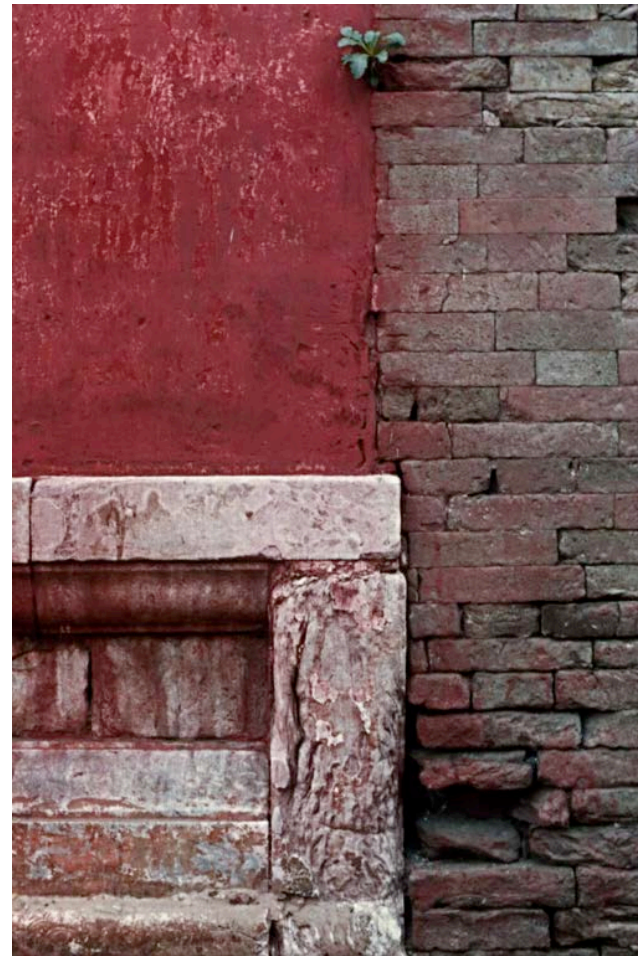# Texture

- What is texture?
- Texture analysis
- Deep Texture
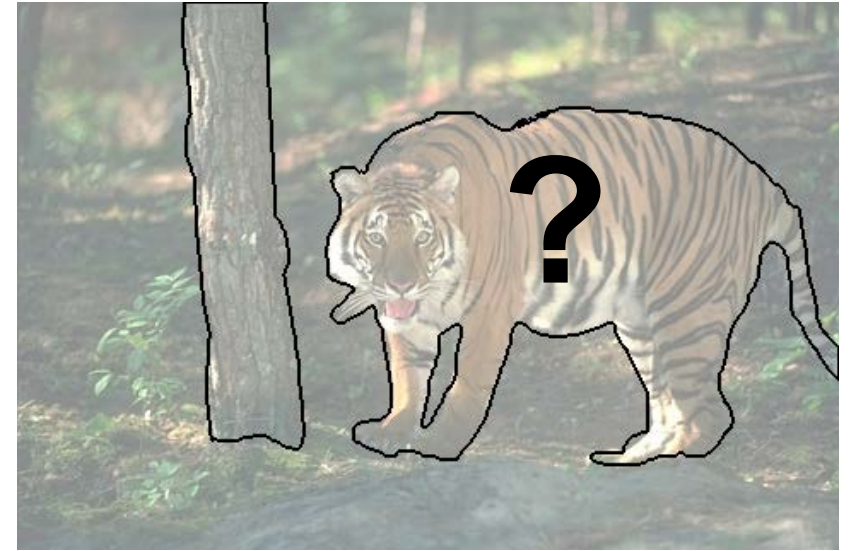
Reminder: Homogeneous or Not?

What is homogeneous in some parts of these images are the statistical properties, not the actual pixel values.
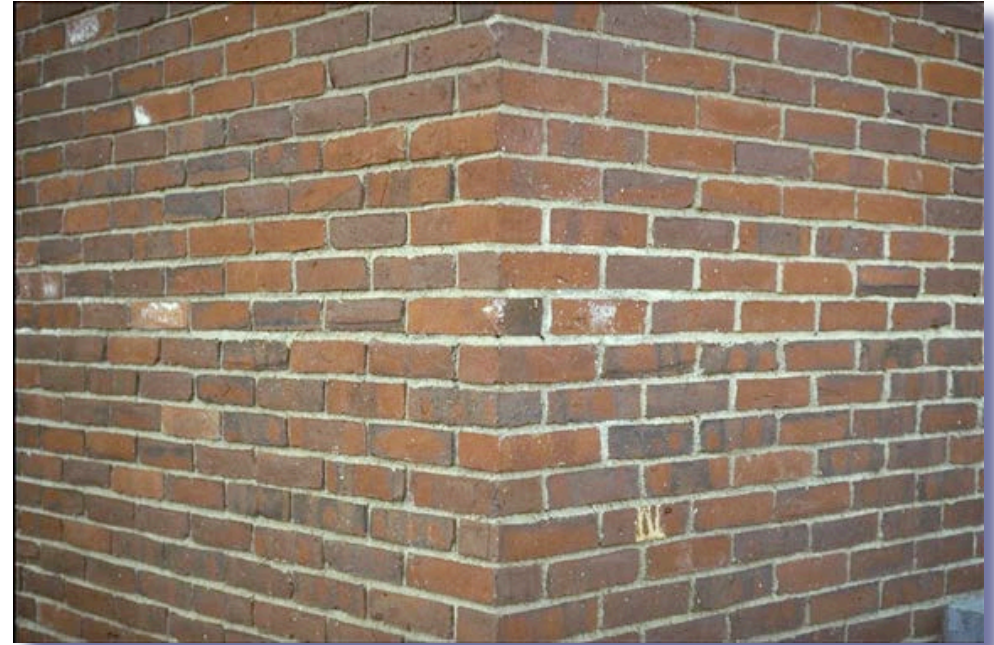
Texture-Based Segmentation

Ideally, we would like to:

- Assign to individual pixels whose texture is similar the same values to form a textural image.

- Evaluate homogeneity both in the original image and in the textural one.

Texture-Based Edges

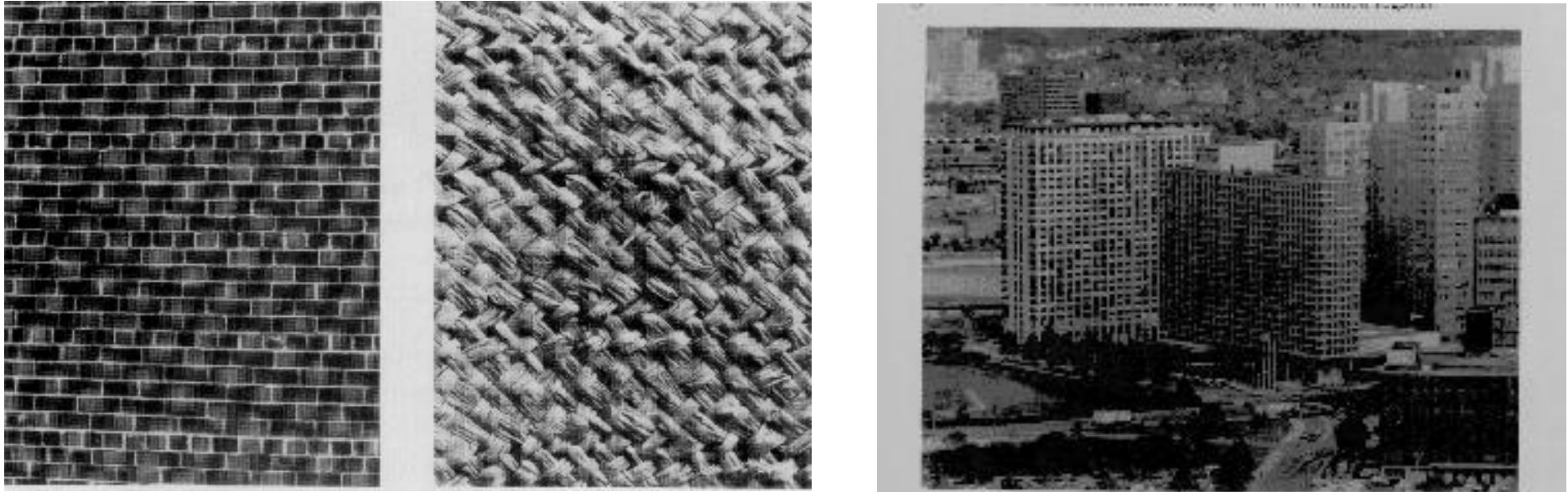Similarly, we would like to be able to find boundaries between textures.

What is Texture?

Repetition of a basic pattern:
- Structural
- Statistical
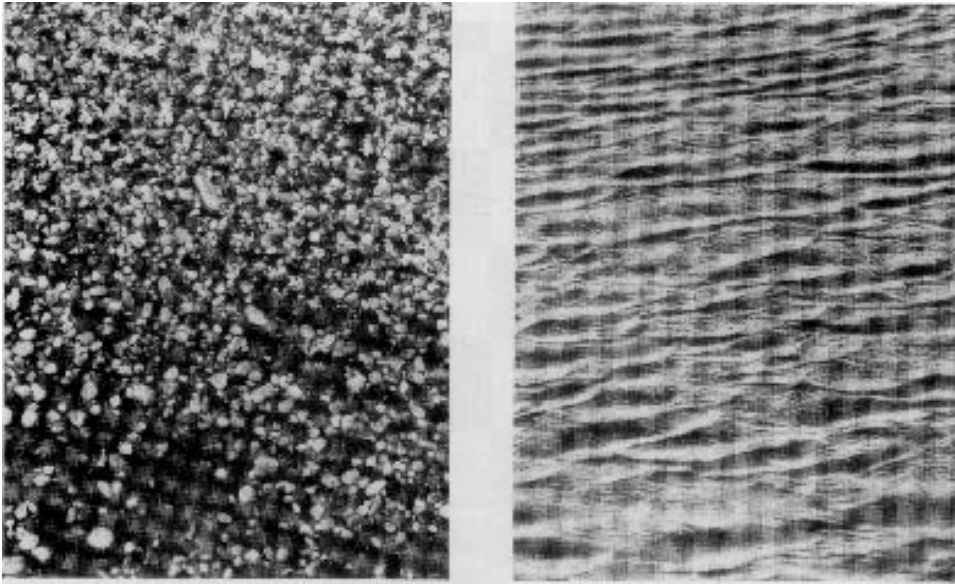→ Non local property, subject to distortions.

Structural Textures

Repetitive Texture Elements (Texels)

A texel represents the smallest graphical element in a two-dimensional texture that creates the impression of a textured surface.

Statistical Textures

Homogeneous Statistical Properties

Textured vs Smooth

A "featureless" surface can be regarded as the most elementary spatial texture:

- Microstructures define reflectance properties.
- They may be uniform or smoothly varying.

