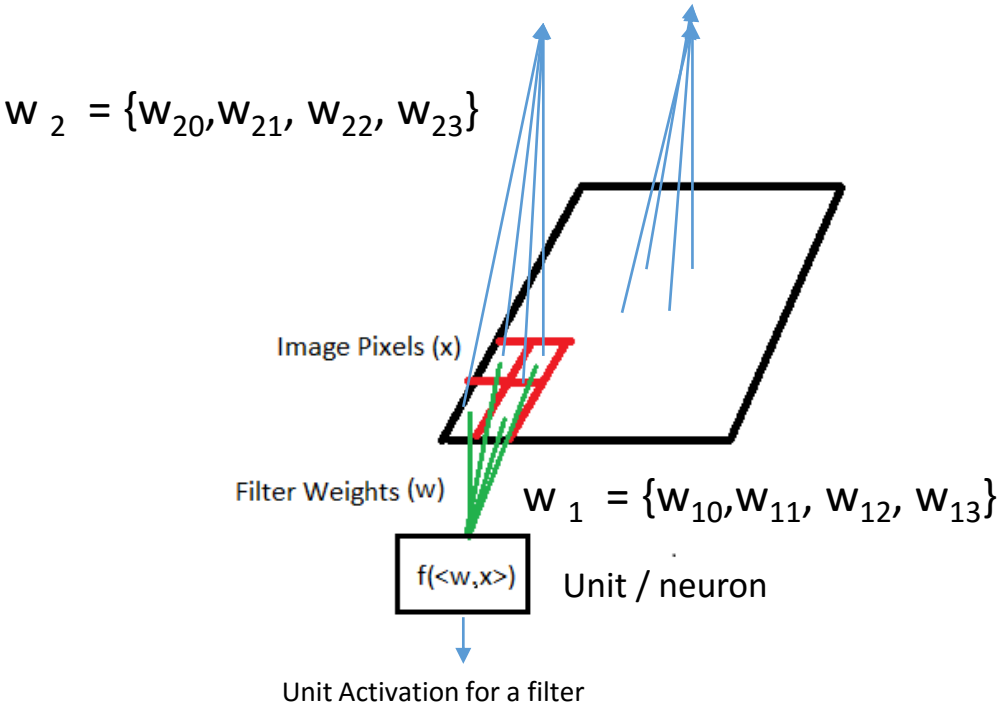
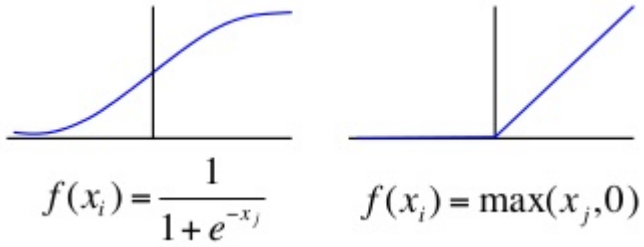
Visualizing Conv-nets

Slide images and text sourced off:
Paper: Object Detectors emerge in Deep Scene CNNs: Zhou et. al
http://cs231n.github.io/
http://colah.github.io/

