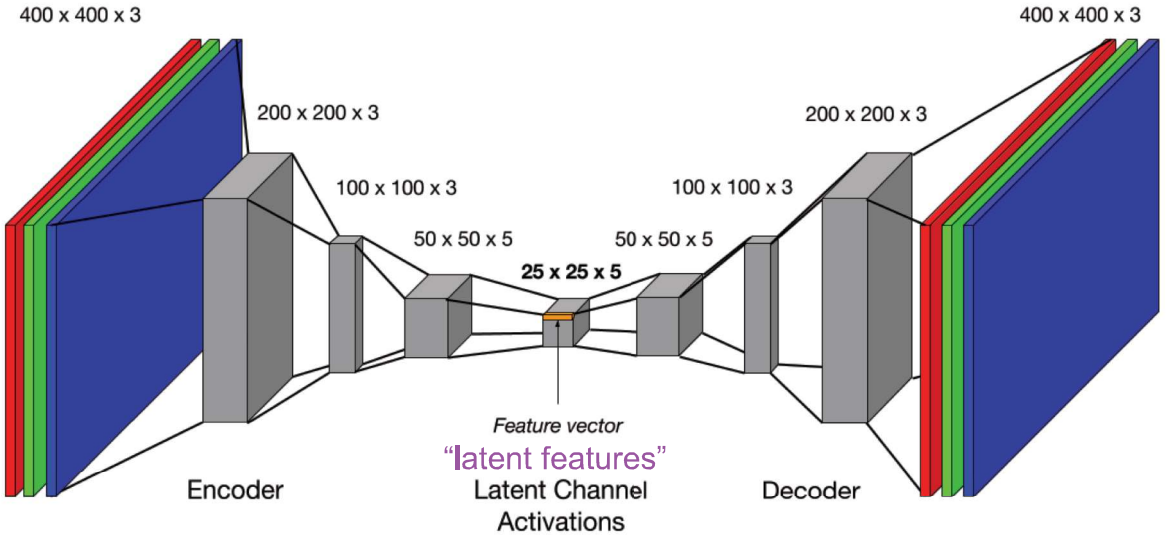
From: http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Image to Image Autoencoder

- Denoising and other applications
- Upsampling through learnable filters

Autoencoder

- CNN is not limited for classification!
- When all the layers are convolution, the output can have the same shape as the input (speech->speech, image->image)
- Autoencoder= Encoder+Decoder
- Encoder: image-> features
- Decoder: features -> image
- Fully convolutional network (FCN)

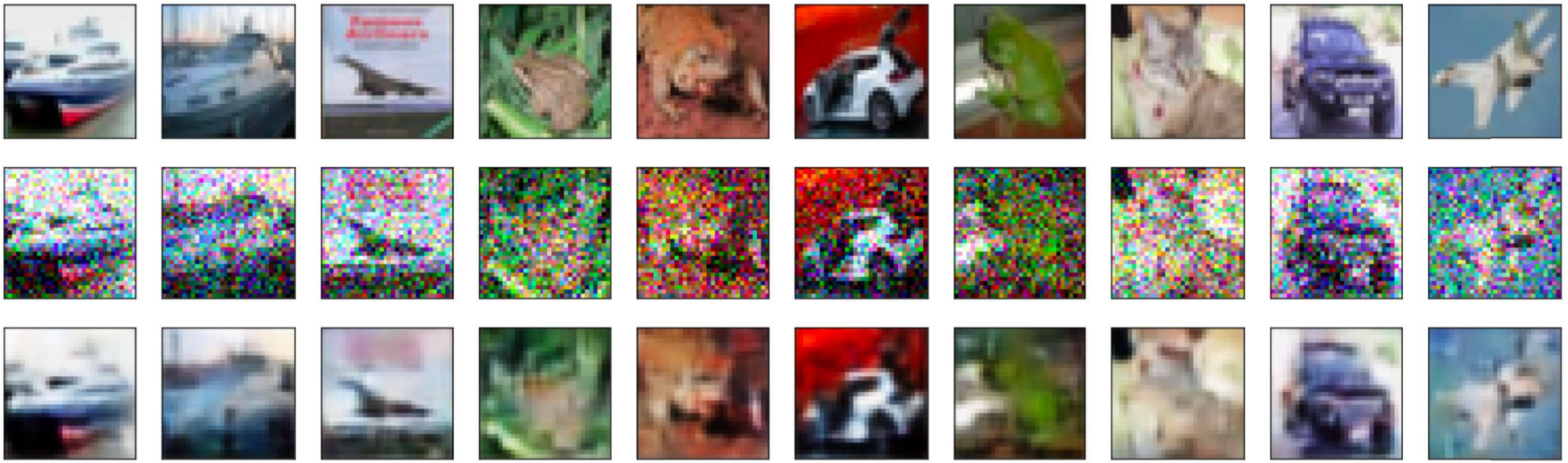


From http://warp.whoi.edu/content/images/2017/08/caearch_shallow.png

Can be applied at the whole image level, or (overlapped) block level.

Autoencoder for image denoising

- Input: noisy image; Output= denoised image
- Need pairs of clean and noisy images as training samples, normalized to range (0,1)
- Following from a simple network (with only three conv layers in encoder and two conv layers in decoder)



- Better image denoising networks:

  Zhang, Kai, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising." *IEEE Transactions on Image Processing* 26, no. 7 (2017): 3142-3155.

  ...

Sample Keras implementation for denoising

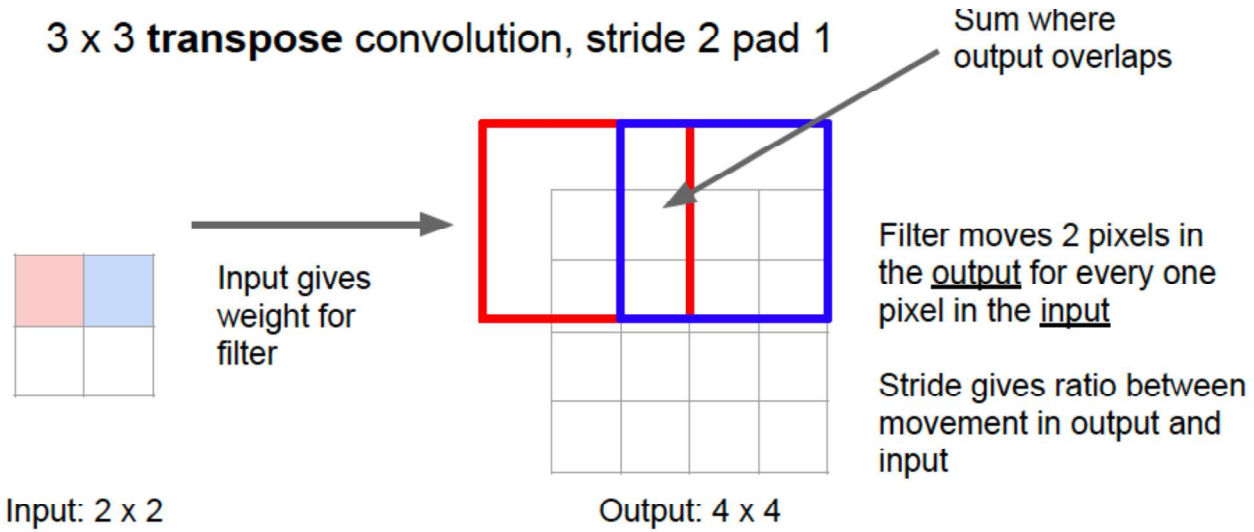
```
from keras.layers import Input, Conv2D, MaxPool2D, UpSampling2D, Dropout
from keras.layers.normalization import BatchNormalization
from keras.models import Model
import keras.backend as K


K.clear_session()
input_img = Input(shape=(32,32,3))
x = Conv2D(16, (3,3), activation='relu', padding='same')(input_img)
x = MaxPool2D((2,2), padding='same')(x)
x = BatchNormalization()(x)
# x = Dropout(0.25)(x)
x = Conv2D(32, (3,3), activation='relu', padding='same')(x)
encoded = MaxPool2D((2,2), padding='same', name='encoded_layer')(x)

x = BatchNormalization()(encoded)
# x = Dropout(0.25)(x)
x = Conv2D(32, (3, 3), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
x = BatchNormalization()(x)
# x = Dropout(0.25)(x)
x = Conv2D(16, (3, 3), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
x = BatchNormalization()(x)
# x = Dropout(0.25)(x)
decoded = Conv2D(3, (3, 3), activation='sigmoid', padding='same')(x)
autoencoder = Model(input_img, decoded)
```

How to perform upsampling?

- Using default upsampling filter (nearest, linear)
- Learn the interpolation filter (transposed convolution or deconvolution)
  - First generate zero-filled image (inserting zeros between known samples)
  - Then apply the filter



From http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Also known as “Deconvolution” (not a proper name!)

Why "Transpose Convolution"?

We can express convolution in terms of a matrix multiplication

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & z & 0 & 0 & 0 \\ 0 & 0 & x & y & z & 0 \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ bx + cy + dz \end{bmatrix}$$

Example: 1D conv, kernel size=3, stride=2, padding=1

Convolution transpose multiplies by the transpose of the same matrix:

$$\vec{x} *^T \vec{a} = X^T \vec{a}$$

$$\begin{bmatrix} x & 0 \\ y & 0 \\ z & x \\ 0 & y \\ 0 & z \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} ax \\ ay \\ az + bx \\ by \\ bz \\ 0 \end{bmatrix}$$

When stride>1, convolution transpose is no longer a normal convolution!

From http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Actually it is a convolution, need to pad zero in between "a" and "b"

DSP explanation of transpose convolution

$$\vec{x} *^T \vec{a} = X^T \vec{a}$$

$$\begin{bmatrix} x & 0 \\ y & 0 \\ z & x \\ 0 & y \\ 0 & z \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} ax \\ ay \\ az+bx \\ by \\ bz \\ 0 \end{bmatrix}$$

No zero filling
Stride 2

X corresponding to filter [x,y,z]
X^T corresponds to reversed filter [z,y,x]

Insert zeros before convolution

$$\begin{bmatrix} x & & & \\ y & x & & \\ z & y & x & \\ & z & y & x \\ & & z & y \\ & & & z \end{bmatrix} \begin{bmatrix} a \\ 0 \\ b \\ 0 \end{bmatrix} = \begin{bmatrix} ax \\ ay \\ az+bx \\ by \\ bz \\ 0 \end{bmatrix}$$

Stride K: upsampling by factor of K, insert K-1 zeros in between every original samples
This is exactly how we perform interpolation in DSP!
Interpolation filter is [z,y,x]

Applications of autoencoders

- Output \= input:
  - Denoising
  - Image completion
  - Super resolution
  - Segmentation (only discuss this)
  - Saliency detection
- Output = input
  - For unsupervised learning: to learn features that can represent an image with reduced dimension
  - For compression: use the quantized latent features to represent an image
- Autoencoder loss depends on the underlying application
- Using adversarial loss can help to make the output look more like the target output (beyond this class)

