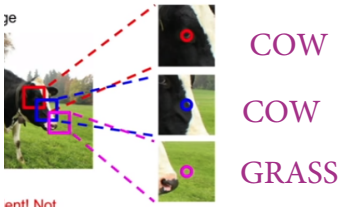
11.md

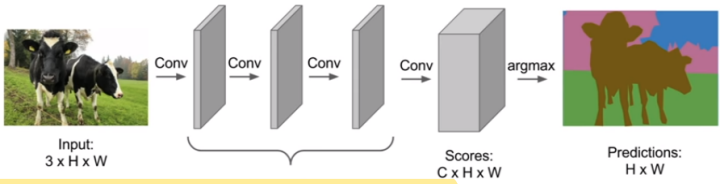
11. Detection and Segmentation

- So far we are talking about image classification problem. In this section we will talk about Segmentation, Localization, Detection.
- **Semantic Segmentation**
  - We want to Label each pixel in the image with a category label.

  - As you see the cows in the image, Semantic Segmentation Don't differentiate instances, only care about pixels.
  - The first idea is to use a **sliding window**. We take a small window size and slide it all over the picture. For each window we want to label the center pixel.
    - It will work but its not a good idea because it will be computational expensive!
    - Very inefficient! Not reusing shared features between overlapping patches.
    - In practice nobody uses this.



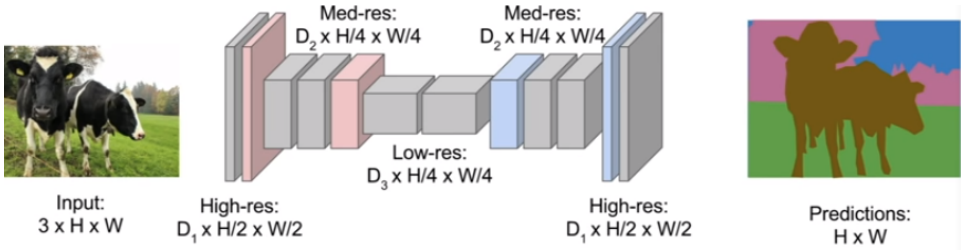
  - The second idea is designing a network as a bunch of Convolutional layers to make predictions for pixels all at once!
    - Input is the whole image. Output is the image with each pixel labeled.
    - We need a lot of labeled data. And its very expensive data.
    - It needs a deep Conv. layers.
    - The loss is cross entropy between each pixel provided.
    - Data augmentation are good here.
    - The problem with this implementation that convolutions at original image resolution will be very expensive.
    - So in practice we don't see something like this right now.



  - The third idea is based on the last idea. The difference is that we are downsampling and upsampling inside the network.

- We downsample because using the whole image as it is very expensive. So we go on multiple layers downsampling and then upsampling in the end.
- Downsampling is an operation like Pooling and strided convolution.
- Upsampling is like "Nearest Neighbor" or "Bed of Nails" or "Max unpooling"
  - **Nearest Neighbor** example:

```
Input:  1  2          Output:  1  1  2  2
        3  4                   1  1  2  2
                               3  3  4  4
                               3  3  4  4
```



  - **Bed of Nails** example:

```
Input:  1  2          Output:  1  0  2  0
        3  4                   0  0  0  0
                               3  0  4  0
                               0  0  0  0
```

  - **Max unpooling** is depending on the earlier steps that was made by max pooling. You fill the pixel where max pooling took place and then fill other pixels by zero.

