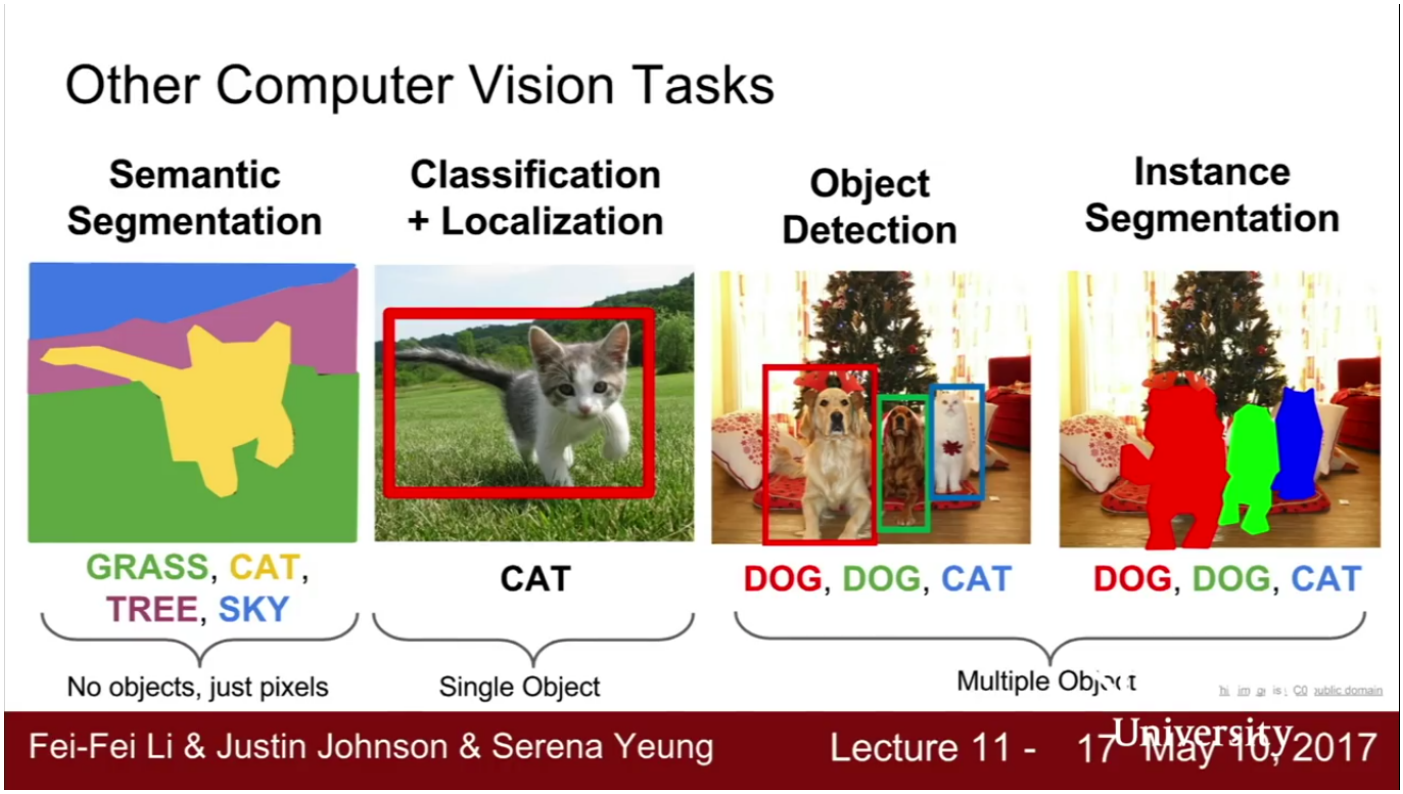
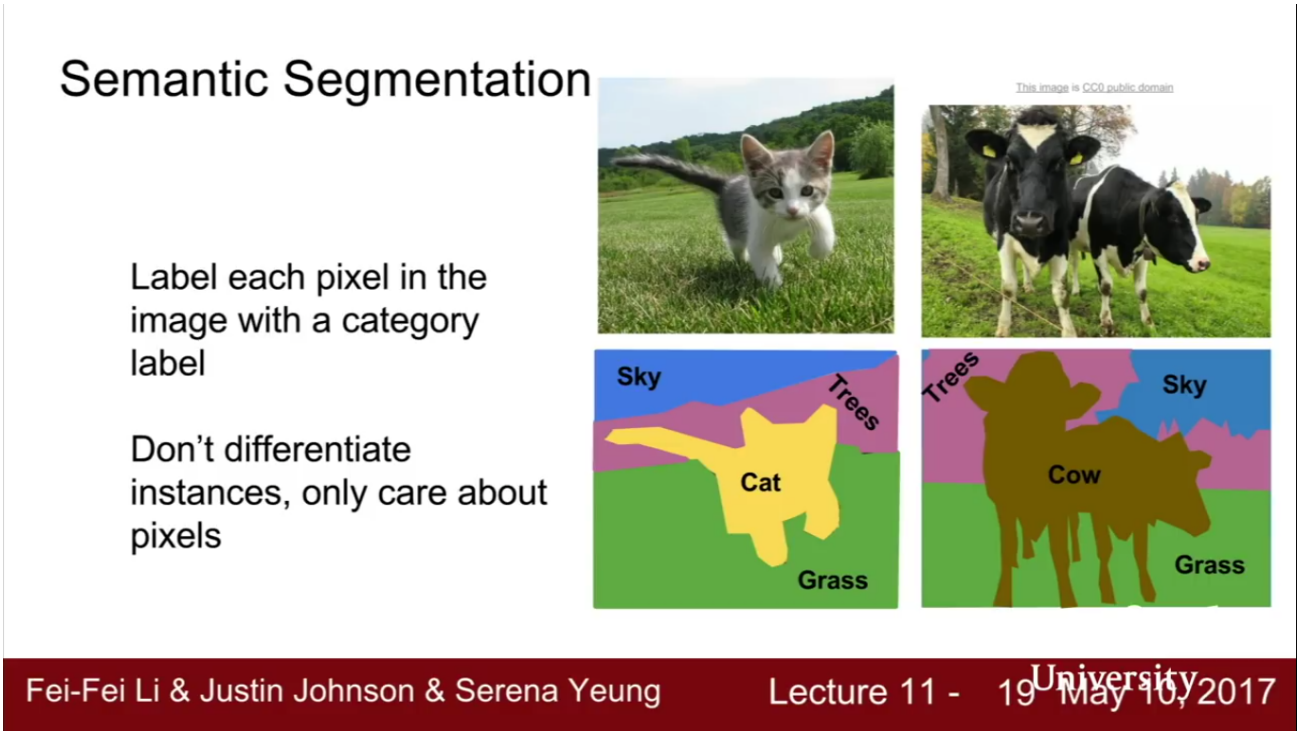
lec11

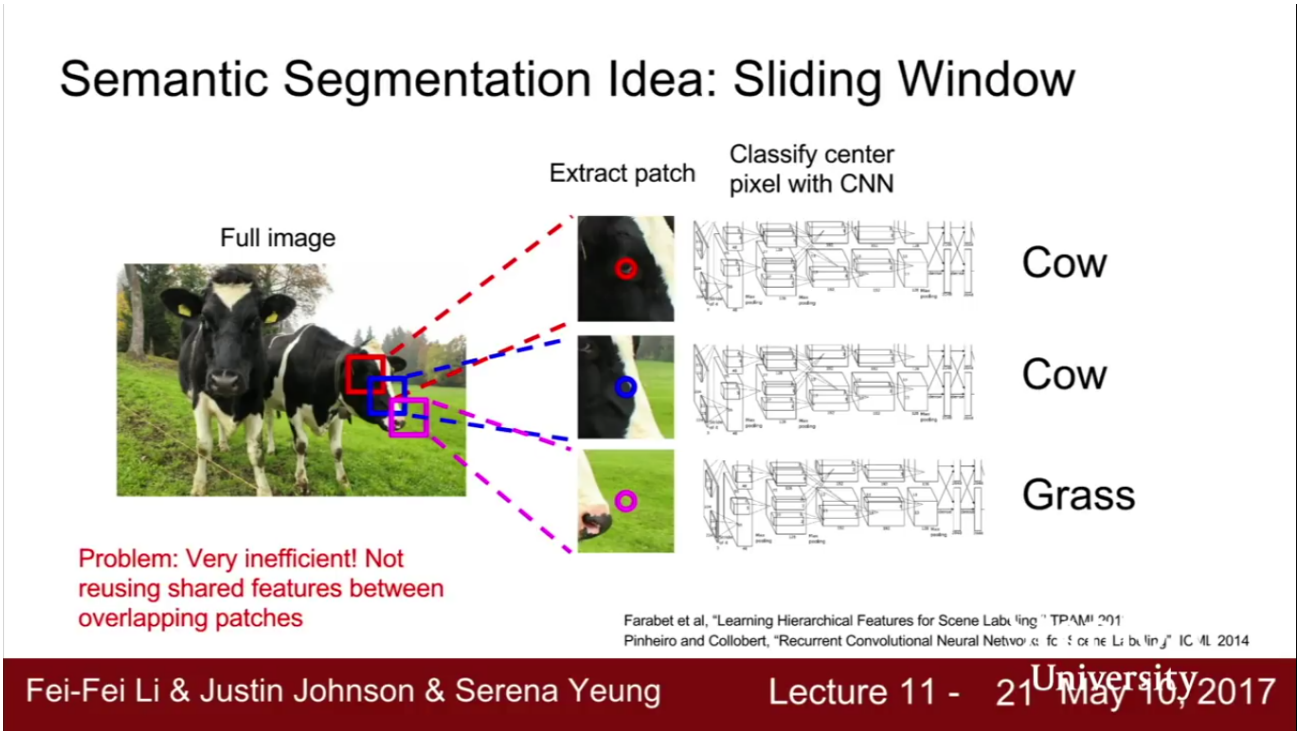
Creation Date: 11/01/2020 16:31
Last Modified Date: 12/01/2020 00:46

Lec 11: Segmentation, Localization and Detection

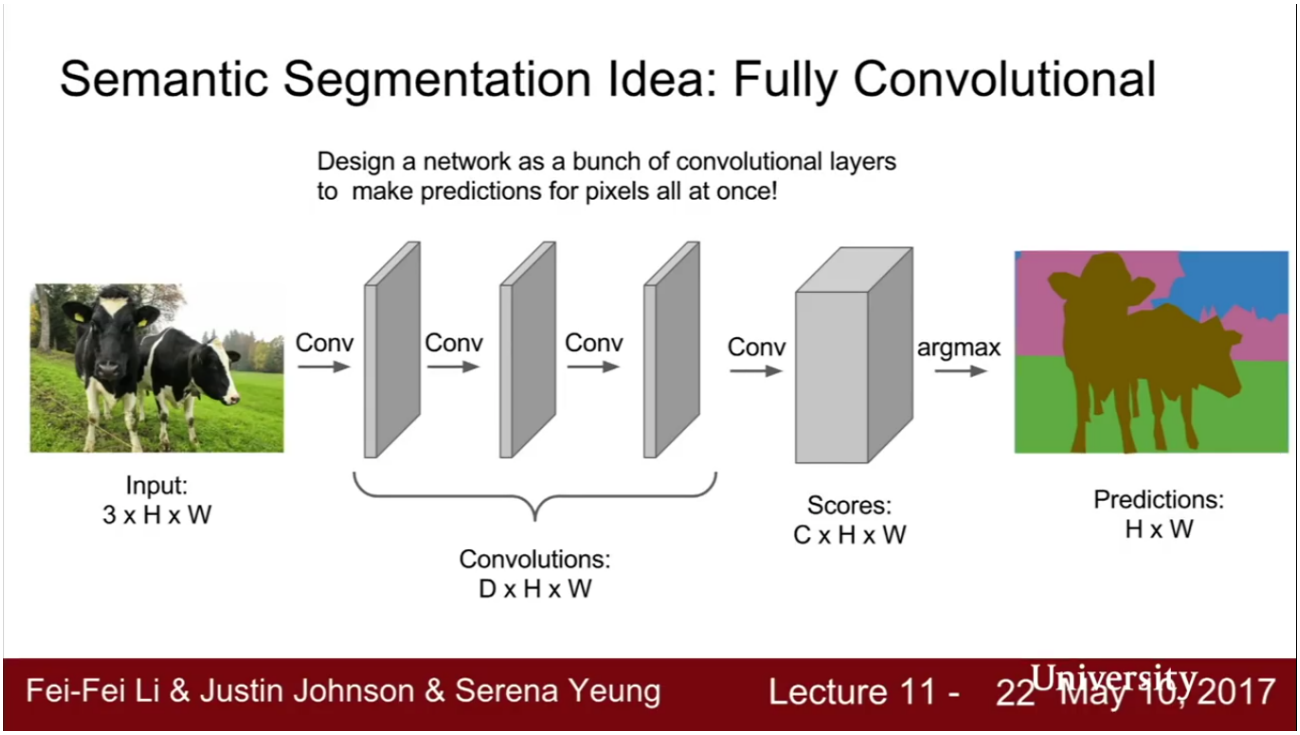


- Semantic Segmentation:
  - The goal is to classify each pixel in the given image.