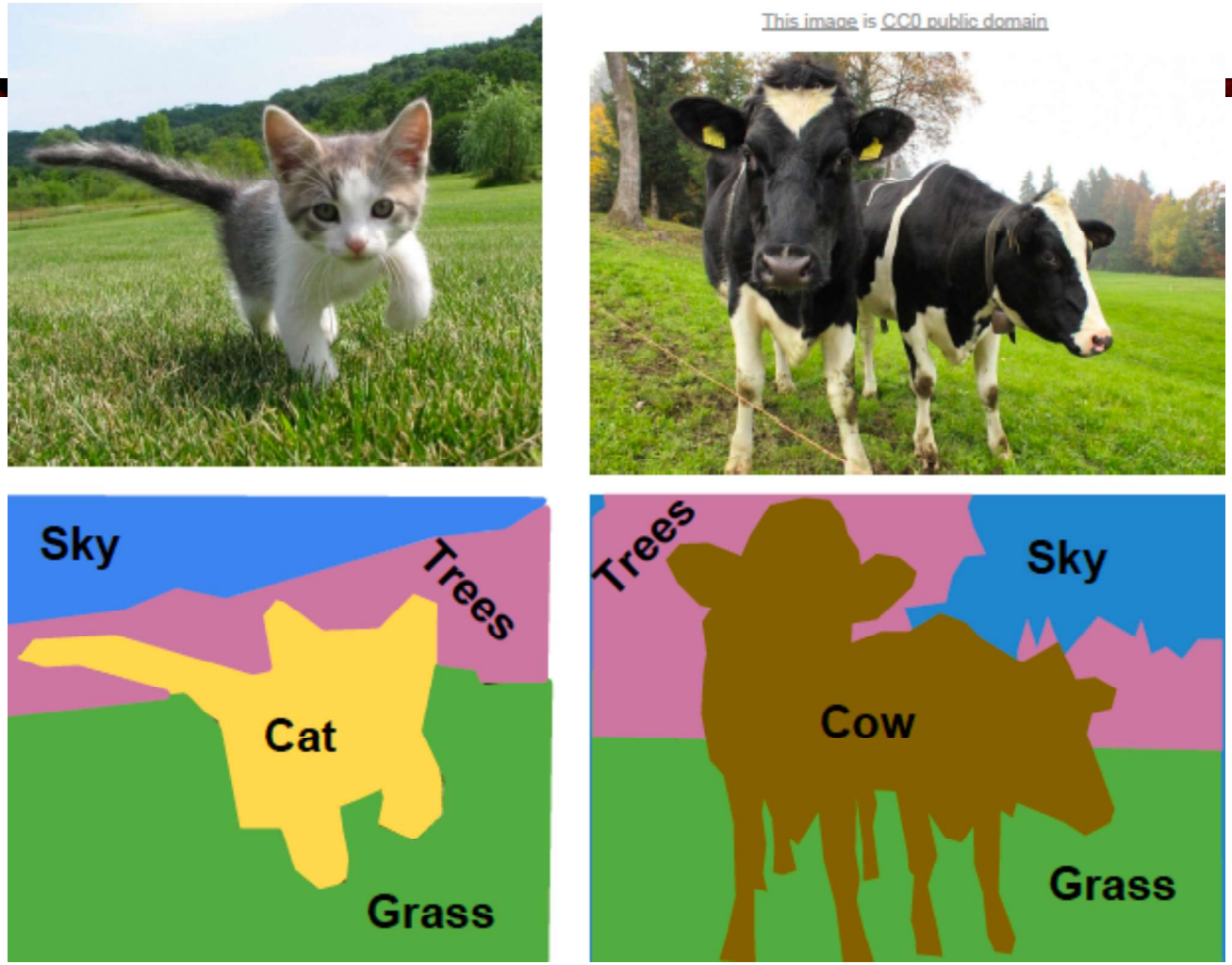
Outline (Part III)

- Image to image autoencoder
- Semantic Segmentation using Multiresolution Autoencoder
- Object detection and classification
- Instance segmentation

Semantic Segmentation

- What does it mean?
- Loss function
- Multiresolution Autoencoder (U-Net)
- Extension for 3D volume (V-Net)
- Work at NYU video lab

Semantic Segmentation



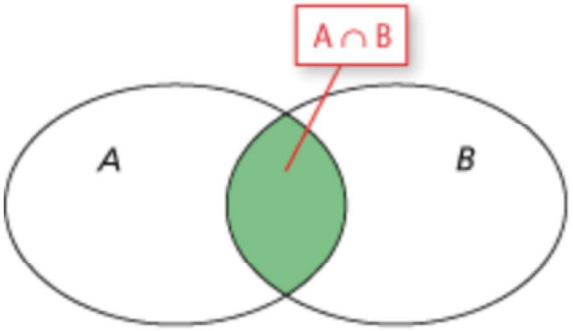
Each pixel is classified into one object class

From http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

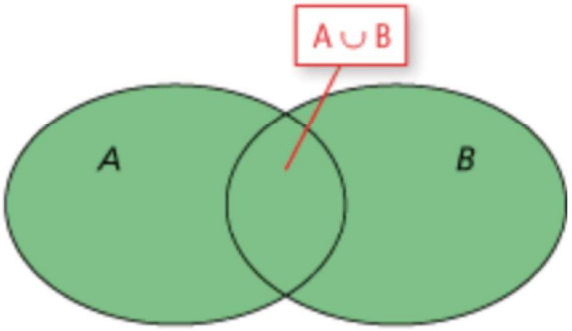
Loss function for segmentation

- Semantic segmentation:
  - Label each pixel as one of the classes
  - Treat as multi-class classification at each pixel
    - Generate multiple segmentation maps as output, one probability map for each class. The probabilities for all classes at each pixel sum to 1
  - Loss:
    - Sum of categorical cross entropy over all pixels for each image, and over all training images
    - DICE = Intersection over union (Non-differentiable)
    - Soft DICE: defined in terms of predicted probability

DICE Loss for Evaluating Binary Segmentation



Intersection of *A* and *B*



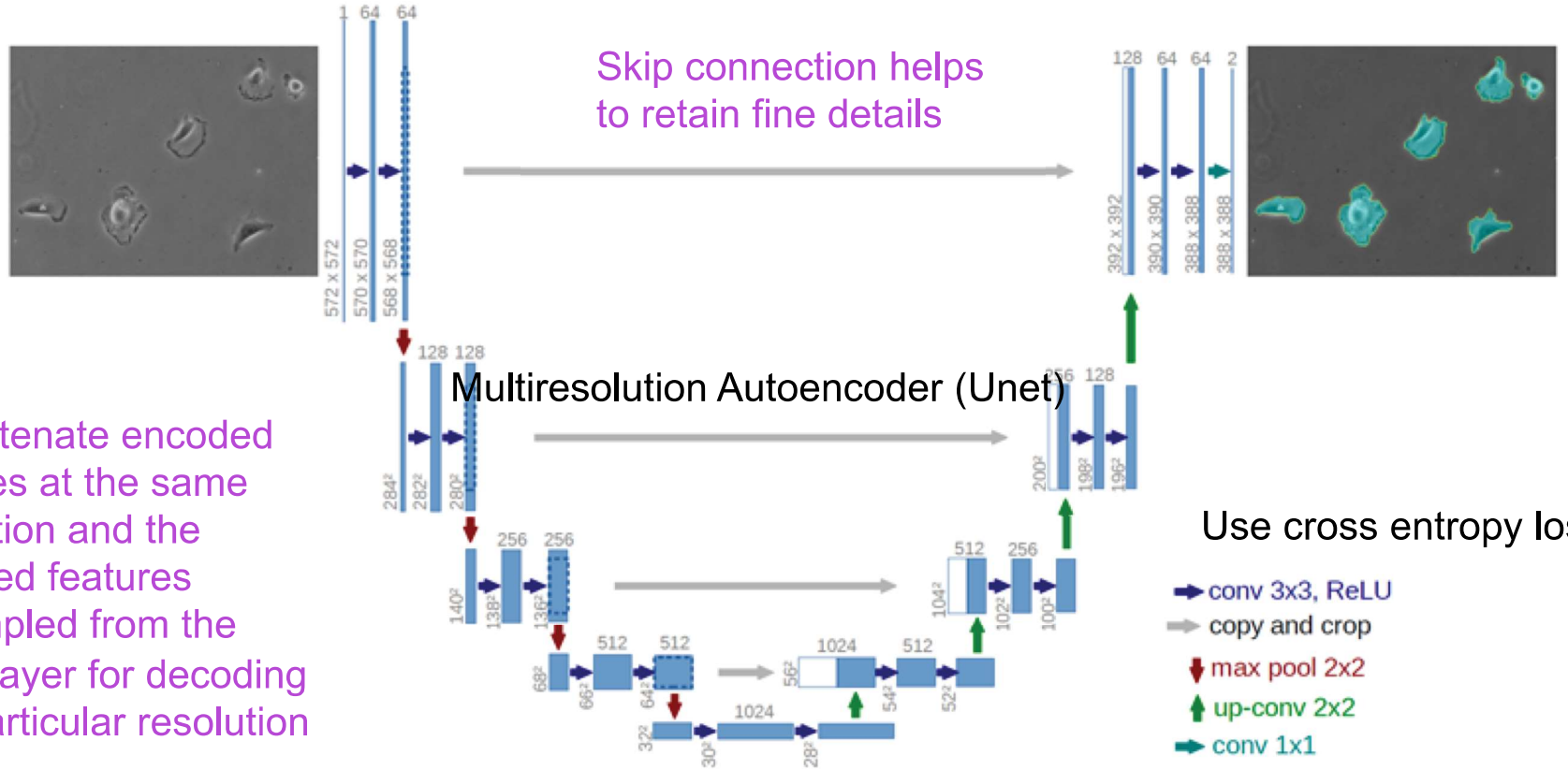
Union of *A* and *B*

A: Ground truth foreground, g_i=1

B: predicted foreground, p_i: probability of pixel i is foreground

- DICE = $\frac{\text{area of } A \cap B}{\text{area of } A \cup B}$ (Intersection over union or IoU, evaluating after thresholding the probability, non-differentiable)
- Soft DICE = $\frac{2\sum_i p_i g_i}{\sum_i p_i^2 + \sum_i g_i^2}$ (differentiable)
- Overcomes the difficulty of using cross entropy loss when the foreground object is much smaller than the background

