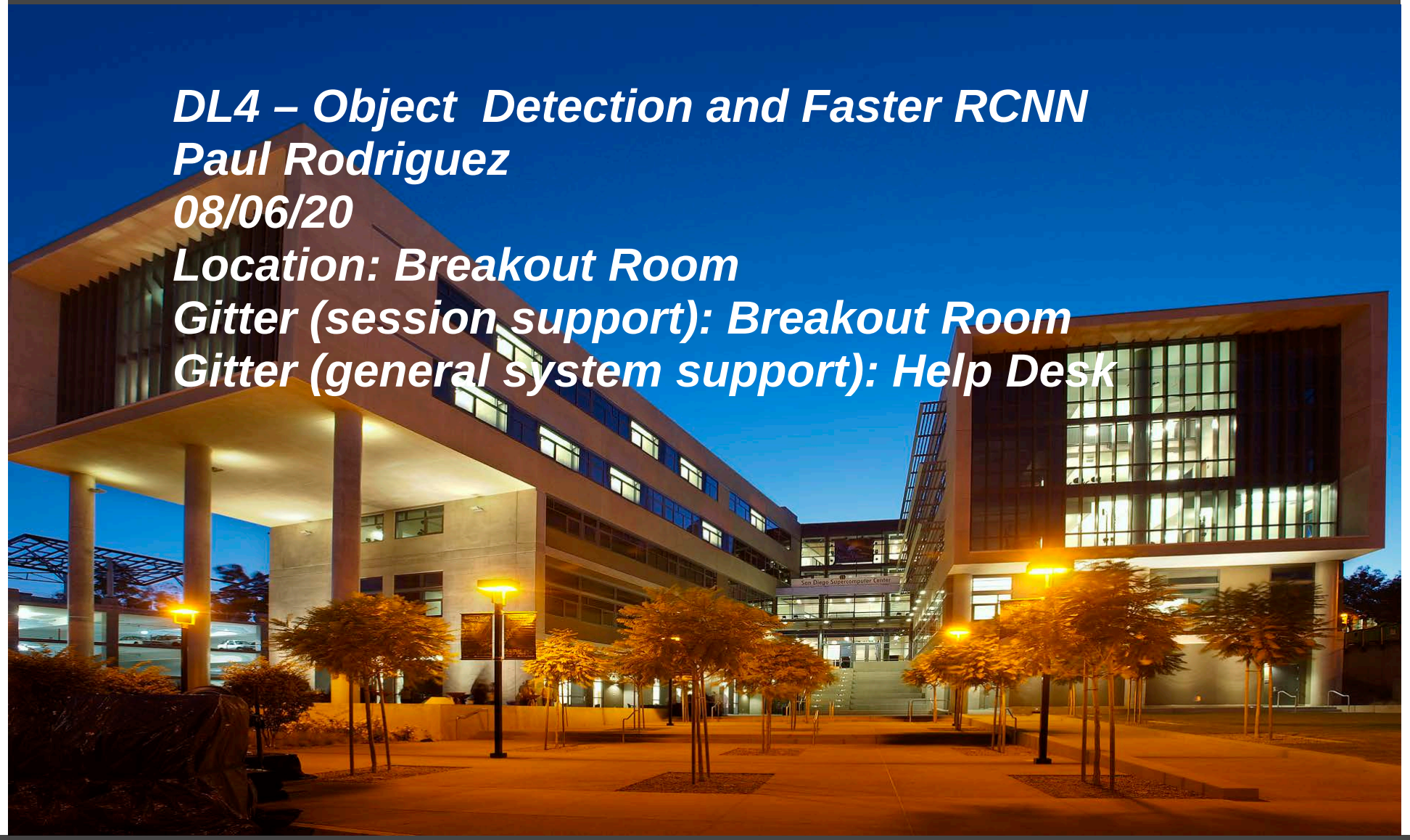
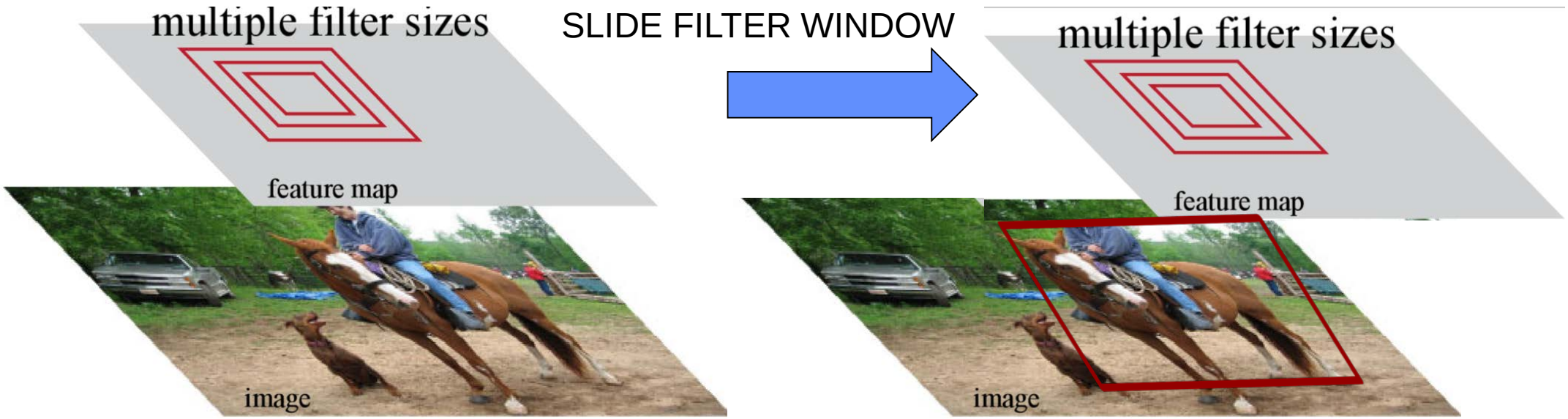
SDSC Summer Institute 2020



DL4 – Object Detection and Faster RCNN
Paul Rodriguez
08/06/20
Location: Breakout Room
Gitter (session support): Breakout Room
Gitter (general system support): Help Desk

CNNs can classifying images - but how do you locate object instances?