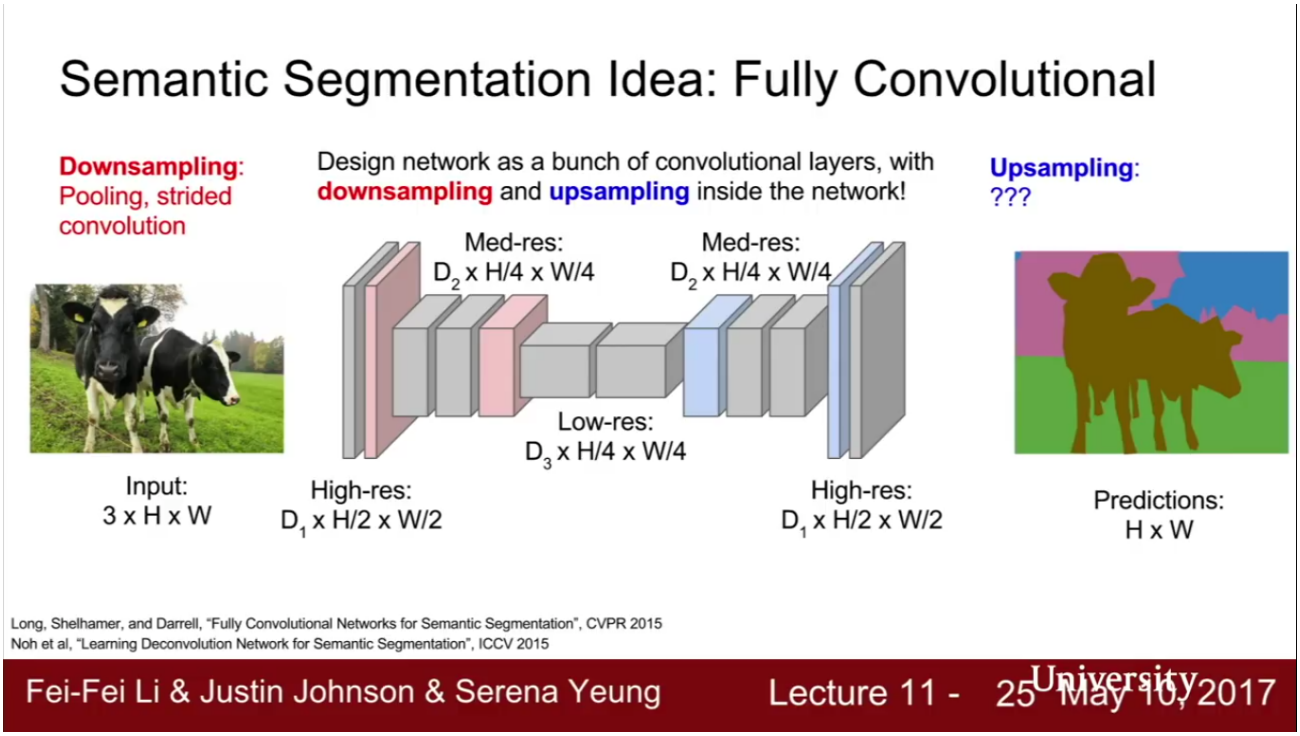
- 

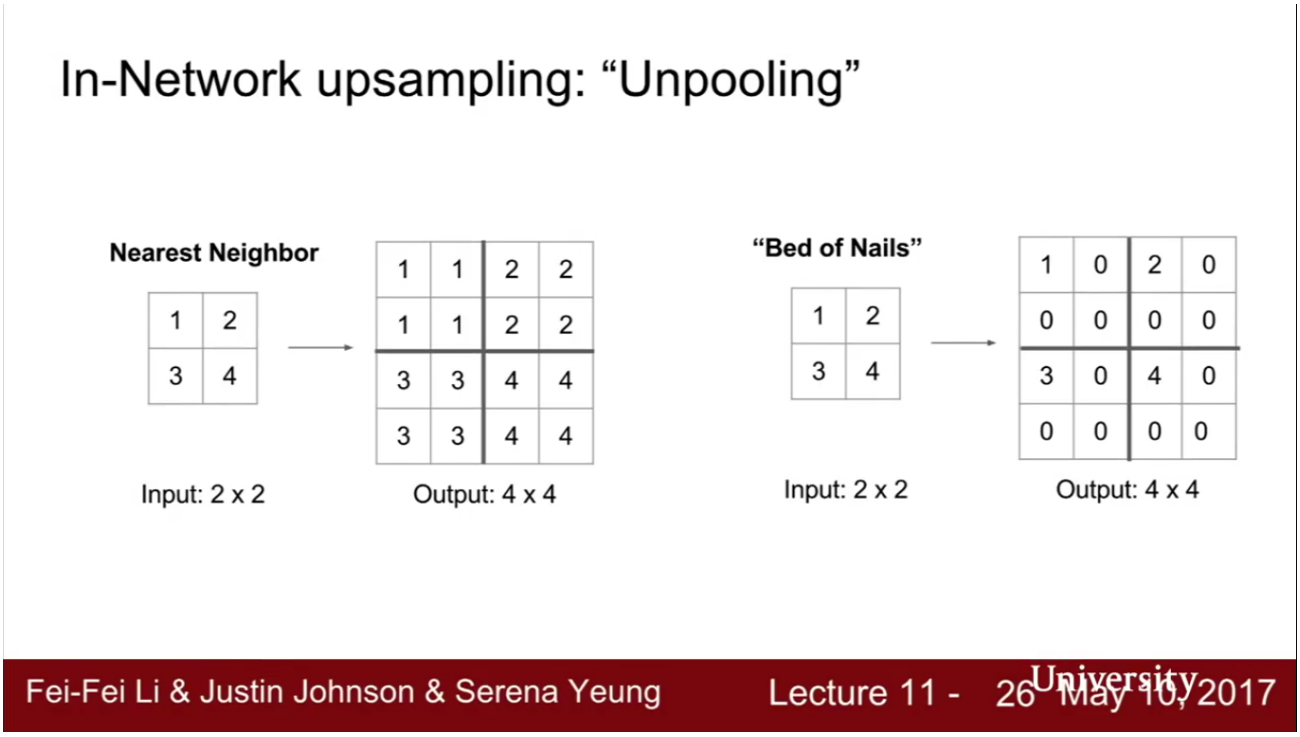
- 

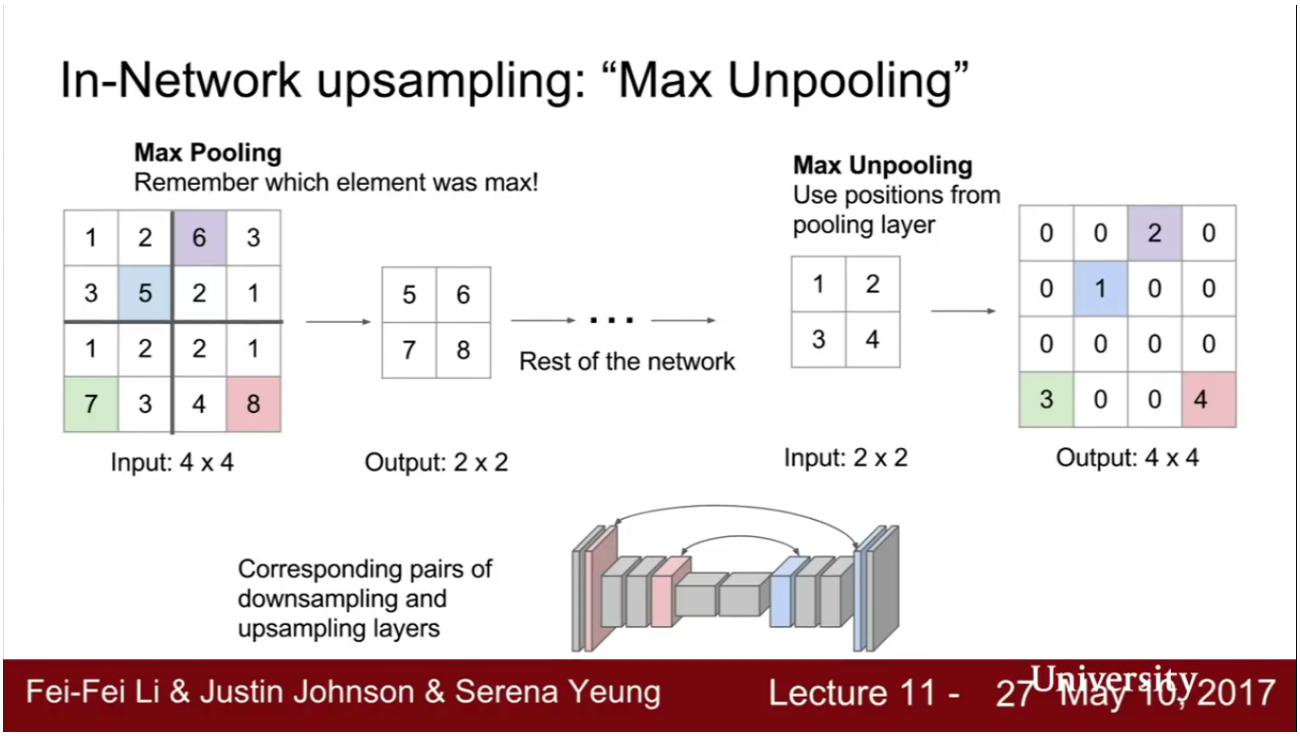
- In sliding window technique we want to know what is the category label for the center pixel of the small slide/patch.



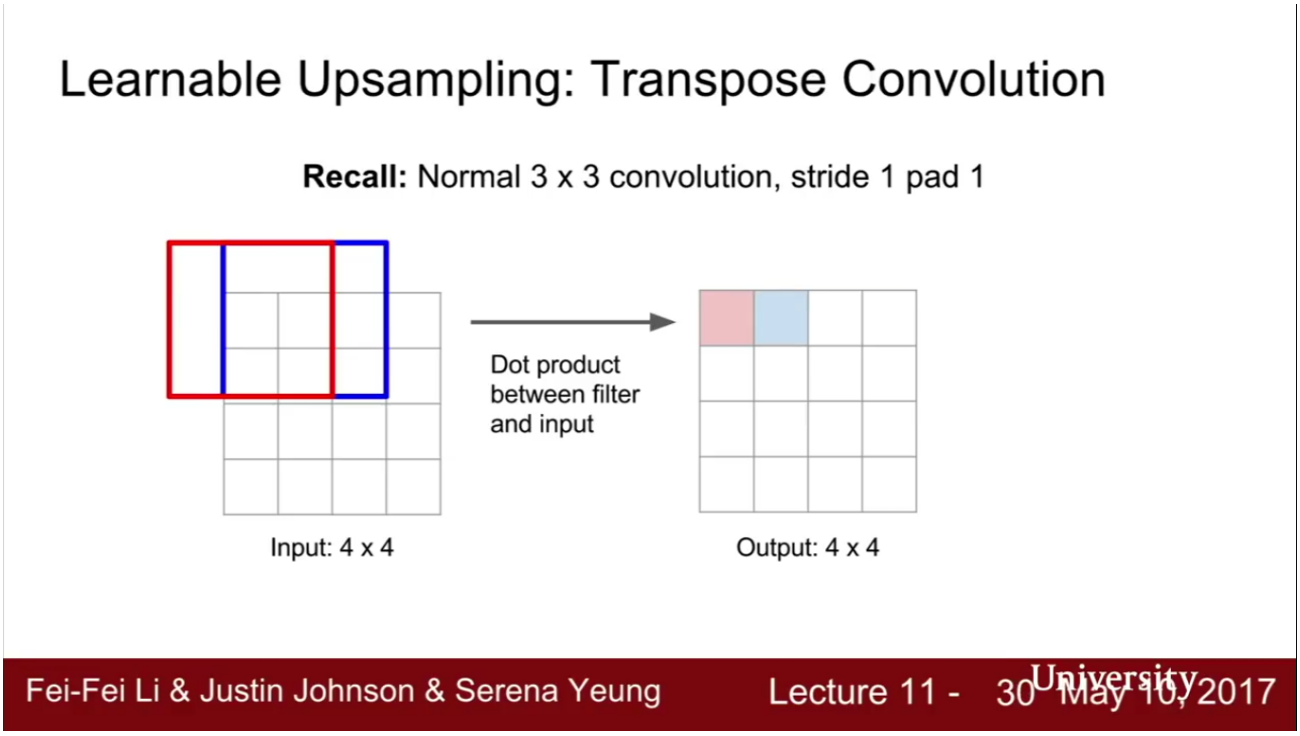
- Here we are preserving the spatial size of the image passing through the network and in the end we will have a tensor where the depth will have C channels which shows the probablity map of each class in it's respective layer.
- During training we have cross entropy loss for each pixel and then sum or average over the minibatch.
- Drawbacks:
  - Extremely heavy model since we will have lots of channels in between and we are also maintaining the spatial resolution of the image through the network.
  - Training time will be high since there are lots of convolutional operations being performed.
  - Solution: **Down-sample and then up-sample the image.**

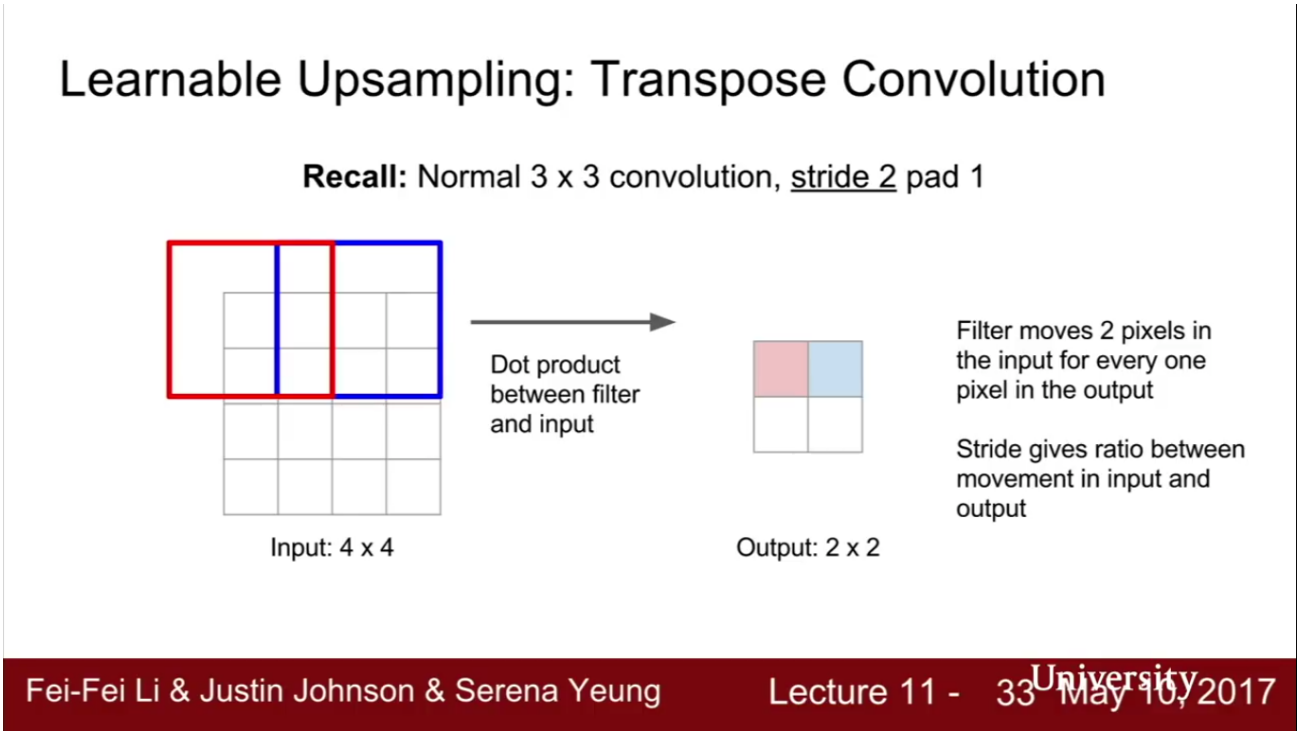


- For Down-sampling we have various techniques like pooling, strided convolutions.
- Most of the networks have a different way of up-sampling.
- 

- 

- If we do not use the max-pool indices we wont be knowing where to put the value, because in up-sampling the dimension will increase and since we have a down-sampled image, we have lost the spatial information from the local receptive field.
- **The techniques such as Nearest-Neighbours, Bed-of-Nails, Max-Unpooling with indices are fixed functions.**
- Why don't we learn the Up-sampling?
- Ans: ***Tranpose Convolution***

- 

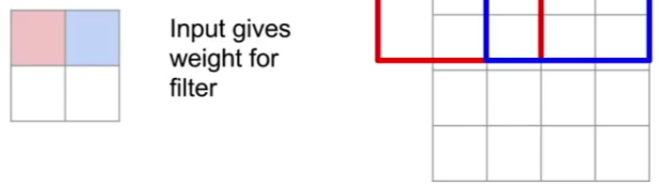
- 

- Transpose Convolution:

Learnable Upsampling: Transpose Convolution

Other names:
-Deconvolution (bad)
-Upconvolution
-Fractionally strided convolution
-Backward strided convolution

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



Input gives weight for filter

Sum where output overlaps

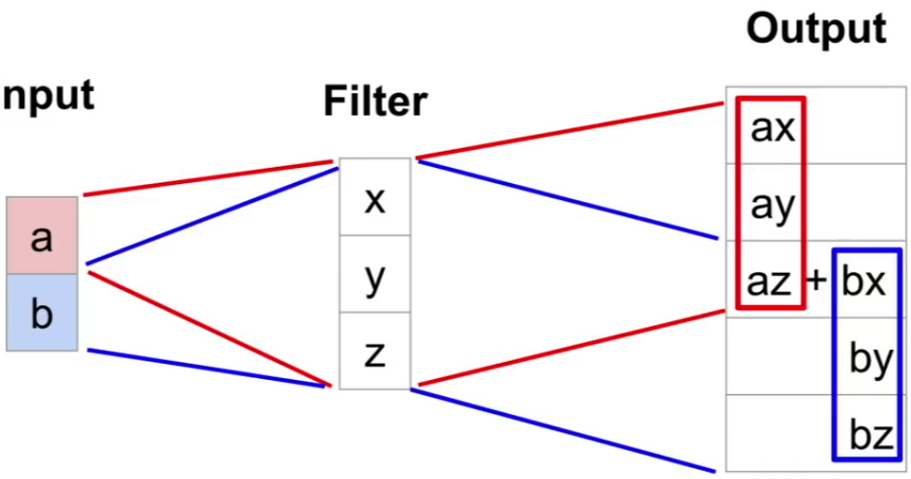
Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

