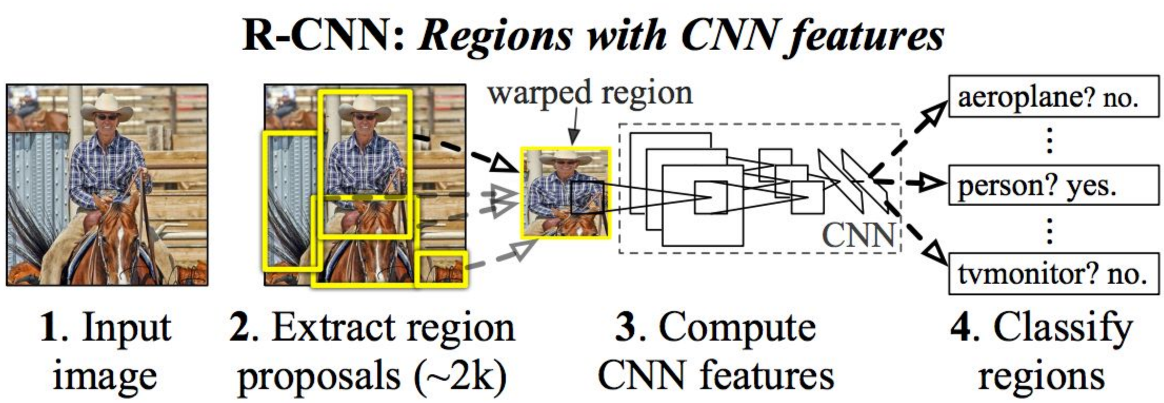
EVOLUTION OF RCNNs

1. R-CNN

- proposes a bunch of boxes in the image and sees if any of them correspond to an object
- Creates region proposal boxes using “Selective Search”
    - Selective search looks at the image through windows of different sizes
    - Tries to group adjacent pixels by texture, color, and intensity. Things with similar attributes are probably one object



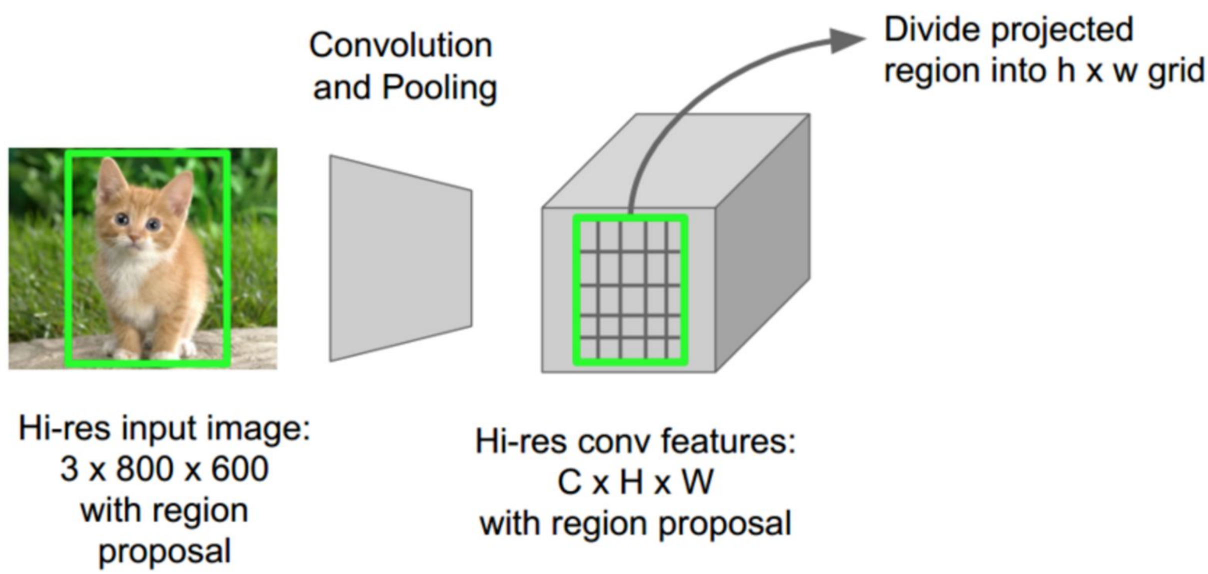
- When all proposals are created each region is warped to a square size and passed through a CNN to extract feature maps.
- Then, an SVM just classifies the feature map.
- Once an object is found, the box can be tightened even more to be more precise
    - Runs a linear regression
    - Input: sub regions of the image that corresponds to each object
    - Output: new bounding box coordinates for each object