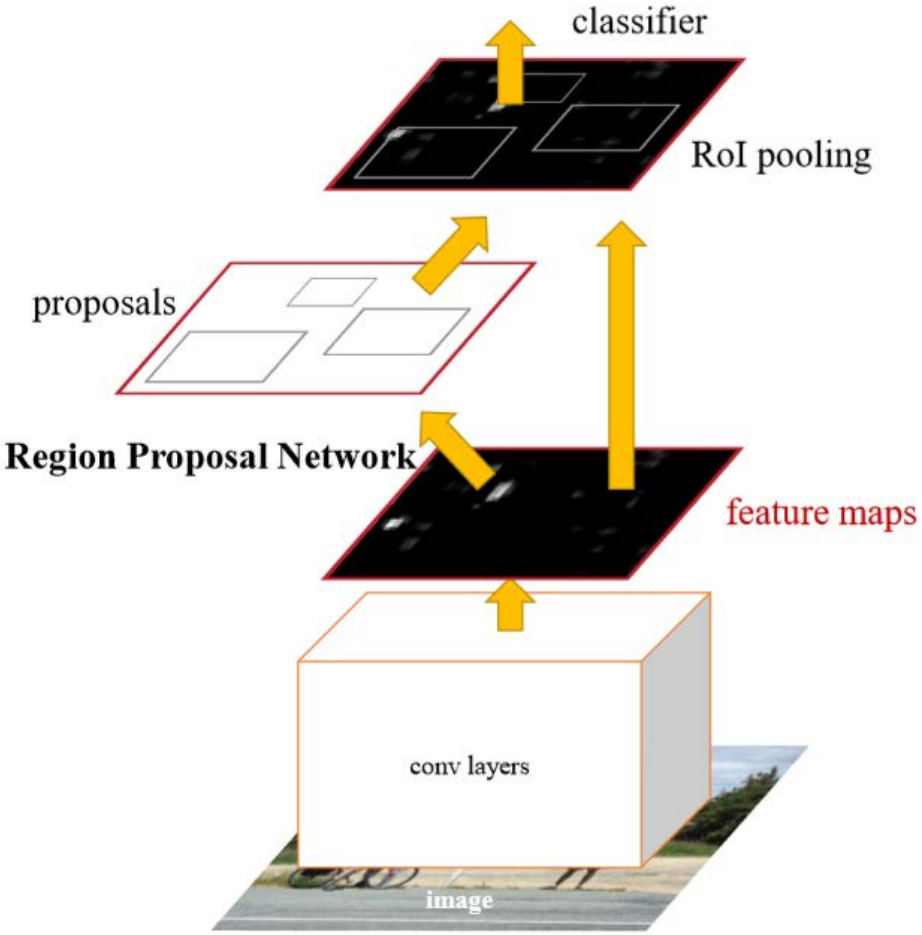
Spatial pyramids and sliding windows helps locate objects of different size



Or, use CNN to classify *AND* detect: Faster RCNN

(Ren,He,Girschick,Sun)

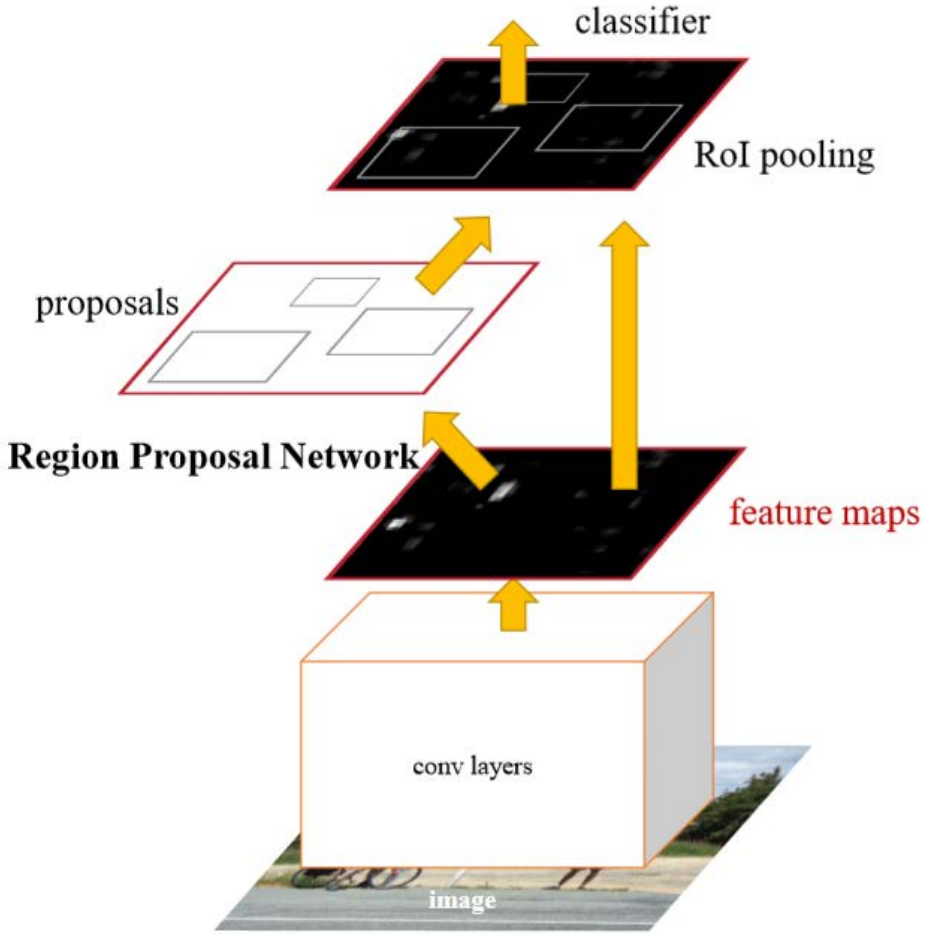
A side path to select Regions



Or, use CNN to classify and detect: Faster RCNN

(Ren,He,Girschick,Sun)

