

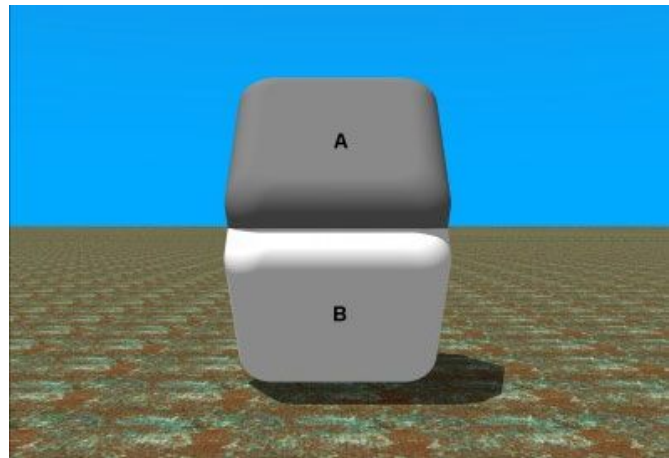
Convolutional Neural Networks

Andrew H. Fagg

Deep Networks for Image Recognition

- Images are composed of large numbers of pixels
- A particular pixel value can vary a lot:
 - Color, illumination
- Objects can vary a lot
 - Size, orientation, perspective

Individual pixels are irrelevant...