Mukul Sati
2/8/2016

Conv-net recap

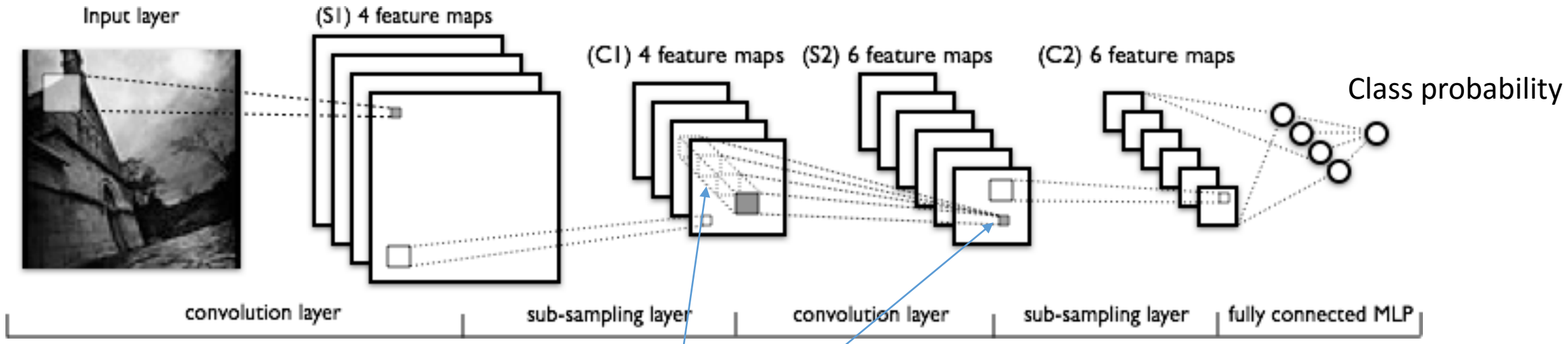




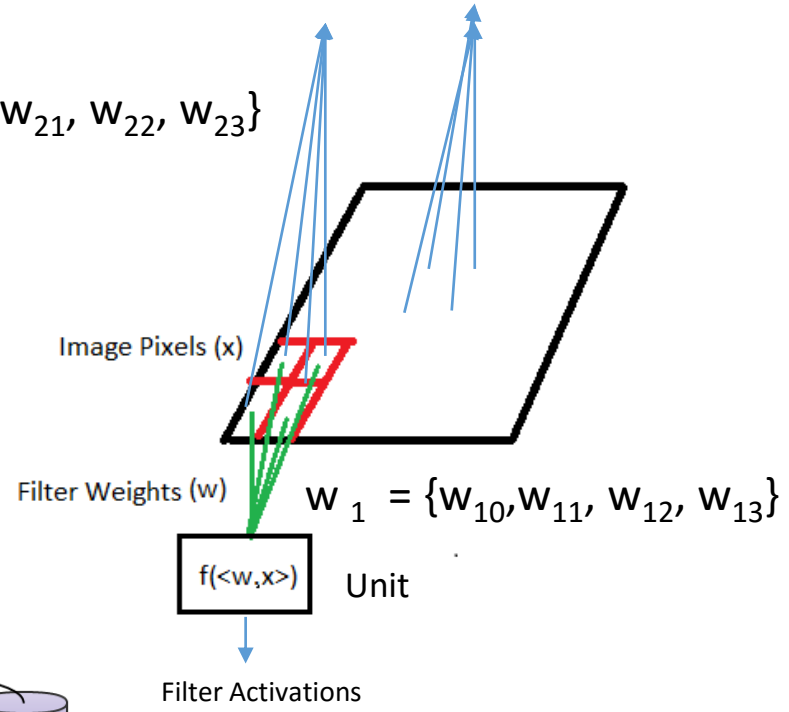
Sigmoid ReLu (Rectified Linear Unit)

Each neuron/unit learns a set of filter weights $\{w_i\}$ that it must weigh spatially local regions by. Each w_i is shared across all the units. Here, 2 filter weight sets w_1 and w_2 are shown.

The linear combination (dot product) of the actual input and the learned weights are passed through a non-linearity to get the activation for a unit for a filter.