1. *Start with 2000 sampled regions*

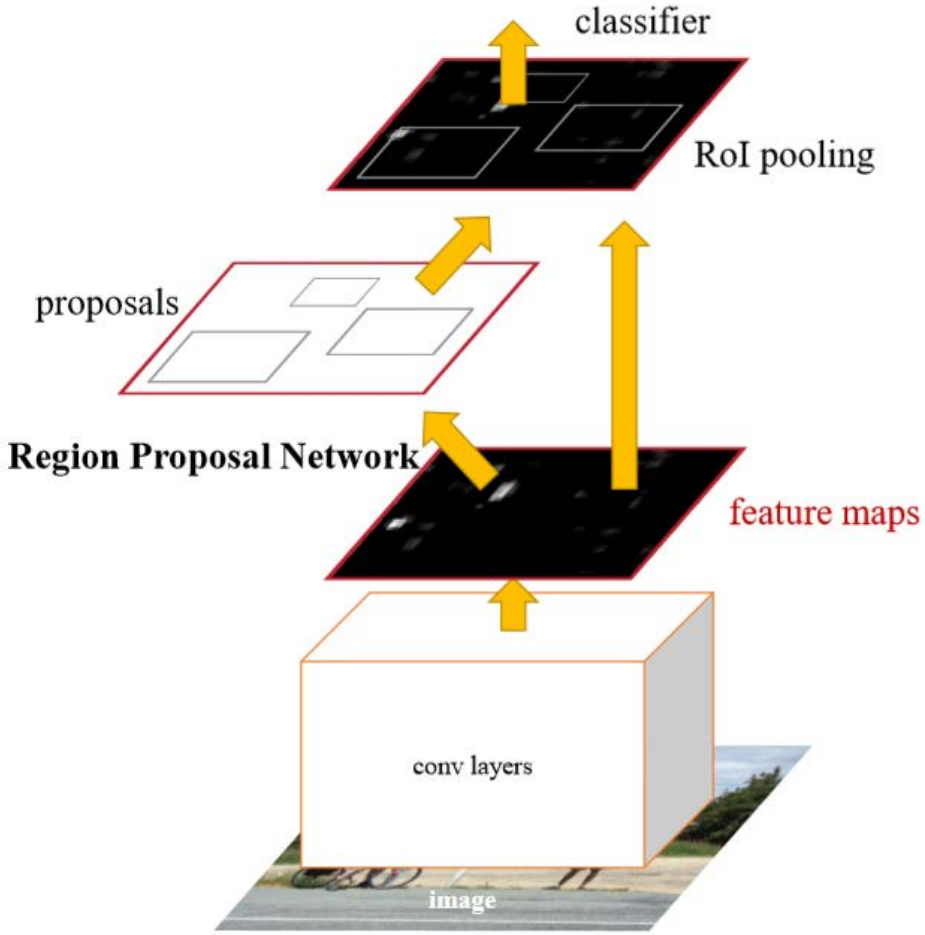


Or, use CNN to classify and detect: Faster RCNN

(Ren,He,Girschick,Sun)

1. *Start with 2000 sampled regions*

2. *Extract feature map values that correspond to regions. Use interpolation and max pooling to downsize into a fixed size*

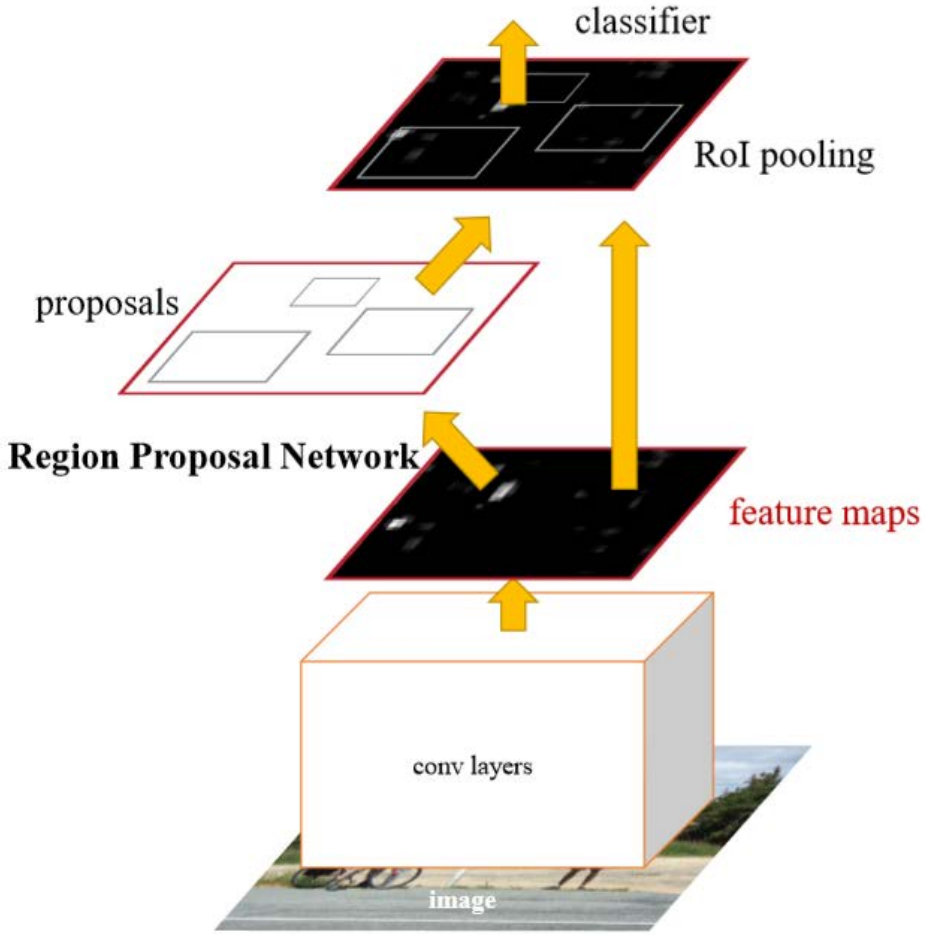


Or, use CNN to classify and detect: Faster RCNN

(Ren,He,Girschick,Sun)