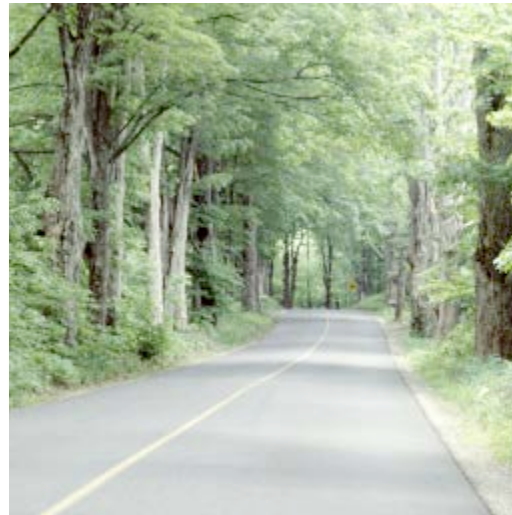→ Texture is a scale dependent phenomenon

Scale Dependence

At these two different scales, the texture seems very different.

Structural vs Statistical

- Segmenting out texels is difficult or impossible in most real images.

What are the fundamental texture primitives in this image?

- Numeric quantities or statistics that describe a texture can be computed from the gray levels or colors alone.

→ The statistical approach is less intuitive, but more effective in practice.
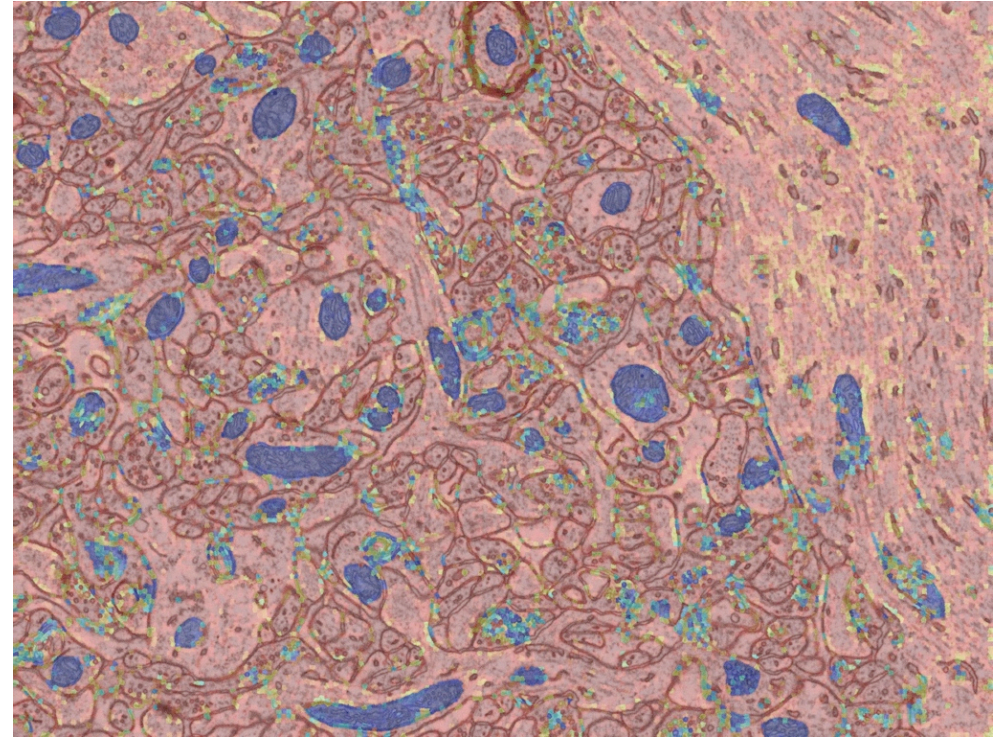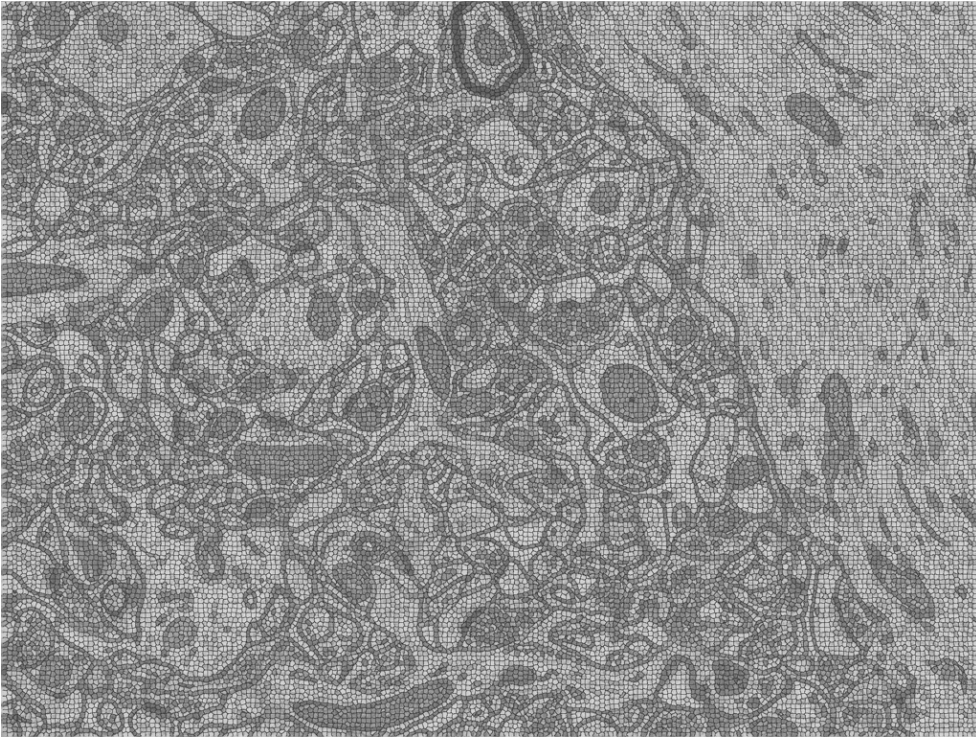
Creating Textural Images

Because texture is non-local, the texture of individual pixels must be estimated using neighborhoods that surround them:
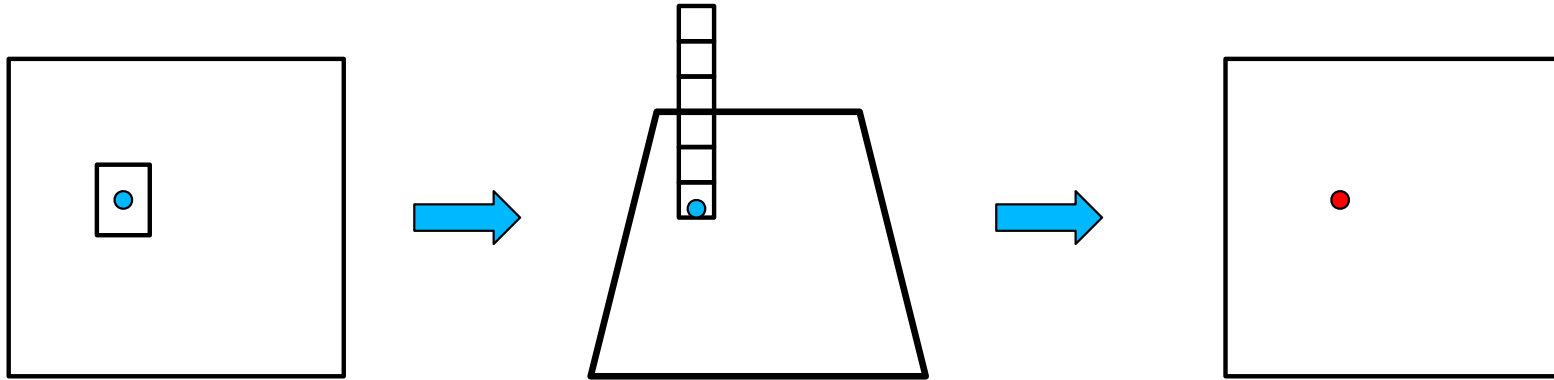
- For each pixel, compute a feature vector using either an image patch or a set of filters.

- Run a classification algorithm to assign a texture value to each pixel.

Reminder: Mitochondria

- Compute image statistics for each superpixel.
- Train a classifier to assign a probability to be within a mitochondria.

—> We used the super pixels to compute local statistics.

Textural Metrics

Spectral metrics:
- Texture is characterized by the properties of its Fourier transform.

Statistical Metrics:
- Texture is as statistical property of the pixels' intensity and color in a region.

Deep Net Metrics:
- They have now mostly superseded the others.
- They encompass the earlier concepts.

Reminder: Discrete Fourier Transform

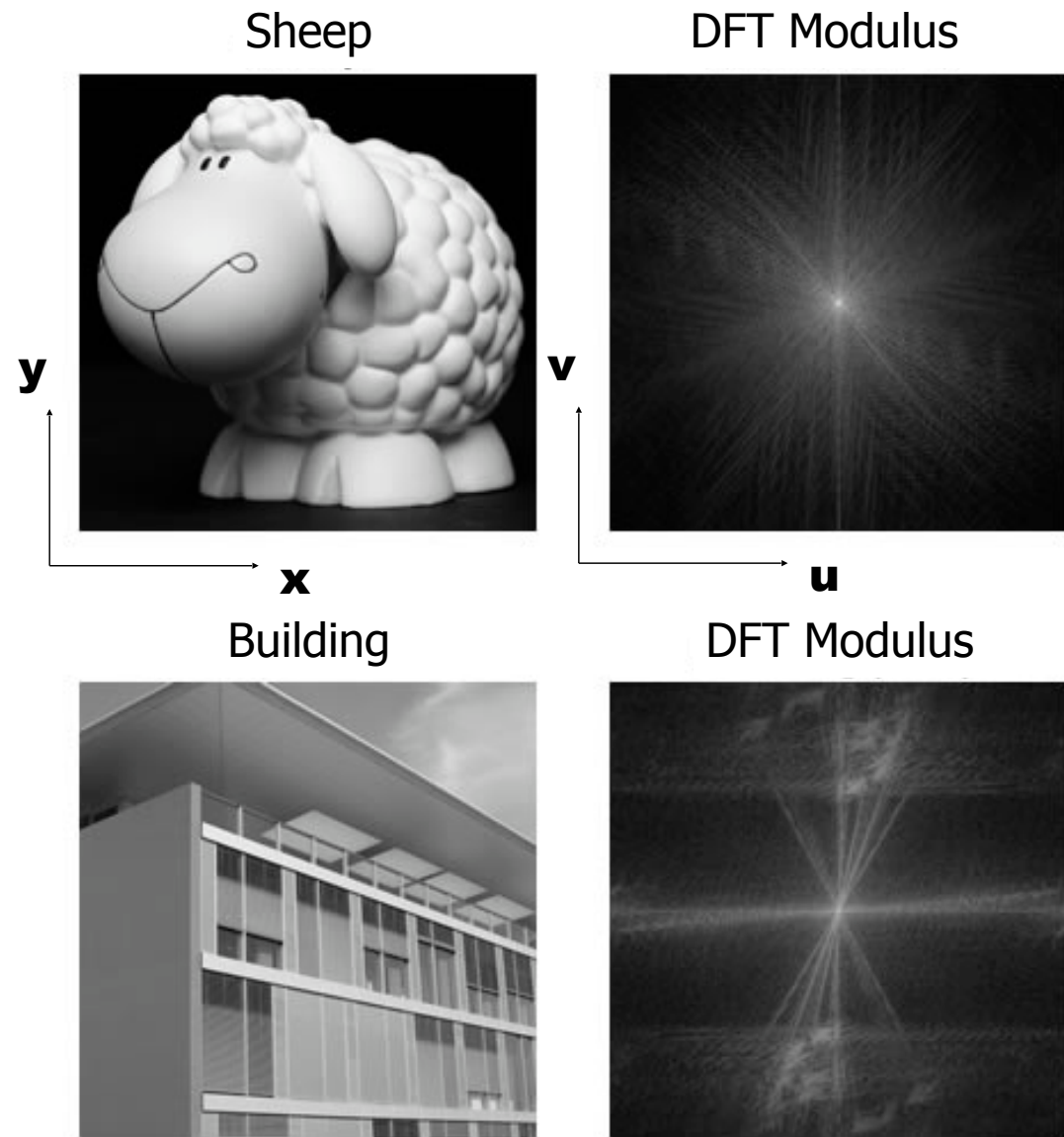$$F(\mu, \nu) = \frac{1}{\sqrt{M*N}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-2i\pi(\mu x/M + \nu y/N)}$$