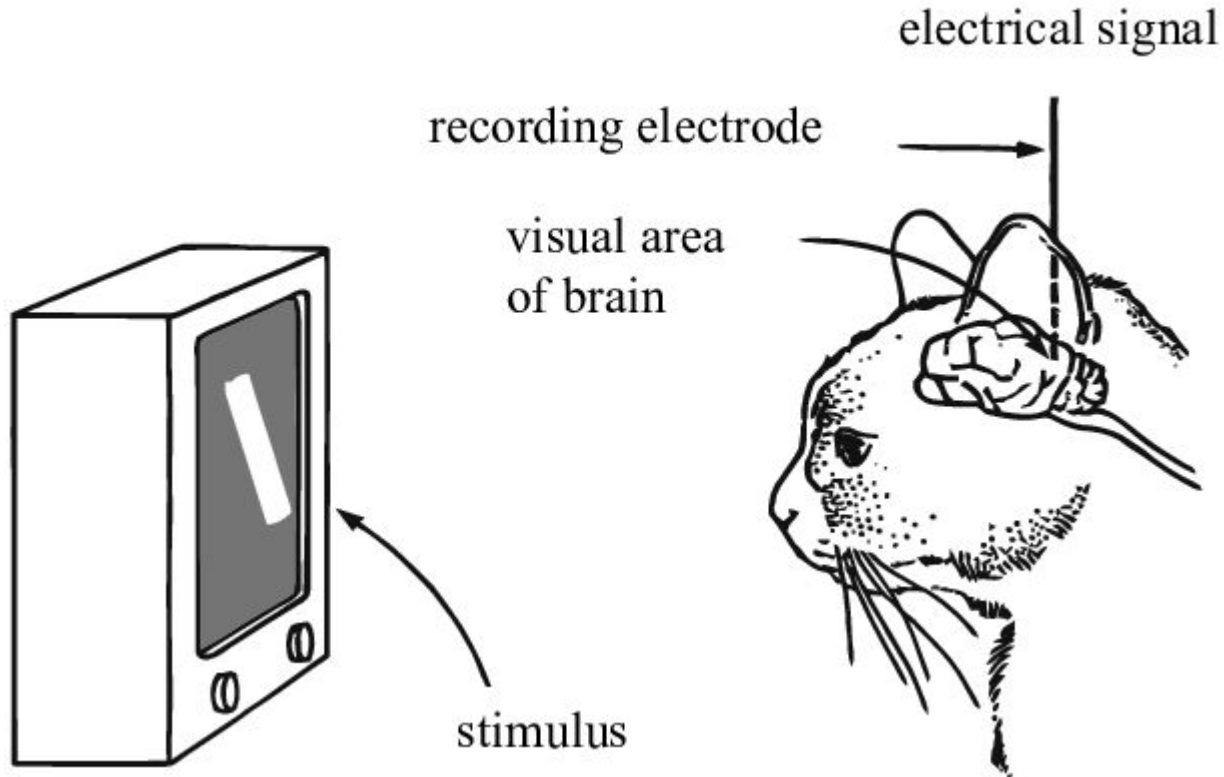


Deep Networks for Image Recognition

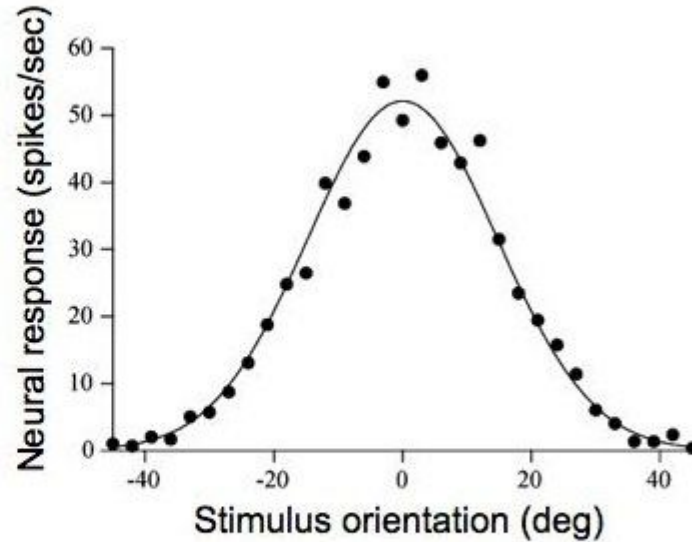
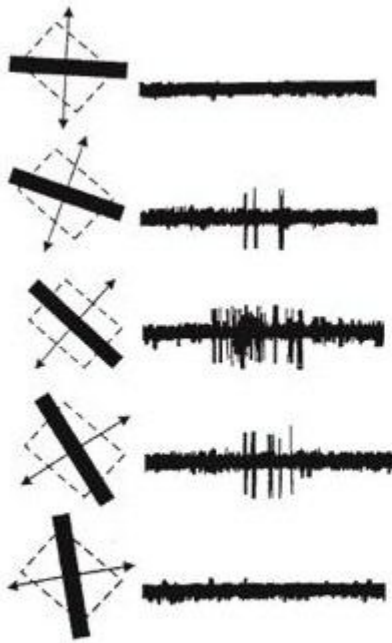
- 1920 (rows) x 1020 (columns) x 3 (channels = RGB) is almost 6 million inputs
- If the next dense layer has 1000 units, then we would have 6 billion parameters!

Need lots of examples and lots of training time. How do we get beyond this?