Input: 2 x 2

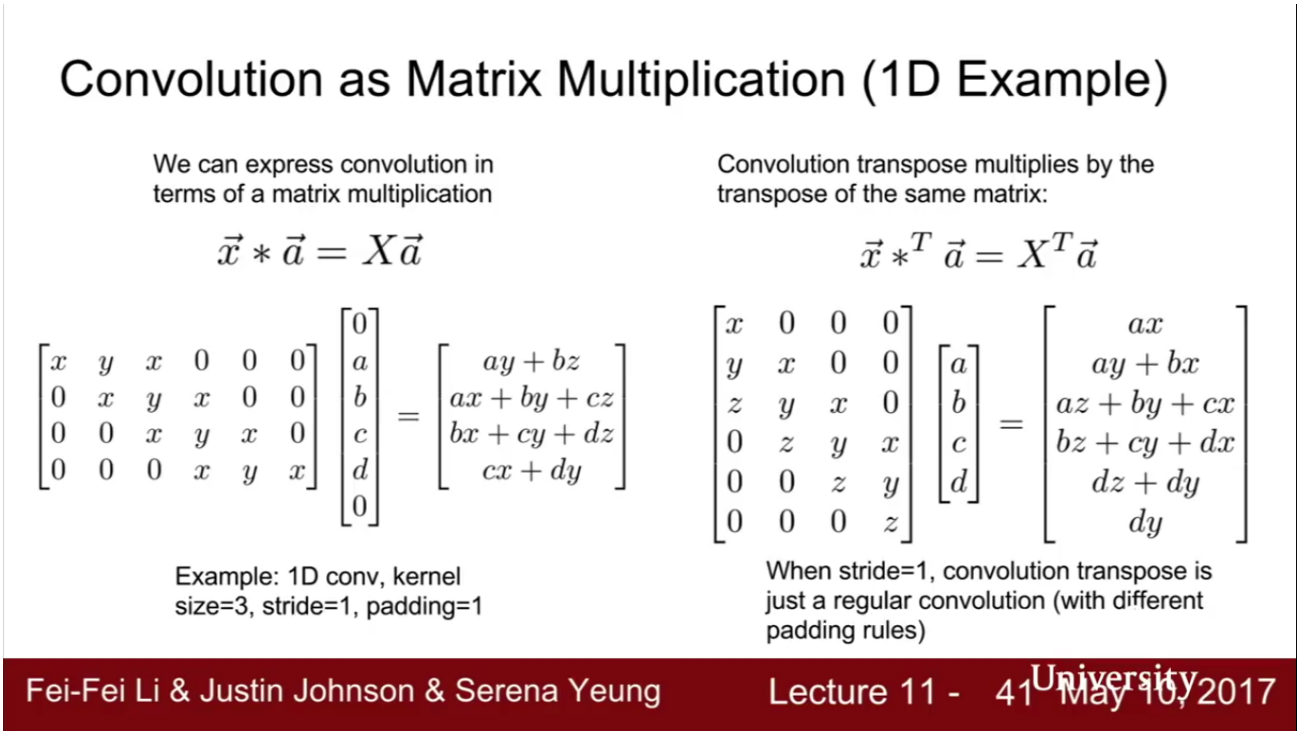
Output: 4 x 4

Transpose Convolution: 1D Example

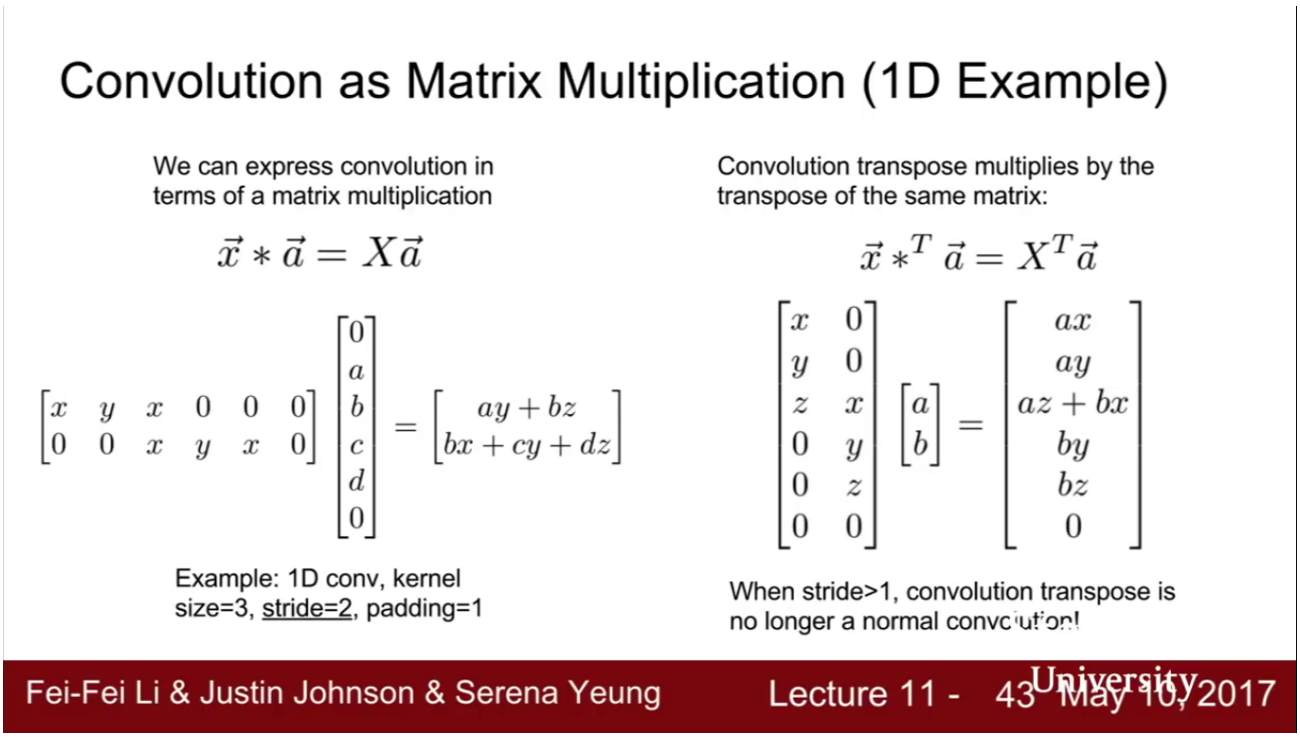


Output contains copies of the filter weighted by the input, summing at where at overlaps in the output

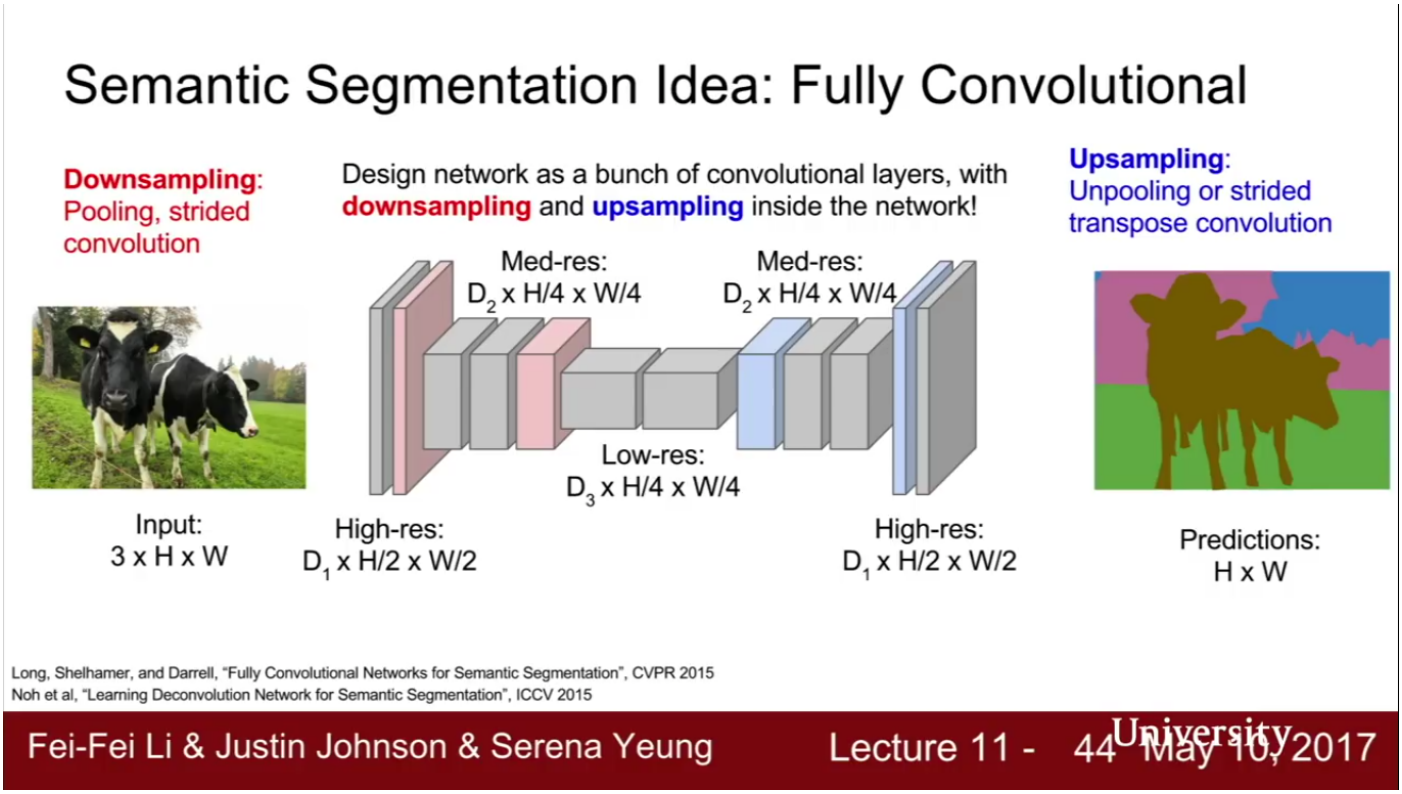
Need to crop one pixel from output to make output exactly 2x input



- Here [a,b,c,d] is the input vector padded with 0, and the kernel is [x,y,z].

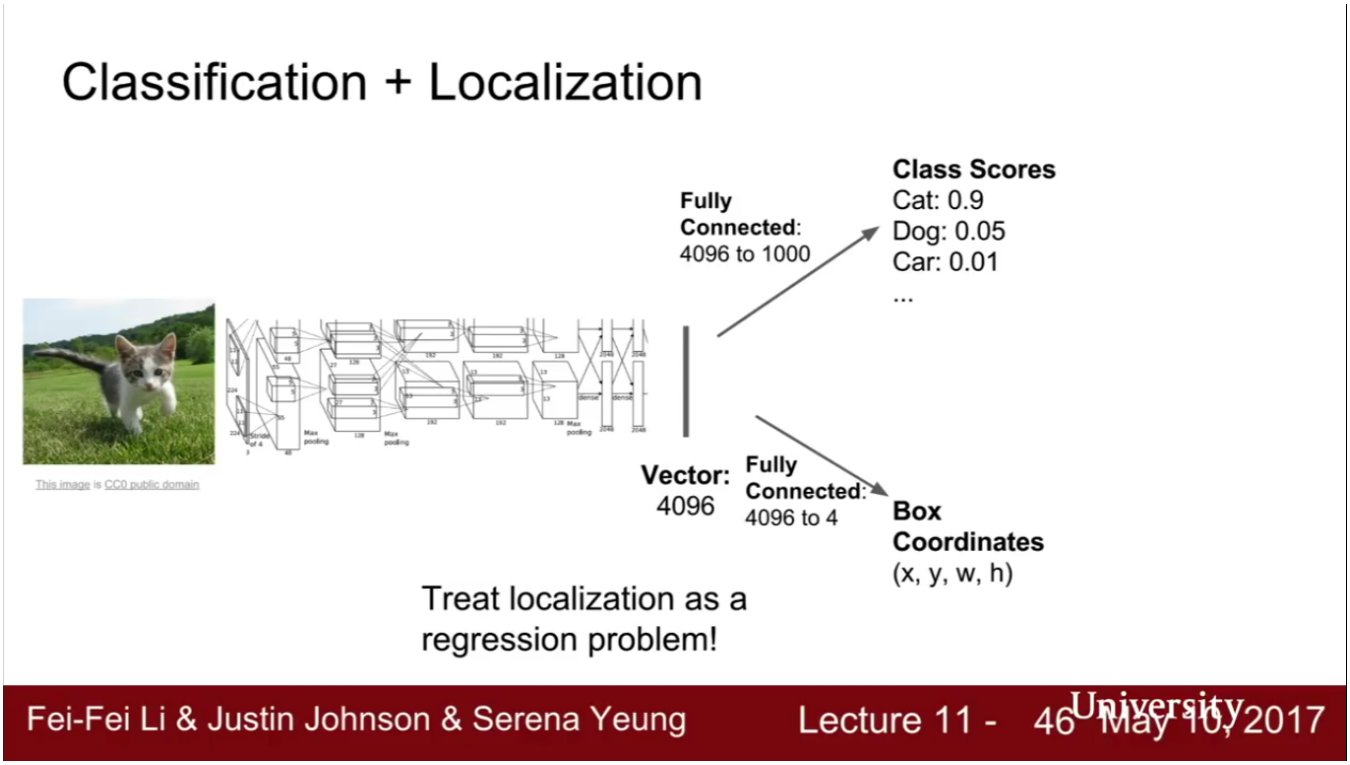


- We see **checkerboard** patterns for higher strides and for filter of odd sizes.

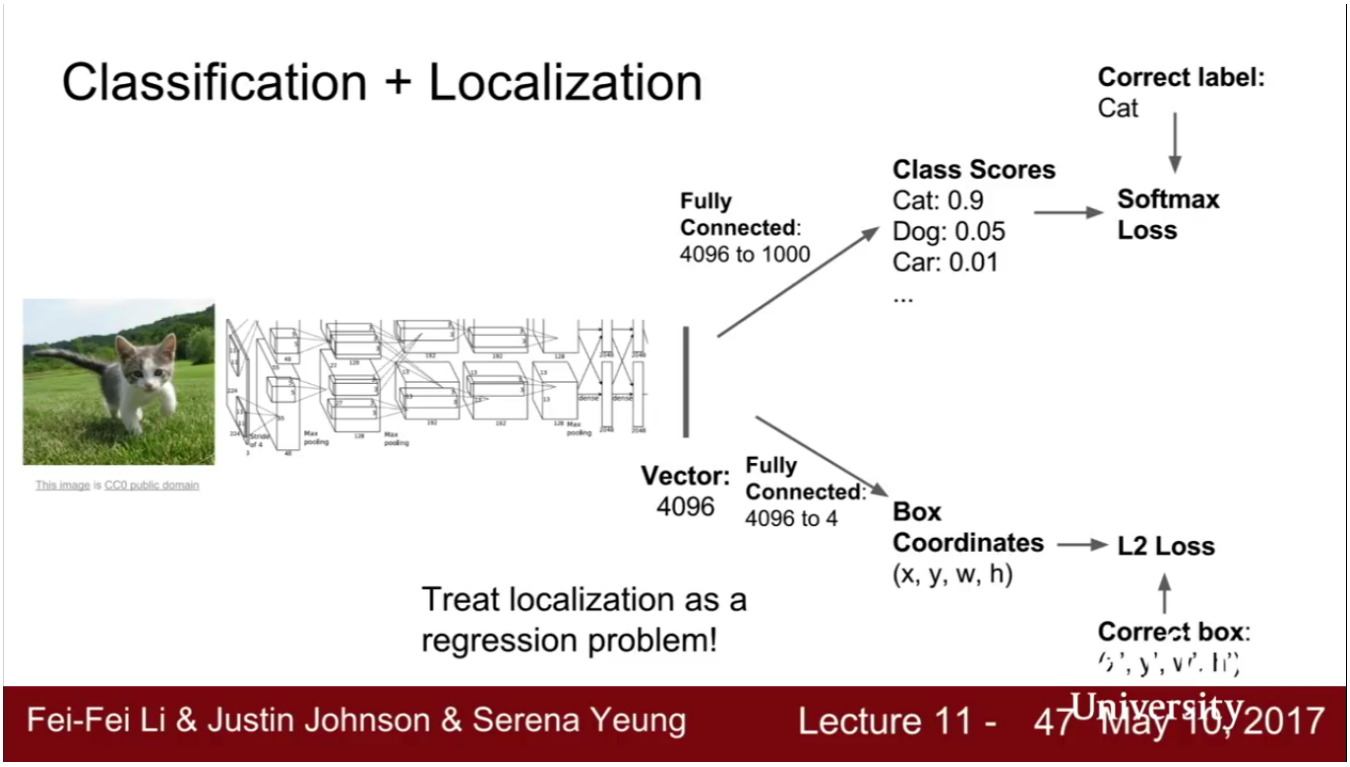


Classification + Localization

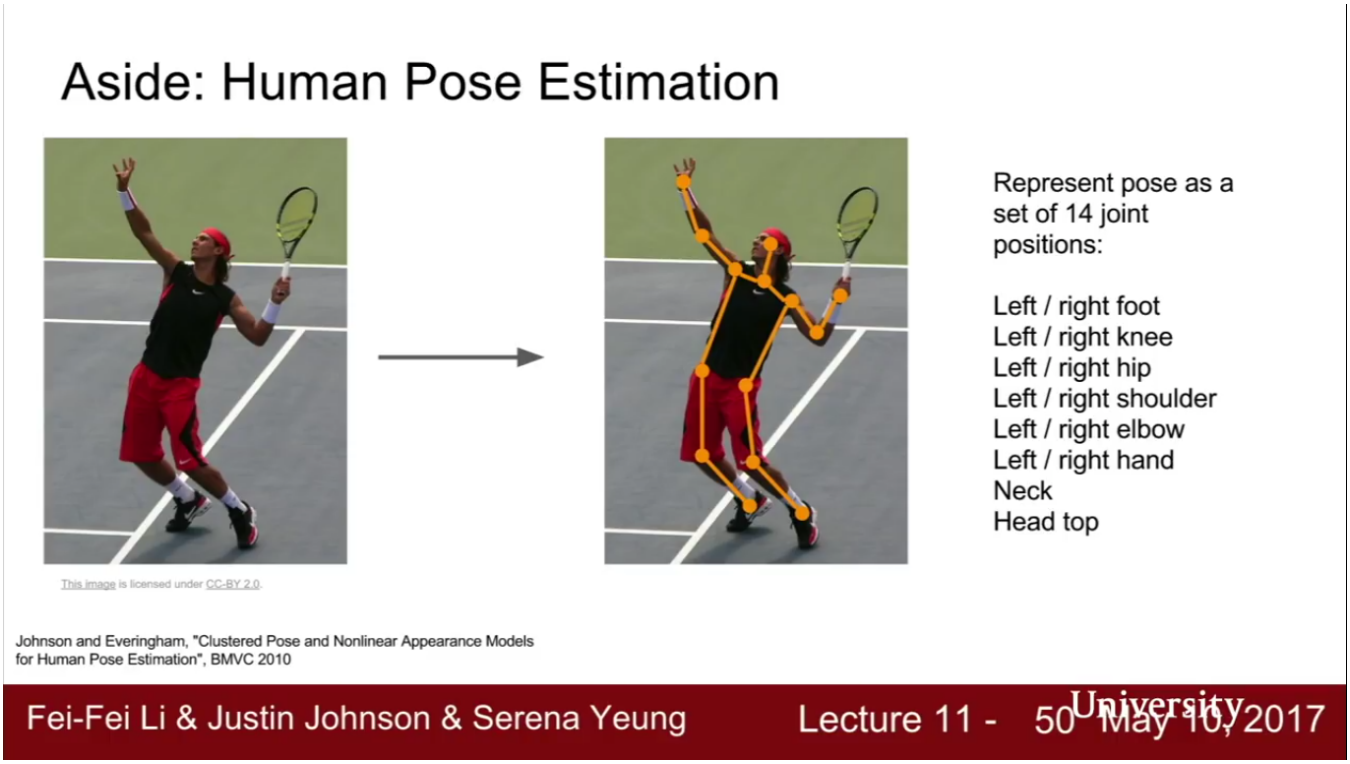
- Here along with classifying an object we also want to draw a bounding box to localize the object present in the image