1. *Start with 2000 sampled regions*

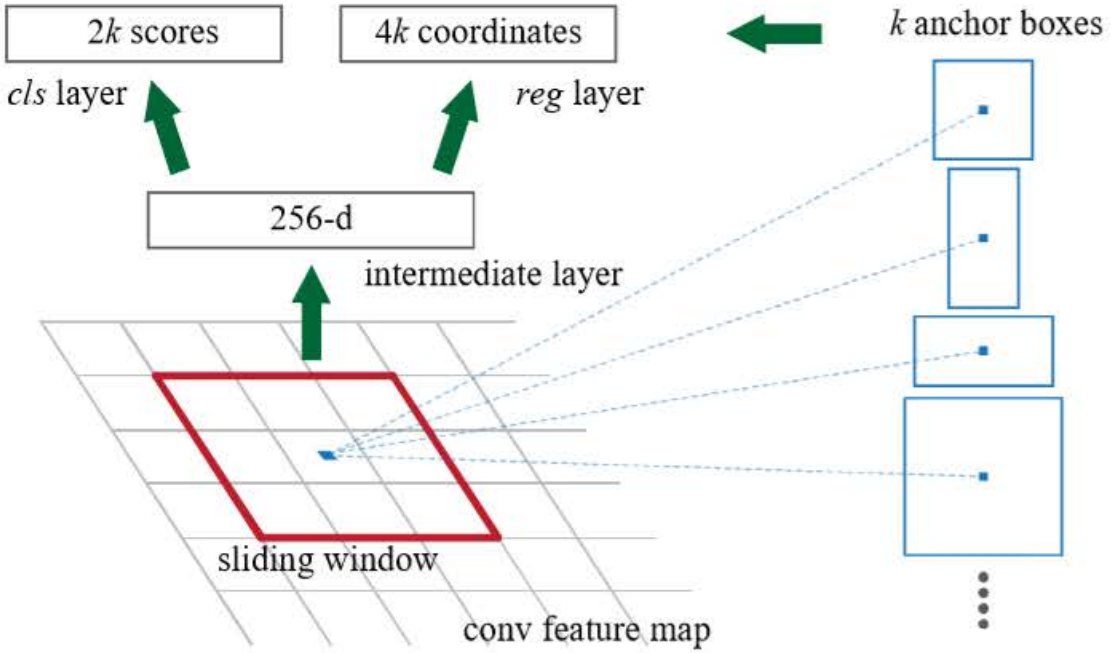
2. *Extract feature map values that correspond to regions. Use interpolation and max pooling to downsize into a fixed size*

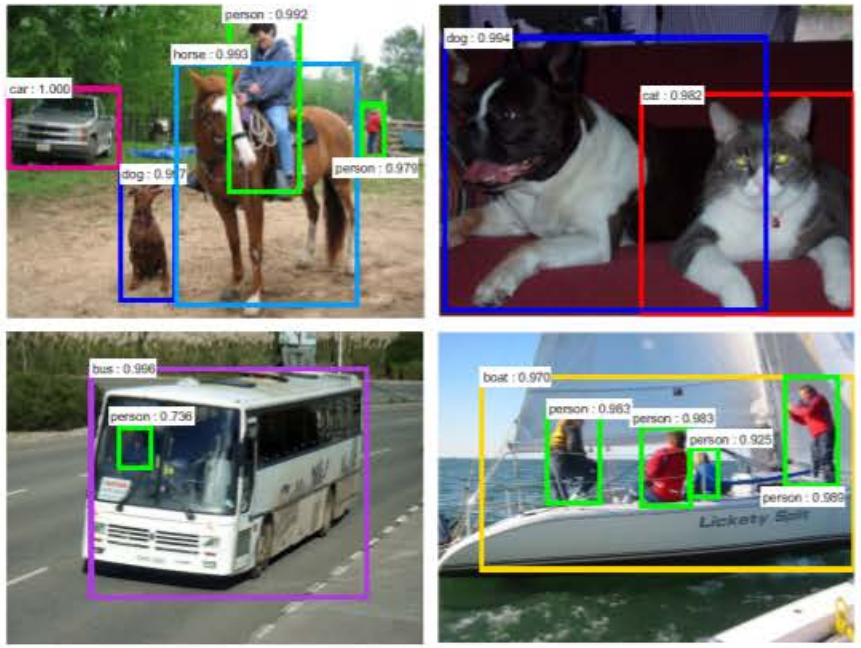


3. *Predict object classes (in CNN top layers) and bounding box coordinates (by regression)*

Region Box size ranges

Output object class prediction from CNN and bounding box information (box center, height, width) from regression model