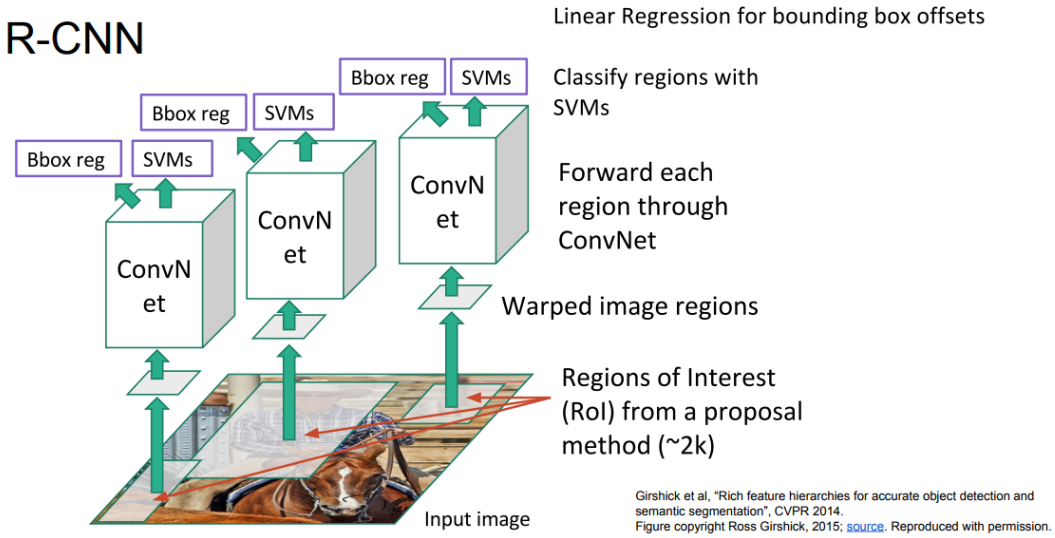
לזכור את מי מערכי הפיקסלים שמרנו, ואז כמו BED OF NAILS אבל
המיקום יקבע לכל פיקסל בנפרד - לפי מי מהם באמת נלקח

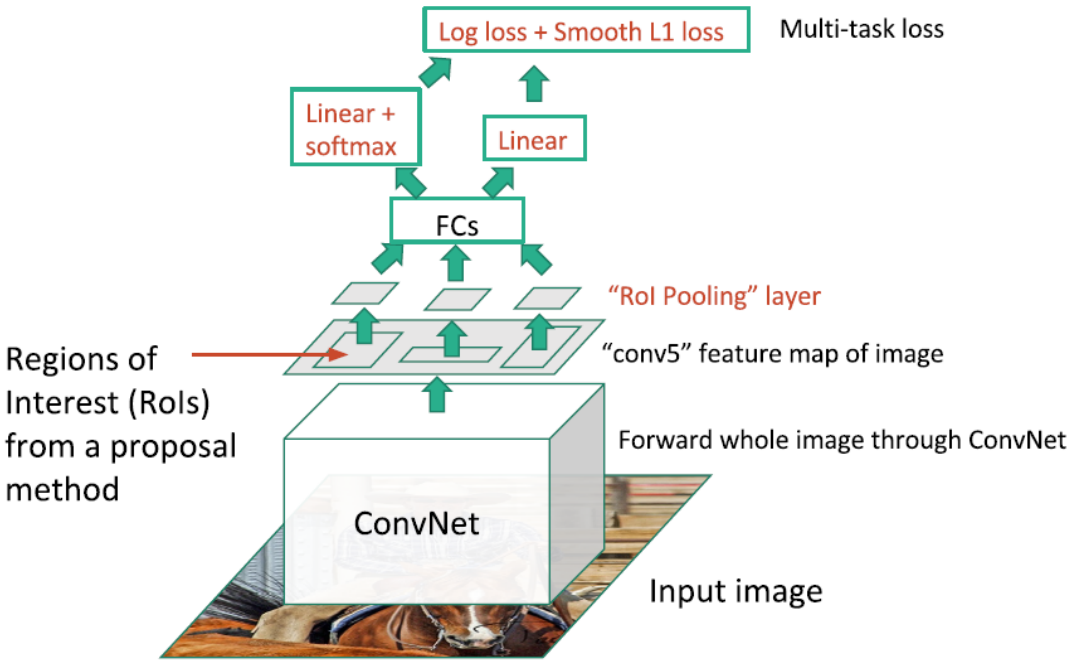
- Max unpooling seems to be the best idea for upsampling.
- There are an idea of Learnable Upsampling called "**Transpose Convolution**"
  - Rather than making a convolution we make the reverse.
  - Also called:
    - Upconvolution.
    - Fractionally strided convolution
    - Backward strided convolution