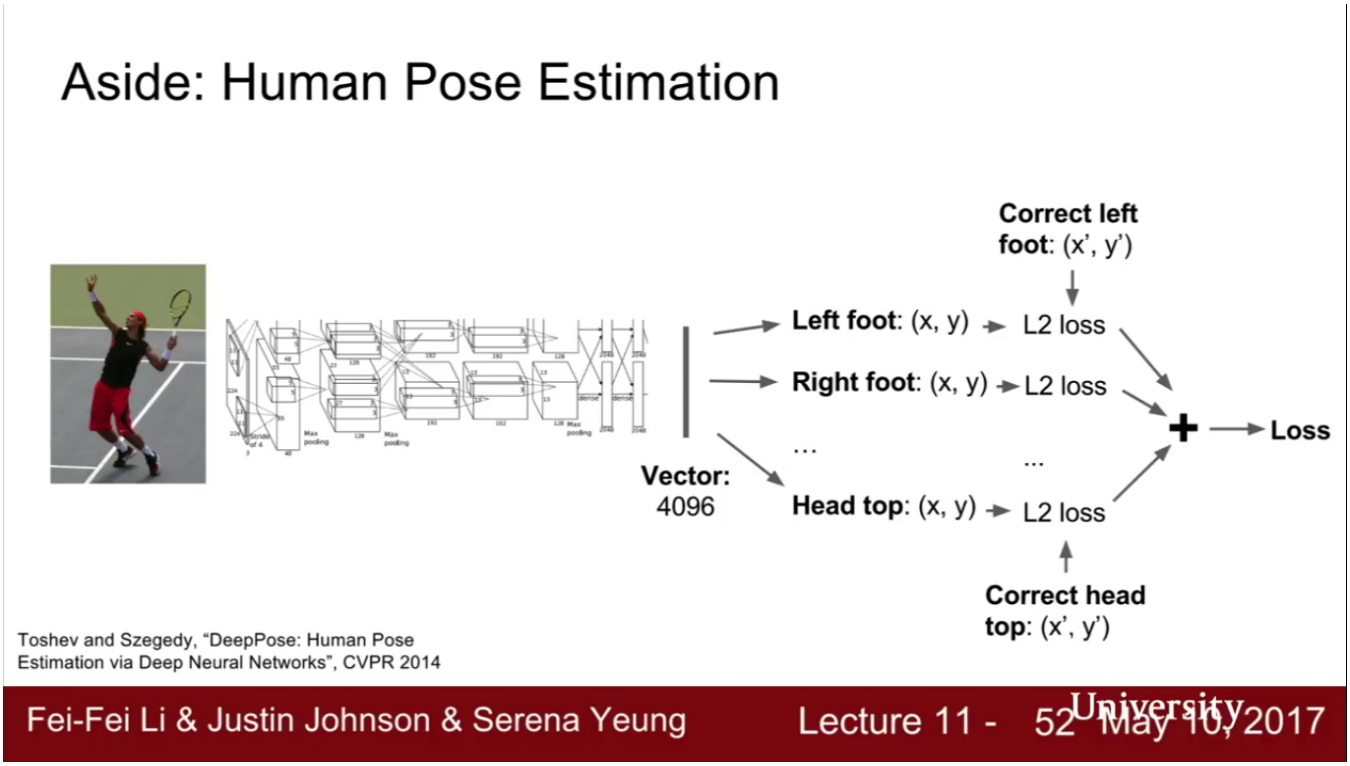


- One set of FC layer can perform the classification whereas the other can generate the Box Coordinates.



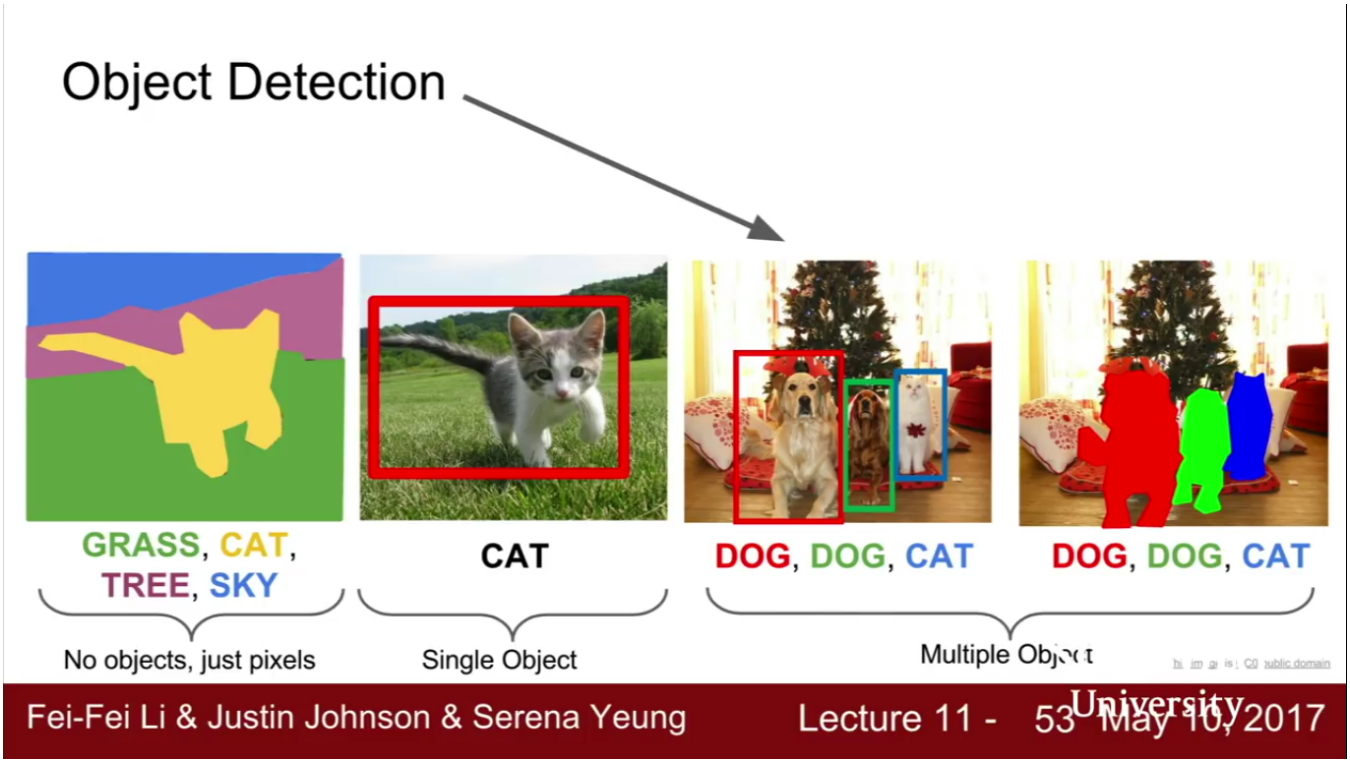
- The L2 loss can be used for the bounding box coordinates and Since we are having only 1 object in the image we can use Softmax Loss.
- These kind of losses in the task is called as **Multi-Task Loss.**
- **In Multi-Task Loss, if we have k loss terms(scalars), we perform a weighted sum to create a single loss scalar.** And we can use this weighted loss terms for finding the gradient.
- So since the weighting parameter is itself a hyperparamter in such scenarios, we are manipulating the absolute value of the losses. So the other hyperparameters like learning rate etc will vary.
- **Hence various approaches look at optimizing the metric associated with the tasks **



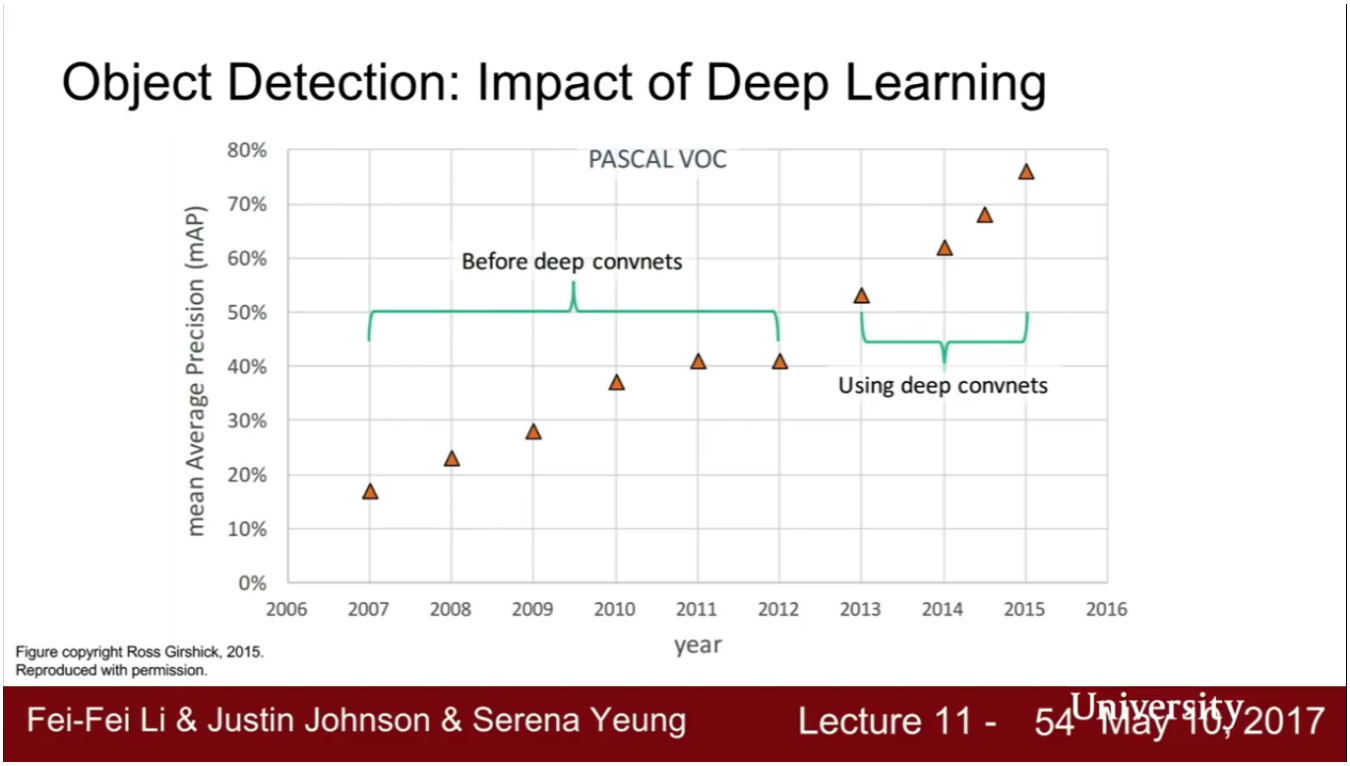


- Regression loss basically means L1,L2 loss etc where the values used are continuous in nature.
- Categorical losses are Cross-Entropy, Softmax, SVM margin loss, are used when we have a fixed number of categories.

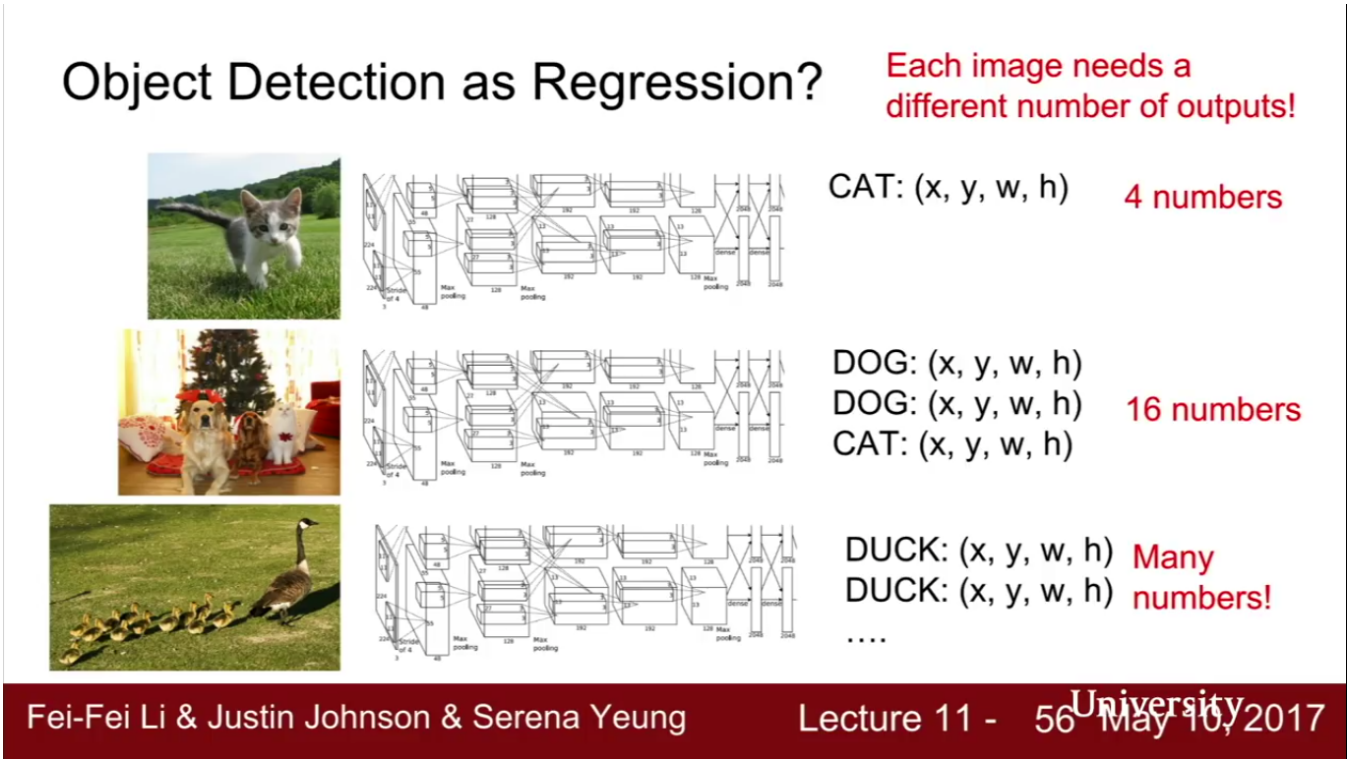
Object Detection



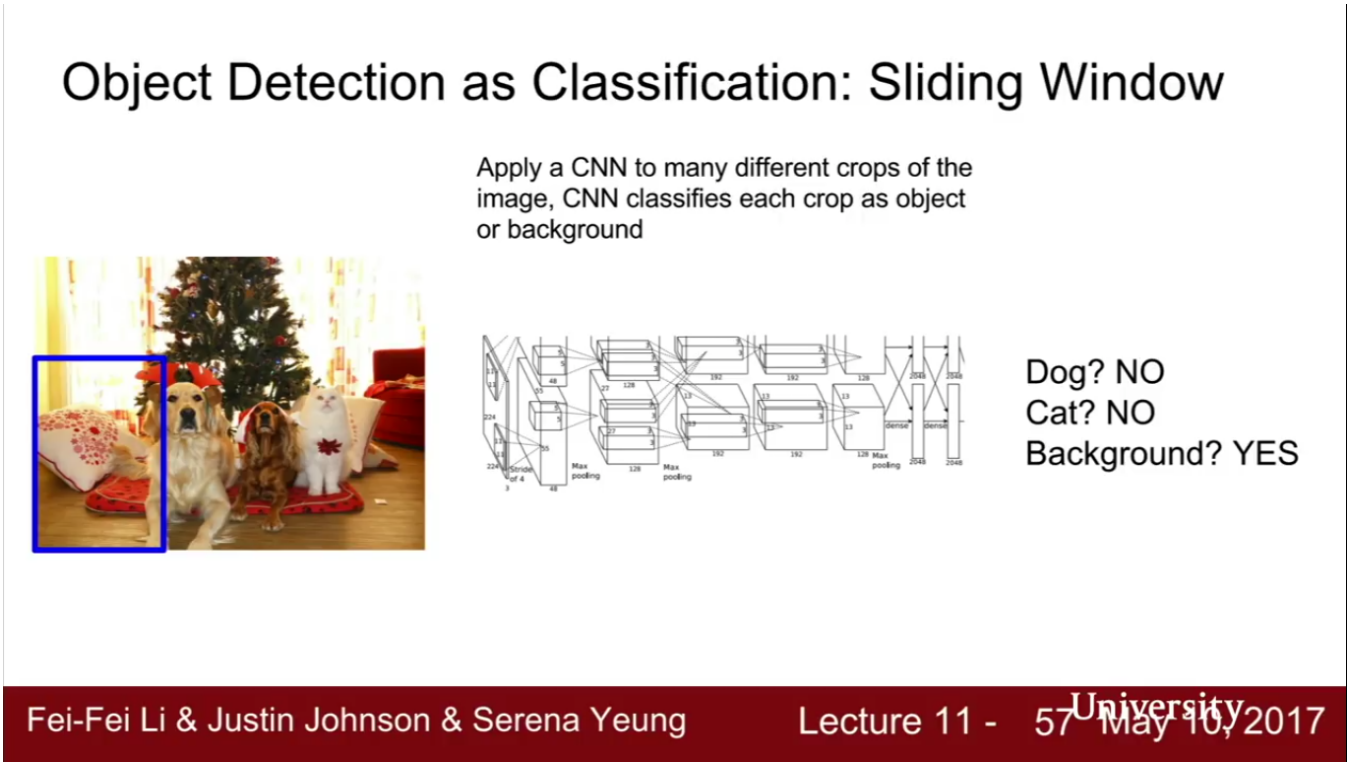
- Object Detection: We have a fixed number of categories, and when a image has any object from those categories, we are supposed to classify it and draw a boundig box around it.
- **Detection varies from Classification+Localization because in detection we will have varying number of detection in each image, i.e. you do not expect how many number of objects are there in the image.**