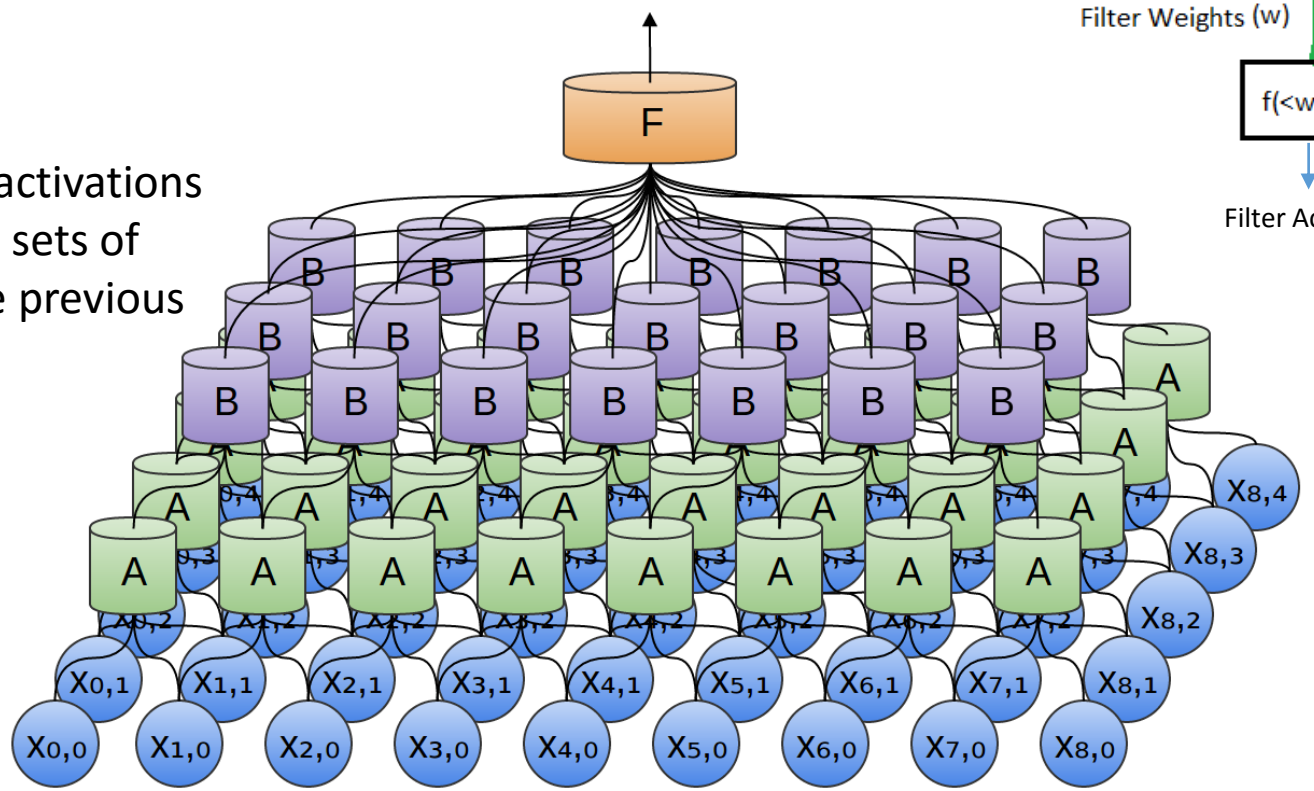


The convolution layers are stacked.
Each unit at a stacked layer learns a set of weights (a set of filters). Each such set identifies one particular way in which the spatially neighboring **activations** of **all** filters/values of all pixel color-channels in the layer below are linearly combined before being passed through the lon-linearilty.



Weights for combining activations of all learned filters in a sets of neighboring units in the previous layer are learned