$$f(x,y) = \frac{1}{\sqrt{M*N}} \sum_{\mu=0}^{M-1} \sum_{\nu=0}^{N-1} F(\mu,\nu) e^{+2i\pi(\mu x/M + \nu y/N)}$$

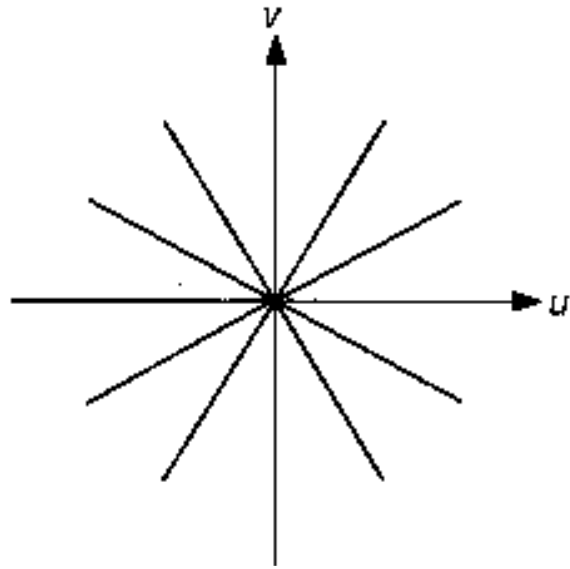The DFT is the discrete equivalent of the 2D Fourier transform:
- The 2D function f is written as a sum of sinusoids.
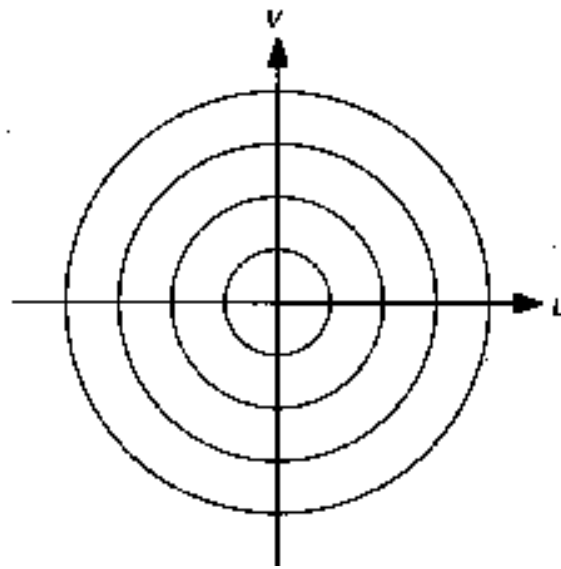- The DFT of f convolved with g is the product of their DFTs.

Spectral Analysis

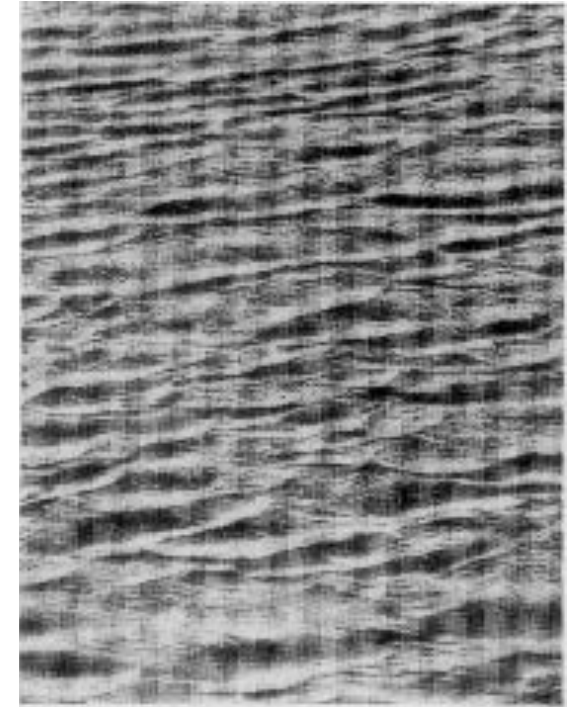Lines in the DFT modulus images capture the main orientations in the image.

Sheep

DFT Modulus

y

x

v

u

Building

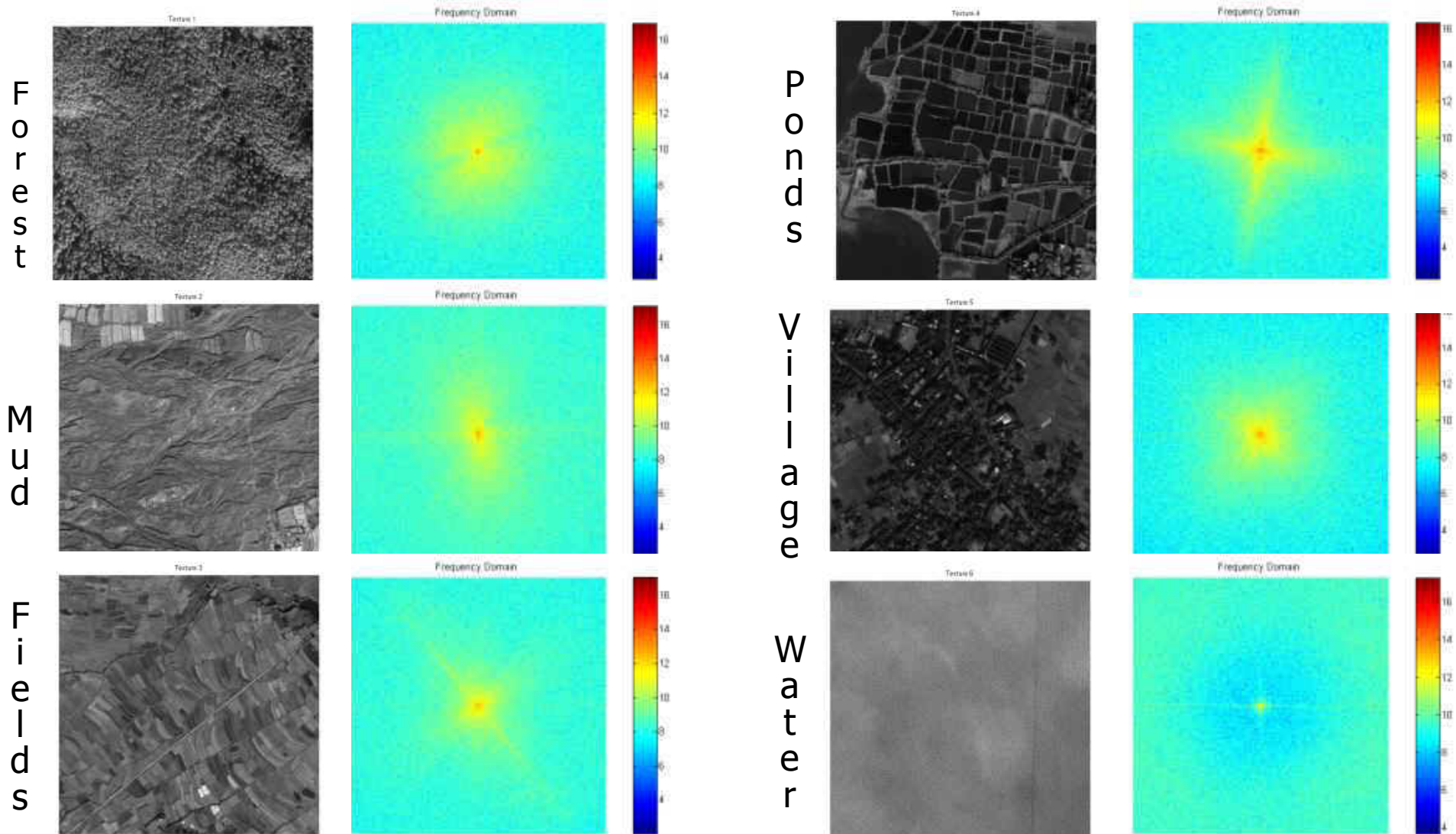DFT Modulus

Texture Analysis

Angular bins
in |DFT| image

Radial bins
in |DFT| image

Angular and radial bins in the Fourier domain capture the directionality and fluctuation speed of an image texture, respectively.

Fourier Texture Classification

- For some types of textures, the Fourier spectra are easily distinguishable.
- A classifier can be trained to tell them appart.
- However, one must have the same texture in the whole image patch.

Limitations

- DFT on small patches is subject to severe boundary effects.

- Only applicable if texture is uniform over large areas.

- Results can be improved by using wavelets instead, but only up to a point.

—> More local metrics are required.