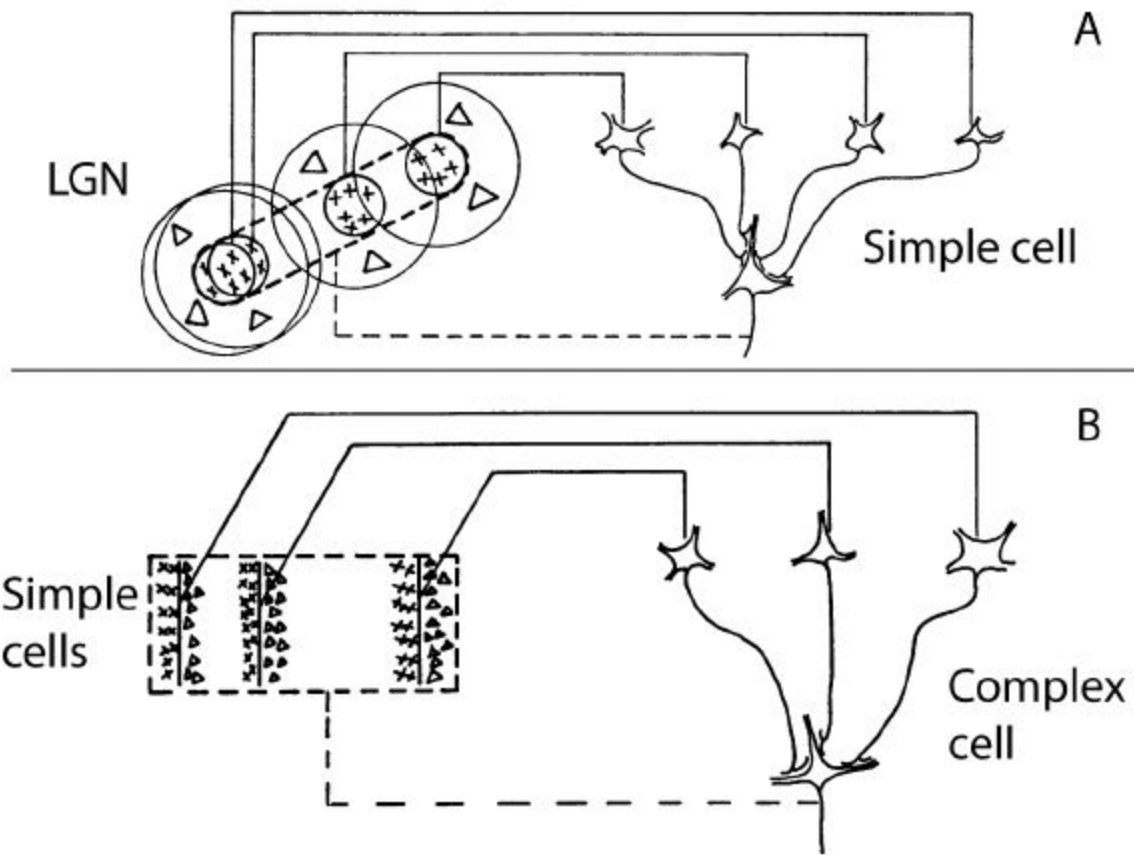
Hubel and Wiesel (1968)