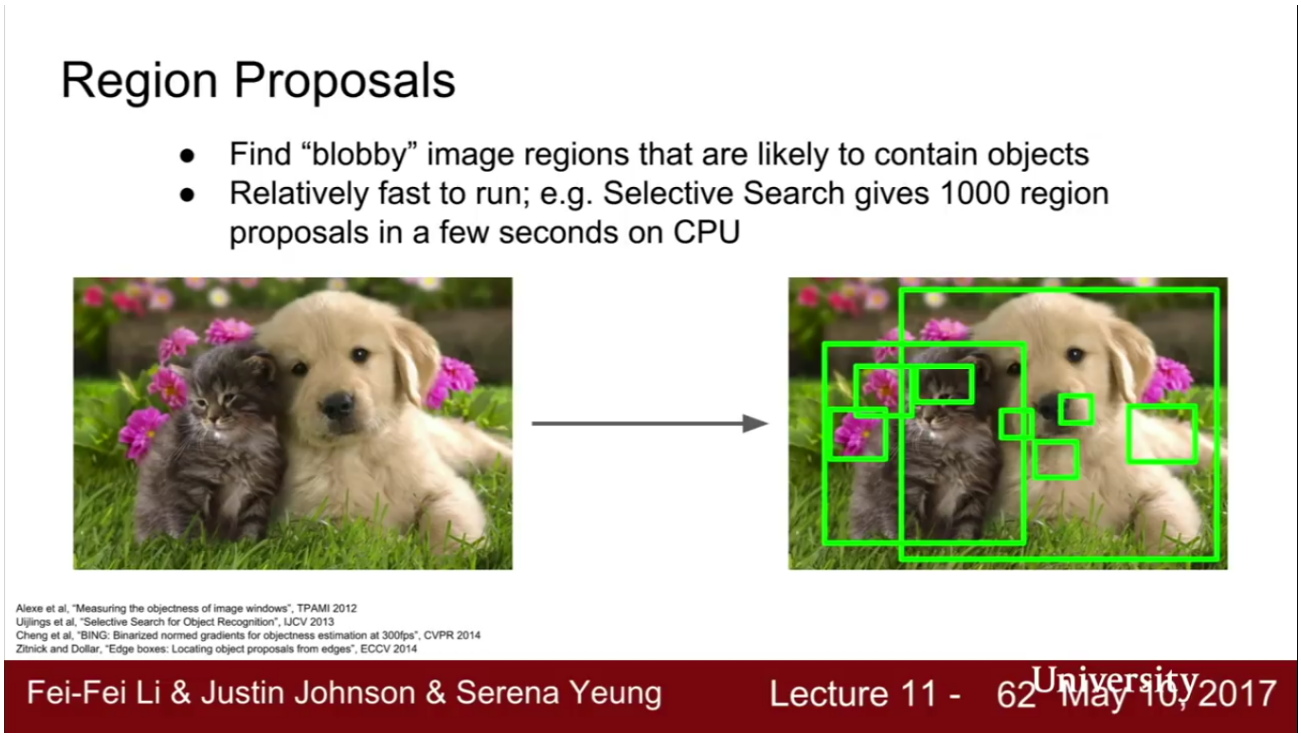
- 

- Before 2013(deep-learning) the best methods were having almost similar results.

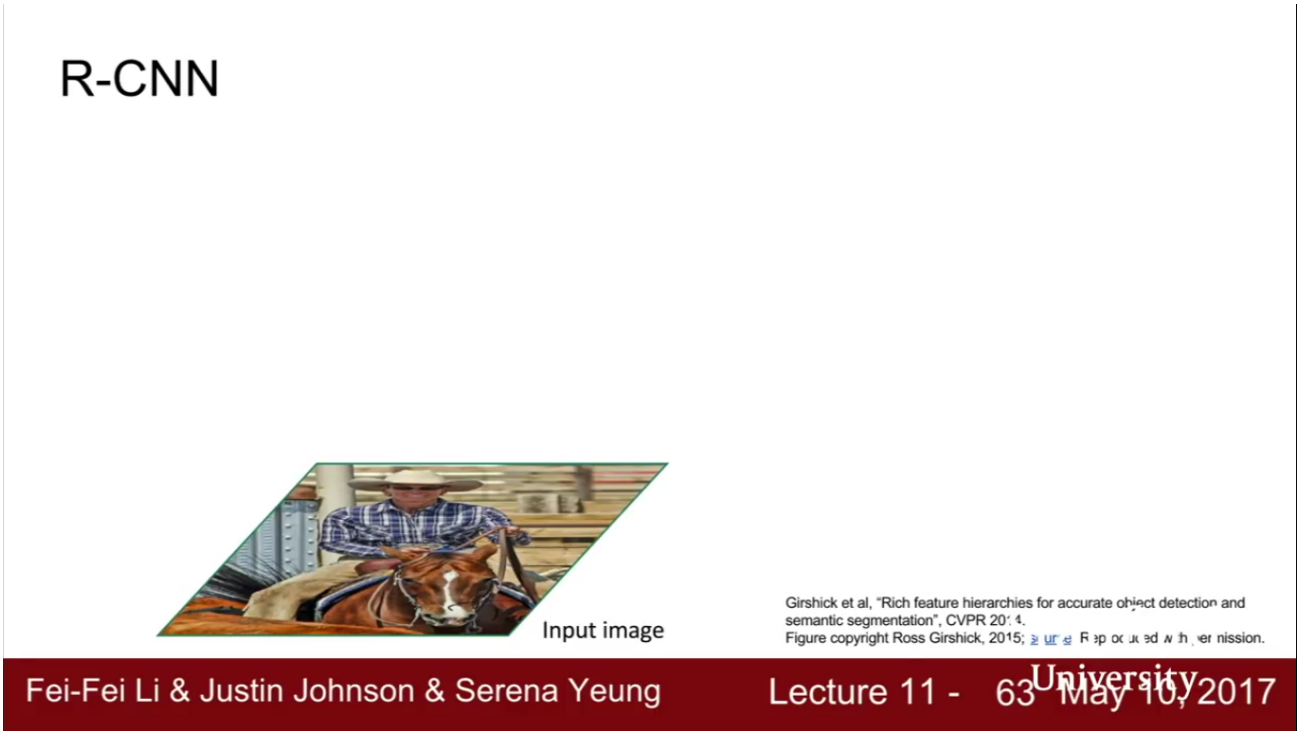
- 

- If we try to solve the Object Detection problem as a regression problem then we have the inherent issue that each image should have varying number of output values.



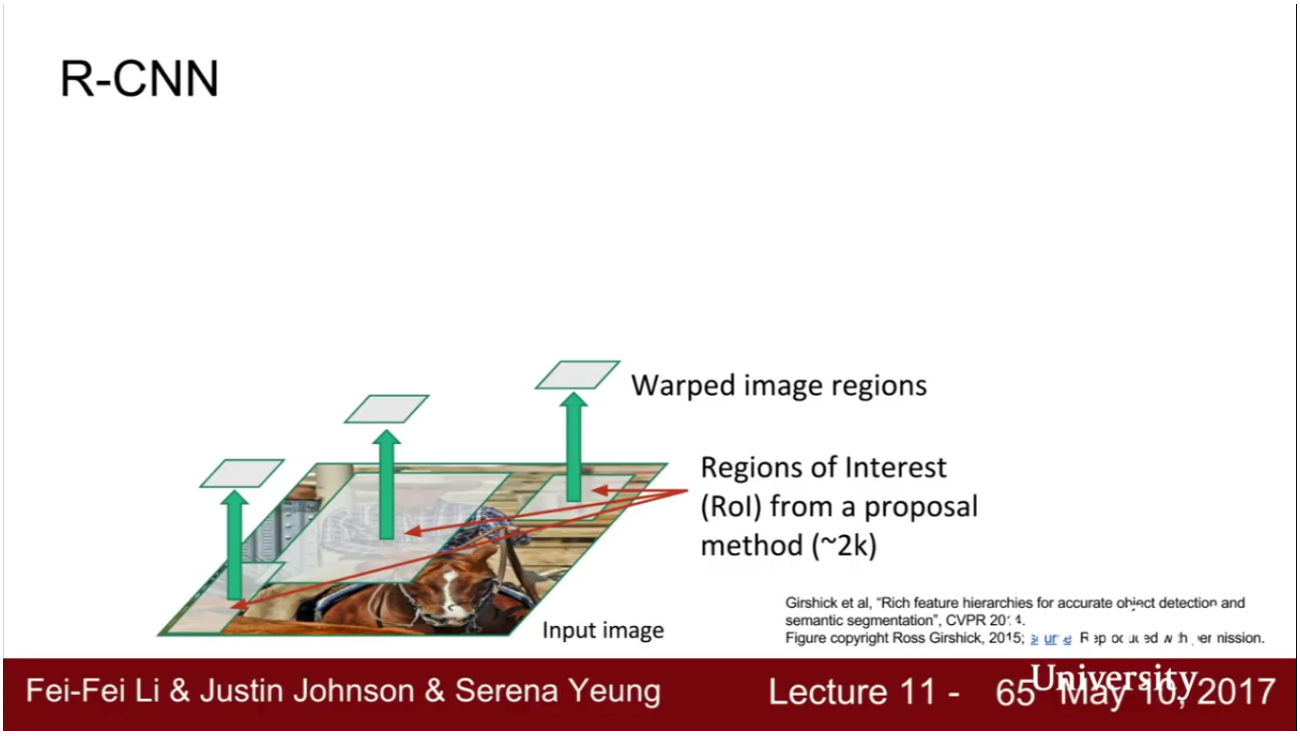
- In sliding window we try to classify the window slice only. Time consuming apporach and lot of spatial redundancy while sliding the window.
- Problem:
  - We need to choose sliding window size.
  - Too much of convolution operation!
  - Too time consuming!
  - Solution: **Region Proposals** to provide slides of images, reducing the unnecessary computations.
- Region Proposals:
  - 

- R-CNN

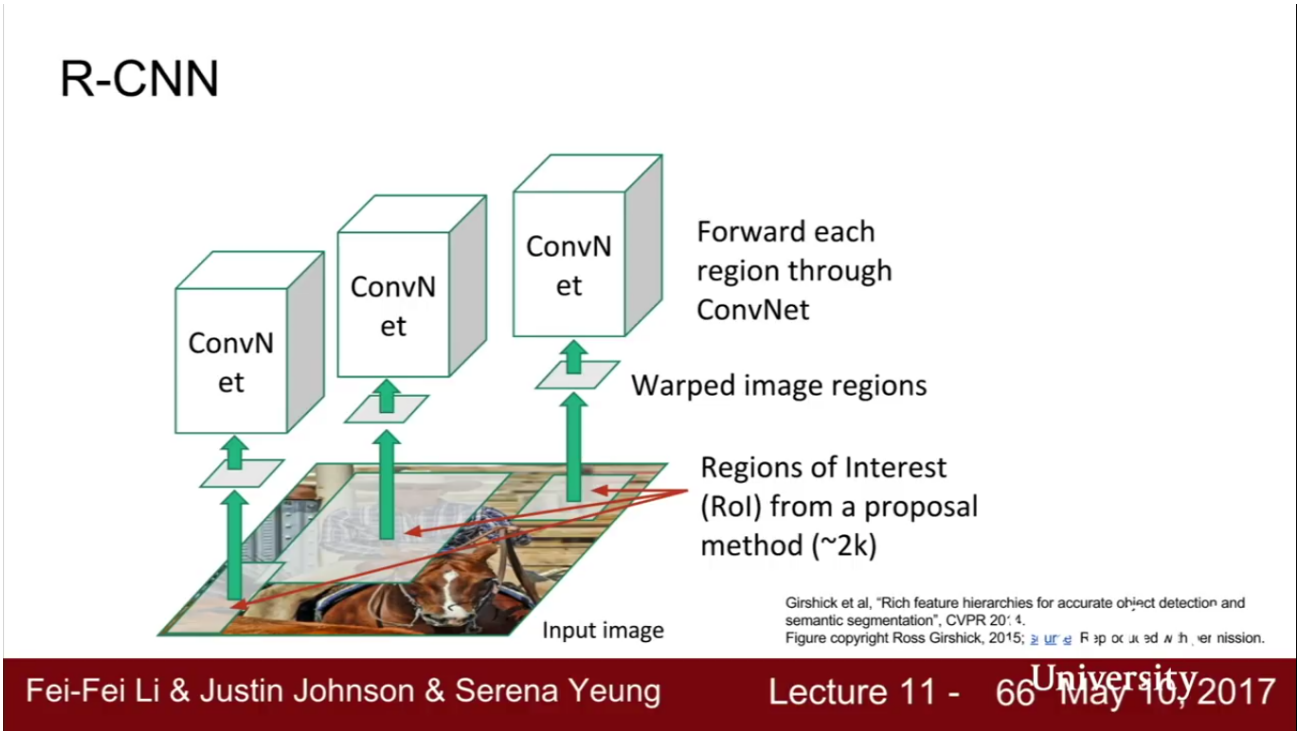
- 

- Given an image we generate Region Proposals also called as Region of Interest(ROIs).
- 

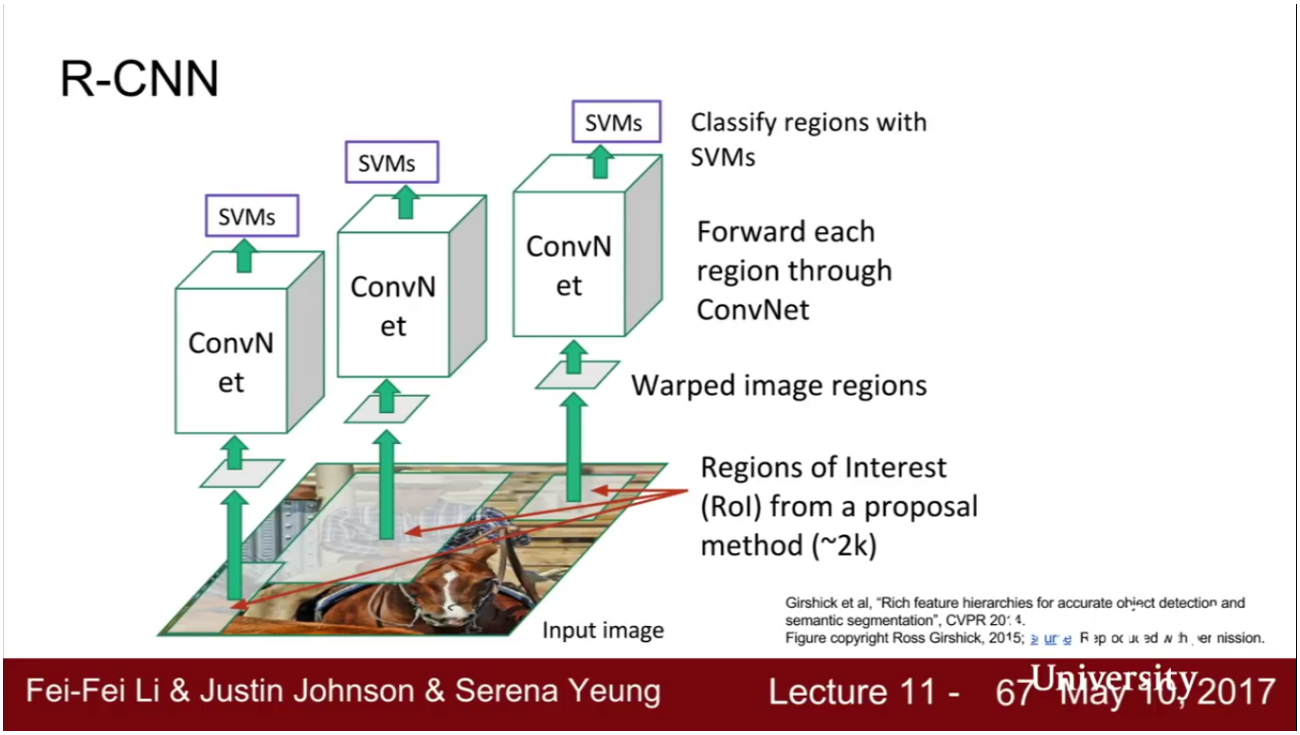
-



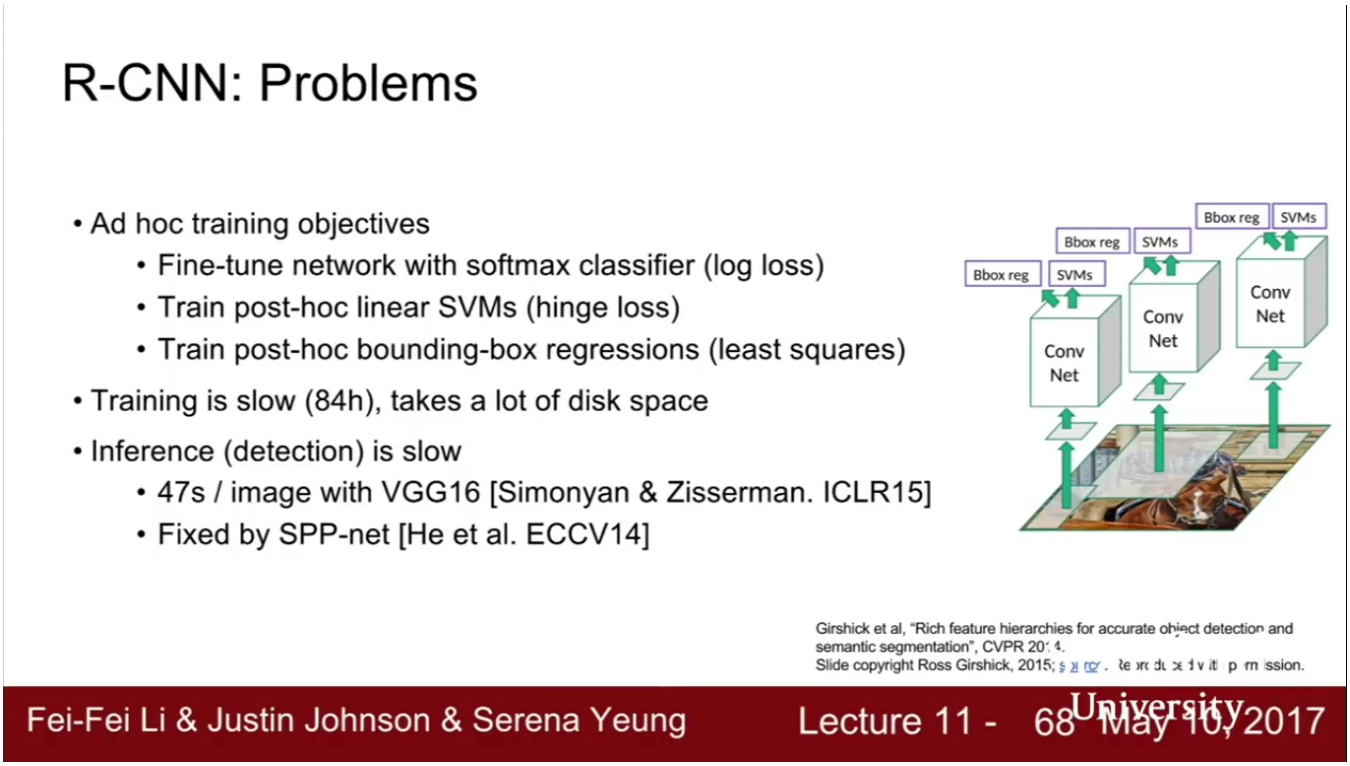
- Since the CNN take images of single size and the ROIs are of different sizes, we have to warp them to a common size which is required by the CNNs.