Statistical Metrics

First order gray-level statistics:
- Statistics of single pixels in terms of histograms.
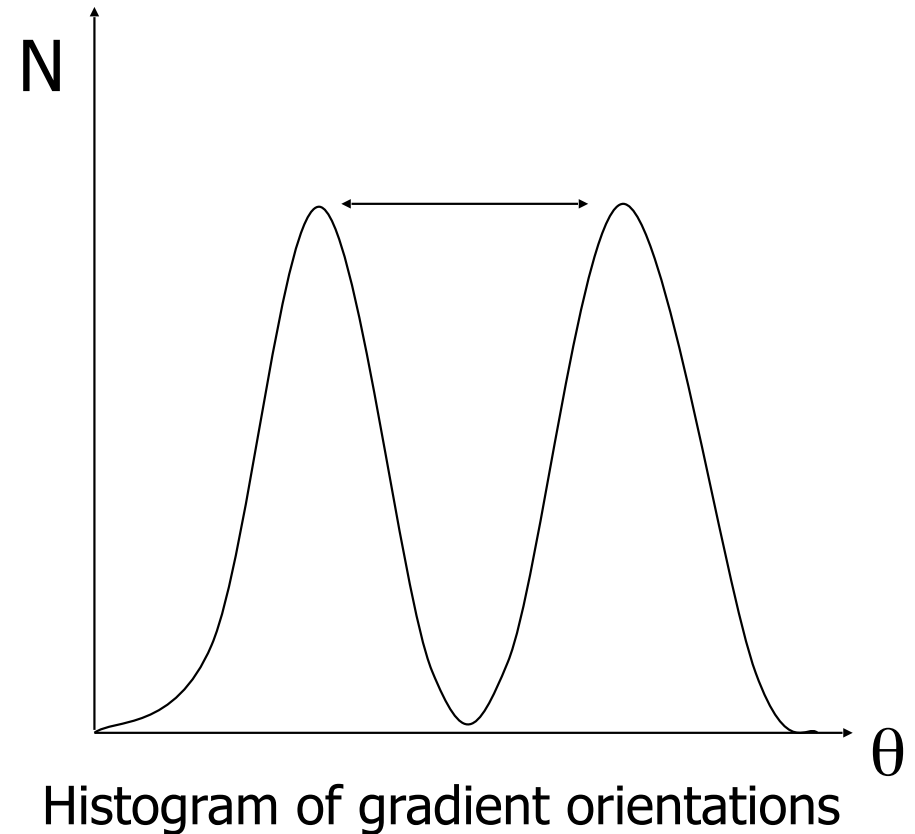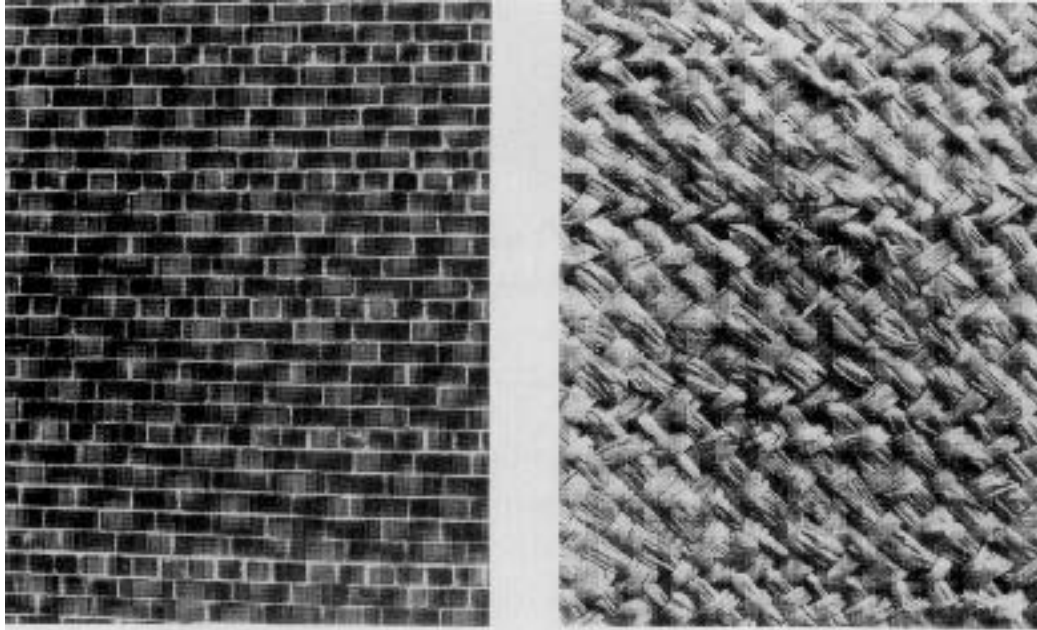- Insensitive to neighborhood relationships.

Second order gray-level statistics:
- Statistics of pixel pairs.

Filter-based measures:
- Statistics of whole neighborhoods.

Simple First Order Measure

Histogram of gradient orientations

The orientation histogram gives a clue to the orientation of the underlying plane.

More First Order Measures

Edge Density and Direction:

- Edge detection as a first step in texture analysis.
- The number of edge pixels in a fixed-size region tells us how busy that region is.
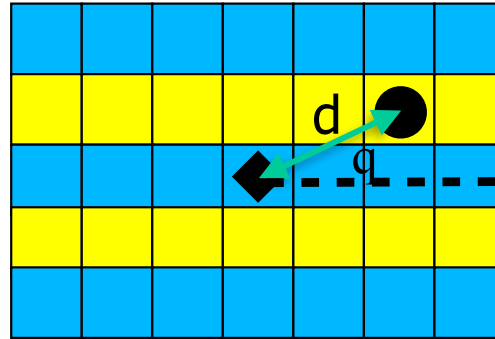- The directions of the edges also help characterize the texture

Edgeness per unit area:

- { p : gradient_magnitude(p) ≥ threshold}| / N where N is the unit area or region.

Edge magnitude and direction histograms:

- { H_G, H_t}

Second Order Measures

Histogram of the co-occurrence of particular intensity values
  in the image.

- Specified in terms of geometric relationships between pixel
  pairs:
  - Distance
  - Orientation

- $P(i,j,d,\theta)$ Frequency with which a pixel with value j occurs at
  distance d and orientation θ from a pixel with value i.

Simple Example

If
$$I = \begin{bmatrix} 0 & 1 & 2 & 3 & 1 \\ 2 & 1 & 3 & 1 & 1 \\ 0 & 0 & 2 & 2 & 1 \\ 1 & 2 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix},$$

then
$$H = \begin{bmatrix} 4 & 1 & 1 & 2 \\ 0 & 1 & 2 & 1 \\ 1 & 2 & 1 & 1 \\ 0 & 3 & 0 & 0 \end{bmatrix},$$

and $P(l,m,1,0) = \dfrac{H(l,m)}{20}$.

Co-Occurrence Matrix

No need to distinguish between

$$P(m, l, \Delta i, \Delta j)$$

and

$$P(l, m, \Delta i, \Delta j)$$

$\rightarrow$ Co-Occurrence matrix C:

$$C = H + H^T$$

Second Order Measures

Contrast:
$$\sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2$$

Dissimilarity:
$$\sum_{i,j=0}^{N-1} P_{i,j}|i-j|$$

Homogeneity:
$$\sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2}$$

Angular Second Moment (Energy, Uniformity):
$$\sum_{i,j=0}^{N-1} P_{i,j}^2$$

Entropy:
$$\sum_{i,j=0}^{N-1} P_{i,j}(-\ln P_{i,j})$$

and many more ....

Landsat Image

r,g,b = ETM+ 4,3,2 (nir,red,green)

112.22W, 51.29N

111.16W, 50.85N
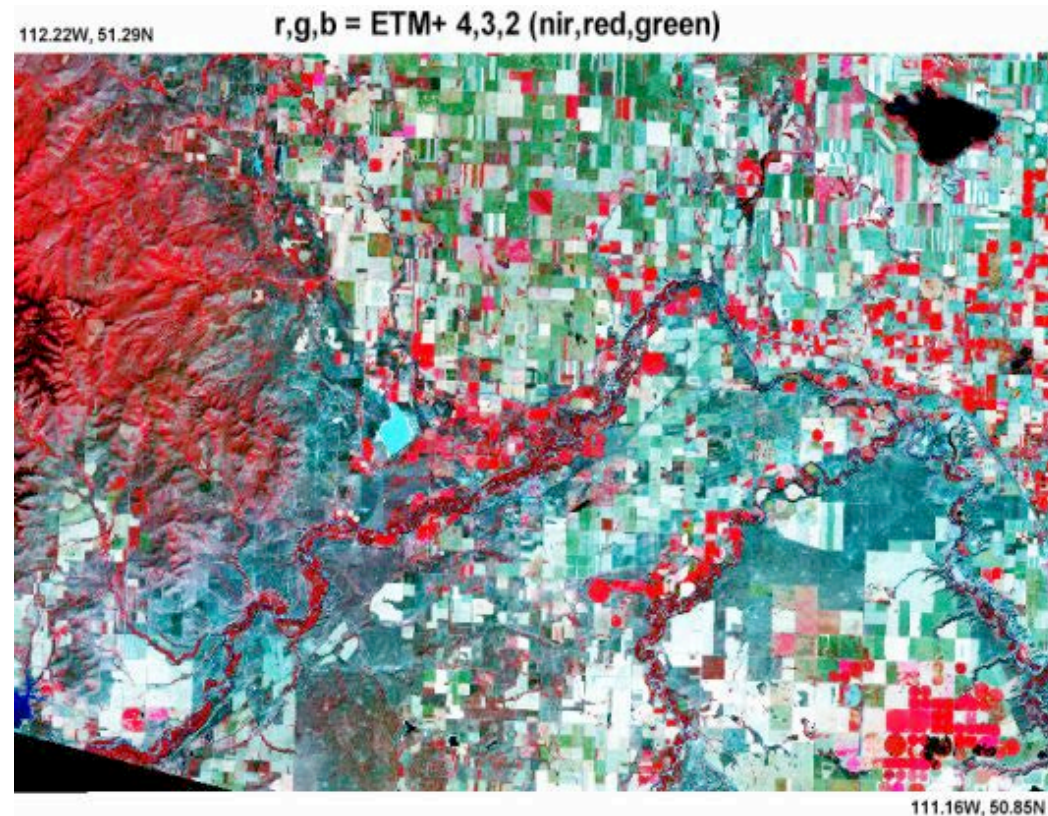
The image is excerpted from Path 41, Row 25 of Landsat 7 ETM+, dated 4 September 1999. This is an area in the Rocky Mountain Foothills near Waterton National Park, Alberta. The western edge of the image contains steep slopes and deep valleys. To the east is both grassland and annual crops, mostly grains. The eastern area is bisected by numerous small streams.