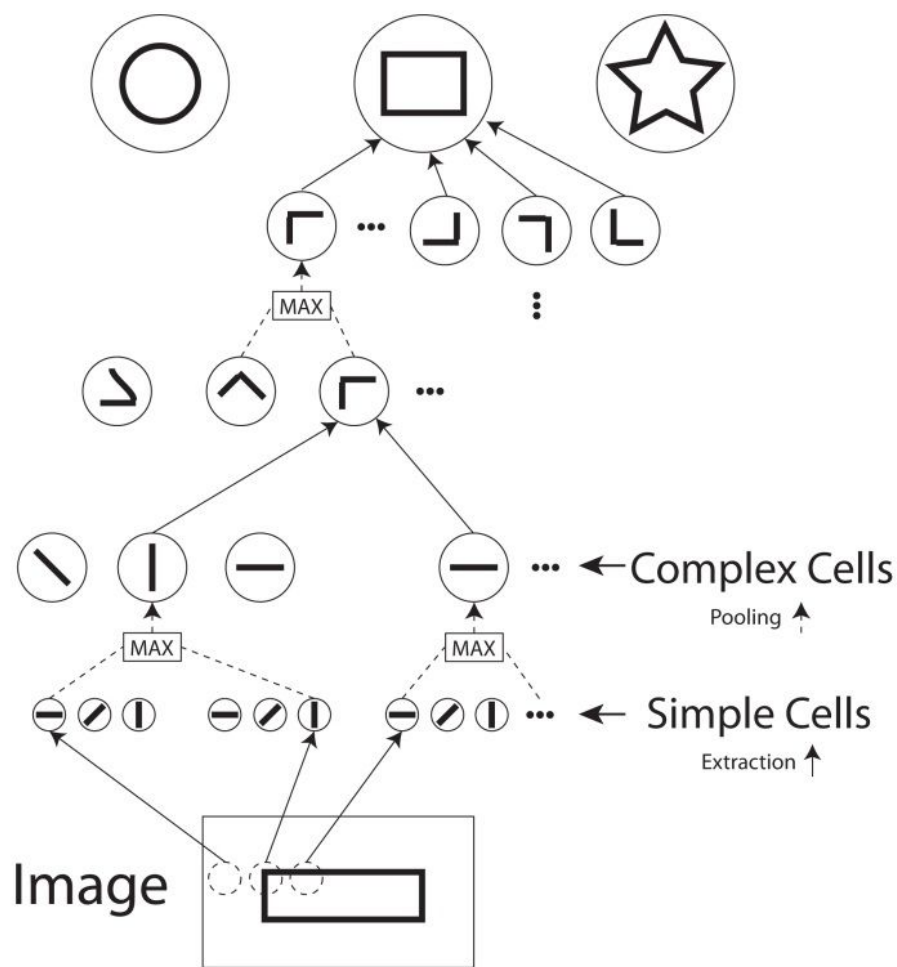


Orientation Sensitivity



Complex Features Formed from Simple Ones

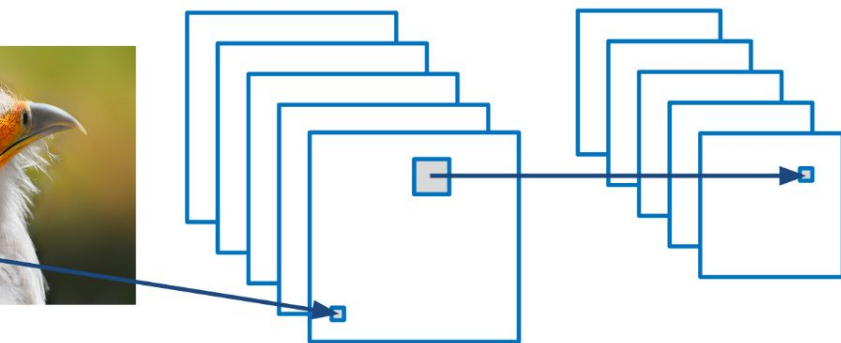




Operator Types

- Feature detection: recognize some pattern over a small grid of inputs
 - Convolution
- Pooling: does there exist some pattern within the grid of inputs?
 - Max
- Scaling: Allows simple feature detection and pooling to apply at multiple visual scales

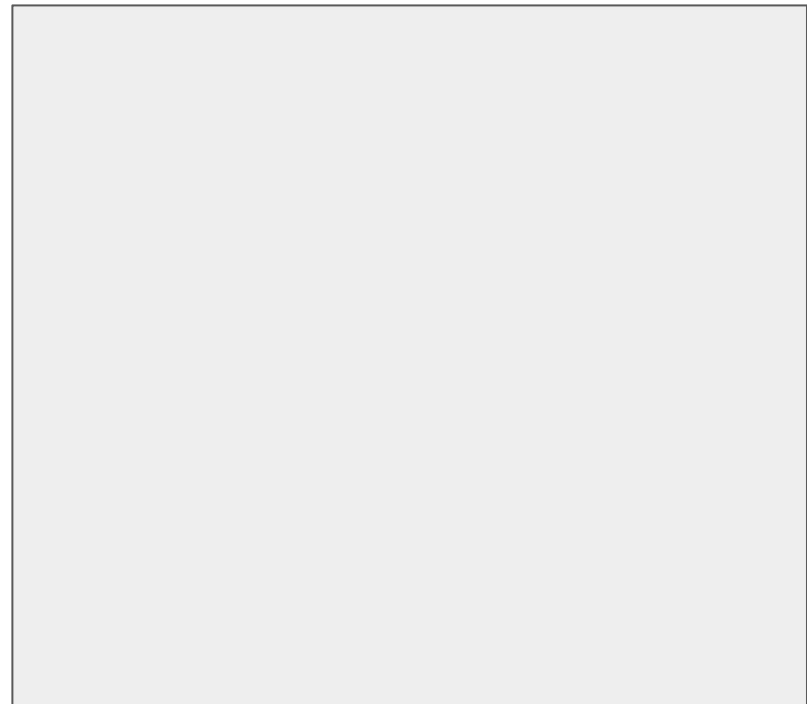
Local Operators



convolution +
nonlinearity

max pooling

convolution + pooling layers



Local Operators

Multiple stacked modules consisting of pattern recognition (convolution), pooling (max) and scaling (striding)