Multi-resolution auto encoder: U-Net



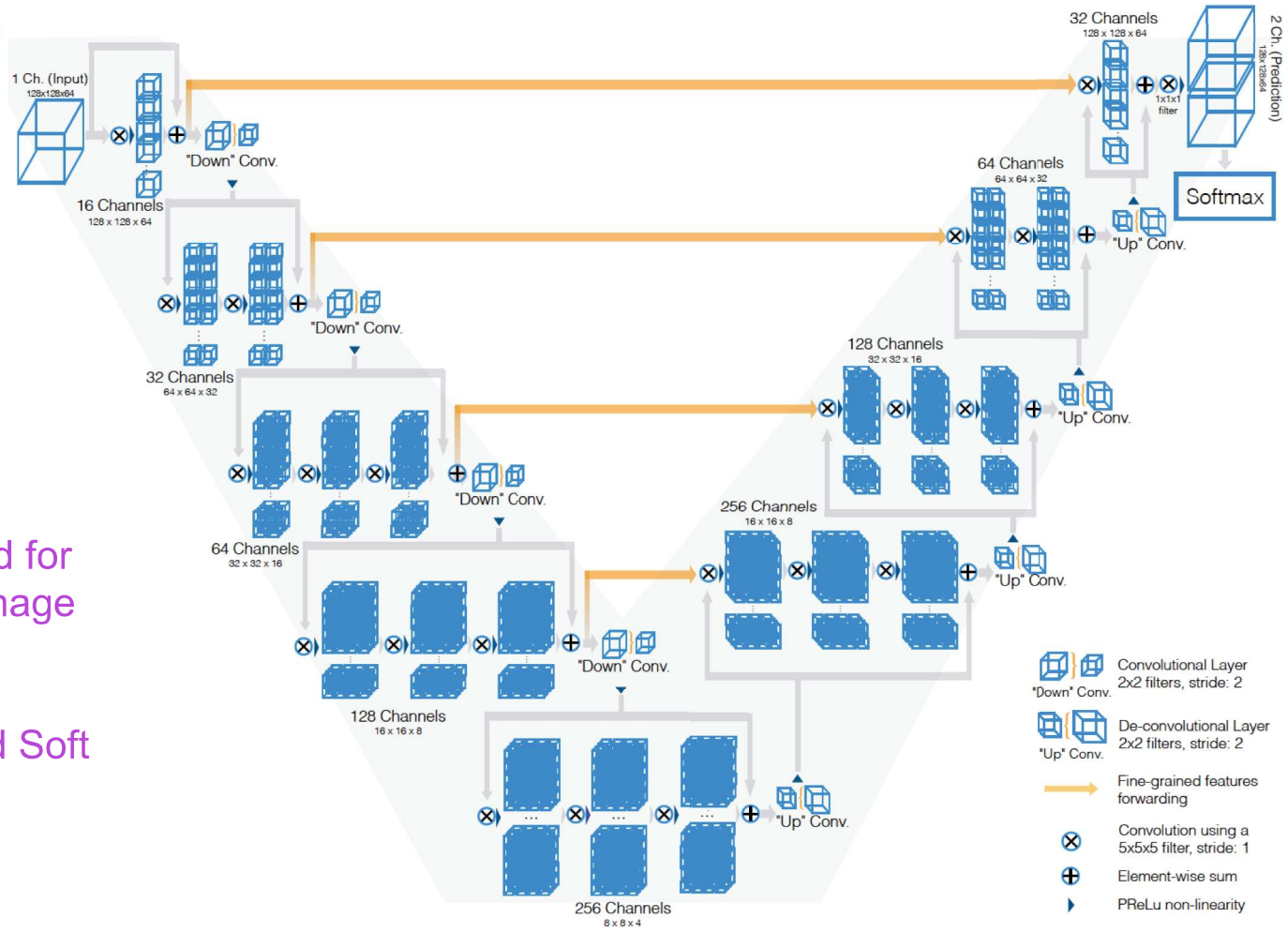
From: https://lmb.informatik.uni-freiburg.de/research/funded_projects/bioss_deeplearning/unet.png

Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham. https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/. Watch the video!

V-Net for Volumetric Image Segmentation

Often used for medical image volume

Introduced Soft DICE



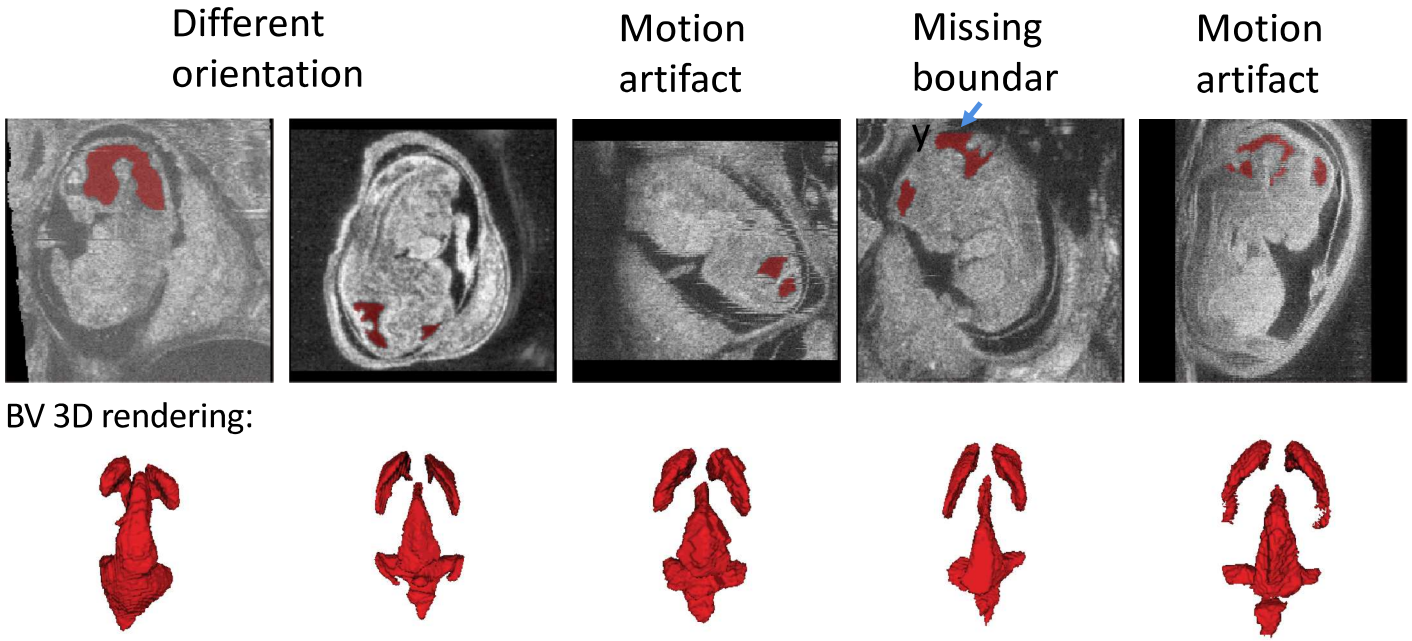
Milletari, Fausto, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-net: Fully convolutional neural networks for volumetric medical image segmentation." *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016. https://arxiv.org/pdf/1606.04797.pdf

Work at NYU Video Lab: Segmentation of High Frequency Ultrasound Images of Mouse Embryos

- Mouse embryo development is a good animal model for studying human brain development.
- High frequency ultrasound can image embryos in vivo.
- Accurate segmentation of the brain ventricles (BVs) from 3D HFU image is essential for assessing the development of the central nervous system of mouse embryos.
- An automated algorithm would permit fast and efficient embryo staging and detection of brain phenotypes.
- Joint work with Lizzi Center for Biomedical Engineering, Riverside Research (Ketterling and Mamou) and NYU School of Medicine (Turnbull and Aristizaba). Supported by NIH R01
  - Tandon students: Ziming Qiu, Jack Langerman, Nitin Nair.

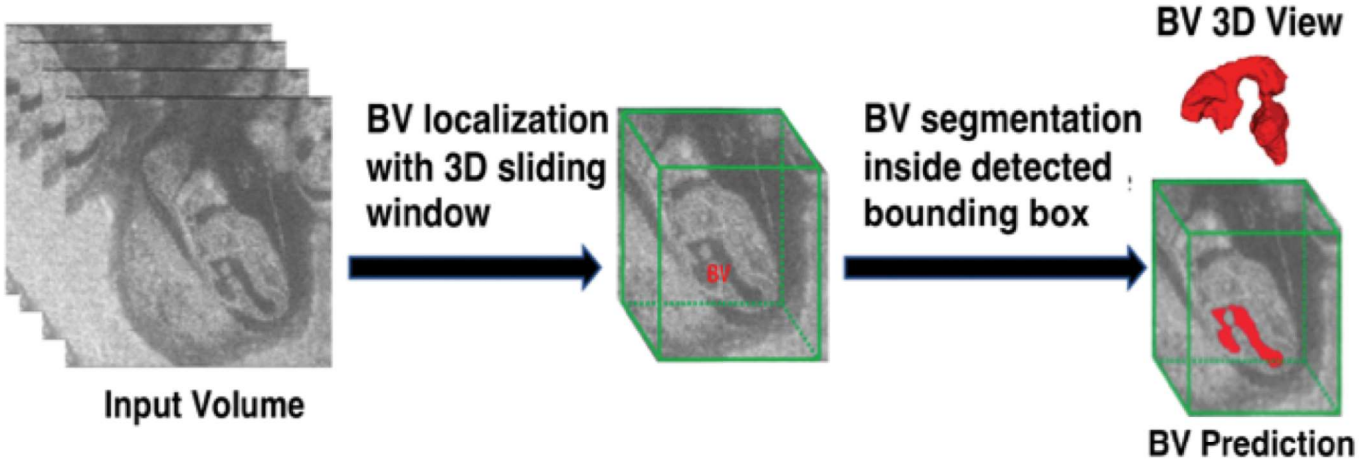
Challenges

- The variety of body posture and orientation.
- The extreme imbalance between BV and background(0.3%).
- Different BV shape and intensity distribution.
- Presence of missing head boundaries and motion artifacts.



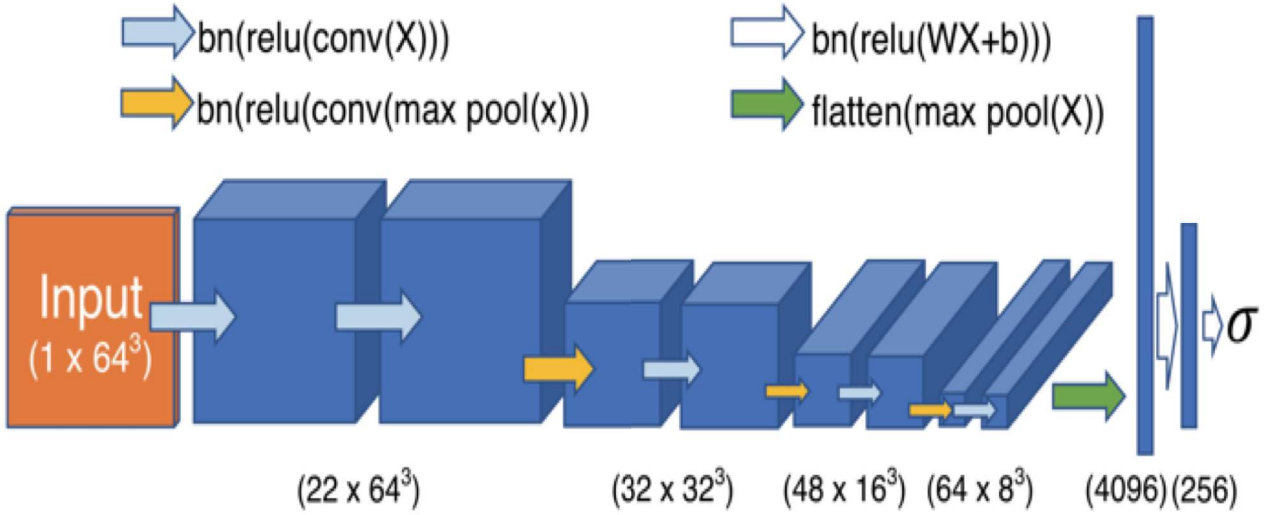
Deep BV: A deep learning based approach

- Manual segmentation of BVs for 370 whole body images
  - Performed by a trained graduate student and verified by a small animal ultrasound imaging expert
  - 259 images captured in 2017 are used for training, 111 in 2016/2018 for testing
- Trained a localization network and a segmentation network