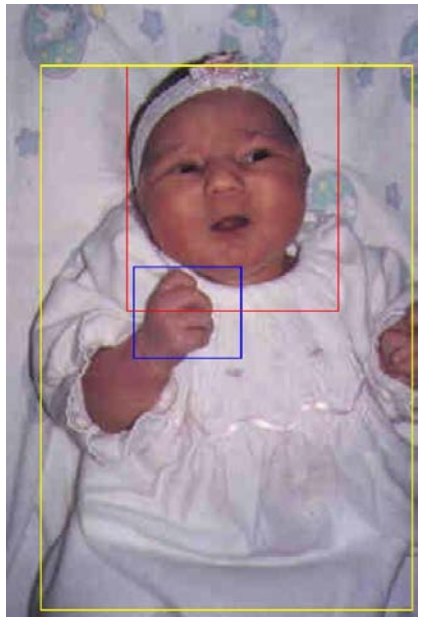


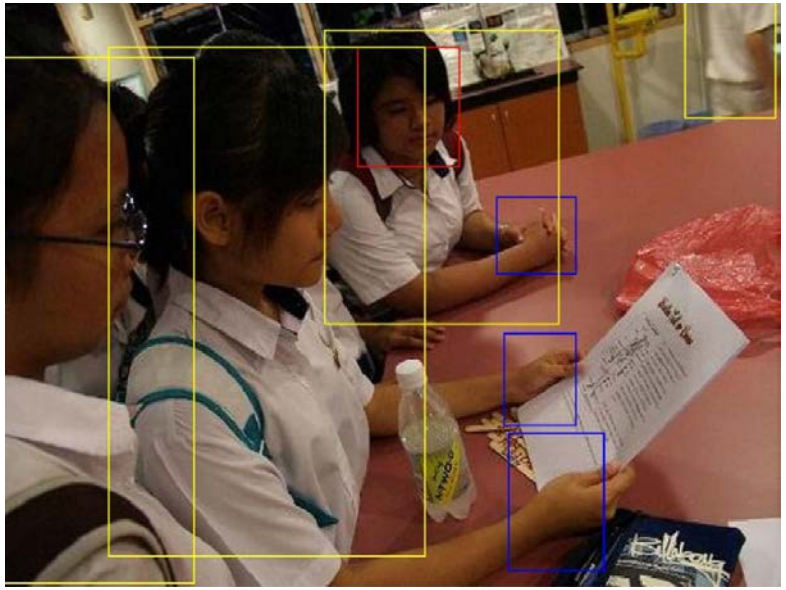


A faster RCNN test with some ‘hand’ labelled data (Mittal etal.)

Person boxes (using “YOLO” CNN, from Redom)

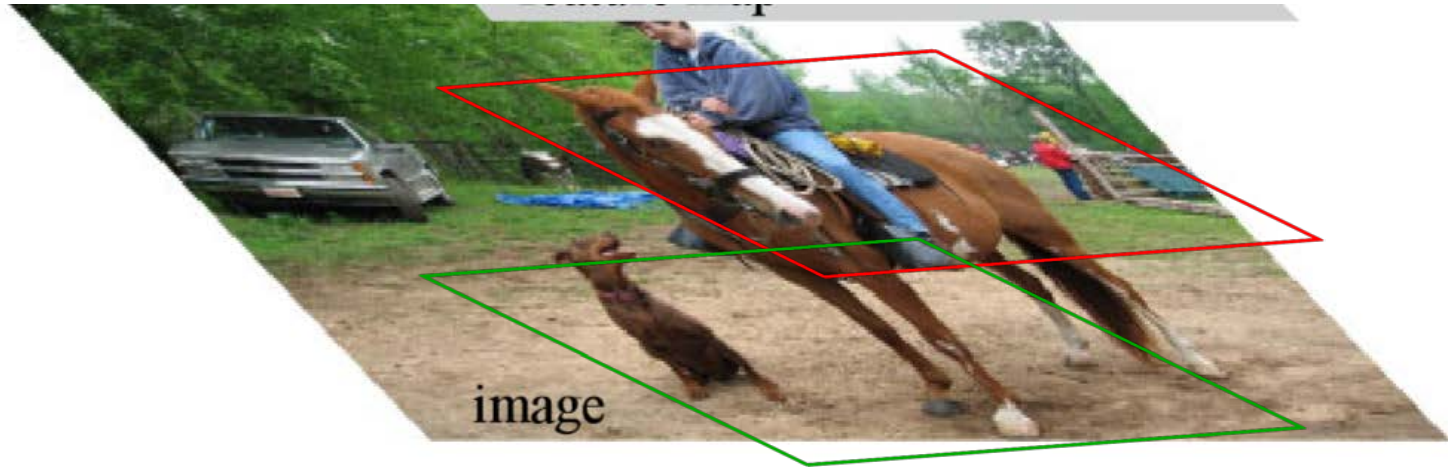
Face detection using google api





Training Data

- **Given positive sample, generate negative samples**
  (and balance sample sizes)