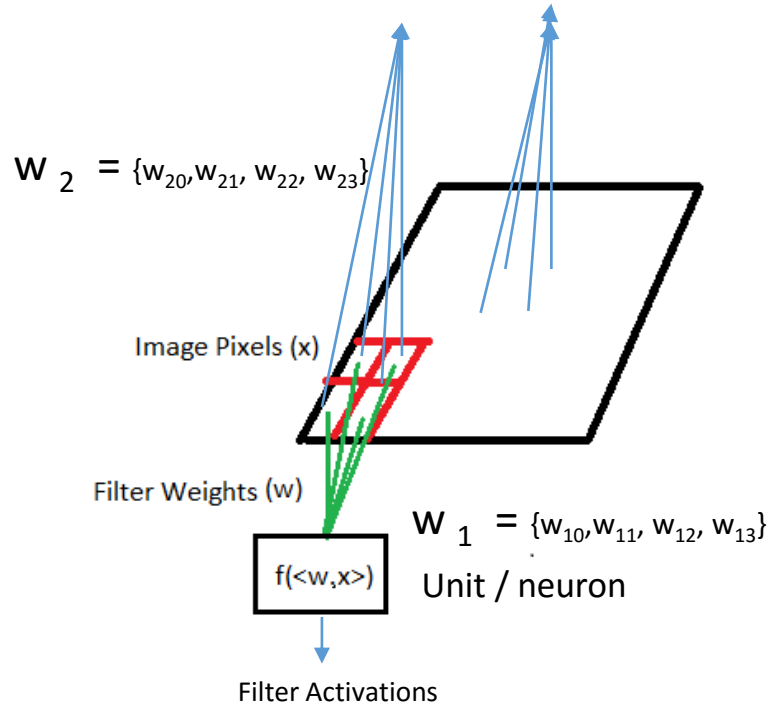
Filters (set of weights) for combining pixel neighbors (R,G,B) channels are learned



Today’s discussion - visualizing learned information

- We have trained the net using back-prop, and are interested in visualizing what the conv-net has learned for enlightenment and wonder ☺.

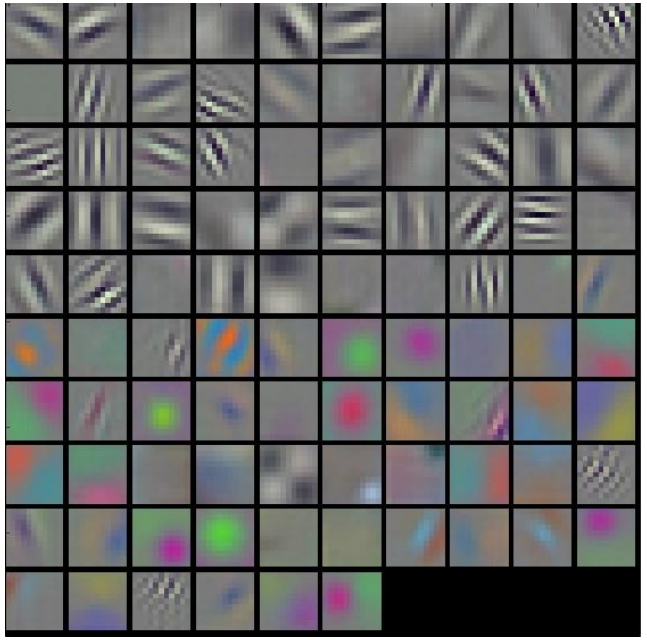
Visualize filter weights



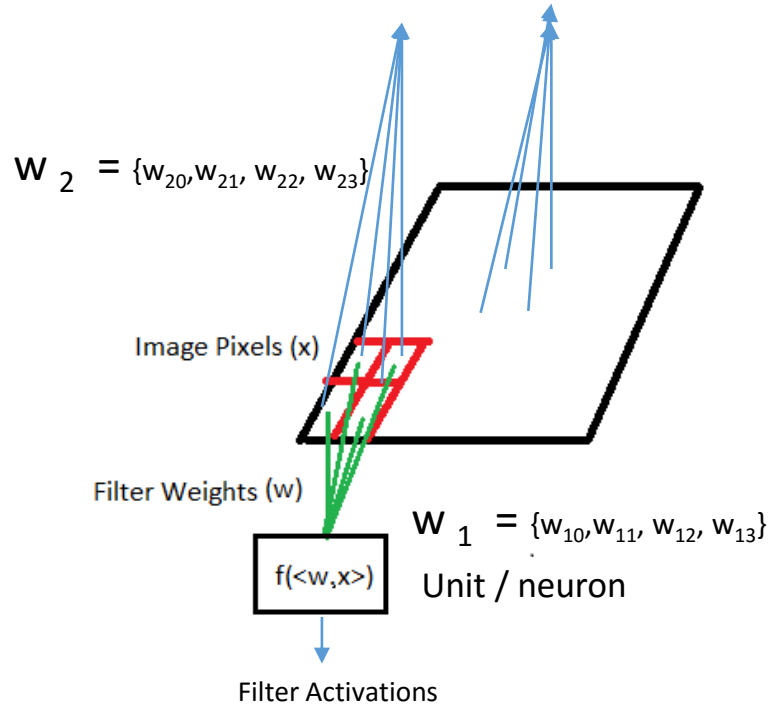
- Easily interpretable at the first layer. Well-trained networks usually display nice and smooth filters without any noisy patterns – mostly, pixels in a spatial locality in an image are correlated.