Why R-CNN sucks:

- Needs a forward pass through the CNN (step 3) for each region proposal… and there are usually a lot of region proposals so that an object isn't missed
    - A lot of region proposals also overlap, leading to useless computation
- It has to train 3 different models separately
    - CNN, the SVM classifier, regression model to tighten box

2. Fast R-CNN

BENEFIT 1: Run the image through the CNN just once per image…

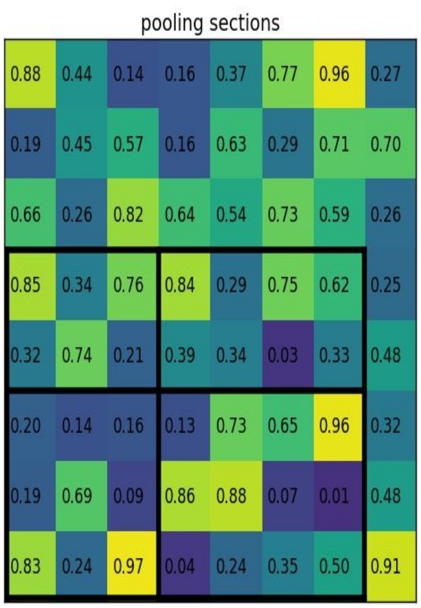
- Region of Interest Pooling (ROI pool):
  - 

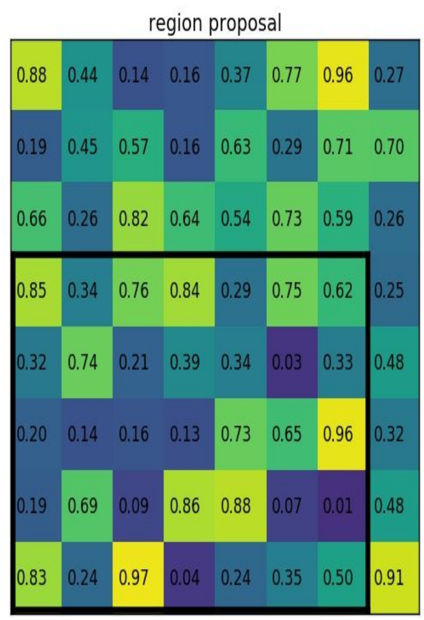
  - ROI pool shares the forward pass for each sub region because it is all done at once
  - features for each region in the original image is obtained by going to the corresponding area in the feature map
  - Then all the features in a region are max pooled, so now the regions with a high value probably have an object in them.

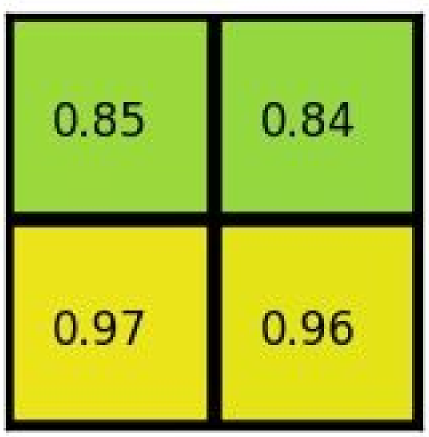
  - IN DEPTH:
    - ROI pooling layer inputs:
      - feature map from deep CNN
      - N x 5 matrix which has a list of ROI, where N in number of regions
        - Each row is a ROI
        - first column = image index