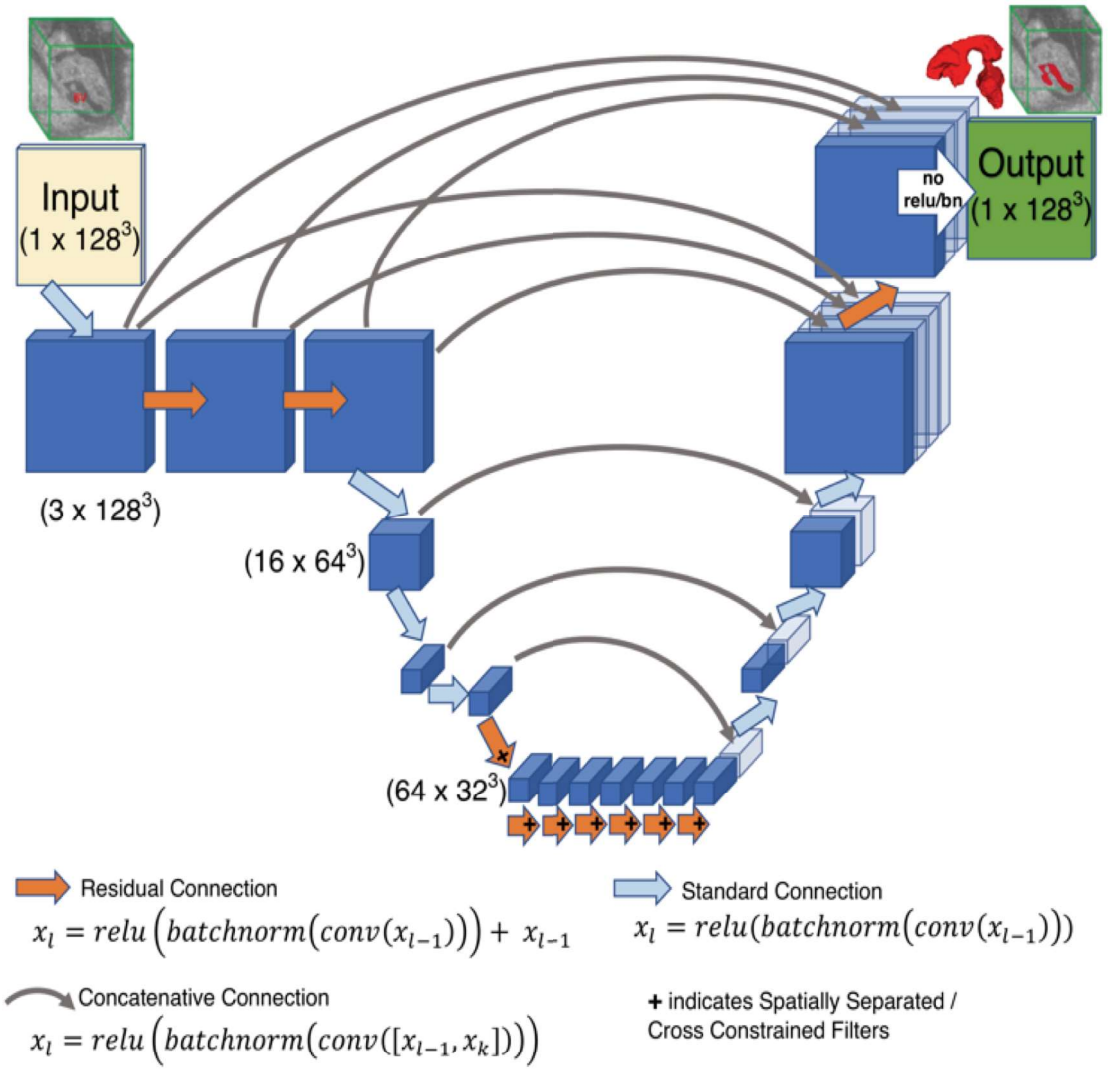
Ziming Qiu, Jack Langerman, Nitin Nair, Orlando Aristizabal, Jonathan Mamou, Daniel H. Turnbull, Jeffrey Ketterling, Yao Wang: DEEP BV: A FULLY AUTOMATED SYSTEM FOR BRAIN VENTRICLE LOCALIZATION AND SEGMENTATION IN 3D ULTRASOUND IMAGES OF EMBRYONIC MICE . To be presented at The 2018 IEEE Signal Processing in Medicine and Biology Symposium. Dec. 2018. https://arxiv.org/pdf/1811.03601.pdf

Localization Network



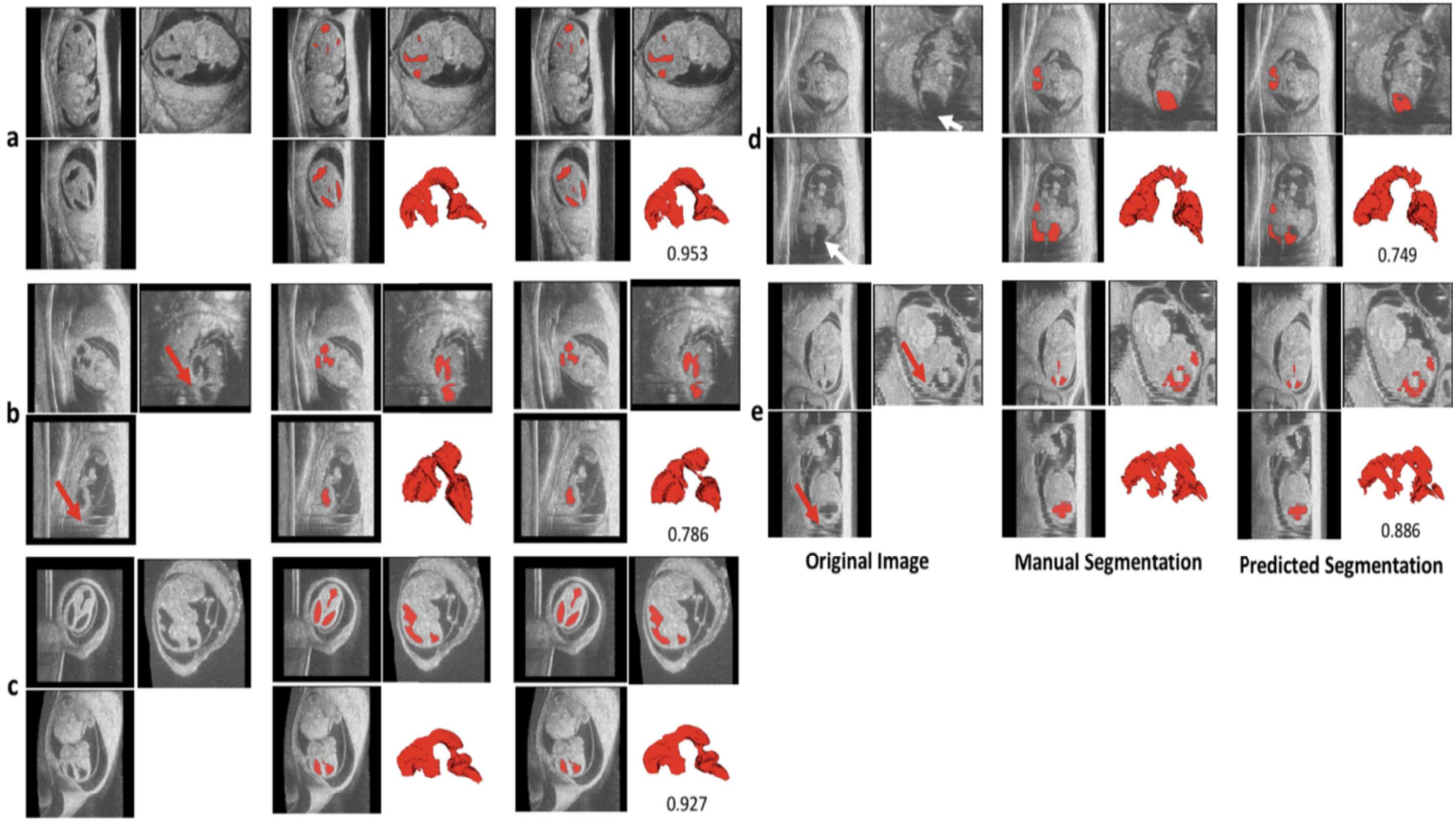
- Use a 10-layer VGG style volumetric CNN network (using 3D Conv) as a sliding window classifier.
- < 80% --negative, > 99% --positive; half a million training samples from 259 images. With additional data augmentation.
- Resulting network is applied to each sliding window (128^3 -> 64^3) of the 3D volume, to determine whether the window contains the BV completely.

Segmentation Network



- Inspired by the Vnet architecture for medical image segmentation.
- Use Soft DICE loss with both positive (BV) loss and negative (non-BV) loss.
- Add additional full resolution layers to extract shape variation at the original resolution.
- Add additional low resolution layers, to increase the effective receptive field (to learn global shape prior).
- Use cross/separated constrained filter structures to reduce the number of parameters.

Sample Results



More Recent Improvement: Simultaneous Segmentation of BV and Body

- Tongda Xu*, Ziming Qiu*, William Das, Chuiyu Wang, Jack Langerman, Nitin Nair, Orlando Aristizabal, Jonathan Mamou, Daniel H. Turnbull, Jeffrey A. Ketterling, Yao Wang, “Deep Mouse: An End-to-end Auto-context Refinement Framework for Brain Ventricle & Body Segmentation in Embryonic Mice Ultrasound Volumes,” in IEEE International Symposium on Biomedical Imaging (ISBI), 2020, * equal contribution, arxiv preprint.

Simultaneous Localization and Segmentation

- Low resolution coarse segmentation to localize regions of interest for both BV and Body.
- Full resolution refined segmentation in each detected region of interest.
- Jointly trained end to end.

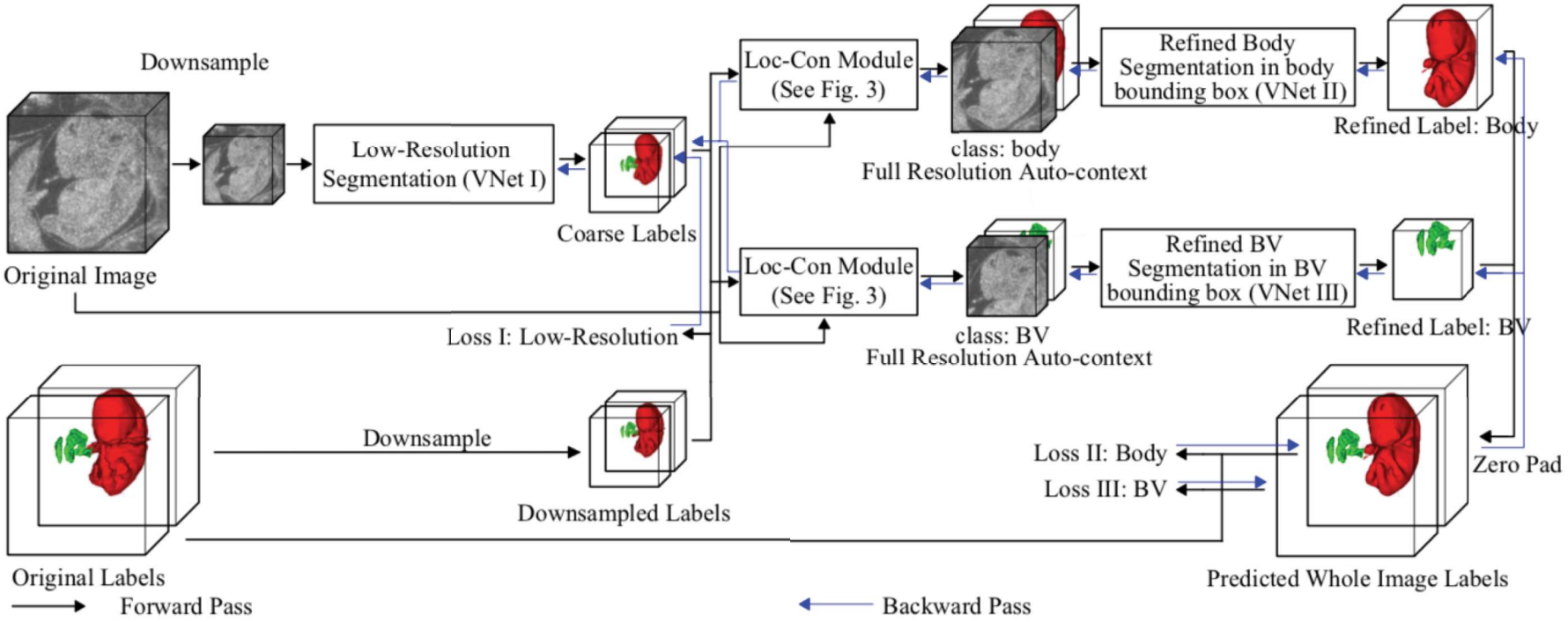


Fig. 2: Diagram of overall pipeline.