- Noisy patterns can be an indicator of a network that hasn't been trained for long enough.

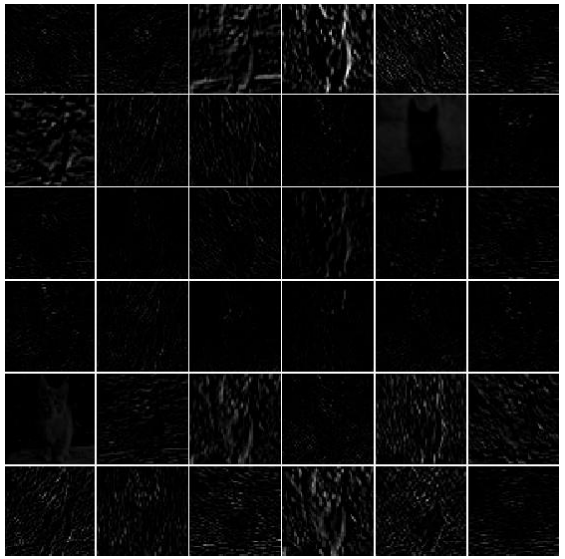
- At first layer, mostly edge-detectors are learned =>



Visualize filter activations



- Visualize filter activations – Select an image. Select a filter. Feed the image forward through the network up-to the desired layer. Find & visualize activations, for that filter, of each unit in the layer.
- The set of all activations for 1 filter => activation map.
- Activations are generally sparse and localized – specially if ReLu is used for non-linearity, which gives true zeros. Thus, activation maps are mostly black when visualized =>



Every box shows an activation map corresponding to some filter.

Occluding parts of images

- A ConvNet classifies an image as a dog. How can we be certain that it's actually picking up on the dog in the image as opposed to some contextual cues from the background or some other miscellaneous object?
- Plot probability of class of interest as a function of the position of an occluder object: Iterate over regions of the image: set the region of the image to be all zero, and look at the probability of the class finally predicted for this new image. Visualize the probability as a 2-dimensional heat map.

The probability of Pomeranian plummets when the occluder covers the face of the dog

Display images that maximally activate a unit

- Take a large dataset of images, feed them through the network and keep track of which images maximally activate some unit (neuron) (can take sum of activation, or max activation over all learned filters for the neuron).
- Visualize the set of images to get an understanding of what the unit is looking for in its receptive field.
- As each filter for the unit has learned a non-linear combination of the activations of different filters in previous layers, images with different content may activate the same unit, but some neurons "learn" to distinguish higher level abstractions automatically.

It can be seen that some neurons are responsive to upper bodies, text, or specular highlights

Object Detectors emerge in Deep Scene CNNs

- Train CNNs to perform scene classification

- As scenes are composed of objects, the CNN automatically discovers meaningful object detectors! <Without ever being taught the notion of objects>

Image	Scene Label
	Bedroom
	Auditorium
	Art Gallery
	Dining Room

Differences in learned representation

Conv nets are trained on two different set of images:
1) ImageNet – Object labels (cats, etc)
2) Places-CNN – Scene labels.