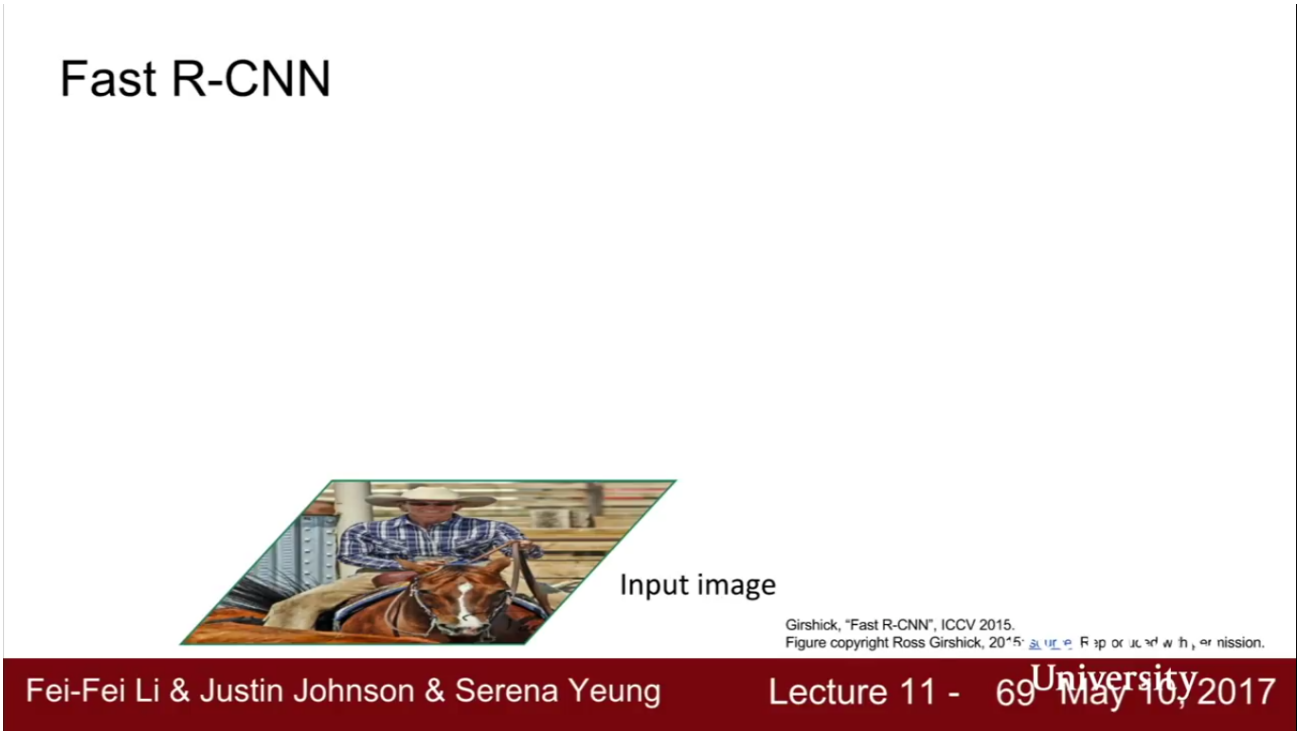
- Then we pass these warped image regions through the CNNs

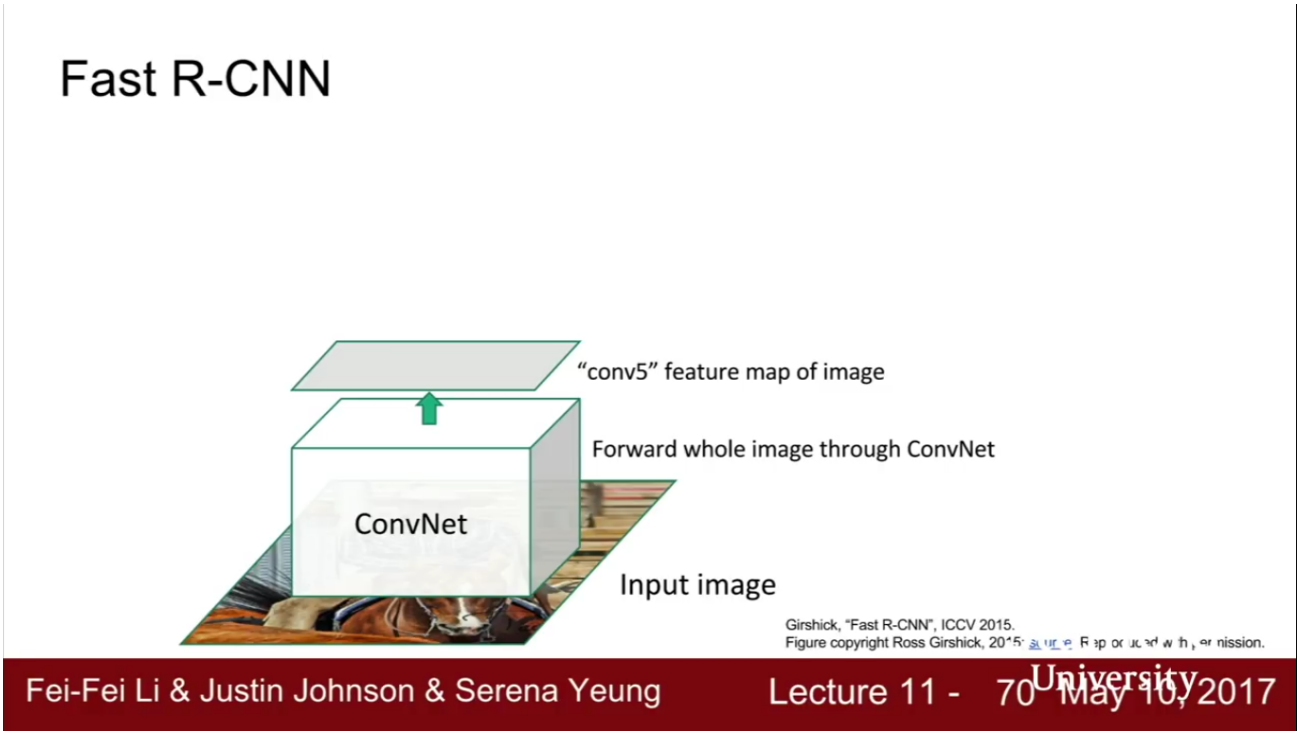


- The output of CNNs are passed to SVM for classification.
- Along with classification we also generate the bounding box coordinates. (missing in the slide)

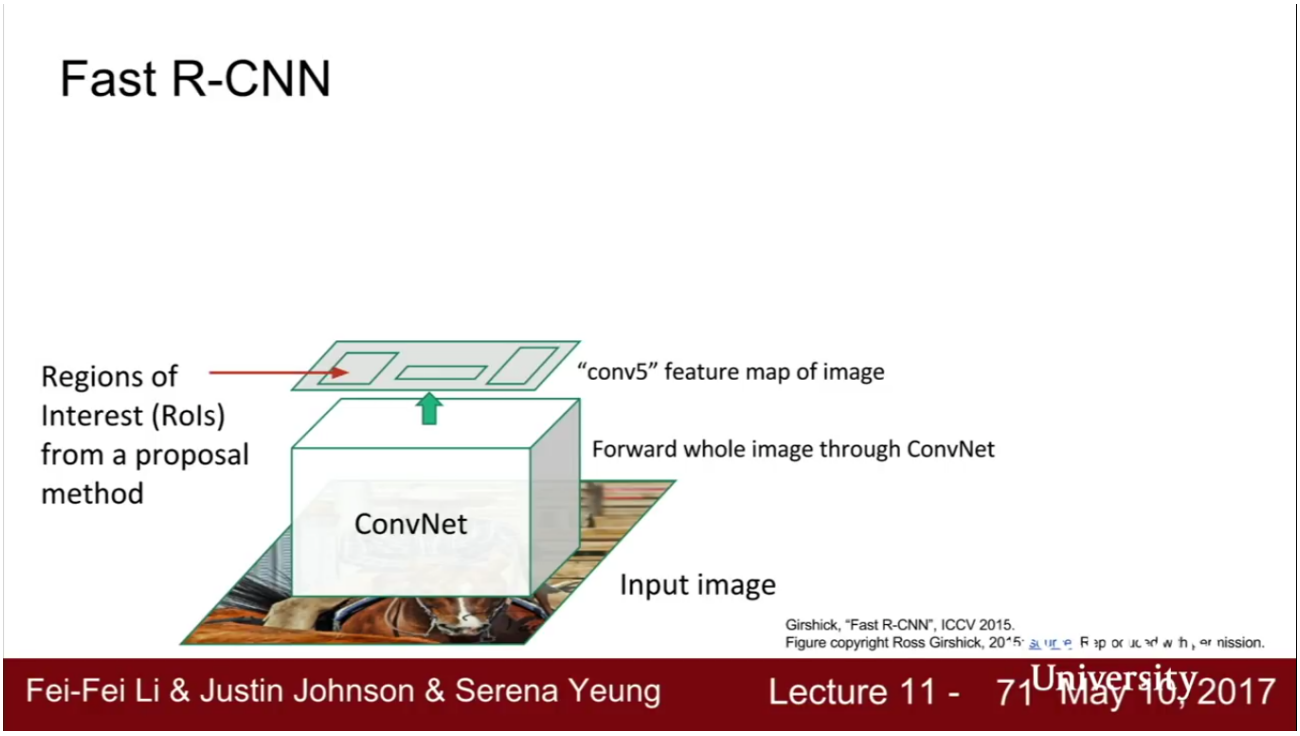


- Fast R-CNN

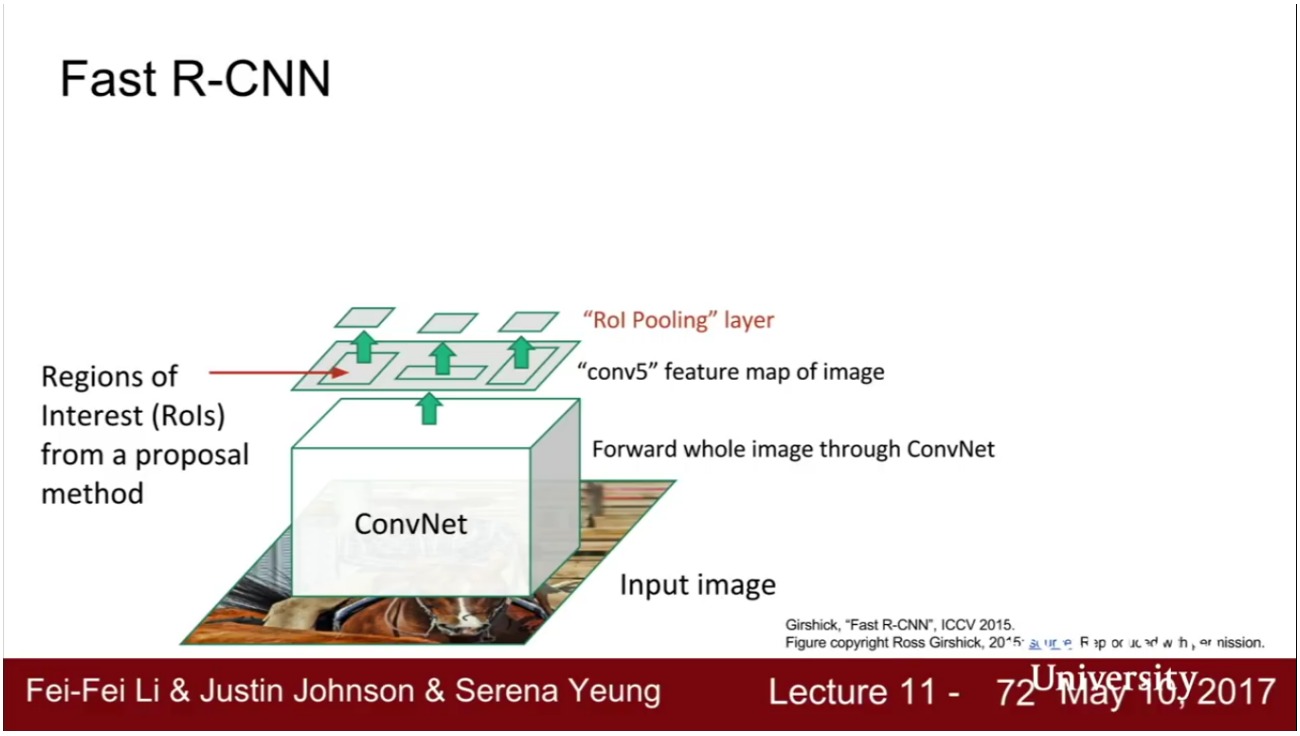
- 

- 

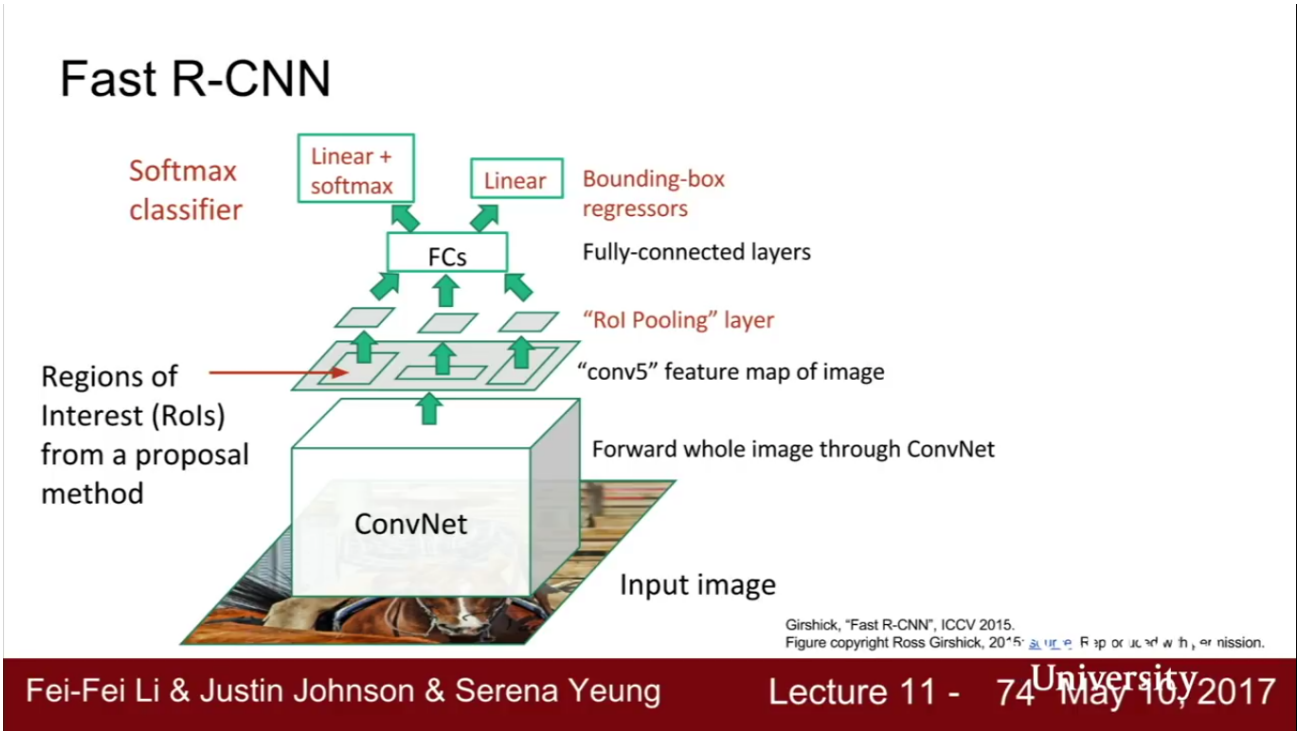
- In this technique instead of using Region Proposal generator to generate ROIs, we pass the entire image through a CNN and use Region Proposal to generate ROIs.