Sample Results

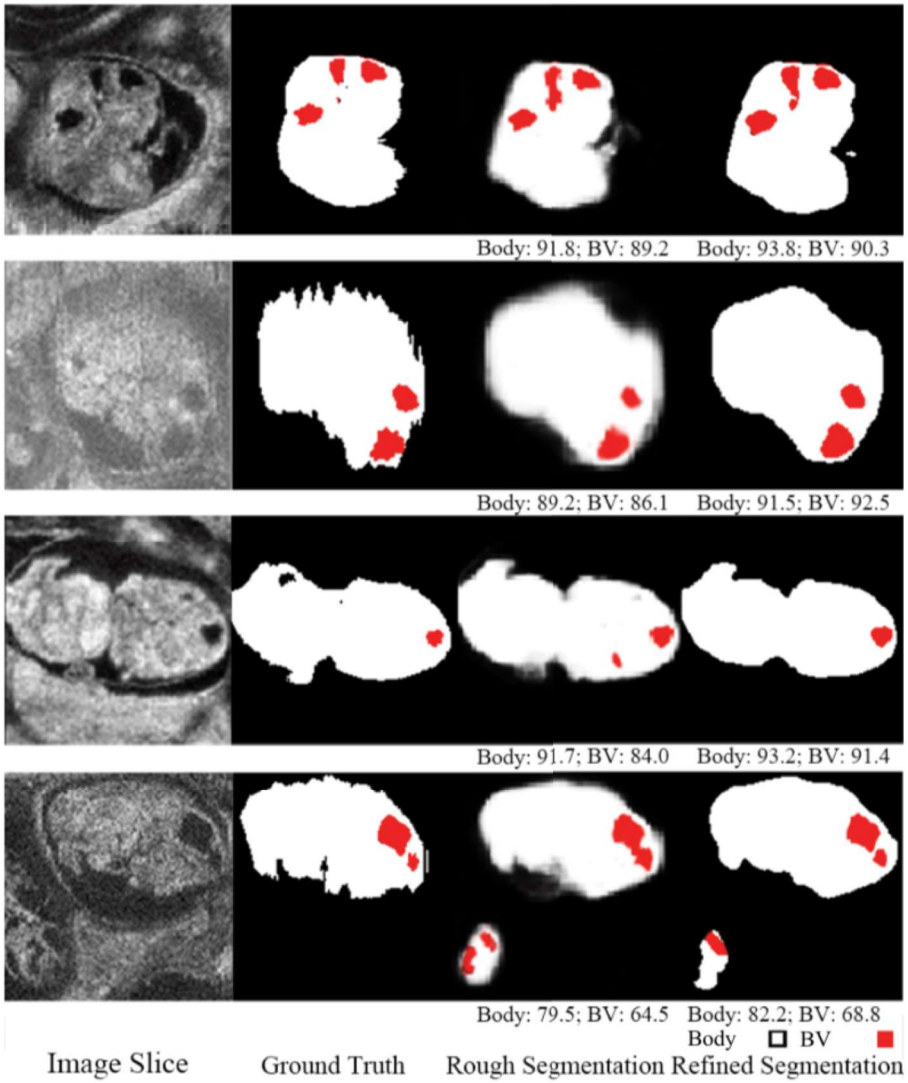


Table 1: The Dice Similarity Coefficient (DSC) and inference time averaged over 46 test volumes for different methods.

Results / Methods	BV DSC	Body DSC	Inference Time *
Benchmark	0.904 [7]	0.932 [8]	102.36s
Initial Segmentation	0.818	0.918	0.006s
Refinement w/o Auto-Context Input	0.878	0.922	0.08s
Refinement w/ Auto-Context Input	0.894	0.924	0.09s
Refinement End-to-end	**0.906**	**0.934**	0.09s

Outline (Part III)

- Image to image autoencoder
- Semantic Segmentation using Multiresolution Autoencoder
- Object detection and classification
- Instance segmentation

Object Detection

- What does it mean?
- Faster R-CNN
  - region proposal network and ROI pooling
- YOLO/SSD
- Work at NYU video lab

Object Localization and Classification

From: Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." In *Advances in neural information processing systems*, pp. 91-99. 2015.

Faster R-CNN

- A region proposal network (RPN) goes through the feature maps generated from the Conv layer and for each location, examines multiple anchor boxes and classify it as Object or not and further more predict deviation of the actual object box location and size from the anchor
- Features corresponding to ”likely object boxes” (RoI pooling) further go through a classification and regression layer (Post-RPN), to classify the object and further refine the box location and size

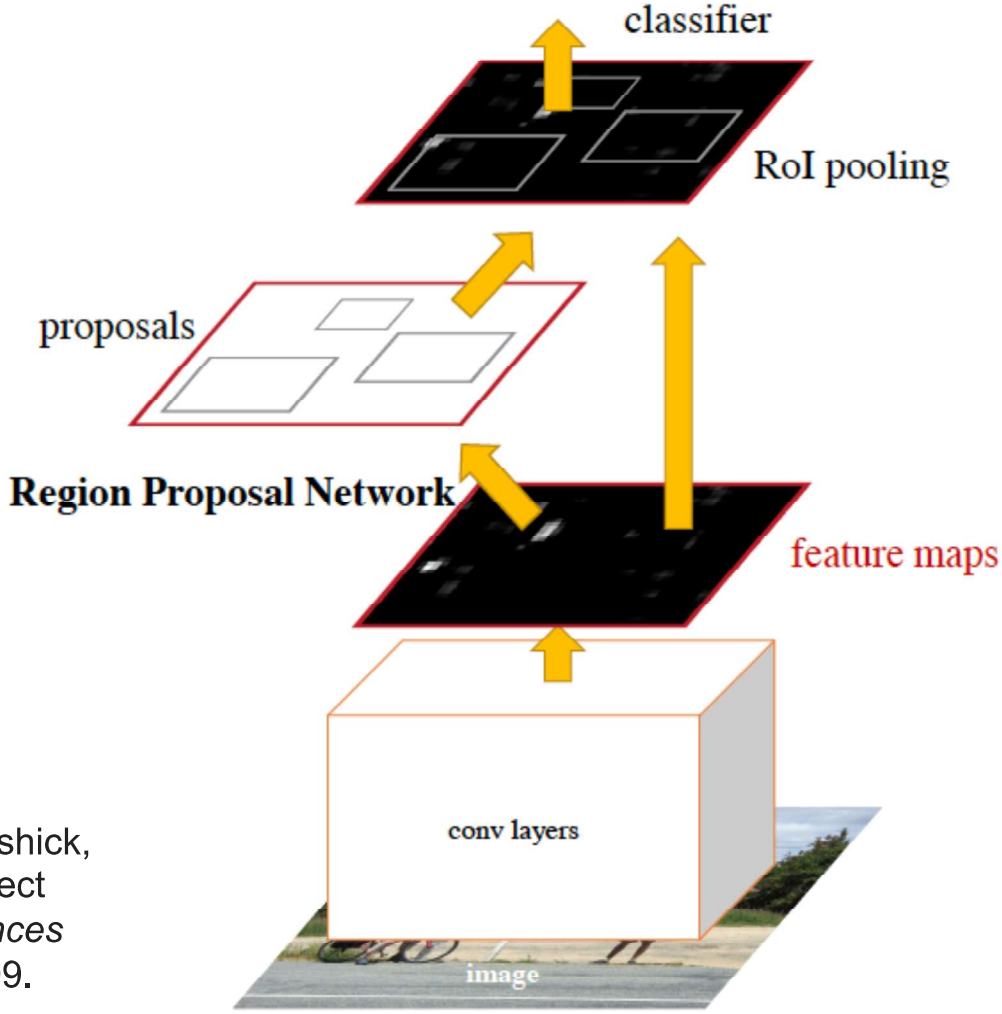


Figure from Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." In *Advances in neural information processing systems*, pp. 91-99. 2015.

Region Proposal Network

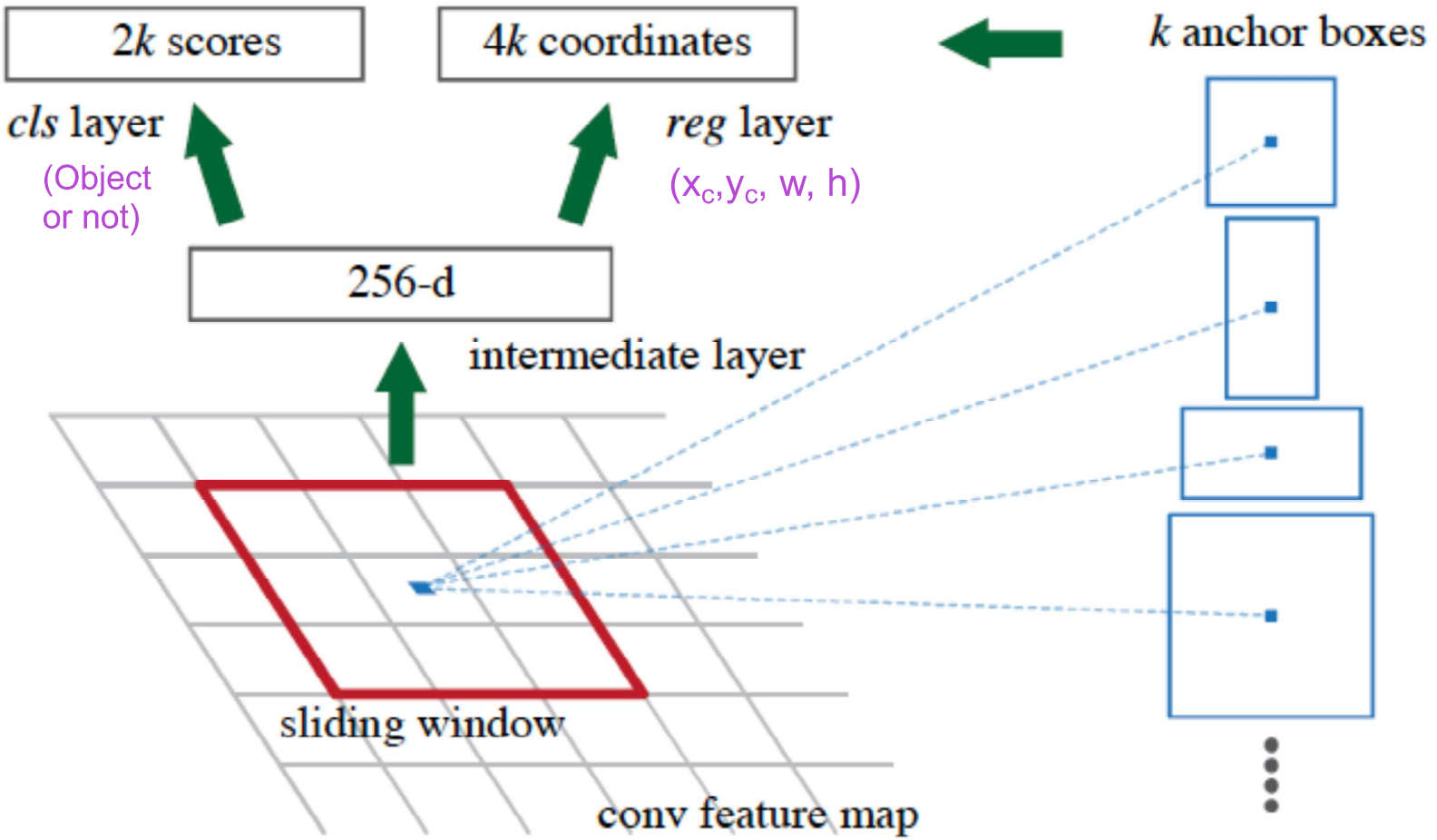


Figure from Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." In *Advances in neural information processing systems*, pp. 91-99. 2015.