Sample and add boxes with < 40% overlap as ‘negative’, > 50% as positive

Matlab Implementation

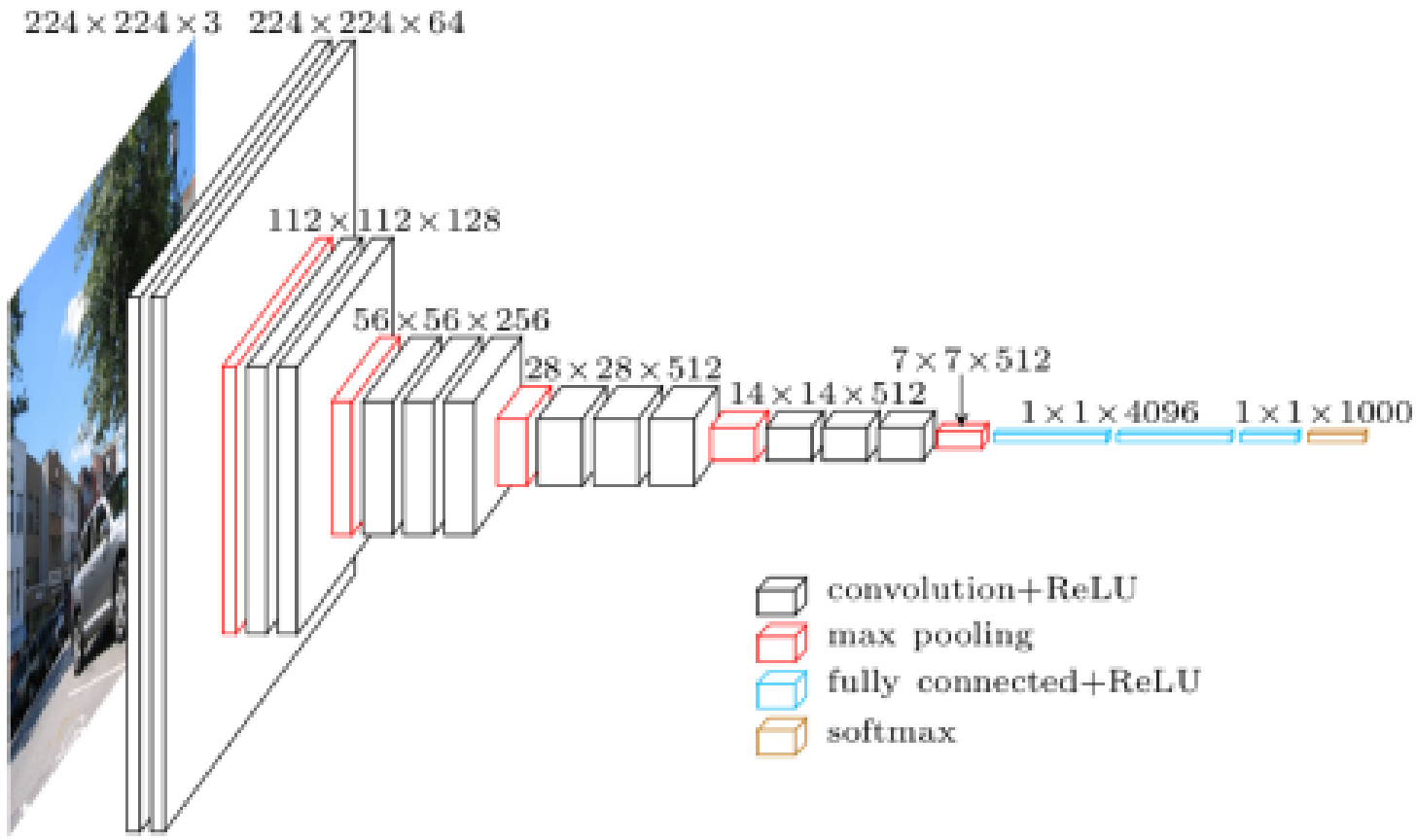
```
detector = trainFasterRCNNObjectDetector(
                    trainingData, vgg19, options, ...
```

~1K image file names
and ~2K boxes

A pretrained
network from Vis. Geom.
Group at Oxford

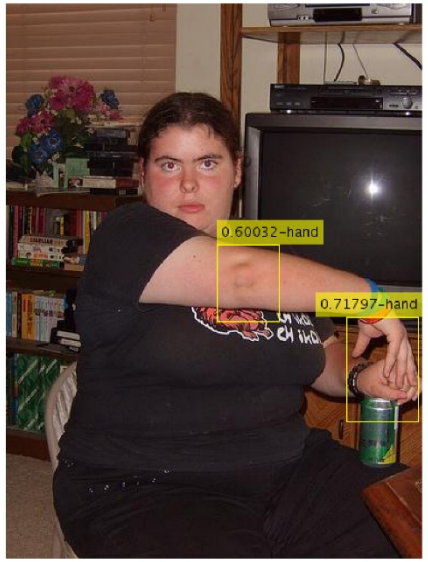
learning rate and epochs

Using VGG19 pretrained network



Results

- Using 50% overlap with true box as correct ~50% TP rate on train
- 4-8 hours on 1 compute node (CPU) 5 epochs on ~1K images