Full Resolution
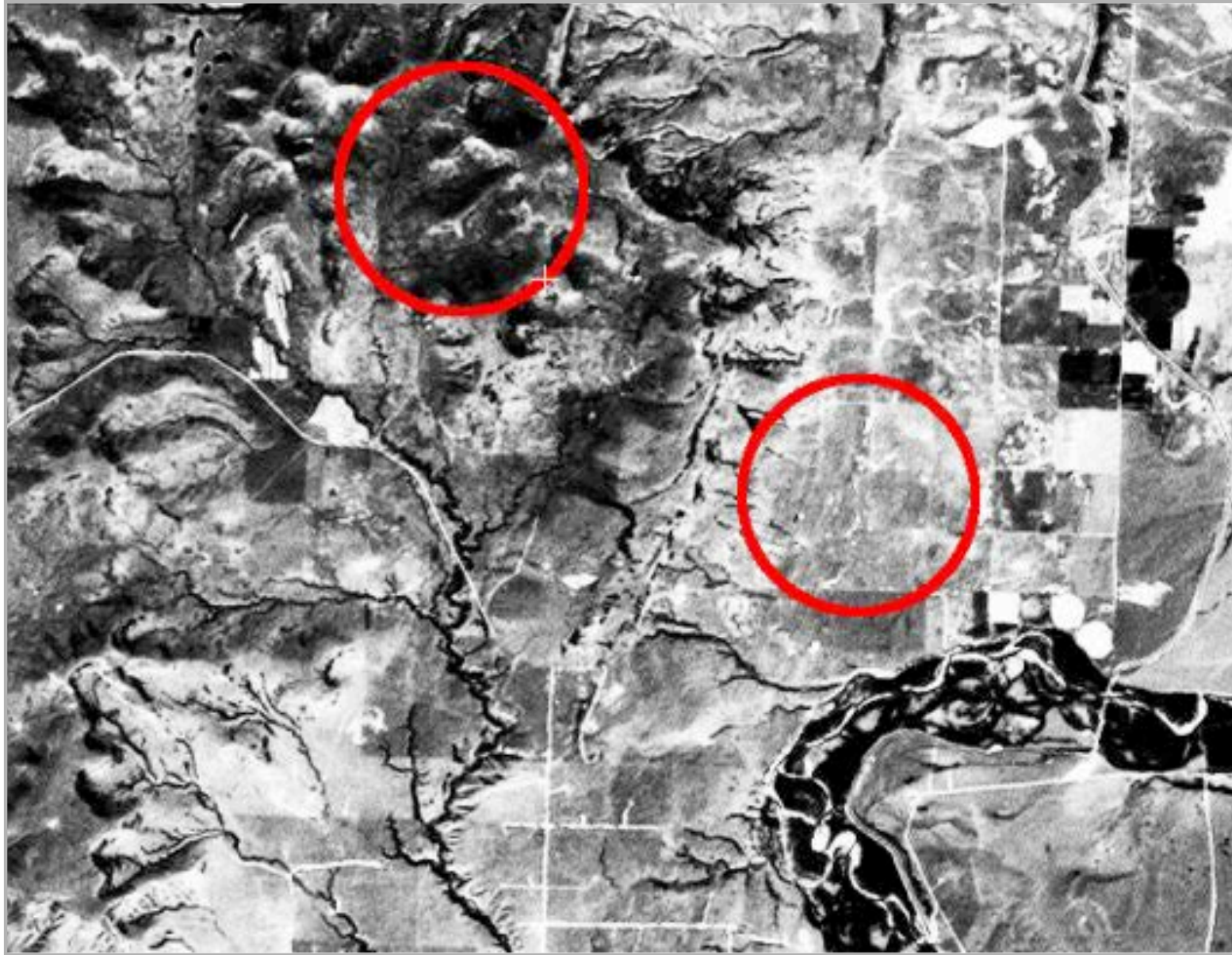
Let us consider two areas, one in the hills, the other in the plain.

Using Second Order Measures
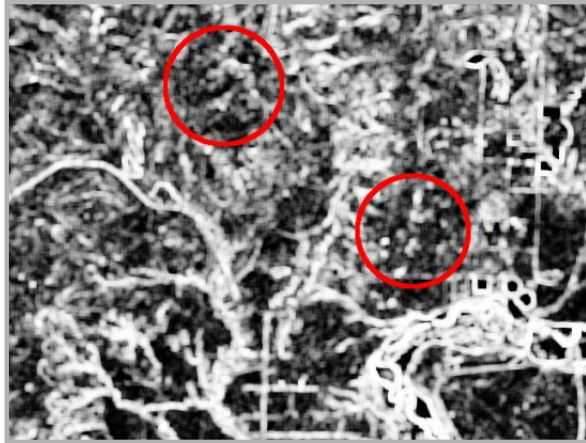
Contrast | Dissimilarity | Homogeneity
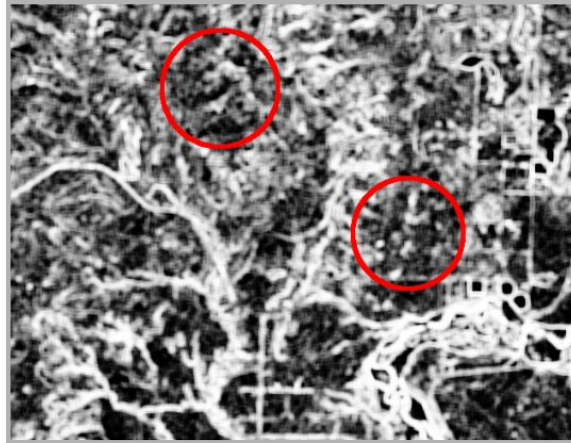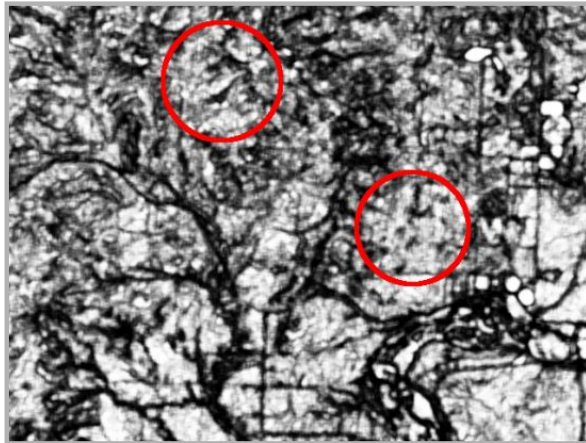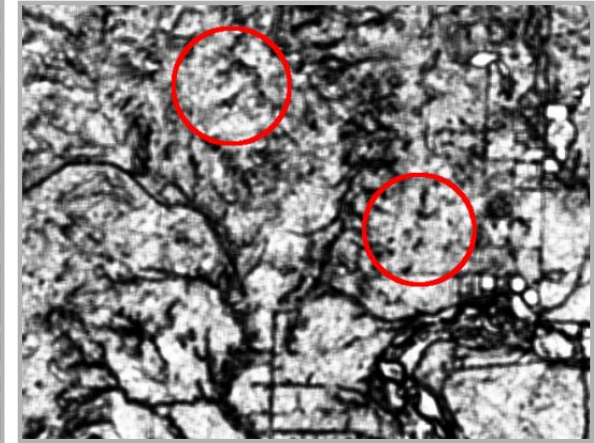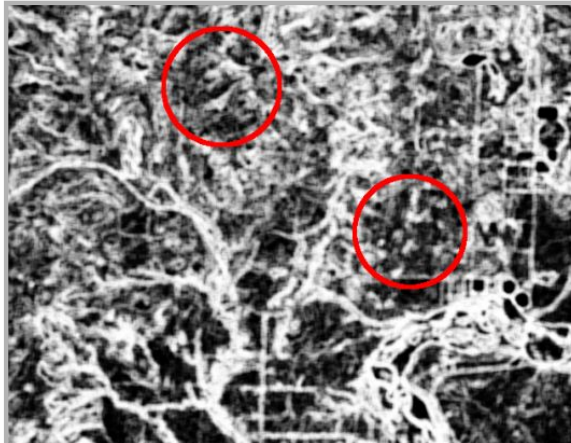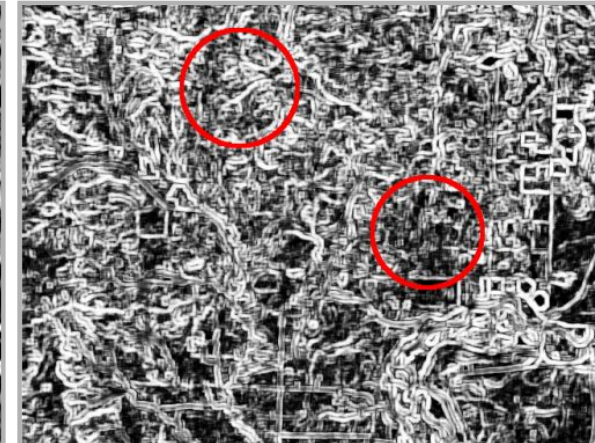
ASM | Entropy | Correlation

The various measures within the two areas are sufficiently different for a classifier to easily distinguish them.

Classification

Used to identify eight terrain classes:
- Old residential
- New residential
- Urban
- Lake
- Swamp
- Scrub
- Wood

Texture Boundary Detection
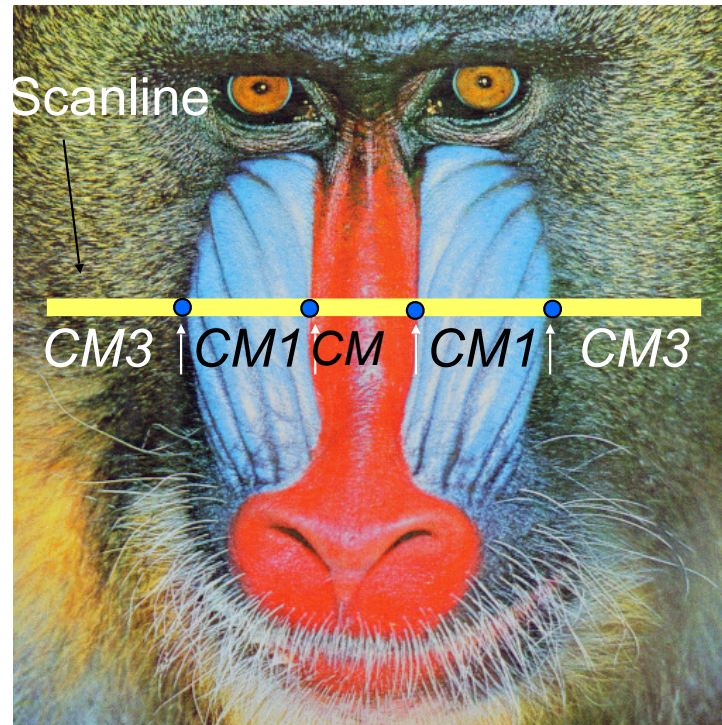
- A Markovian model of order 1 defines the probability of each pixel intensity given the previous pixel intensity along a line.

- These probabilities can be represented by a coocurrence matrix for which each element CM(i,j) gives the probability of having adjacent pixels of intensity i and j.