Training of Faster R-CNN

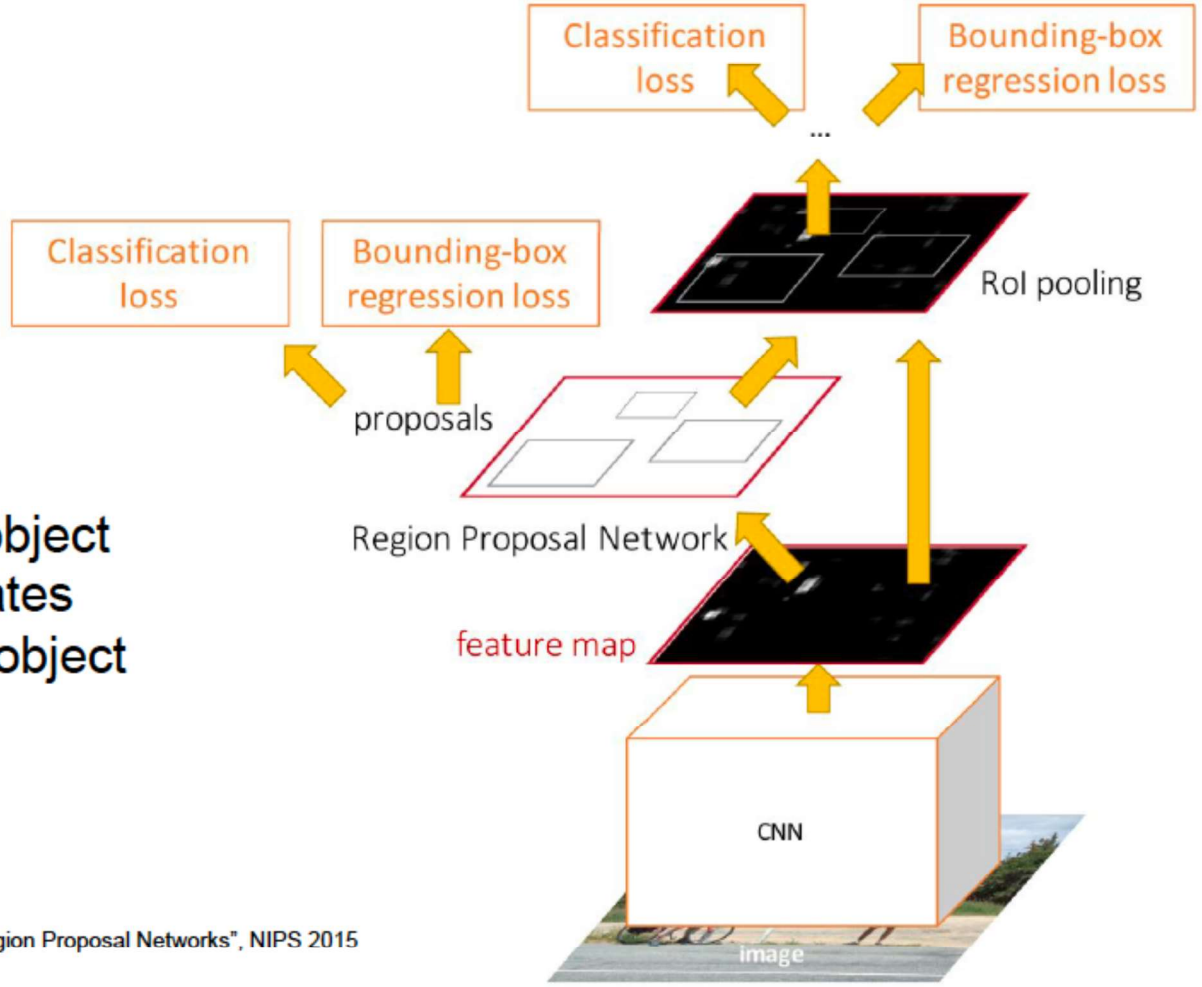
Faster R-CNN:

Make CNN do proposals!

Insert **Region Proposal Network (RPN)** to predict proposals from features

Jointly train with 4 losses:

1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates



Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

From http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

More Sample Results

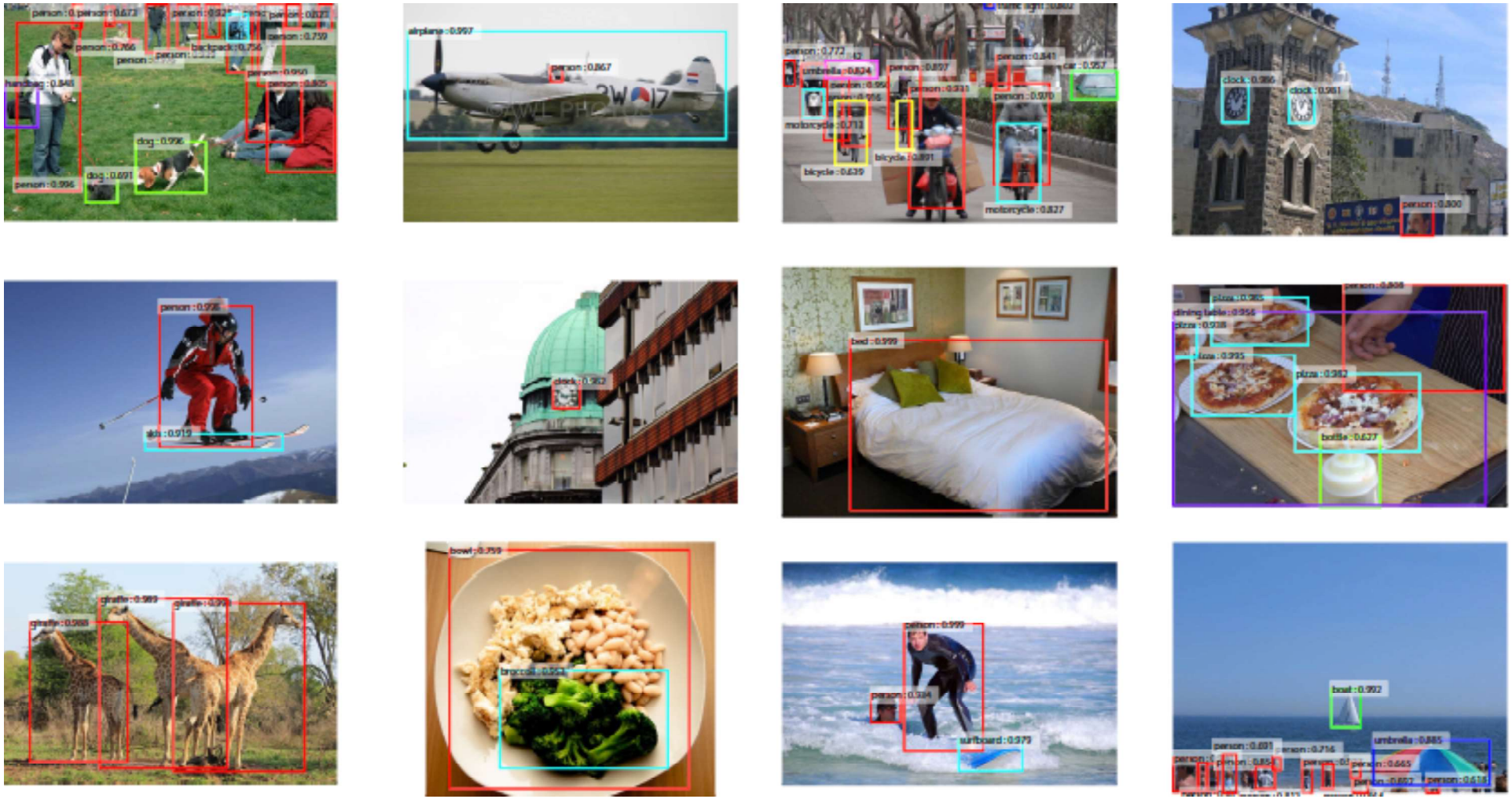
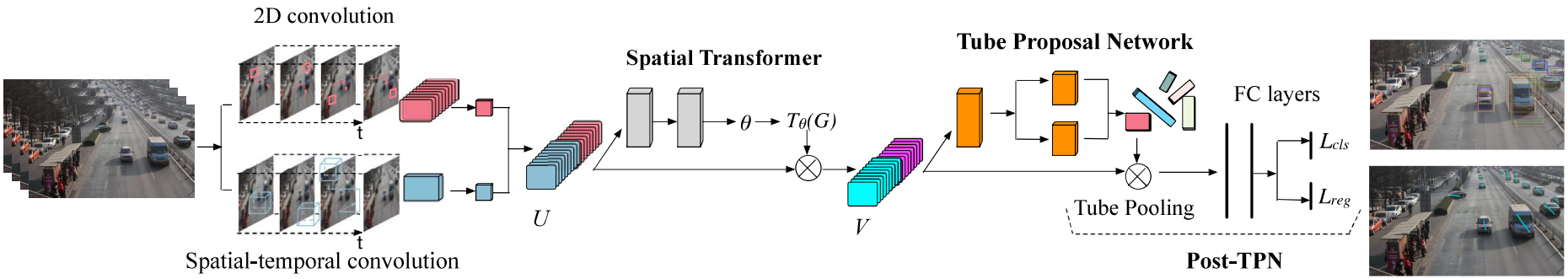


Figure from Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." In *Advances in neural information processing systems*, pp. 91-99. 2015.

Work at NYU Video Lab: Robust Vehicle Tracking at Urban Intersections

- Vehicle detection and tracking at very congested urban intersections
  - Traffic accidence analysis based on vehicle trajectory data
  - Joint project with NYU Center for Urban Science + Progress (CUSP)
- Challenges
  - Low resolution NYC Dept of Transportation surveillance videos
  - Severe occlusion in dense traffic
  - Vanishing point (non-bird eye view) viewing angles
  - Shadows and illumination changes
- Developing a deep learning network that can simultaneously detect and track a video object
  - Detect bounding tubes that cover moving objects in short video segments
  - Extension of faster region-CNN, which detects bounding boxes in individual frames
- Tandon Student: Chenge Li (Ph.D. 2019)