In Summary

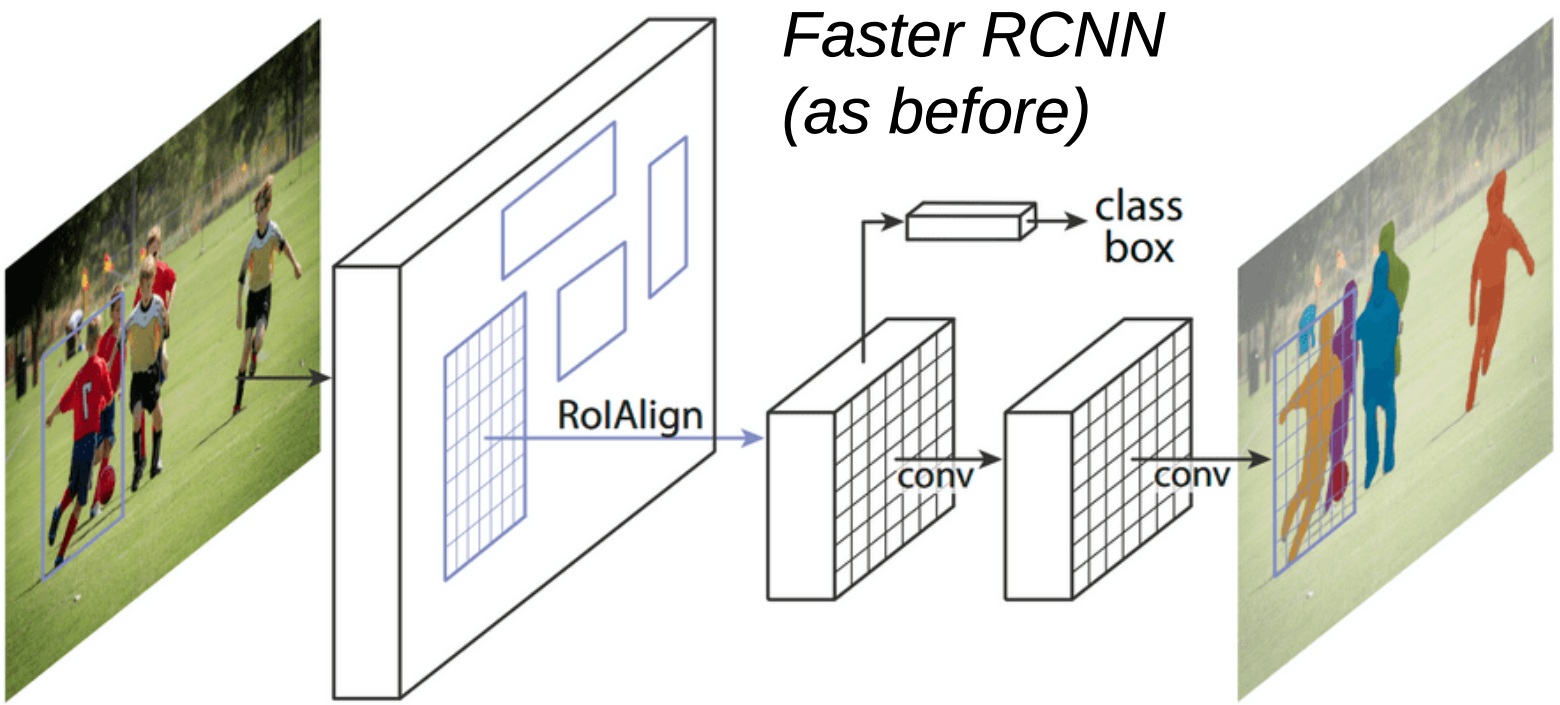
proposing regions takes much time

Matlab fasterRCNN easy to use but could use more options

(like turning off weight changes for transfer RCNN learning, reading in images faster, etc..)

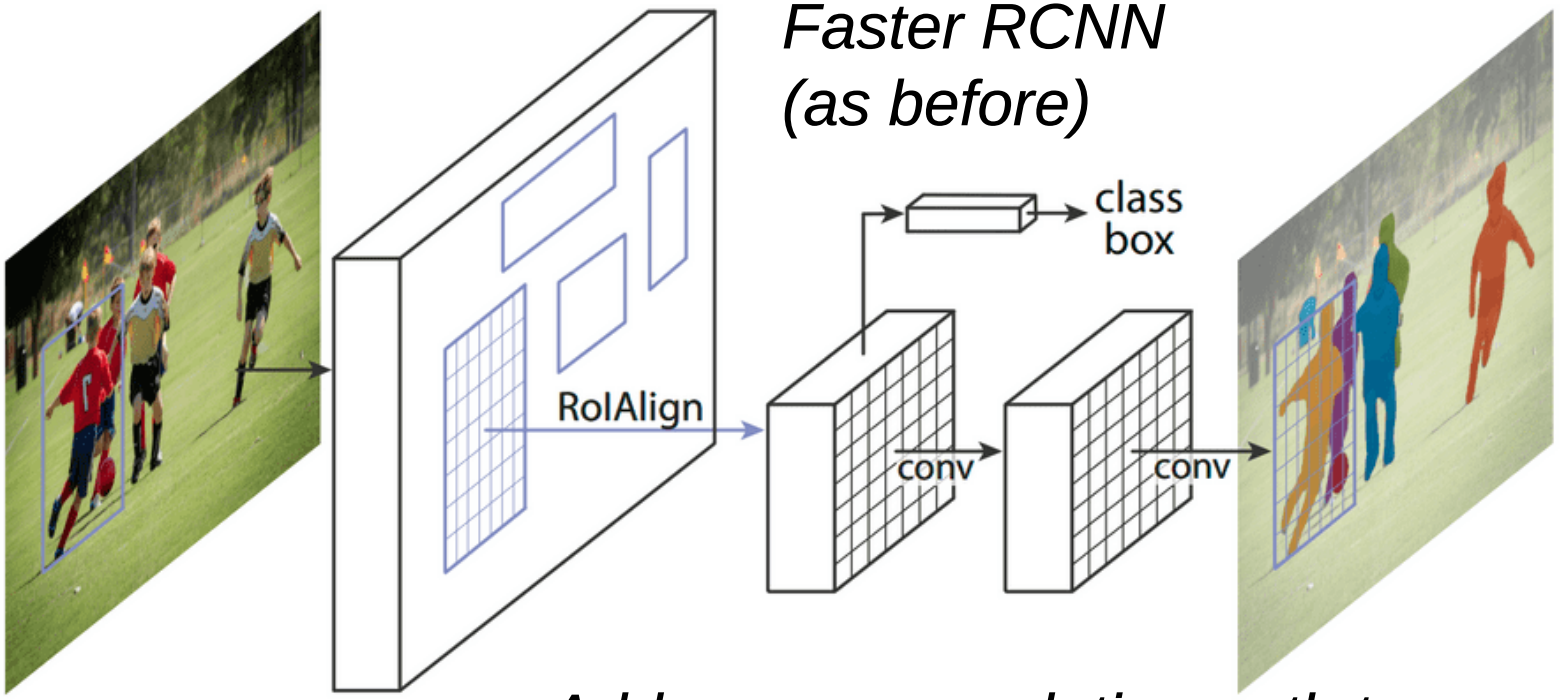
Segmentation with Mask RCNN

(He,Gkioxari,Dollar,Girschick 2018)



Segmentation with Mask RCNN

(He,Gkioxari,Dollar,Girschick 2018)