        - last 4 cols = coordinates for corners of region

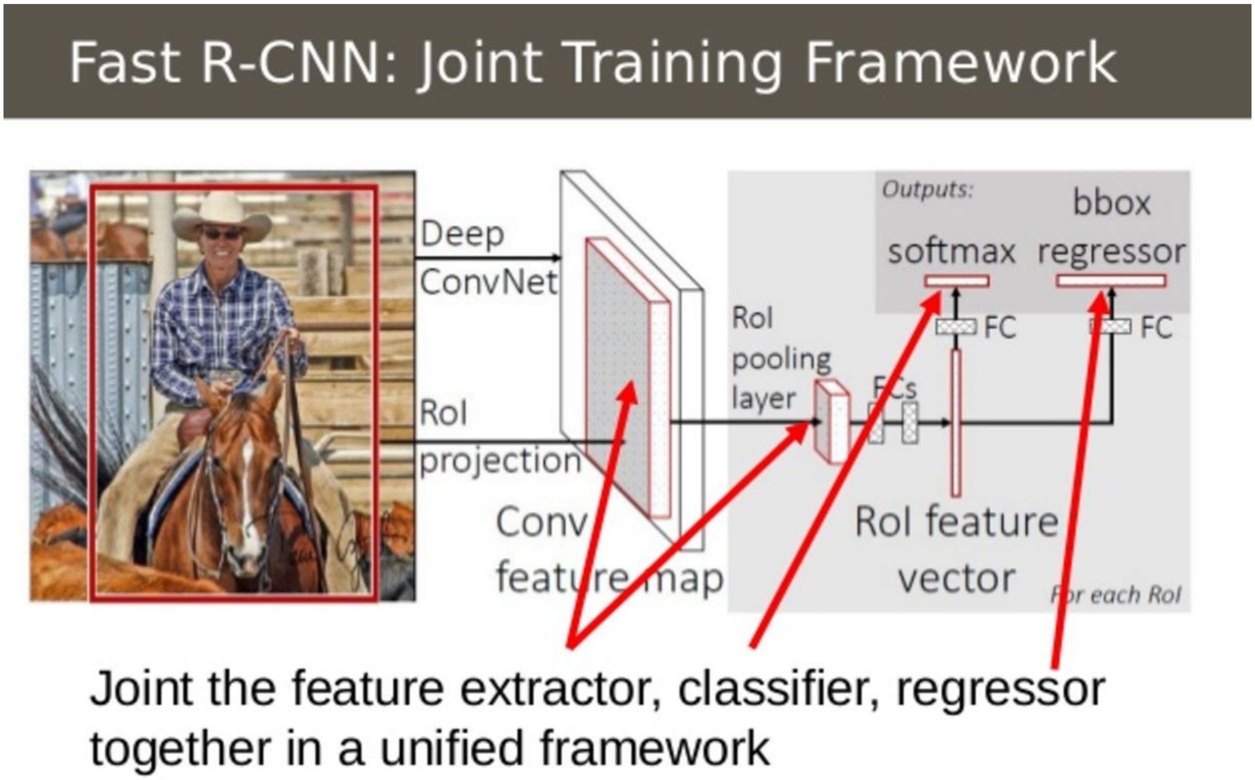
    - For every inputted region of interest, it takes the corresponding section of the feature map and scales it to some predefined size (p x p)
        - Divide the ROI into ‘p’ number of equal sized sections
        - Max pool each section
- This is fast because you can use the same feature map for every region proposal, no useless recalculating.