TrackNet Model Overview



Feature Extraction

We found using C3D alone is not as good as using both C3D and VGG

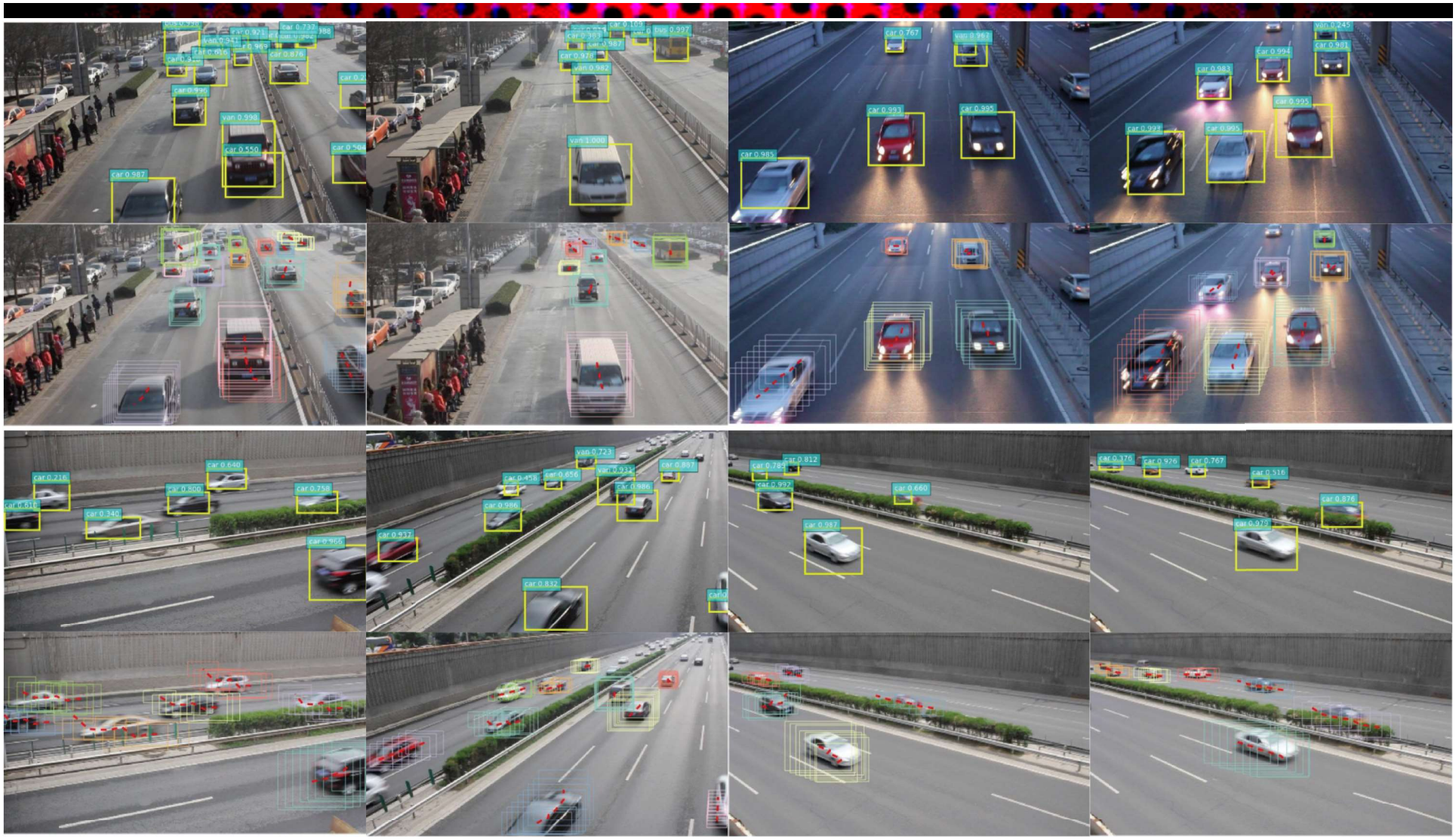
Transform feature maps so that traffic videos from different view angles have similar feature representation

At each candidate location, test multiple anchor tubes: generate a "objectness score" and the offset from the original anchor tube location and shape

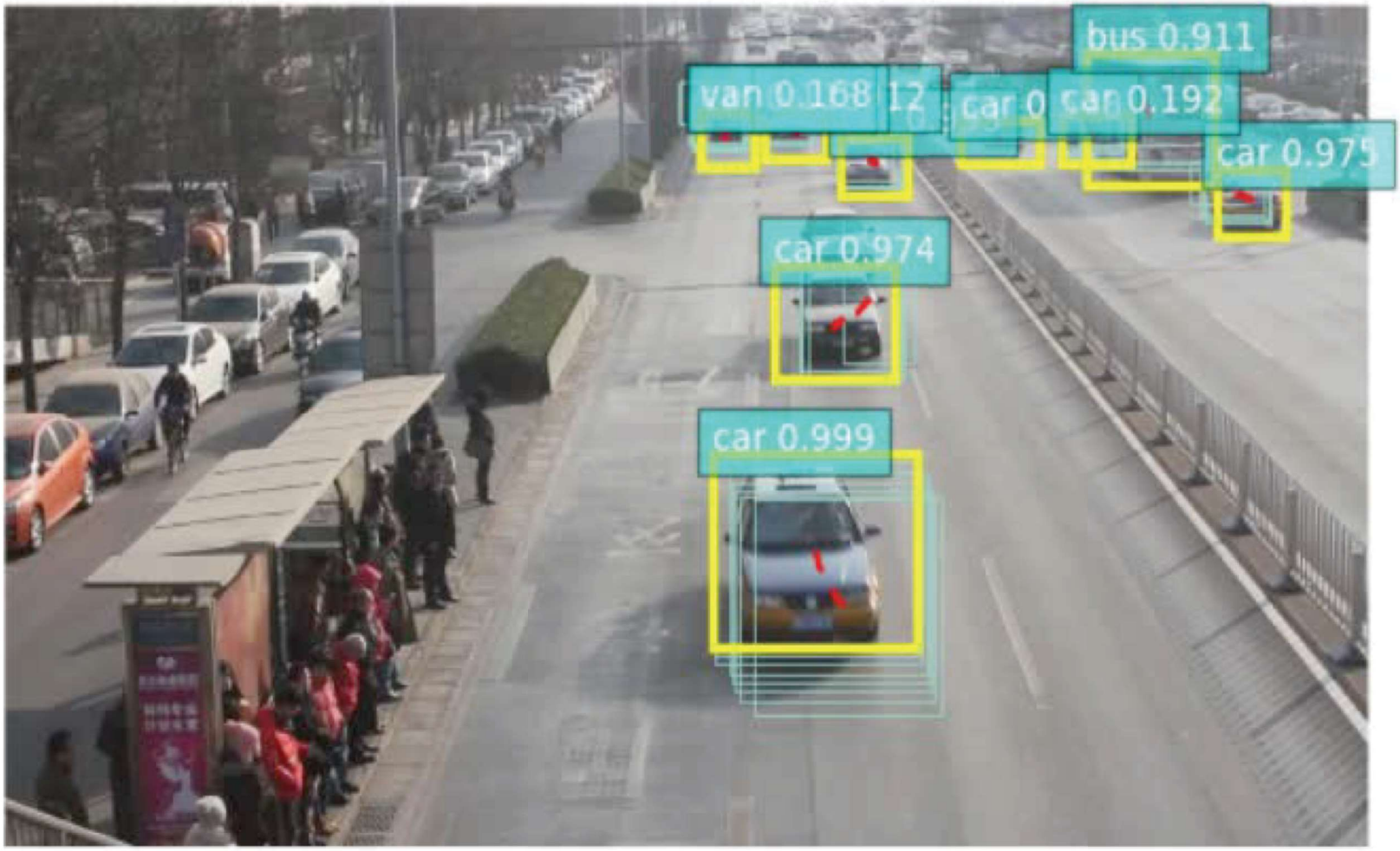
For each proposal tube with high "objectness score", further refine the tube parameters, and classify it into one of predefined object category

Chenge Li, Gregory Dobler, Xin Feng, Yao Wang "**TrackNet: TrackNet: Simultaneous Detection and Tracking of Multiple Objects**", https://arxiv.org/abs/1902.01466

Sample Results: Snap Shot

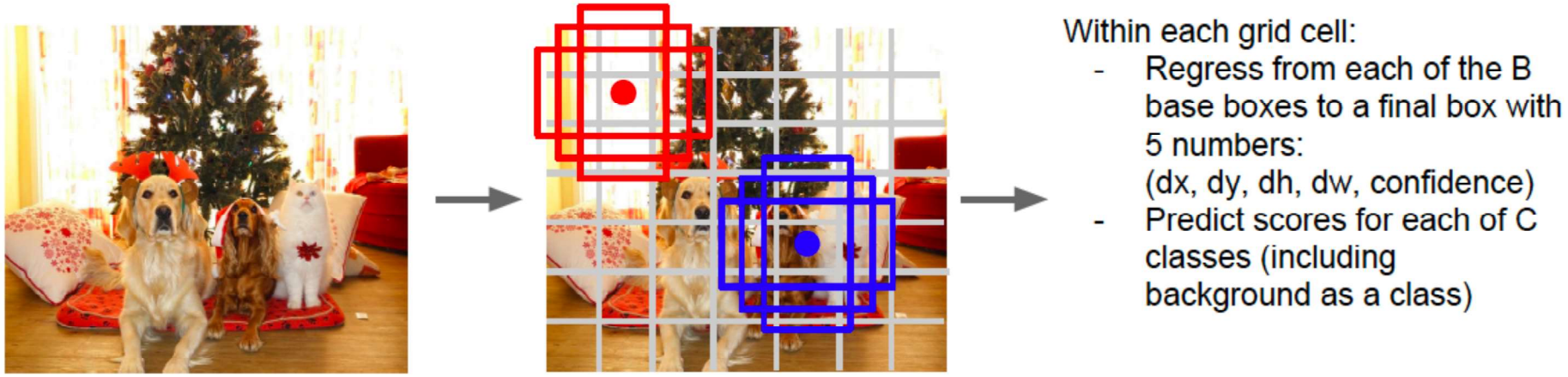


Sample Results: Video



Detection without Proposals: YOLO / SSD

Go from input image to tensor of scores with one big convolutional network!



Input image
3 x H x W

Divide image into grid
7 x 7

Image a set of **base boxes** centered at each grid cell
Here B = 3

Within each grid cell:
- Regress from each of the B base boxes to a final box with 5 numbers: (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)

Output:
7 x 7 x (5 * B + C)

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al. "SSD: Single-Shot MultiBox Detector", ECCV 2016

From http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Faster than faster R-CNN, but not as accurate

- Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788. 2016.
- Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "SSD: Single shot multibox detector." In *European conference on computer vision*, pp. 21-37. Springer, Cham, 2016.
  In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788. 2016.

Outline (Part III)

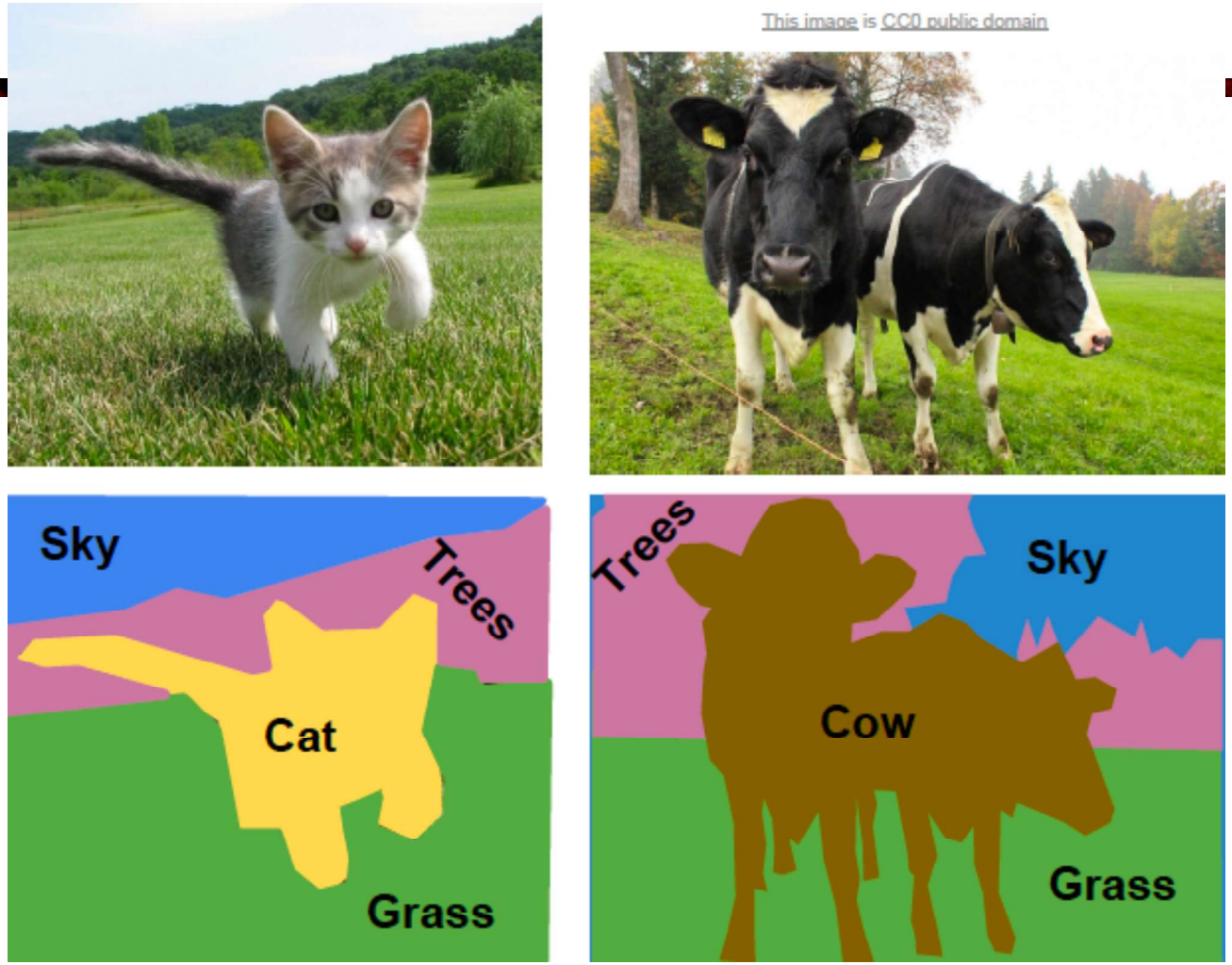
- Image to image autoencoder
- Semantic Segmentation using Multiresolution Autoencoder
- Object detection and classification
- Instance segmentation