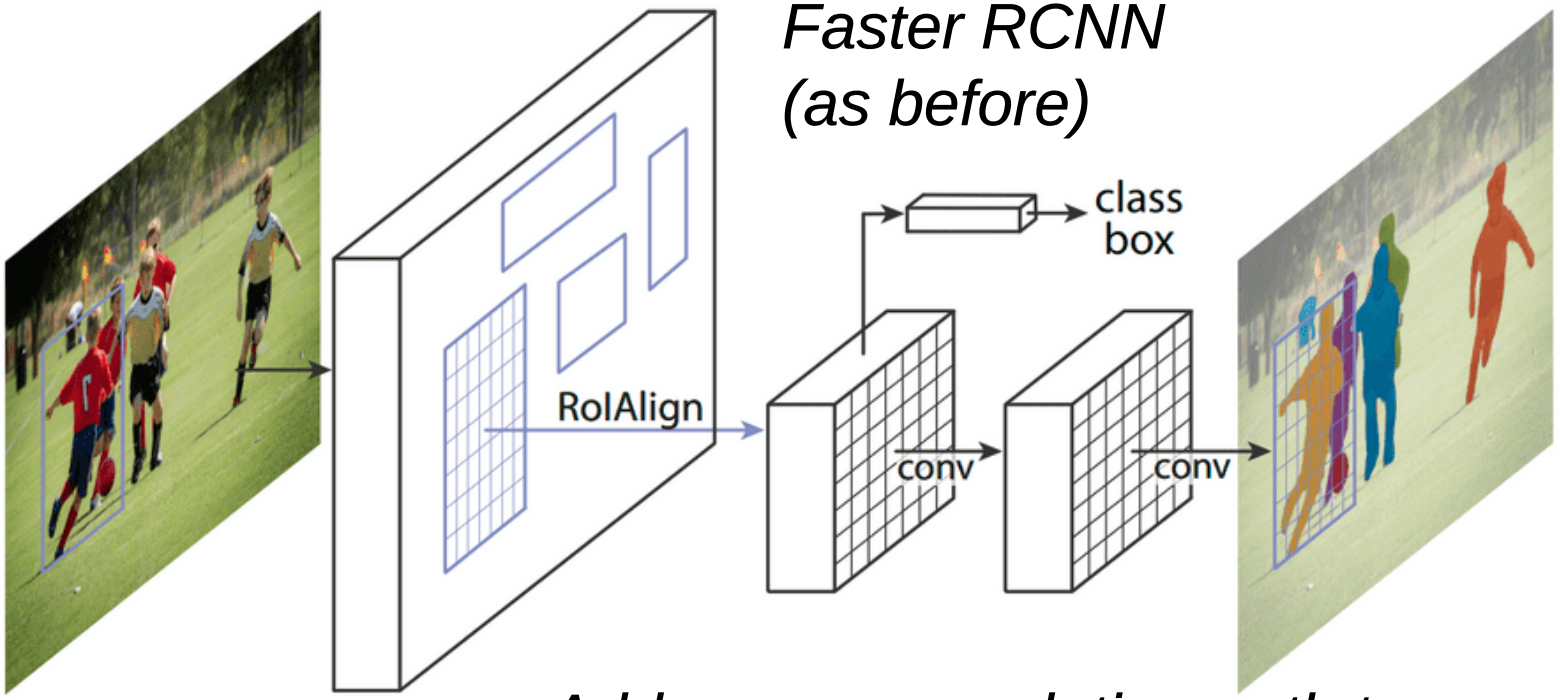


Faster RCNN
(as before)

Add a new convolution path to output pixelwise mask prediction

Segmentation with Mask RCNN

(He,Gkioxari,Dollar,Girschick 2018)



Faster RCNN (as before)

Add a new convolution path to output pixelwise mask prediction

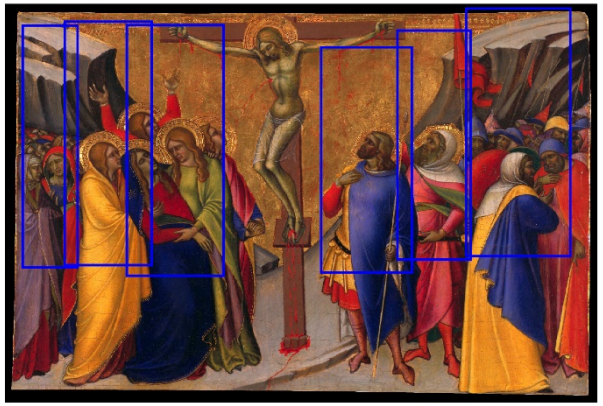
Note: the Loss for the whole model is

$L_{mask} + L_{class} + L_{box}$

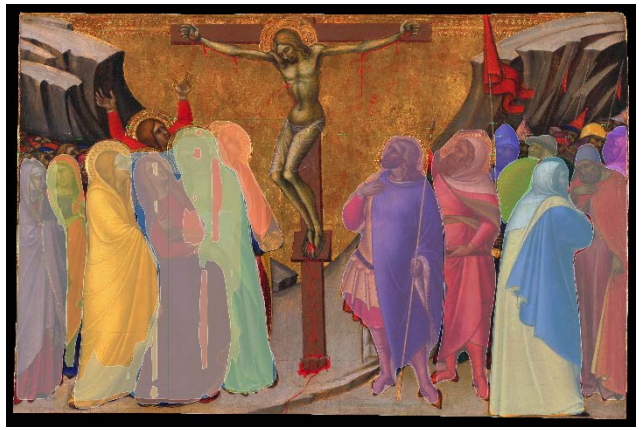
(the feature maps used by all paths will become more generic to whole task)

Caffe2, Facebook “Detectron” networks

Object Detection
ie getting a region
bounding box
(rcnn)



Object
Segmentation
ie getting a mask
(mask-rcnn)



Object Parts
ie getting keypoints
(keypoint-rcnn)

Caffe2 quick overview

- Facebook took over Caffe, and built Caffe2 on top of pyTorch
- Keras is easier to learn, Caffe2 better for production (supposedly)
- CNNs are built as defined-nets (ie network configurations)
- CNNs are run as prediction-nets
- Network activity directly available as “blobs” (like tensors)
- Caffe2 ‘brew’ library has Keras-like higher level API

References

- **Book: https://mitpress.mit.edu/books/deep-learning**
- **Documentation: https://keras.io/**
- **Tutorials I used (borrowed):**
  - http://cs231n.github.io/convolutional-networks/
  - https://hackernoon.com/visualizing-parts-of-convolutional-neural-networks-using-keras-and-cats-5cc01b214e59
  - https://github.com/julienr/ipynb_playground/blob/master/keras/convmnist/keras_cnn_mnist.ipynb