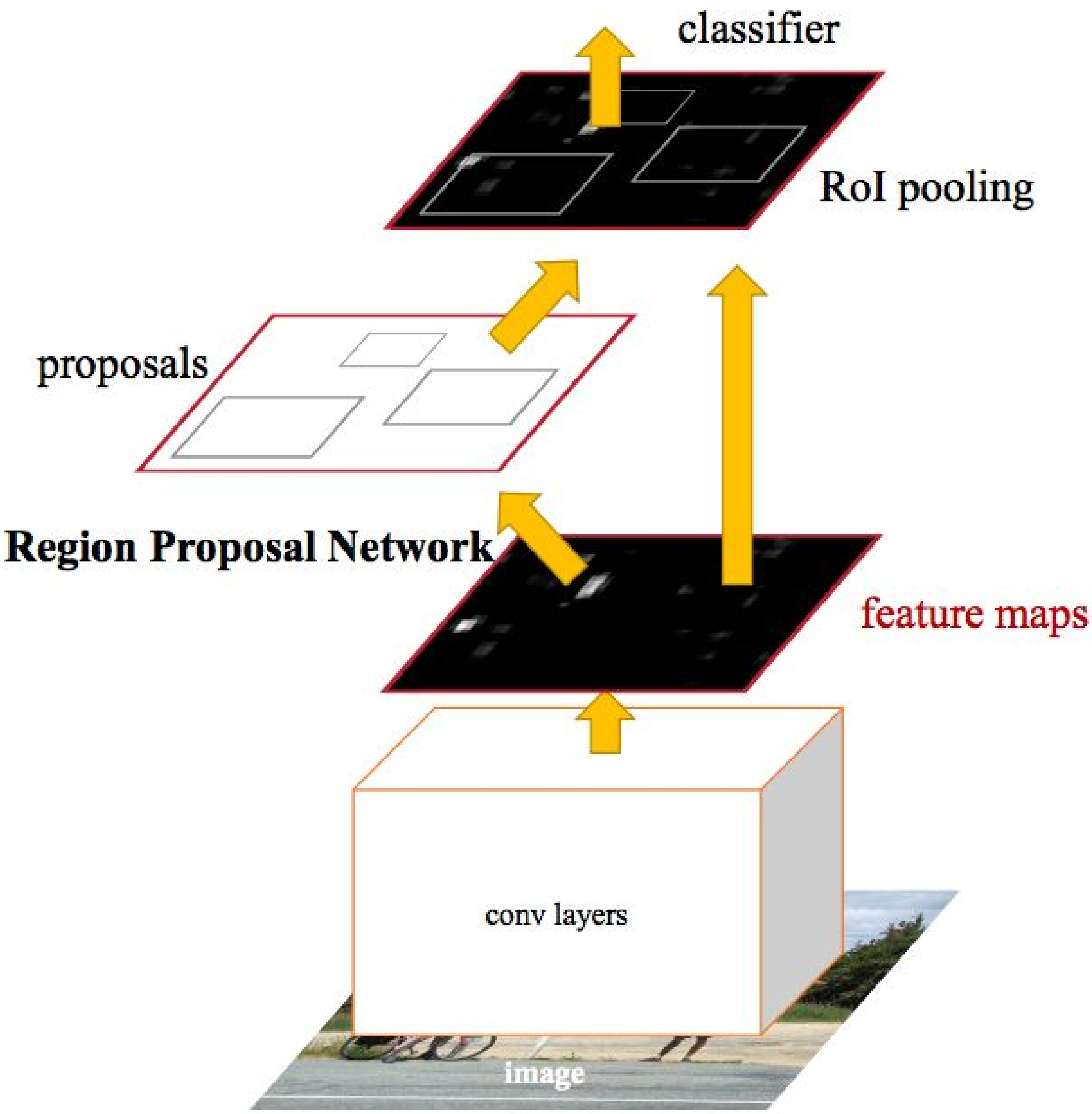
BENEFIT 2: Combine all models into just one network



- SVM classifier has been replaced with a softmax layer after the CNN.
- The regression layer is now parallel to the softmax layer.
- Inputs: Images with region proposals
- Outputs: classifications and tighter boxes