- Therefore, coocurrence matrices can be used to detect the boundary between two different textures

Boundary Tracking

Approximate projection of the model using the pose in previous frame …

It is used to define scanlines to search for body/background boundaries …

Using traditional gradient maxima

Using concurrence matrices

Real Time Tracking

Parameter Choices

Using co-occurence matrices requires choosing:

- window size,
- direction of offset,
- offset distance,
- what channels to use,
- what measures to use.

How do we choose these parameters?

- Critical question for **all** statistical texture methods.
- Can be addressed using Machine Learning.

Filter Based Measures
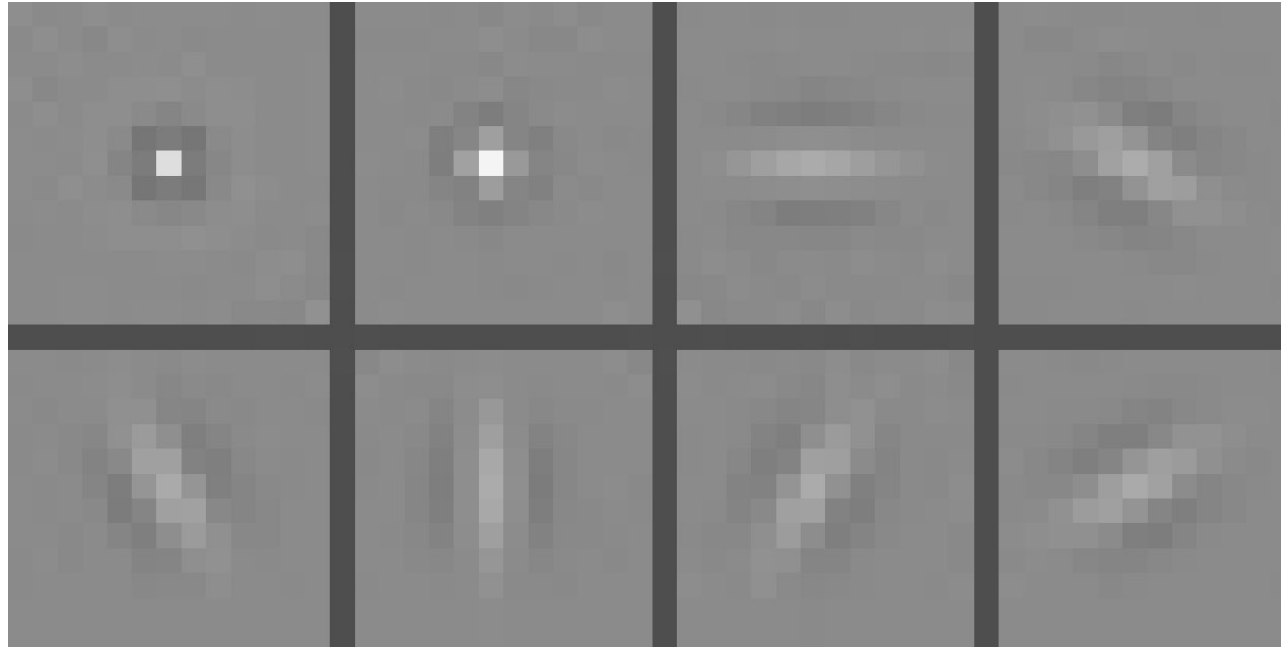
Represent image textures using the responses of a collection of filters.

- An appropriate filter bank will extract useful information such as spots and edges.

- Traditionally one or two spot filters and several oriented bar filters.

Gaussian Filter Derivatives
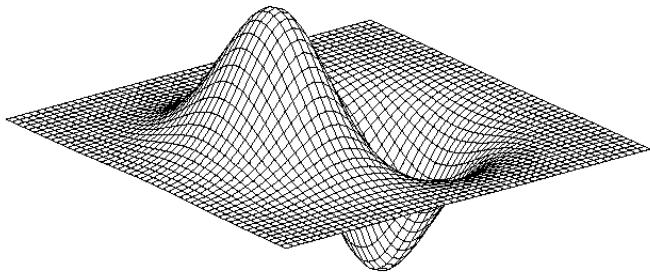
Gaussian Derivative   x and y derivatives at different scales
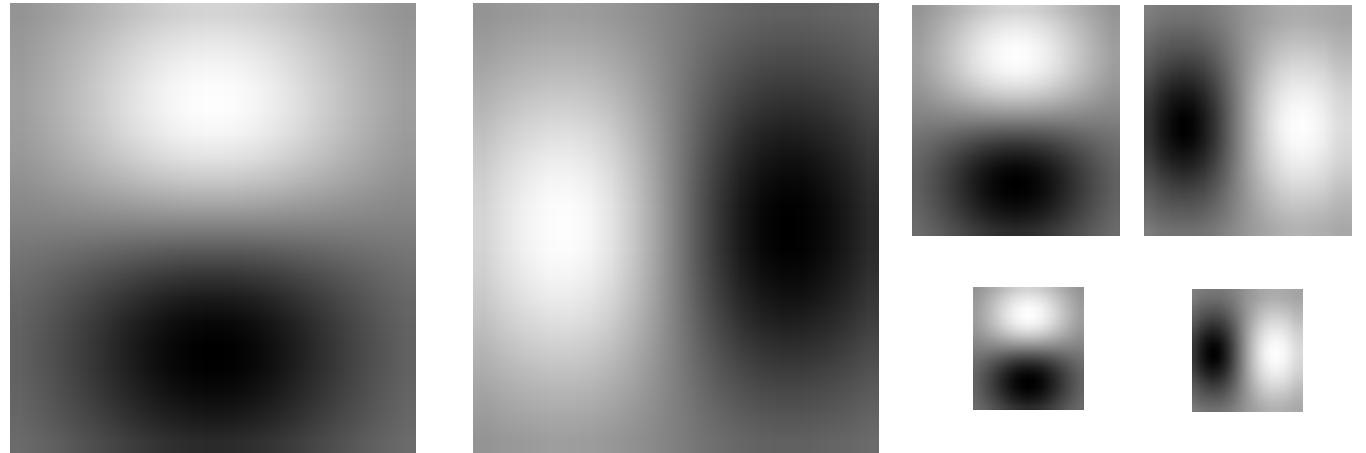
These filters respond to horizontal and vertical edges.

Horizontal and Vertical Structures

vertical filter

Squared responses

Spatially blurred

image

horizontal filter

Threshold squared, blurred responses, then categorize texture based on those two bits
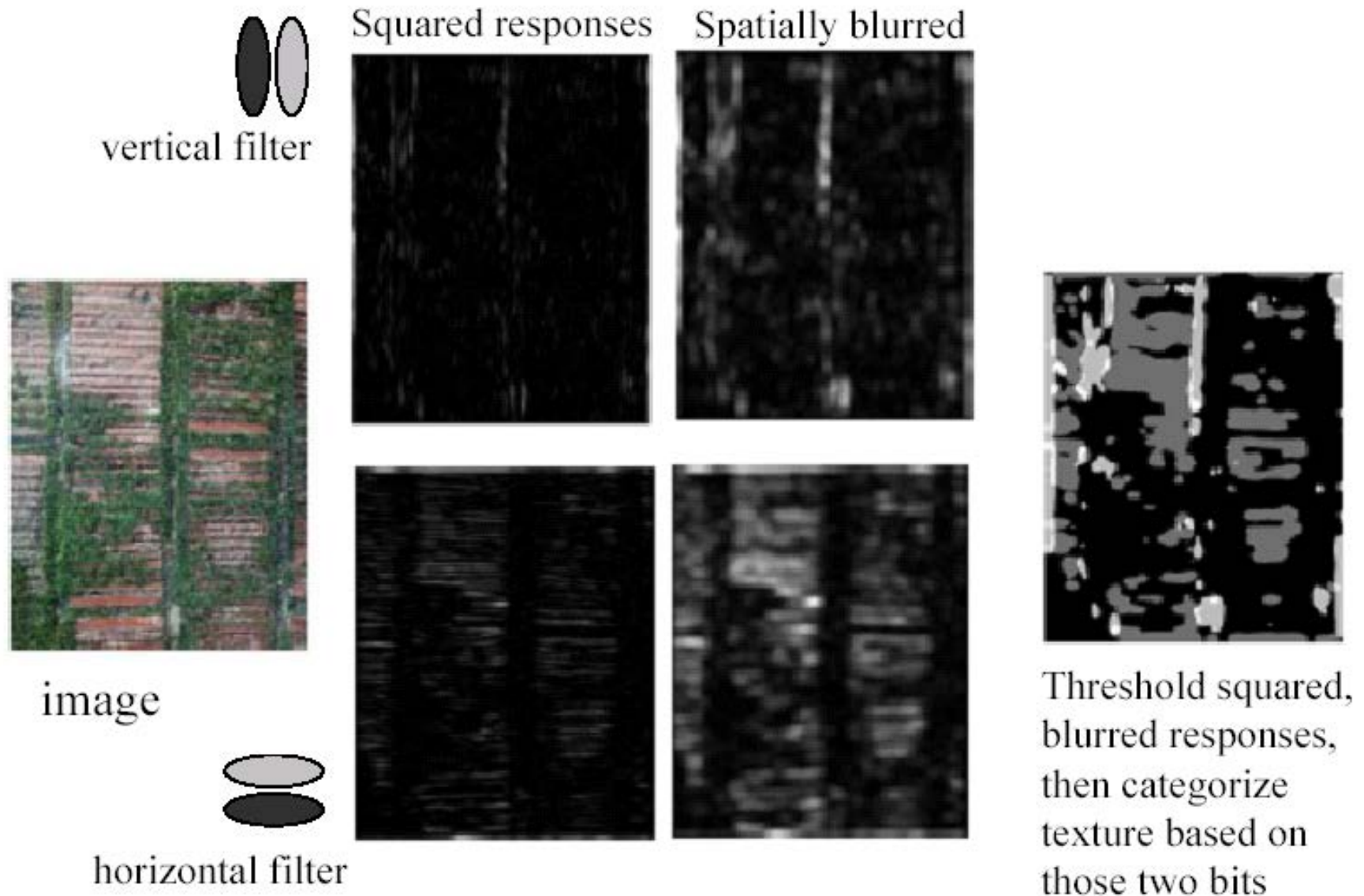
Oriented Filters