Instance Segmentation

Each instance of the same type of object is given a different color!

From: He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn." In *Proceedings of the IEEE international conference on computer vision*, pp. 2961-2969. 2017. https://arxiv.org/pdf/1703.06870.pdf

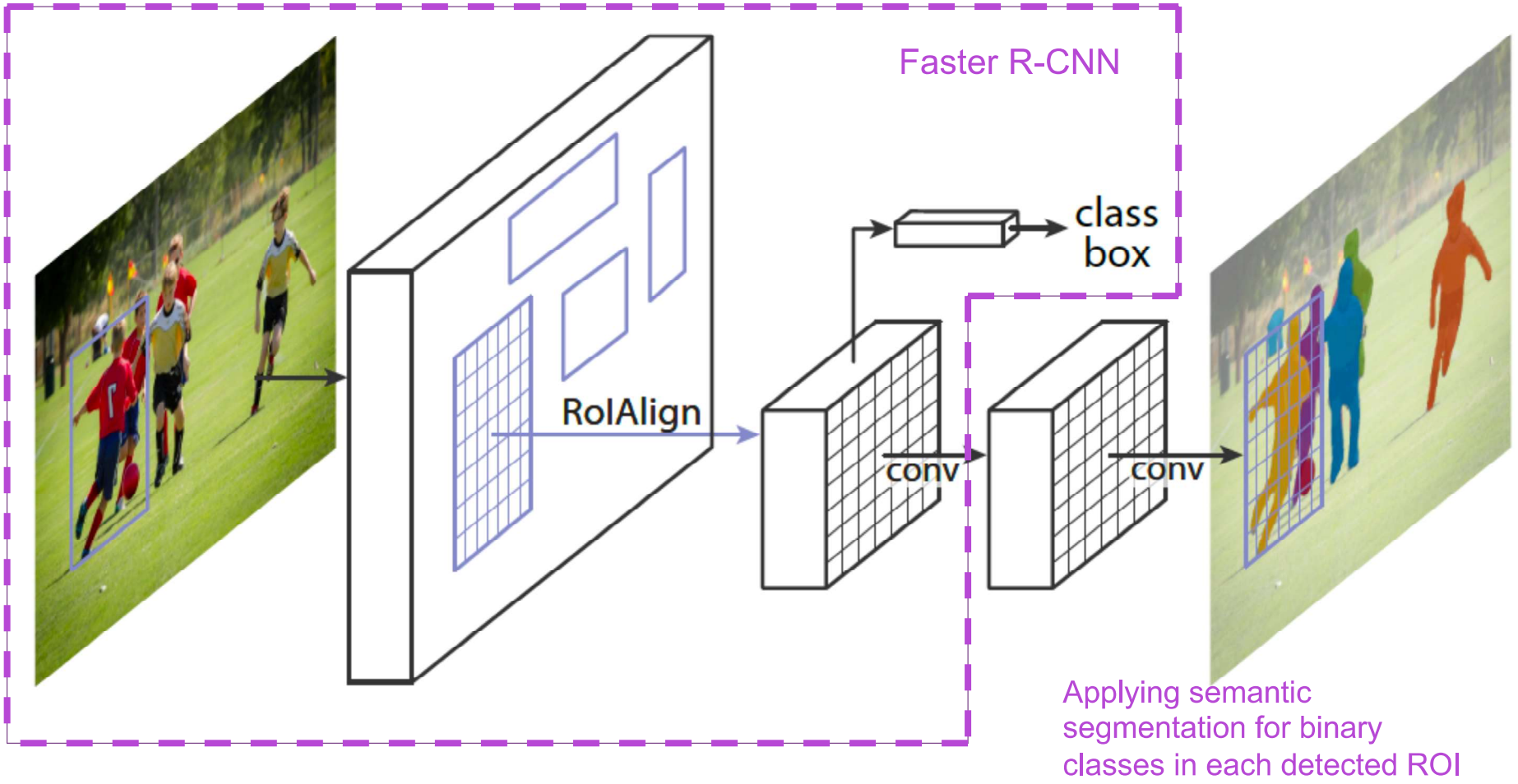
Semantic Segmentation



Each pixel is labeled into one object class: The two cows have the same color!

From http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Mask-RCNN for Instance Segmentation



From: He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn." In *Proceedings of the IEEE international conference on computer vision*, pp. 2961-2969. 2017. https://arxiv.org/pdf/1703.06870.pdf

Summary

- Autoencoder and image processing applications
- Multiresolution autoencoder for segmentation (U-Net, V-net)
  - Combine features at multiple resolutions
- Simultaneous region detection and labeling (R-CNN)
- Instance segmentation (Mask R-CNN)

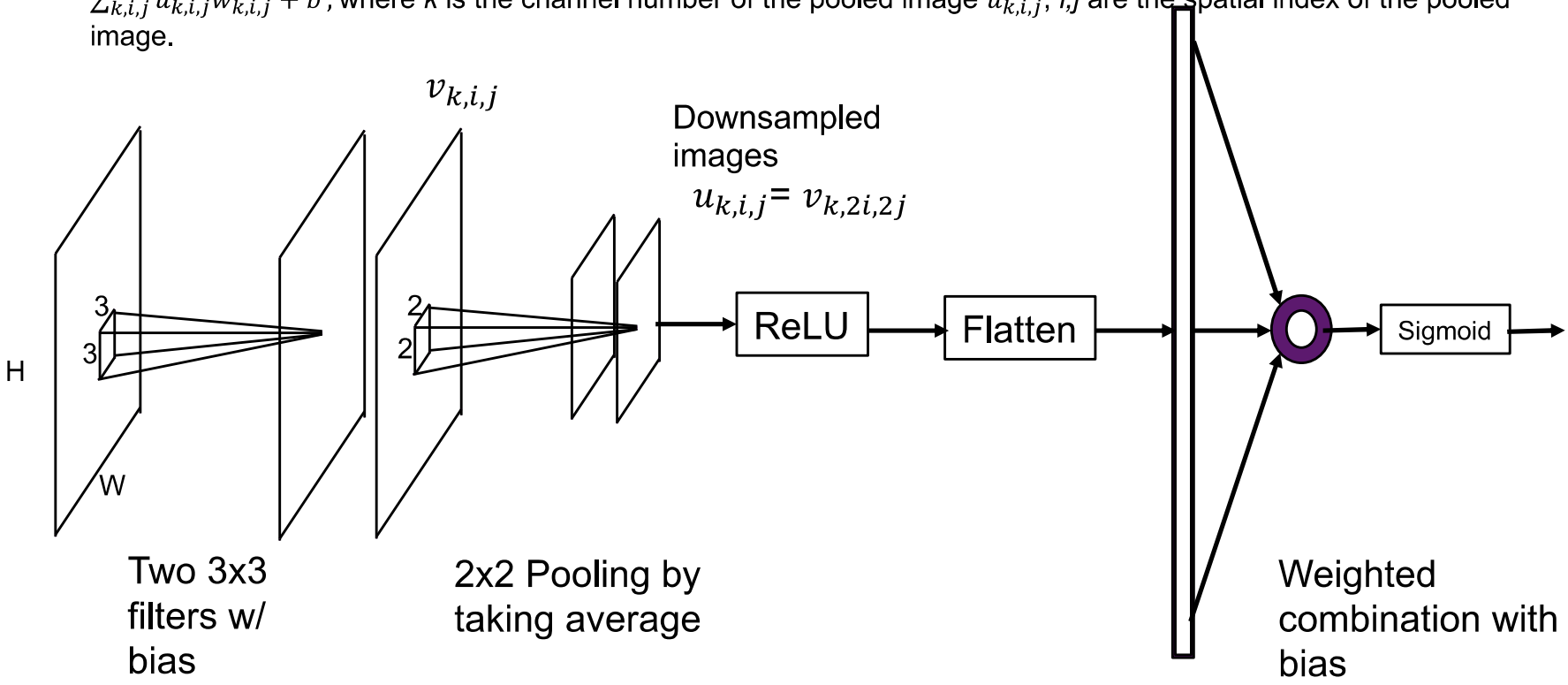
Recommended Readings

- For vision applications:
  - Stanford course by Feifei Li, et al: CS231n: Convolutional Neural Networks for Visual Recognition, Spring 2018: http://cs231n.stanford.edu/
  - Objection detection and segmentation: http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf
  - Popular network case studies: http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture09.pdf
  - Learning GPU and PyTorch and TensorFlow: http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture08.pdf
  - Video available for previous offerings:
    - https://www.youtube.com/playlist?list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYvhttps://www.youtube.com/playlist?list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv

Written Assignment

1. For the simple network structure shown below (1 conv layer followed by downsampling and 1 fully connected layer), determine the number of trainable parameters. Assume the input image is gray scale (one channel, with height=H, width=W).

2. (Bonus point, optional) Assuming we use the binary cross entropy loss. Derive the gradients of the loss for one sample (input=one image=x(m,n) with true label y) with respect to all the parameters in the network. (Hint: to simplify your derivation, index the weights of the fully connected layer by the image index, i..e, $u_O = \sum_{k,i,j} u_{k,i,j} w_{k,i,j} + b$, where *k* is the channel number of the pooled image $u_{k,i,j}$, *i,j* are the spatial index of the pooled image.



Written Assignment

3. Consider the following ”very simple” segmentation network. Assuming we use SOFT Dice as the loss function. Derive the gradients with respect to the filter weights and bias of the last layer.

4. Why does “residual connection” help the network training?

5. Consider a network with 2 convolutional layers. 1) Suppose each layer uses 3x3 filters without dilation, what is the perceptive field size of each output pixel? 2) Suppose the first layer uses 3x3 filter without dilation, and the second layer uses 3x3 filters with a dilation rate of 2. What is the perceptive field of each output pixel? 3) Now suppose each layer uses 3x3 filters without dilation, but there is a 2x2 downsampling after the first layer. What is the perceptive field? Discuss the pros and cons of these methods.