3. EVEN FASTER R-CNN

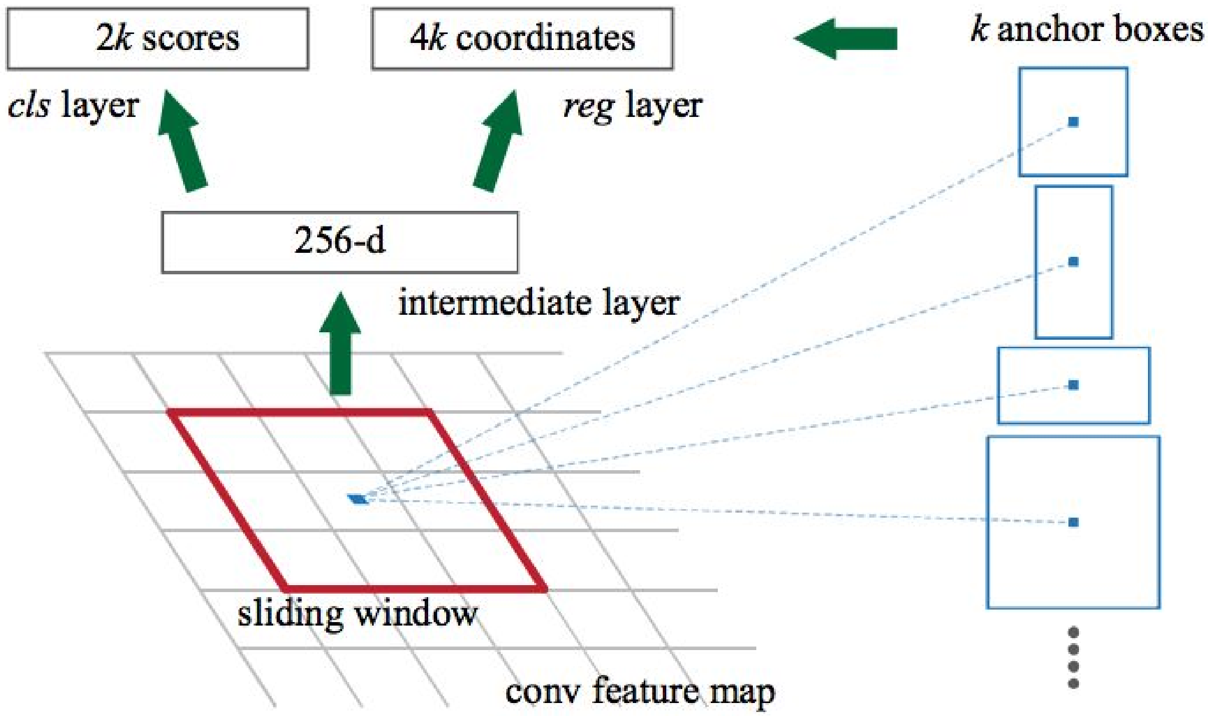
- **Problems with Fast R-CNN**
  - Region proposal process of selective search is still very slow
- Faster R-CNN uses the feature map extracted from the CNN both to do classification and do extract region proposals


classifier
RoI pooling
proposals
Region Proposal Network
feature maps
conv layers
image

- only one CNN needs to be trained
- **Inputs:**
  - just the raw image, no need to inputs region proposal boxes since we just generate those with the same feature map
- **Outputs:**
  - classifications and bounding boxes of objects

But how are the regions generated with just a feature map?!?!?!?!?!

- Region Proposal Network
  - Another CNN operating on the features extracted from the first CNN



- ○ Passes a sliding window over the CNN feature map (kind of like a convolution but not rly…)
    - ■ At each window location, the network outputs a score (for whether or not its an object) and the bounding box per anchor
- ○ In the end, outputs 'k' number of potential bounding boxes and scores for each box.

- Anchor boxes:
    - Some common aspect rations for the thing we are looking for
    - Example: we are looking for a ball
        - The aspect ratio of a ball will never be extremely thin so one anchor box could be 1:1
    - For each anchor box like that you output one bounding box and score per position in the image.
        - This lets the network know where the object we are looking for is in the image
    - **Inputs:**
        - CNN feature map
    - **Outputs:**
        - 1 bounding box per anchor, and a score representing how likely the thing in there is an object