Image Segmentation / Object Detection
Image Segmentation / Object Detection

• Find all instances of (a) particular type(s) of object(s) within an image.

• Examples:
  - Find all cats in an image
  - Find all cats, dogs, persons, chairs, ...
Image (binary) classification:

- 6 Filters 129 x 129 x 3
- 4 Filters 65 x 65 x 6
- Flatten and fully connected to 100 neurons
- Fully connected to 10 neurons
- Fully connected to 1 neuron

Estimate class of 256 x 256 RGB image.
Cat or not Cat
Chair or not chair
Example: Cats

Binary Image Cat Classifier

Input: 256 x 256 RGB Image

Output: Probability image is of a cat

<table>
<thead>
<tr>
<th>Cat</th>
<th>- Cat</th>
</tr>
</thead>
<tbody>
<tr>
<td>.994</td>
<td>.006</td>
</tr>
</tbody>
</table>
Example: Cats

Binary Image
Cat Classifier

Input:
256 x 256 RGB Image

Output:
Probability image is of a cat

<table>
<thead>
<tr>
<th>Cat</th>
<th>- Cat</th>
</tr>
</thead>
<tbody>
<tr>
<td>.994</td>
<td>.006</td>
</tr>
</tbody>
</table>
Example: Cats

Binary Image Cat Classifier

Input:
256 x 256 RGB Image

Output:
Probability image is of a cat

<table>
<thead>
<tr>
<th>Cat</th>
<th>- Cat</th>
</tr>
</thead>
<tbody>
<tr>
<td>.0019</td>
<td>.9981</td>
</tr>
</tbody>
</table>
Approach 1: Sliding Classifier

- Take fixed size image classifiers trained to identify desired objects.
  - Example 256 x 256 RGB
- Apply them to each segment of the picture.
- Returns a pixel-by-pixel map giving probability that the specified area around given pixels contain the object.
• Start at the top left corner...
• Slide the classifier square over the image, getting readings for each location.
• Slide the classifier square over the image, getting readings for each location.
• The output will be a probability value for each location tested.
• Note that we use our CNN classifier as a filter on the larger image.
• Using multiple image classifiers as multiple filters allows us to find different types of objects in an image
Approach 2: FCN

- The sliding classifier needs an inference run for every location
  - Very inefficient
- We can improve efficiency by using a FCN or 'deep filter'.
Fully connected layer = Maximal filter

- A neuron in a fully connected layer is equivalent to a filter of equal size to its input layer.
  - It is a non-linear function of the weighted sum of all input layer outputs.
An alternative view of our cat network.
• If we explicitly specify the fully connected layers as maximal filters we obtain a 'deep filter'.

• It runs on the original size image, producing a single output.

• It will also run on larger images, producing a matrix of outputs centered at different pixels.
• The output of the FCN is equivalent to sliding the original CNN over larger pictures.
• But it is much more efficient since it avoid large amounts of redundant calculations.