$$\frac{\partial I}{\partial \theta} = cos(\theta)\frac{\partial I}{\partial x} + sin(\theta)\frac{\partial I}{\partial y}$$
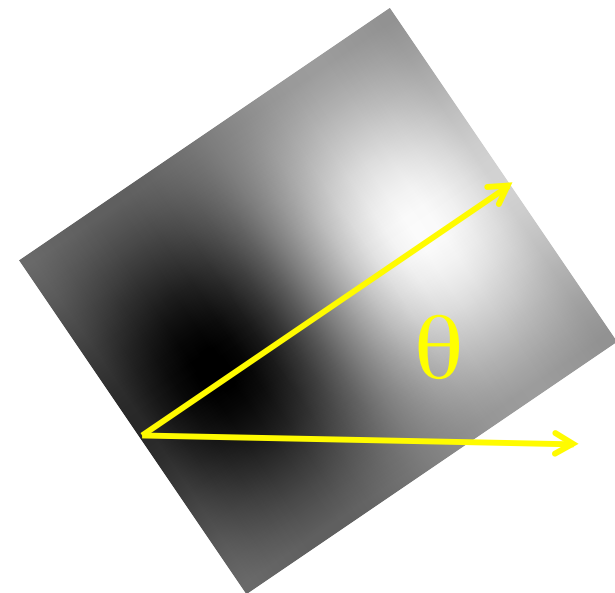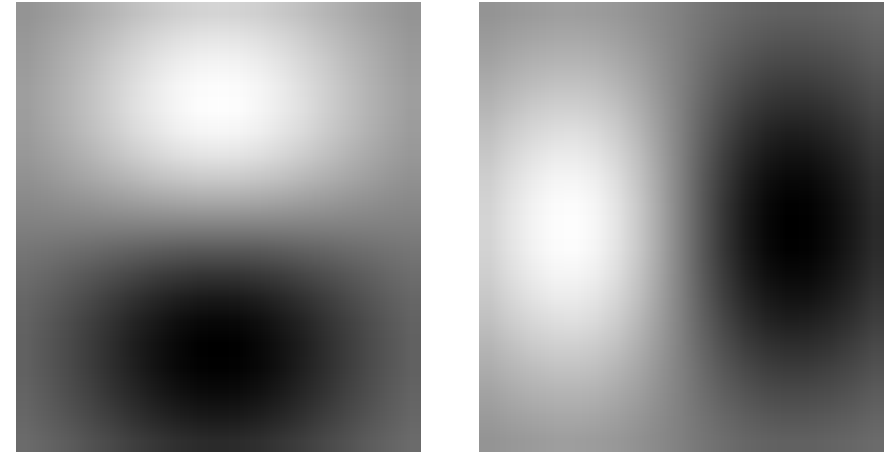
Directional Gradients
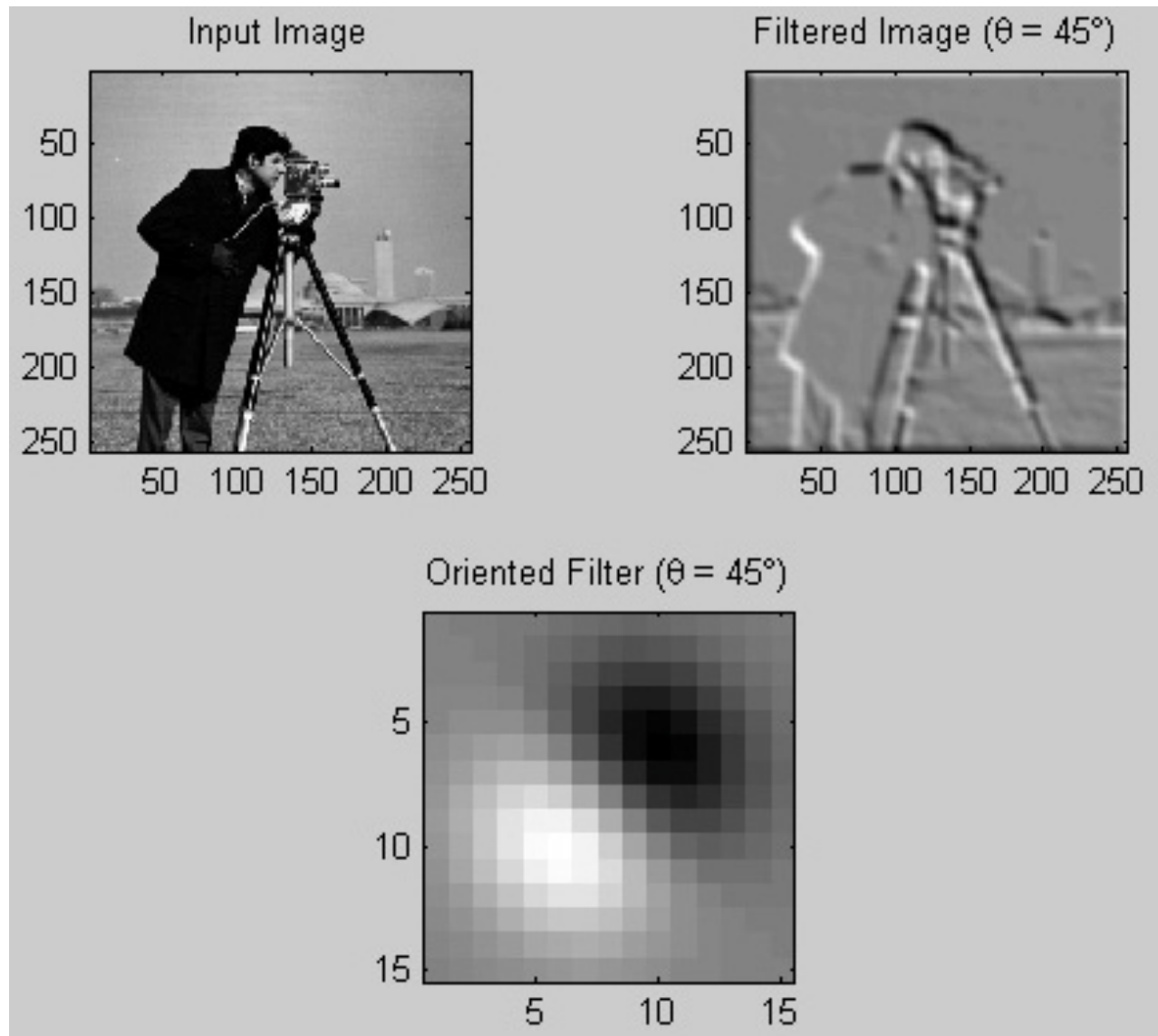
Oriented filters respond to edges in a specific direction.

Higher Order Derivatives
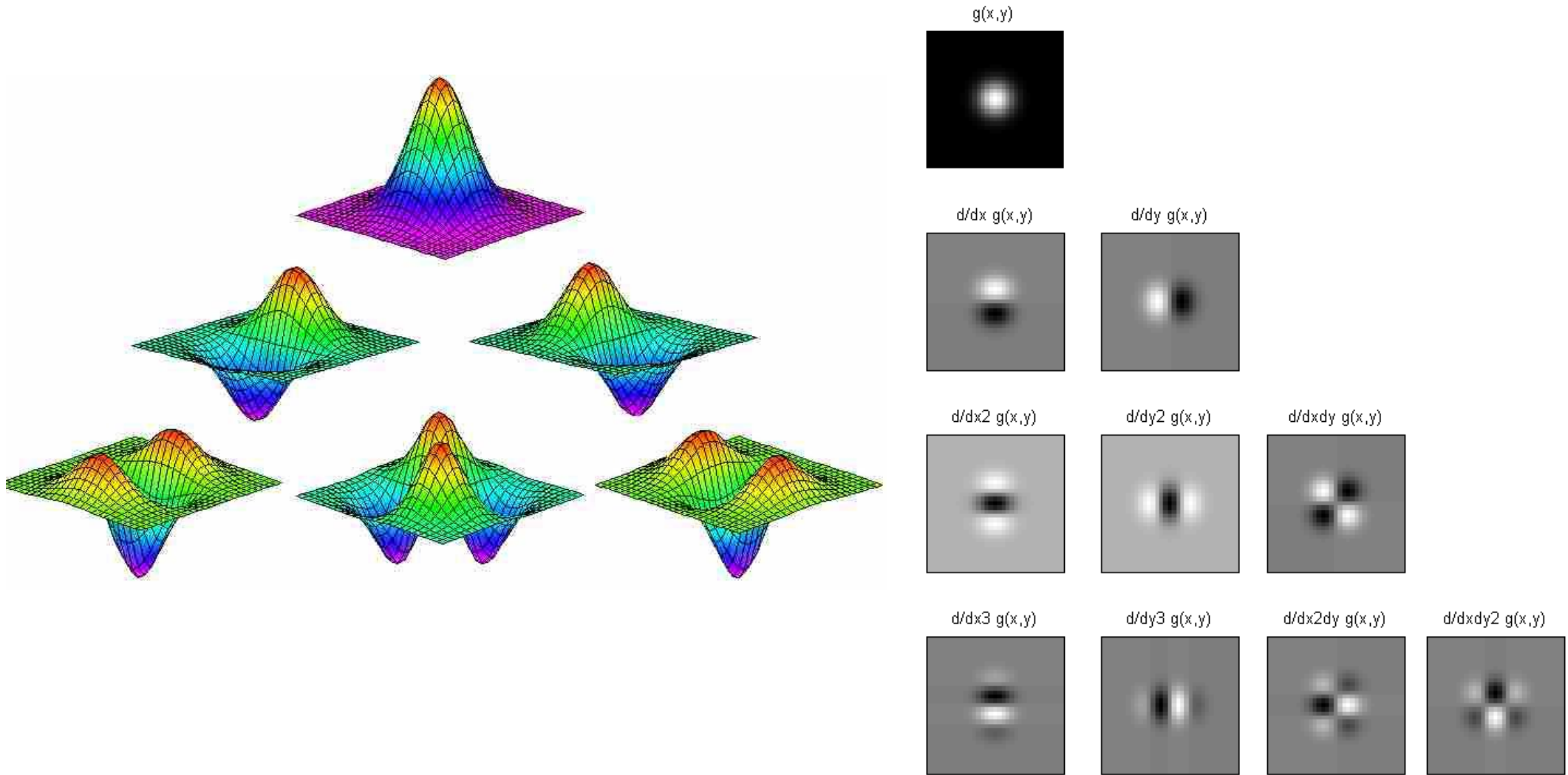
Higher-order derivatives of the Gaussian filters can be used to compute higher-order image derivatives.