Figure 1: Top 3 images producing the largest activation of units in each layer of ImageNet-CNN (top) and Places-CNN (bottom). (The activations in a layer are summed)

Observe that the earlier layers such as pool1 and pool2 prefer similar images for both networks while the later layers tend to be more specialized to the specific task of scene or object categorization.

Visualizing discriminating information for scenes

- Given an image that is correctly classified by the network, simplify this image such that it keeps as little visual information as possible while still having a high classification score for the same category. This simplified image (named minimal image representation) will highlights the elements that lead to the high classification score.

- Given an image, create a segmentation of edges and regions and remove segments from the image iteratively. At each iteration, remove the segment that produces the smallest decrease of the correct classification score and do this until the image is incorrectly classified.

Visualizing discriminating information for Scenes

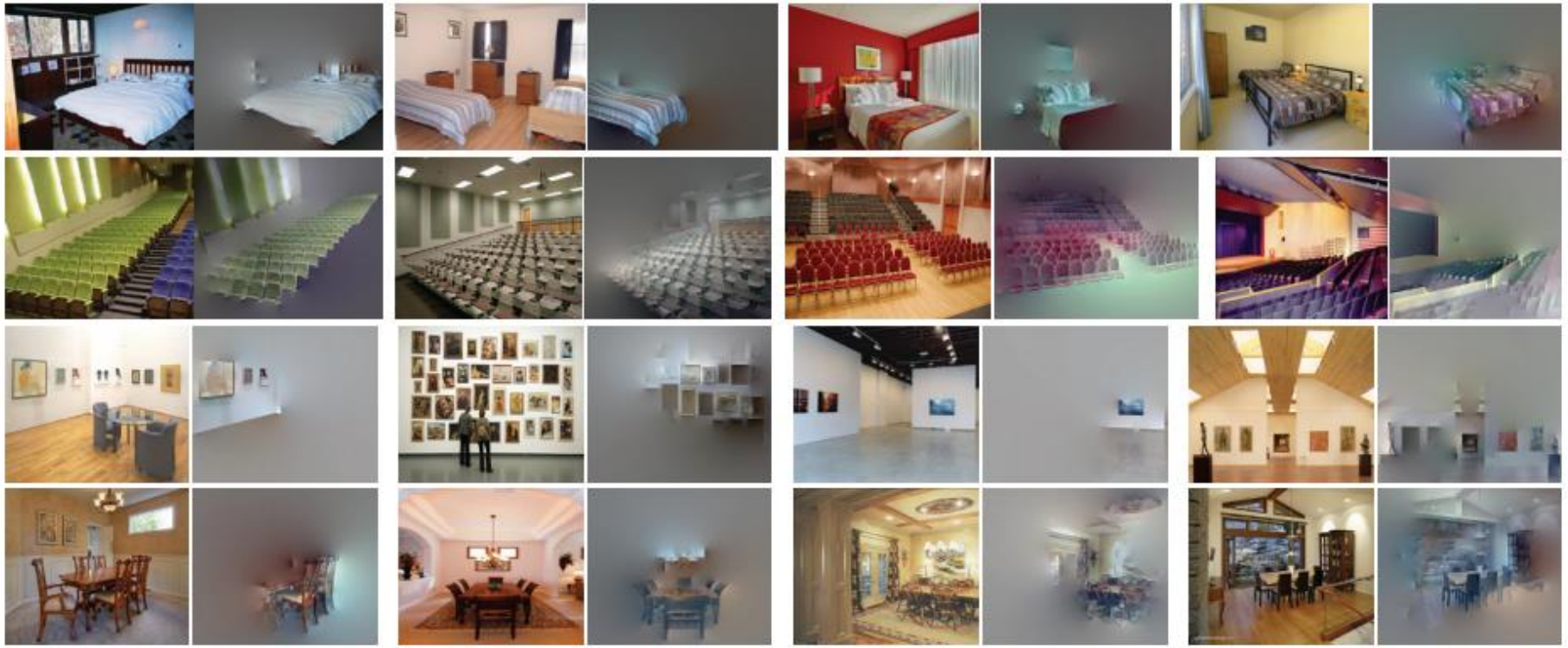


Figure 2: Each pair of images shows the original image (left) and a simplified image (right) that gets classified by the Places-CNN as the same scene category as the original image. From top to bottom, the four rows show different scene categories: bedroom, auditorium, art gallery, and dining room.