קונבולוציה מגרעין אחד שמוציא 3 על 3

  - Learn the artimitic of the upsampling please refer to chapter 4 in this paper.

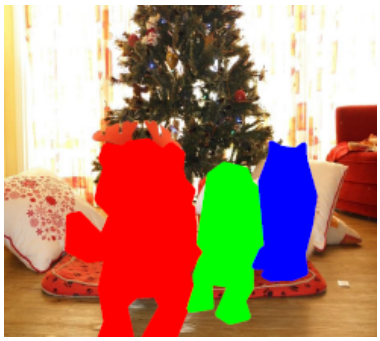
- **Classification + Localization**:
  - In this problem we want to classify the main object in the image and its location as a rectangle.
  - We assume there are one object.
  - We will create a multi task NN. The architecture are as following:
    - Convolution network layers connected to:
      - FC layers that classify the object. `# The plain classification problem we know`
      - FC layers that connects to a four numbers `(x,y,w,h)`
        - We treat Localization as a regression problem.
  - This problem will have two losses:
    - Softmax loss for classification
    - Regression (Linear loss) for the localization (L2 loss)
  - Loss = SoftmaxLoss + L2 loss
  - Often the first Conv layers are pretrained NNs like AlexNet!
  - This technique can be used in so many other problems like: Human Pose Estimation.
- **Object Detection**
  - A core idea of computer vision. We will talk by details in this problem.

- The difference between "Classification + Localization" and this problem is that here we want to detect one or mode different objects and its locations!
- First idea is to use a sliding window
  - Worked well and long time.
  - The steps are:
    - Apply a CNN to many different crops of the image, CNN classifies each crop as object or background.
  - The problem is we need to apply CNN to huge number of locations and scales, very computationally expensive!
  - The brute force sliding window will make us take thousands of thousands of time.
- Region Proposals will help us deciding which region we should run our NN at:
  - Find **blobby** image regions that are likely to contain objects.
  - Relatively fast to run; e.g. Selective Search gives 1000 region proposals in a few seconds on CPU
- So now we can apply one of the Region proposals networks and then apply the first idea.
- There is another idea which is called R-CNN
  - 

  - The idea is bad because its taking parts of the image -With Region Proposals- if different sizes and feed it to CNN after scaling them all to one size. Scaling is bad
  - Also its very slow.
- Fast R-CNN is another idea that developed on R-CNN
  - 

        - It uses one CNN to do everything.
    - Faster R-CNN does its own region proposals by Inserting Region Proposal Network (RPN) to predict proposals from features.
        - The fastest of the R-CNNs.
    - Another idea is Detection without Proposals: YOLO / SSD
        - YOLO stands for you only look once.
        - YOLO/SDD is two separate algorithms.
        - Faster but not as accurate.
    - Takeaways
        - Faster R-CNN is slower but more accurate.
        - SSD/YOLO is much faster but not as accurate.

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)

- **Denese Captioning**
    - Denese Captioning is "Object Detection + Captioning"
    - Paper that covers this idea can be found here.
- **Instance Segmentation**
    - This is like the full problem.
    - 
    - Rather than we want to predict the bounding box, we want to know which pixel label but also distinguish them.
    - There are a lot of ideas.
    - There are a new idea "Mask R-CNN"
        - Like R-CNN but inside it we apply the Semantic Segmentation
        - There are a lot of good results out of this paper.
        - It sums all the things that we have discussed in this lecture.
        - Performance of this seems good.