- 

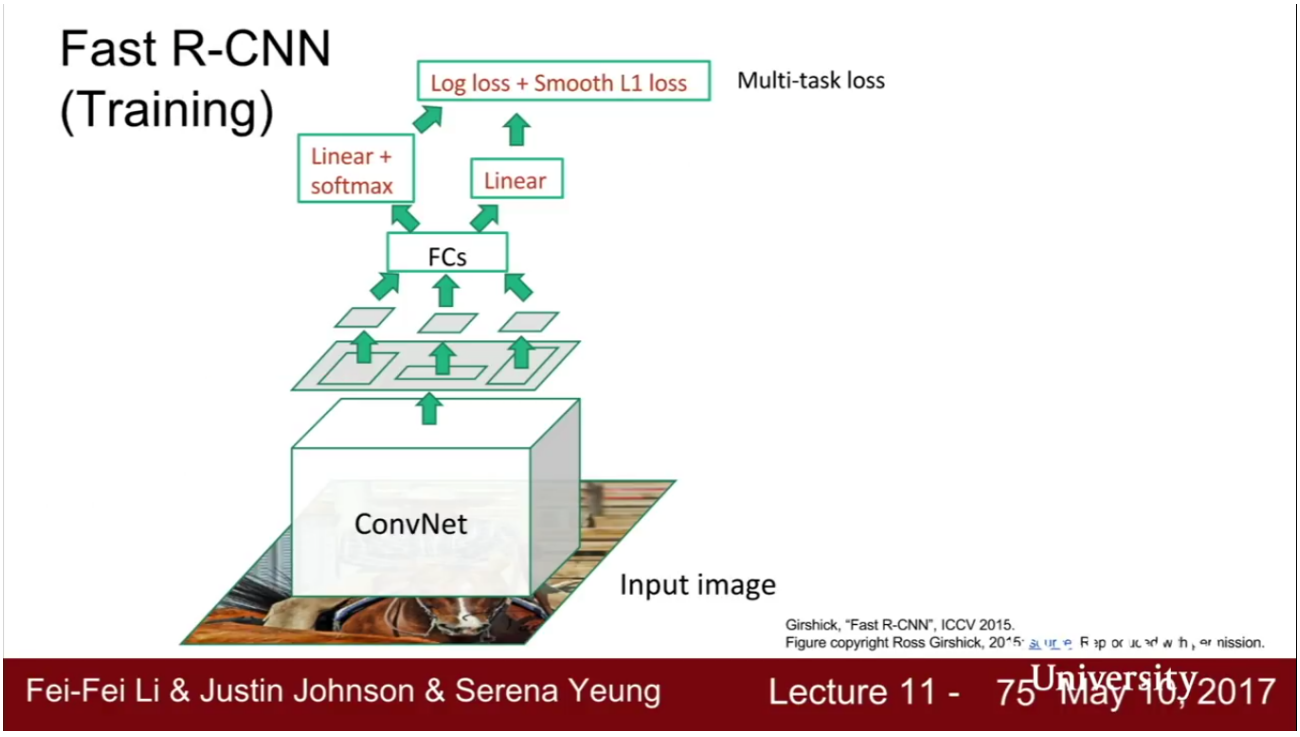
- Advantage is that we are reducing the use of Convolution operation.

- 

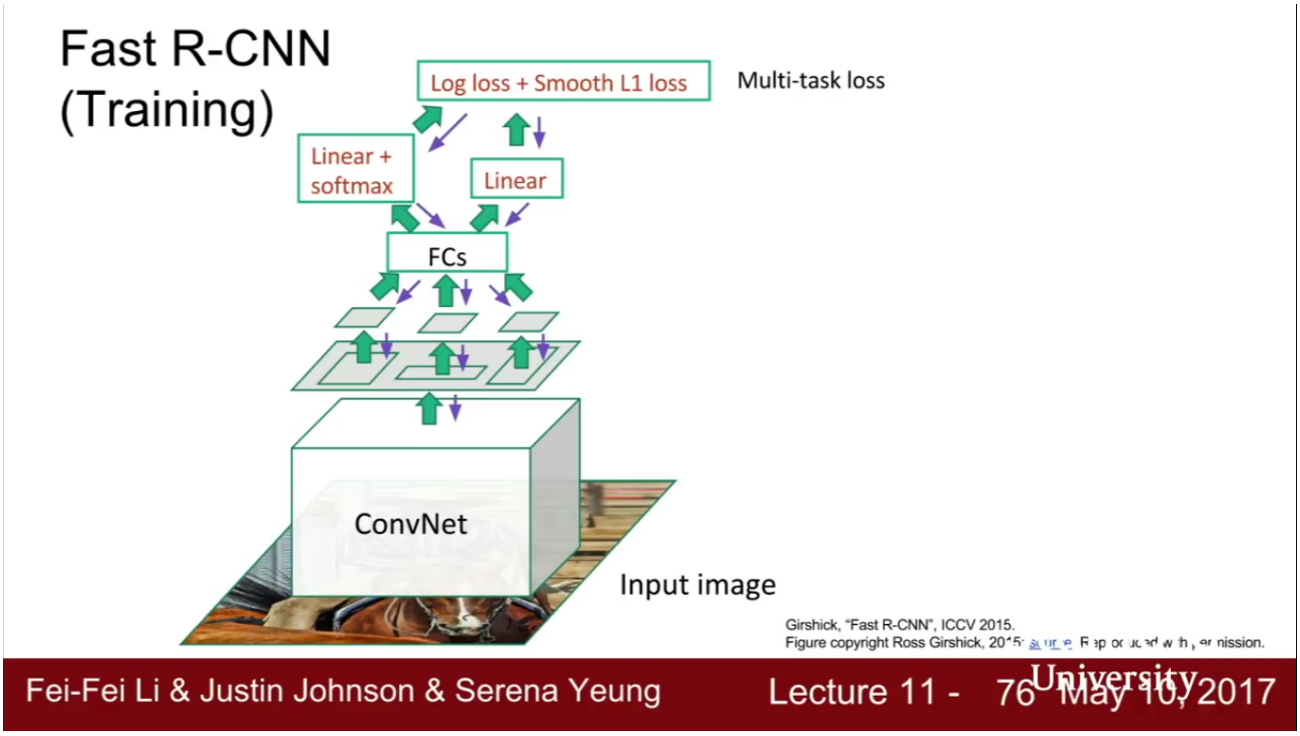
- Since the ROIs are of different sizes, we use ROI Pooling to make them of same size.

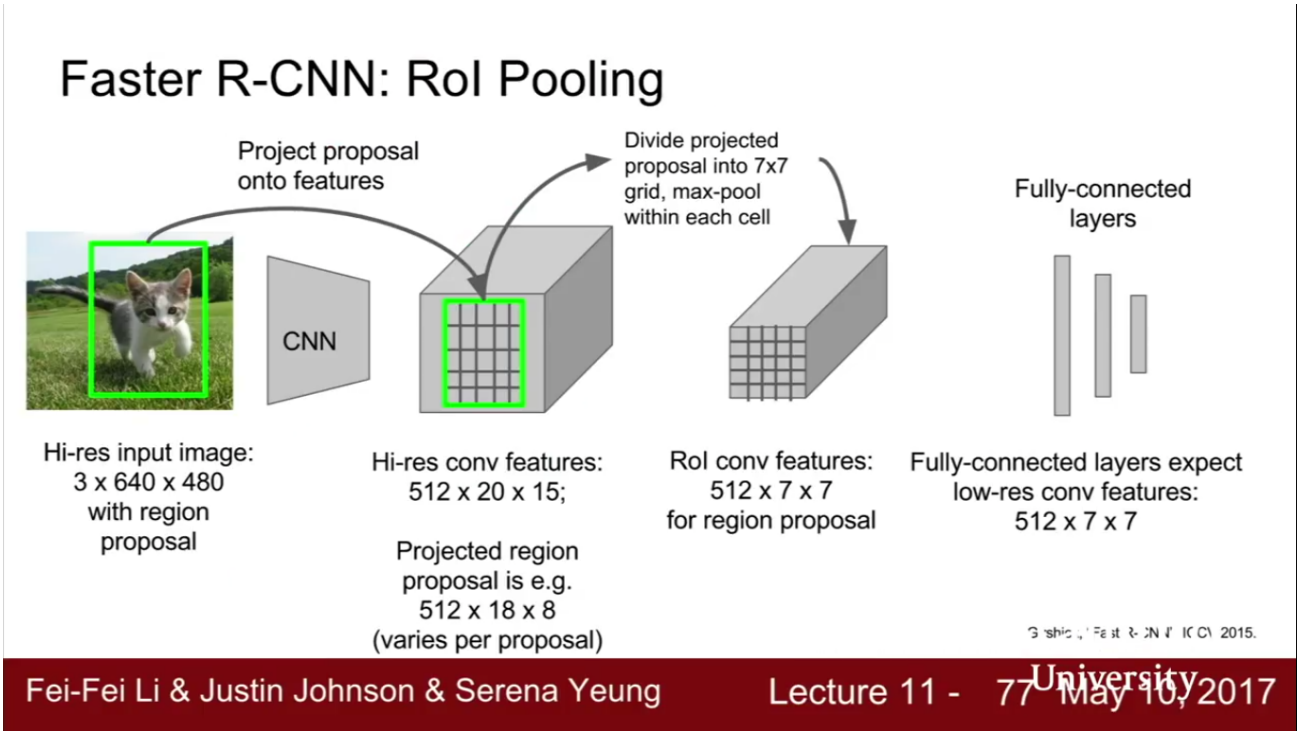
- 

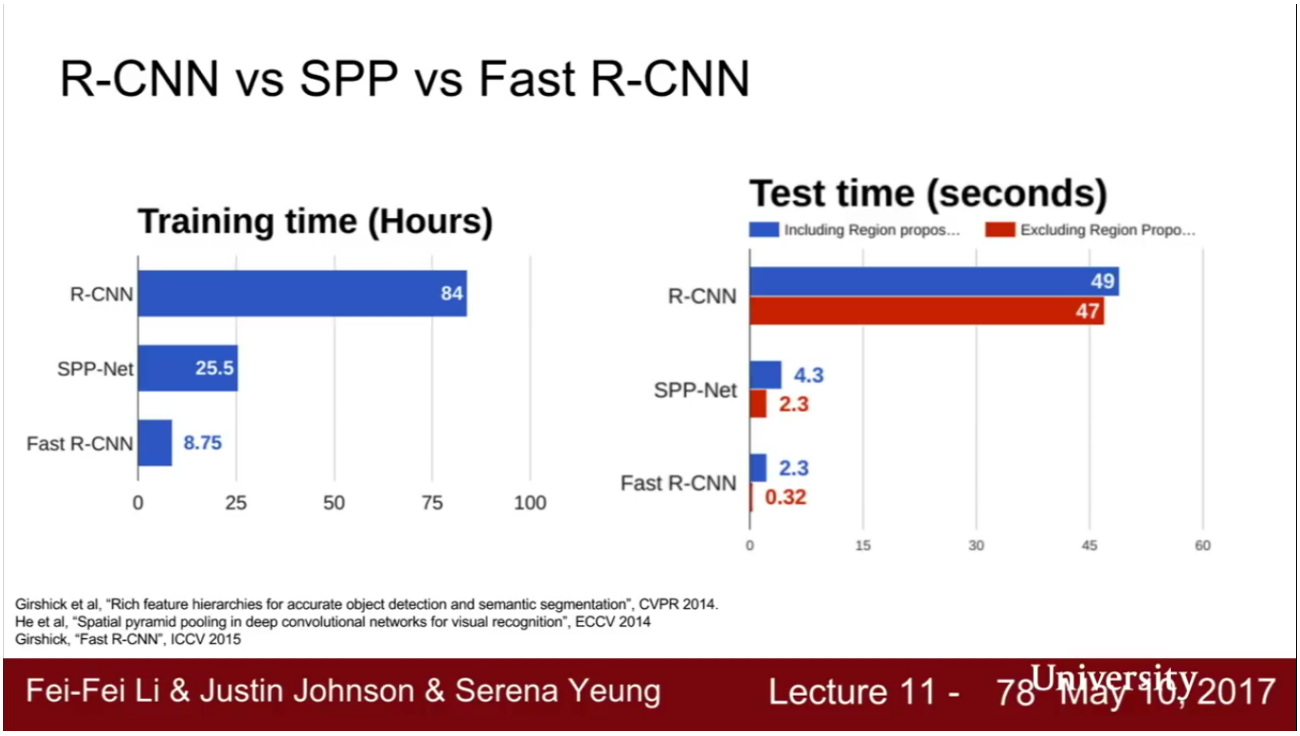
- We can use the FC layer output for classification and regression to generate Category label and Bounding box co-ordinates.

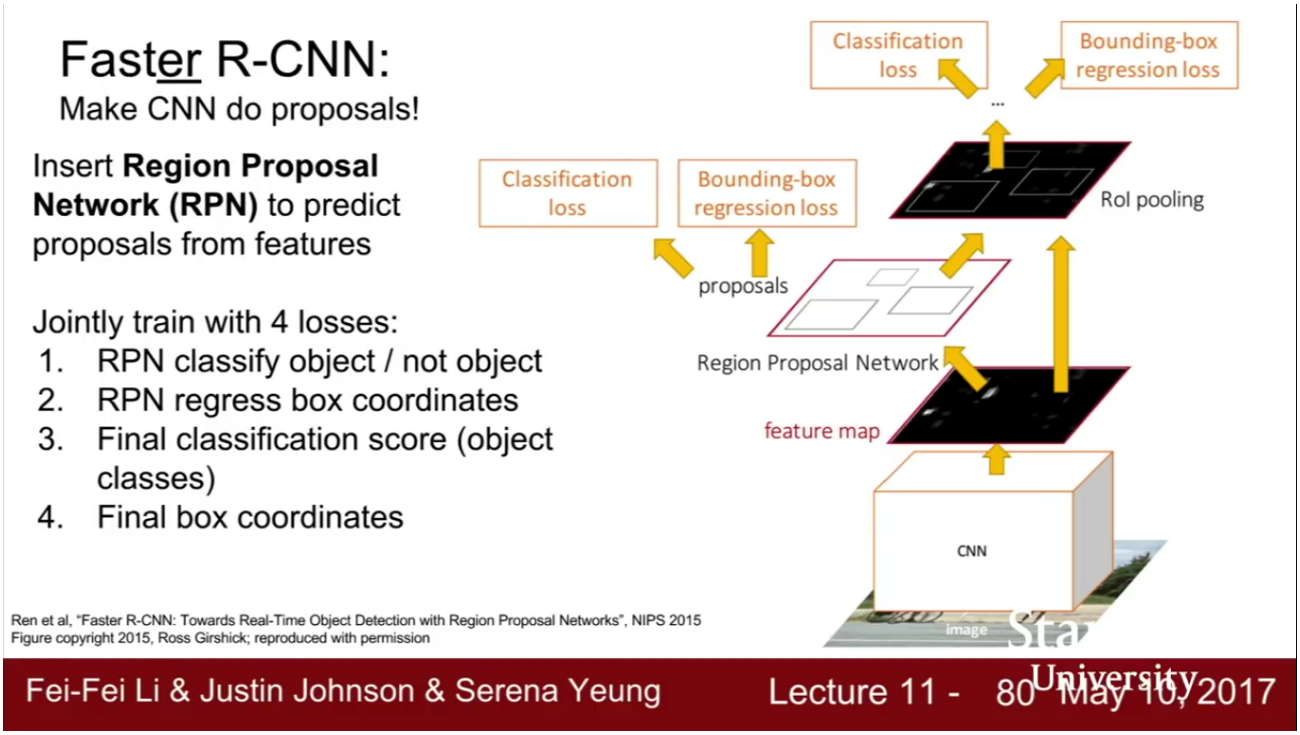
- 

- Since we have 2 different loss terms we combine them.