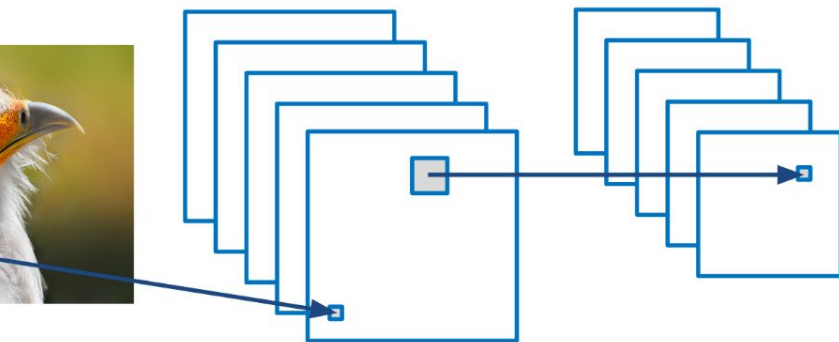
- With each module, our representation becomes more and more abstract
 - Ultimately: feathers, eyes, beaks ...
 - All have specific visual patterns, though there may be many variations of each
- Putting these larger scale features together: how do we handle the big variations in positioning of eyes relative to beaks (and other features)?

Beyond the Primitives

How should the primitives be combined to form more of a semantic representation (dog, cat, grandma, etc.)?

- After computing the primitives in the first layers of our deep network, employ dense layers to allow for arbitrary combinations of the primitives

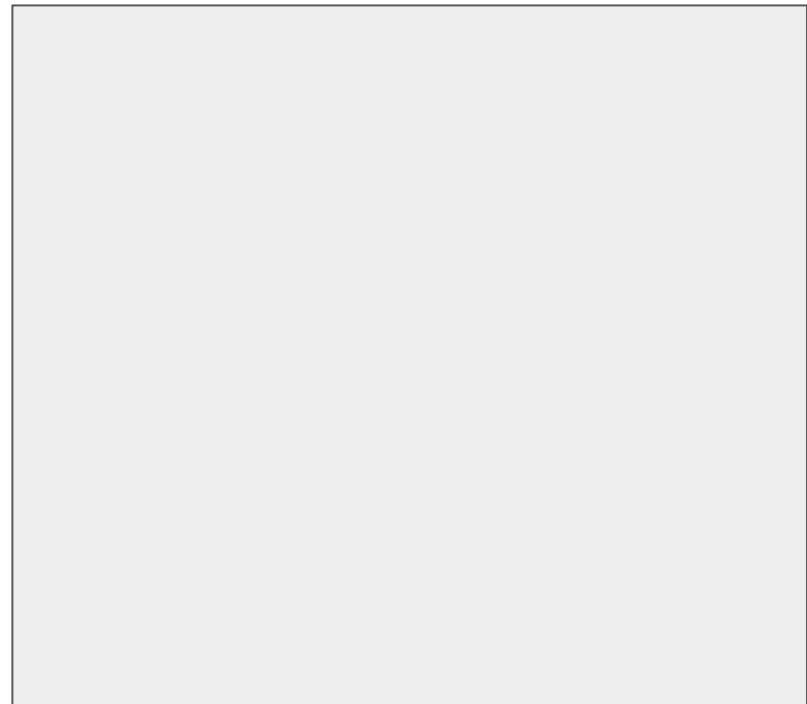
Local Operators



convolution +
nonlinearity

max pooling

convolution + pooling layers



Simple Image Recognition Architecture

- Multiple modules of convolution, pooling and scaling
- Flatten the feature representation
 - Flatten: old way of doing things (still maintains some spatial info)
 - Global Max Pooling: Does there exist a feature in the image?
- Multiple layers of densely connected neurons
- Final layer:
 - Compute a score for each class that we are recognizing
 - Compute a softmax over the scores: yields class probabilities

Combining Local Operators to Recognize Global Patterns