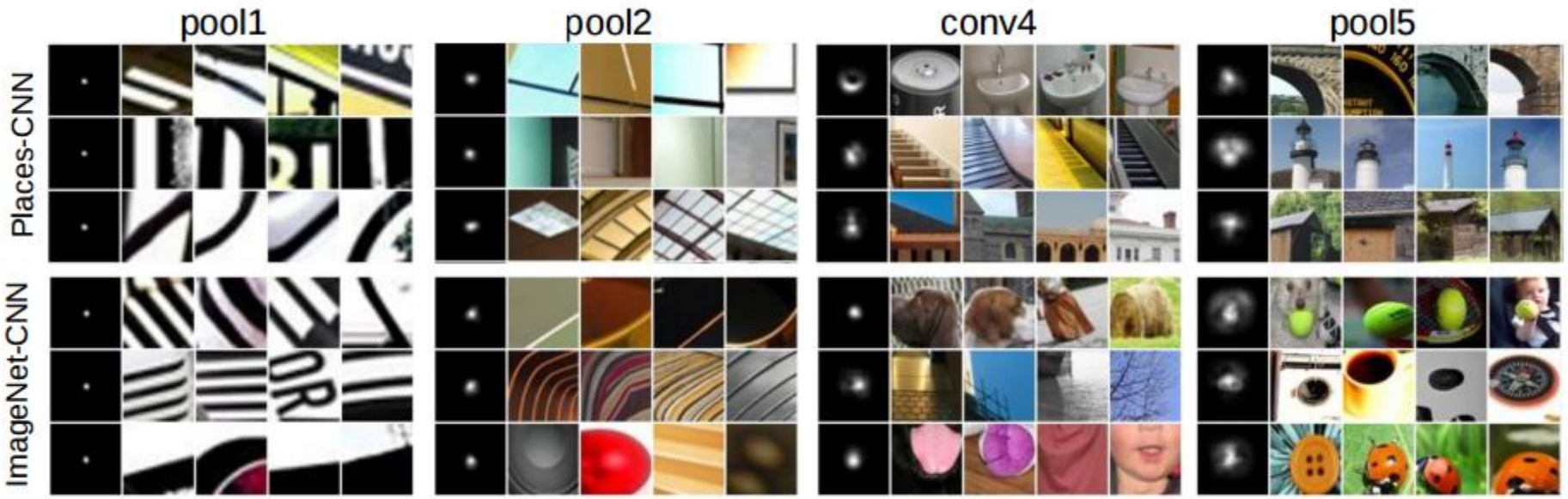
Receptive field size estimation for a unit ‘u’

- What pixels in an image are being used to activate a neuron?
- Calculable theoretically. Here, how to do this empirically.
- Select a unit. Select the top K images with the highest activations for the unit.
- *Identify image regions for activation:* Want to identify exactly which regions of the image lead to the high unit activations. Replicate each image many times with small random occluders at different locations in the image. This results in many occluded images per original image. Feed all the occluded images into the same network and record the change in activation as compared to using the original image. If there is a large discrepancy, we know that the given patch is important and vice versa. This allows us to build a discrepancy map for each image for a unit. Similar to =>

Receptive field size estimation for a unit ‘u’

- *Consolidate:* For each image, find out the unit ‘v’ which gives highest activation (this may be different from the current unit ‘u’). Find the theoretical spatial location of the image corresponding to unit ‘v’. Center the discrepancy map for this image here. Average over all discrepancy maps.