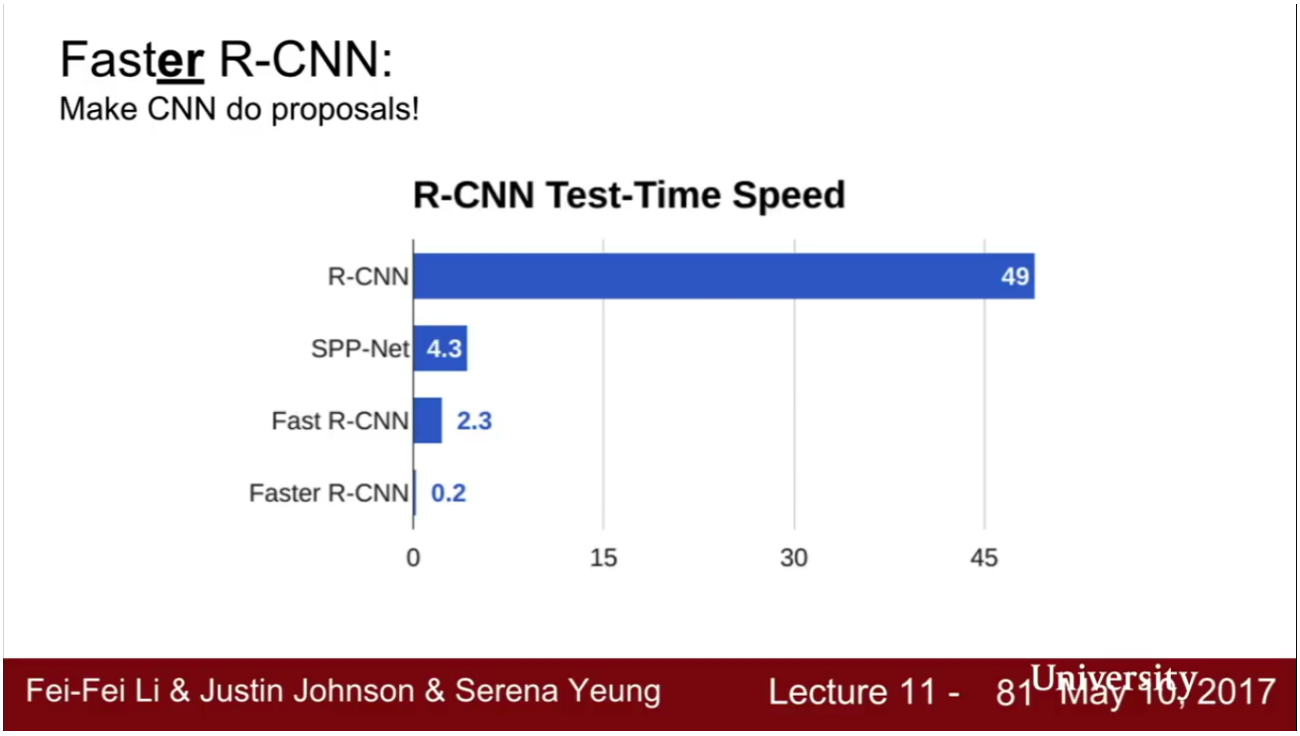
- 

- We can use this combined Multi-Task loss to backpropagate.
- 

- 

- Faster R-CNN:
  - 

    - Computing the Region Proposals using a fixed technique was the time consuming task in previous networks.
    - So in Faster R-CNN we make the network does it region proposals.
    - So here, we run the image through a CNN, and then use a higher feature map as input to the RPN to generate the ROIs.



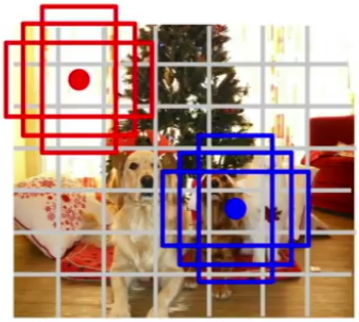
- R-CNN, Fast R-CNN, Faster R-CNN are **region-based methods**

- There is another set of approaches which are in a sense of all feed forward in a single pass.

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

Object Detection: Lots of variables ...

Base Network
VGG16
ResNet-101
Inception V2
Inception V3
Inception ResNet
MobileNet

Object Detection architecture
Faster R-CNN
R-FCN
SSD

Image Size
 # Region Proposals
...

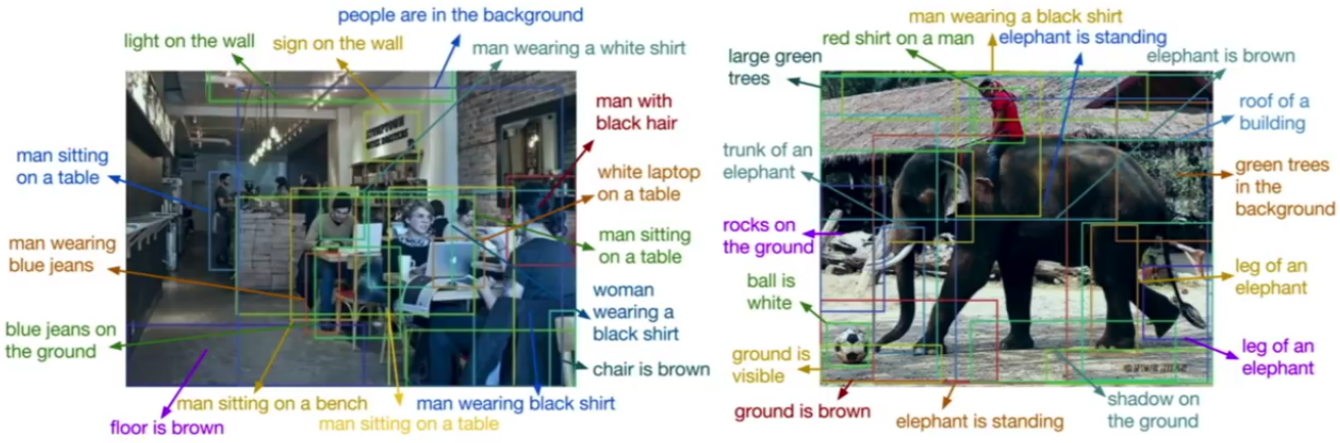
Takeaways
Faster R-CNN is slower but more accurate

SSD is much faster but not as accurate

Huang et al, "Speed/accuracy trade-offs for modern convolutional object detectors", CVPR 2017

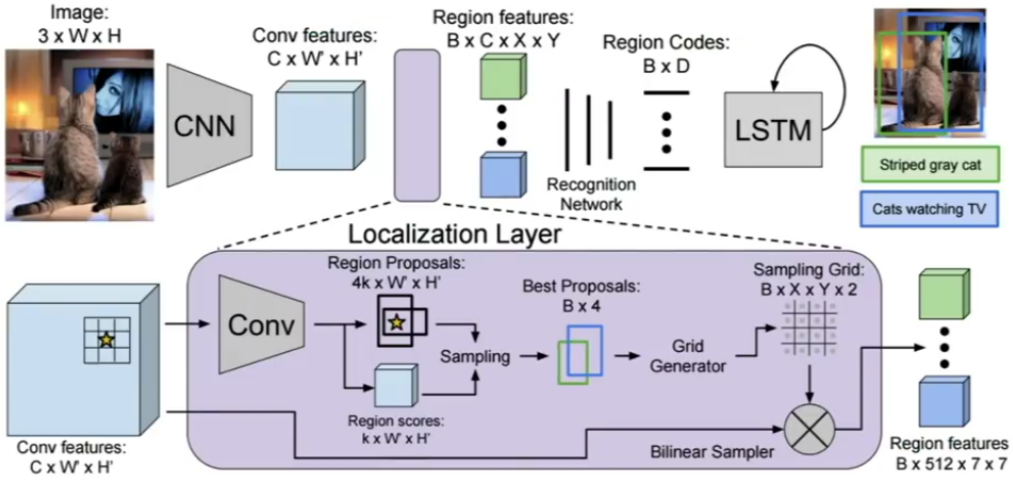
R-FCN: Dai et al, "R-FCN: Object Detection via Region-based Fully Convolutional Networks", NIPS 2016
Inception-V2: Ioffe and Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", ICML 2015
Inception V3: Szegedy et al, "Rethinking the Inception Architecture for Computer Vision", arXiv 2016
Inception ResNet: Szegedy et al, "Inception-V4, Inception-ResNet and the Impact of Residual Connections on Learning", arXiv 2016
MobileNet: Howard et al, "Efficient Convolutional Neural Networks for Mobile Vision Applications", arXiv 2017

Aside: Object Detection + Captioning = Dense Captioning



Johnson, Karpathy, and Fei-Fei, "DenseCap: Fully Convolutional Localization Networks for Dense Captioning", CVPR 2016
Figure copyright IEEE, 2016. Reproduced for educational purposes.

Aside: Object Detection + Captioning = Dense Captioning



Johnson, Karpathy, and Fei-Fei, "DenseCap: Fully Convolutional Localization Networks for Dense Captioning", CVPR 2016
Figure copyright IEEE, 2016. Reproduced for educational purposes.

Instance Segmentation

Instance Segmentation

GRASS, CAT, TREE, SKY

No objects, just pixels

CAT

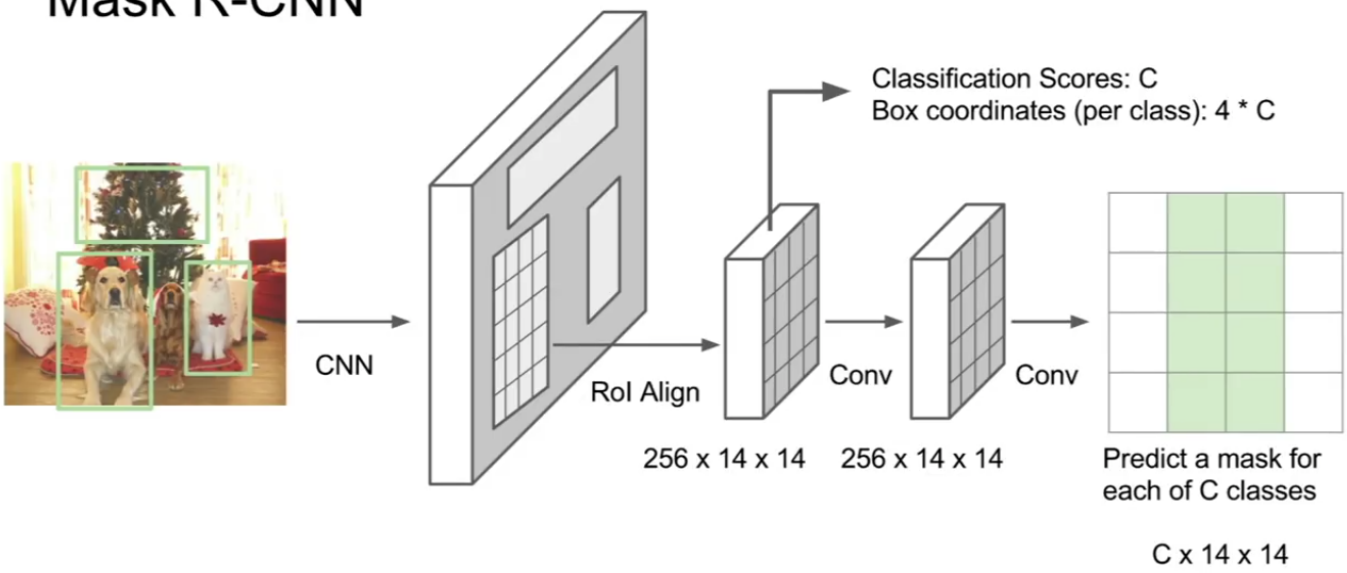
Single Object

DOG, DOG, CAT

DOG, DOG, CAT

Multiple Object

Mask R-CNN



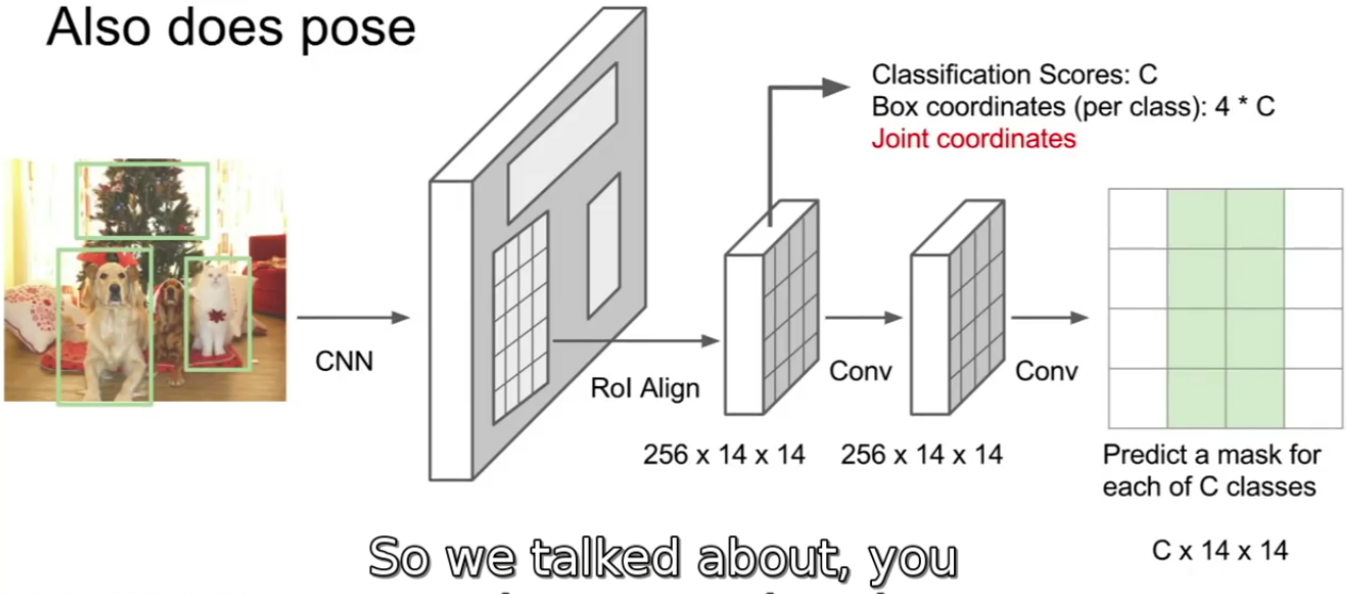
He et al, "Mask R-CNN", arXiv 2017

Subtitle track: Disable

Mask R-CNN: Very Good Results!

He et al, "Mask R-CNN", arXiv 2017
Figures copyright Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, 2017.
Reproduced with permission.

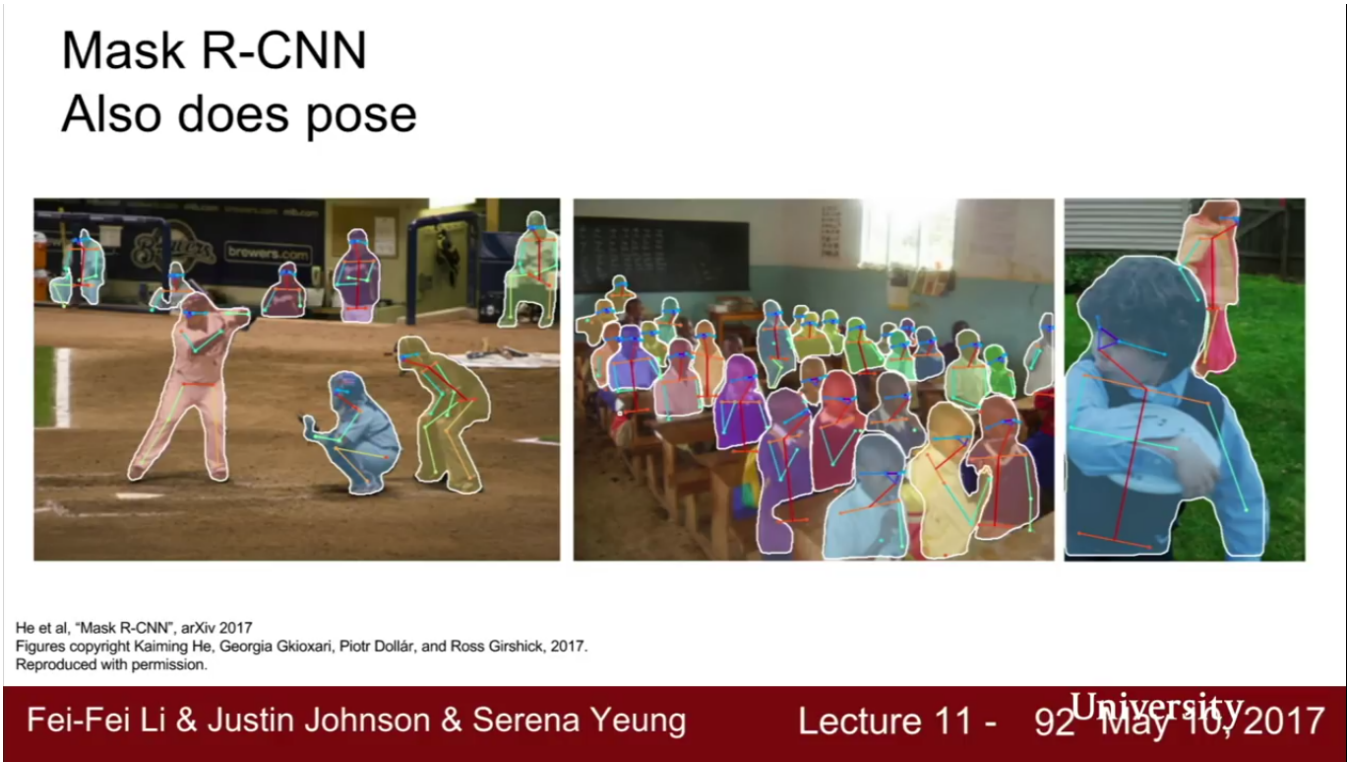
Mask R-CNN
Also does pose



He et al, "Mask R-CNN", arXiv 2017

So we talked about, you can do pose estimation

- To do pose detection along with classification and box coordinates we have to generate joint coordinates.

- 

- Mask R-CNN with pose runs at 5 fps on GPUs.