Receptive field size estimation for a unit



The RFs of 3 units of pool1, pool2, conv4, and pool5 layers respectively for ImageNet and Places-CNNs.

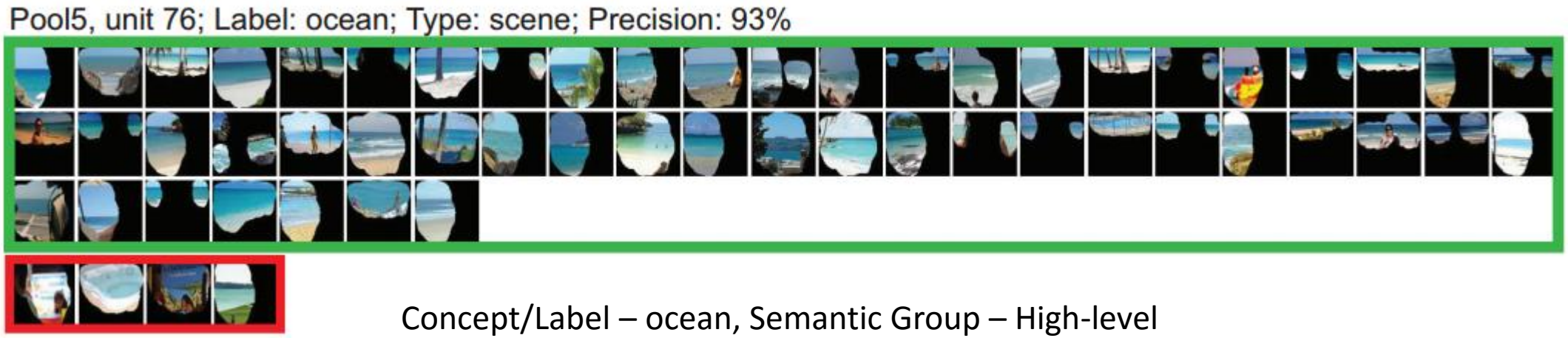
As the layers go deeper, the RF size gradually increases and the activation regions become more semantically meaningful.

Identifying if object level semantics are learned by units

- Goal - understand and quantify the precise semantics learned by each unit.
- Show workers on Mechanical Turk top 60 segmented images that most strongly activate the unit (and also some very low activating images for cross-validation) and ask them to:
  - **Identify the concept, or semantic theme or label:** e.g., car, blue, vertical lines, etc,
  - Mark the set of images that do not fall into this theme,
  - **Categorize the concept in one of 6 semantic groups ranging from low-level to high-level**:
    Simple elements and colors (e.g., horizontal lines, blue)
    Materials and textures (e.g., wood, square grid),
    Regions and surfaces (e.g., road, grass),
    Object parts (e.g., head, leg),
    Objects (e.g., car, person), and
    Scenes (e.g., kitchen, corridor).
- This allows both the semantic information for each unit, as well as the level of abstraction provided by the labeled concept to be obtained.

Identifying if object level semantics are learned by units

- Precision - the percentage of highly activated images that were selected as fitting the labeled concept.
- A large number of units in pool5 are devoted to detecting high level semantic groups - objects and scene-regions.



Concept/Label – ocean, Semantic Group – High-level

Identifying if object level semantics are learned by units

- A large number of units in pool5 are devoted to detecting objects and scene-regions. But what categories of objects are found? Is each category mapped to a single unit or are there multiple units for each object class? Can this information be used to segment a scene into objects?

Object level semantics…

- Some units from the Places-CNN grouped by the type of object class they seem to be detecting.
- Each row shows the top five images for a particular unit that produce the strongest activations.
- The segmentation shows the regions of the image for which the activation of the unit is above a certain threshold.
- Each unit seems to be selective to a particular appearance of the object. For instance, there are 6 units that detect lamps, each unit detecting a particular type of lamp providing finer-grained discrimination; there are 9 units selective to people, each one tuned to different scales or people doing different tasks.

Thanks!