Filter Bank
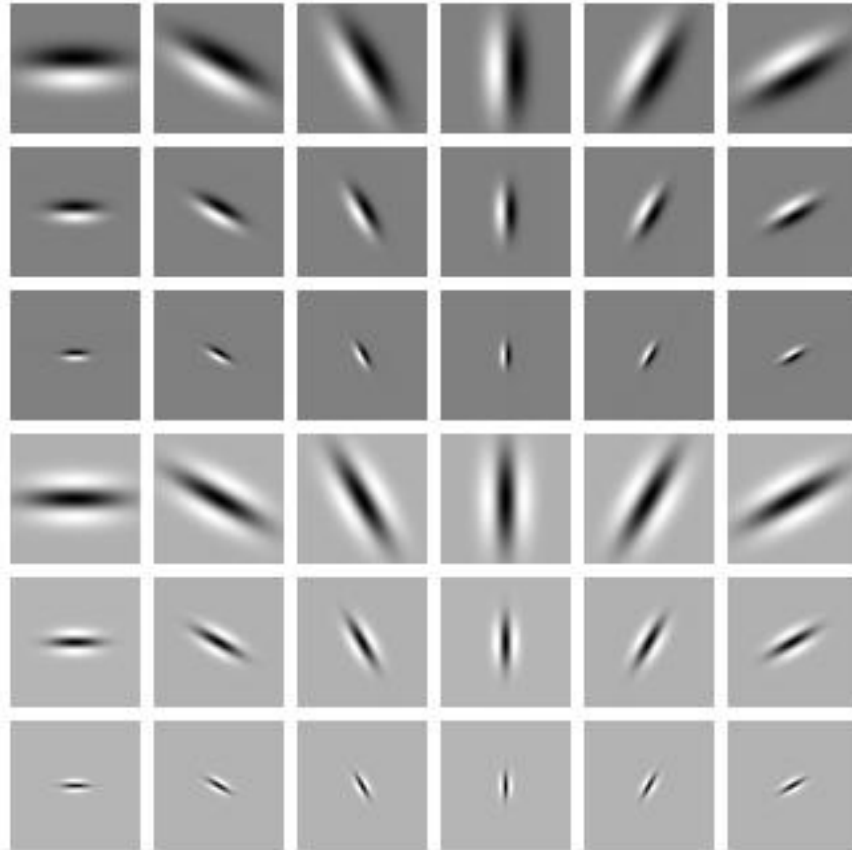
- Different scales.
- Different orientations.
- Derivatives order 0, 1, 2 ..
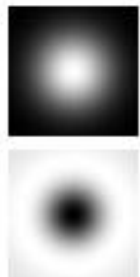
—> For every image pixel, compute a vector of responses to each filter.

Filter Responses: Small Scales
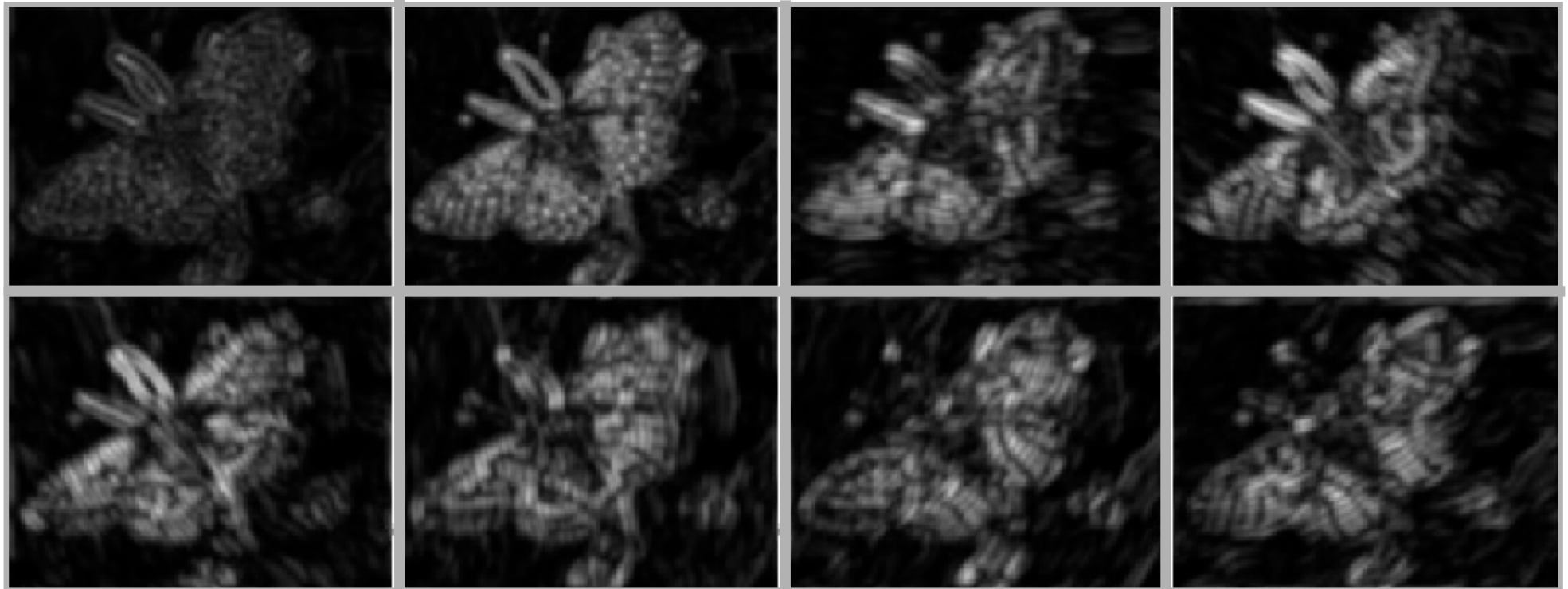
Gaussian filters with a small σ. Capture local details.

Filter Responses: Large Scales
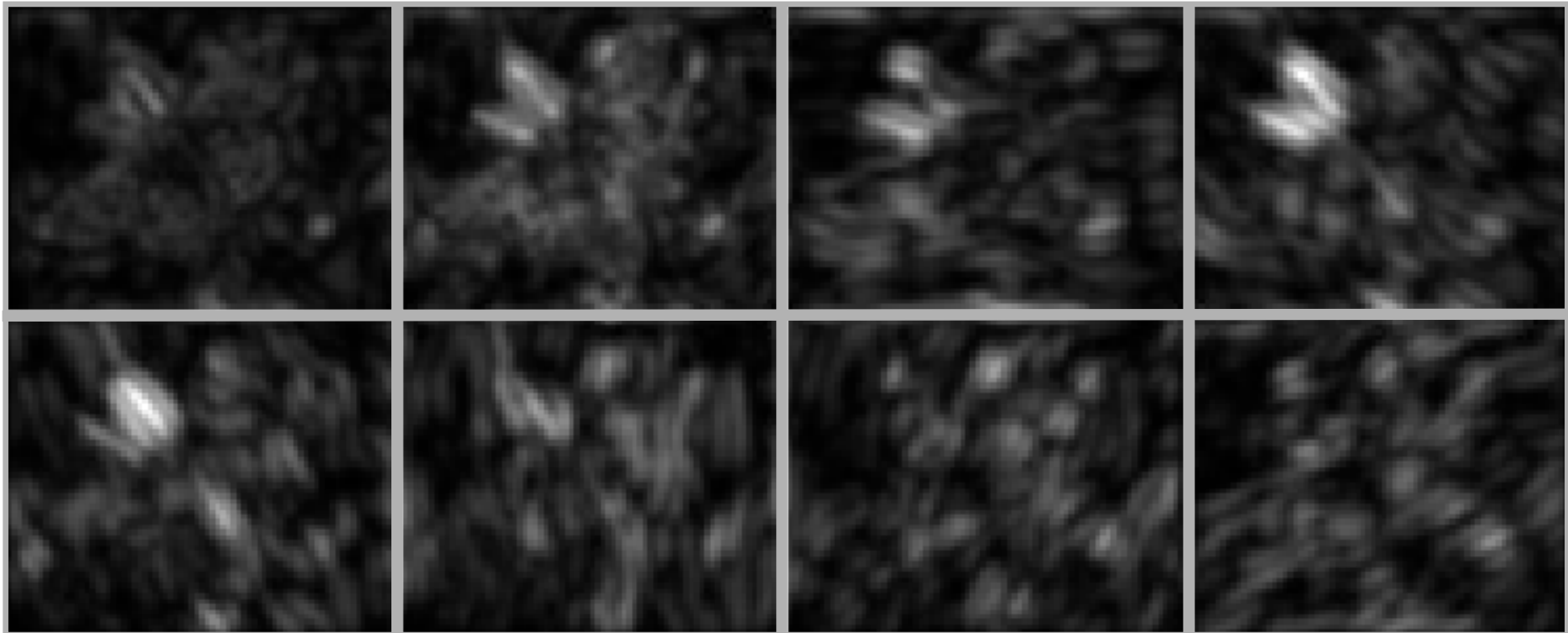
Gaussian filters with a large σ. Capture larger details.

Gabor Filters

Gabor filters are the products of a Gaussian filter with oriented sinusoids. They come in pairs, each consisting of a symmetric filter and an anti-symmetric filter:

$$G_{\text{sym}}(x, y) = \cos(k_x x + k_y y) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

$$G_{\text{asym}}(x, y) = \sin(k_x x + k_y y) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

where k_x and k_y determine the spatial frequency and the orientation of the filter and σ determines the scale.

→ A filter bank is formed by varying the frequency, the scale, and the filter orientation
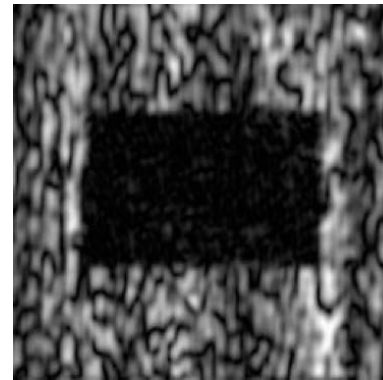
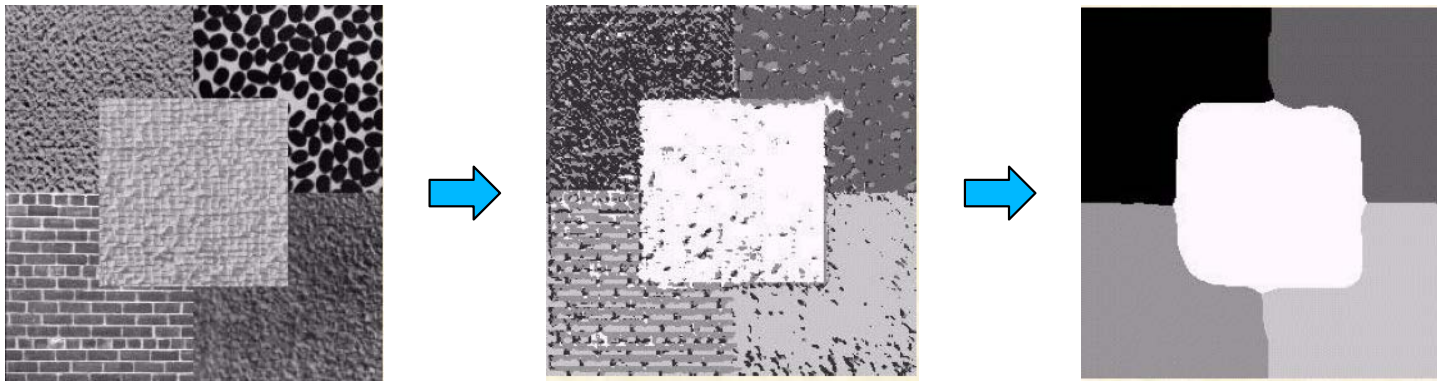# Vertical Derivatives
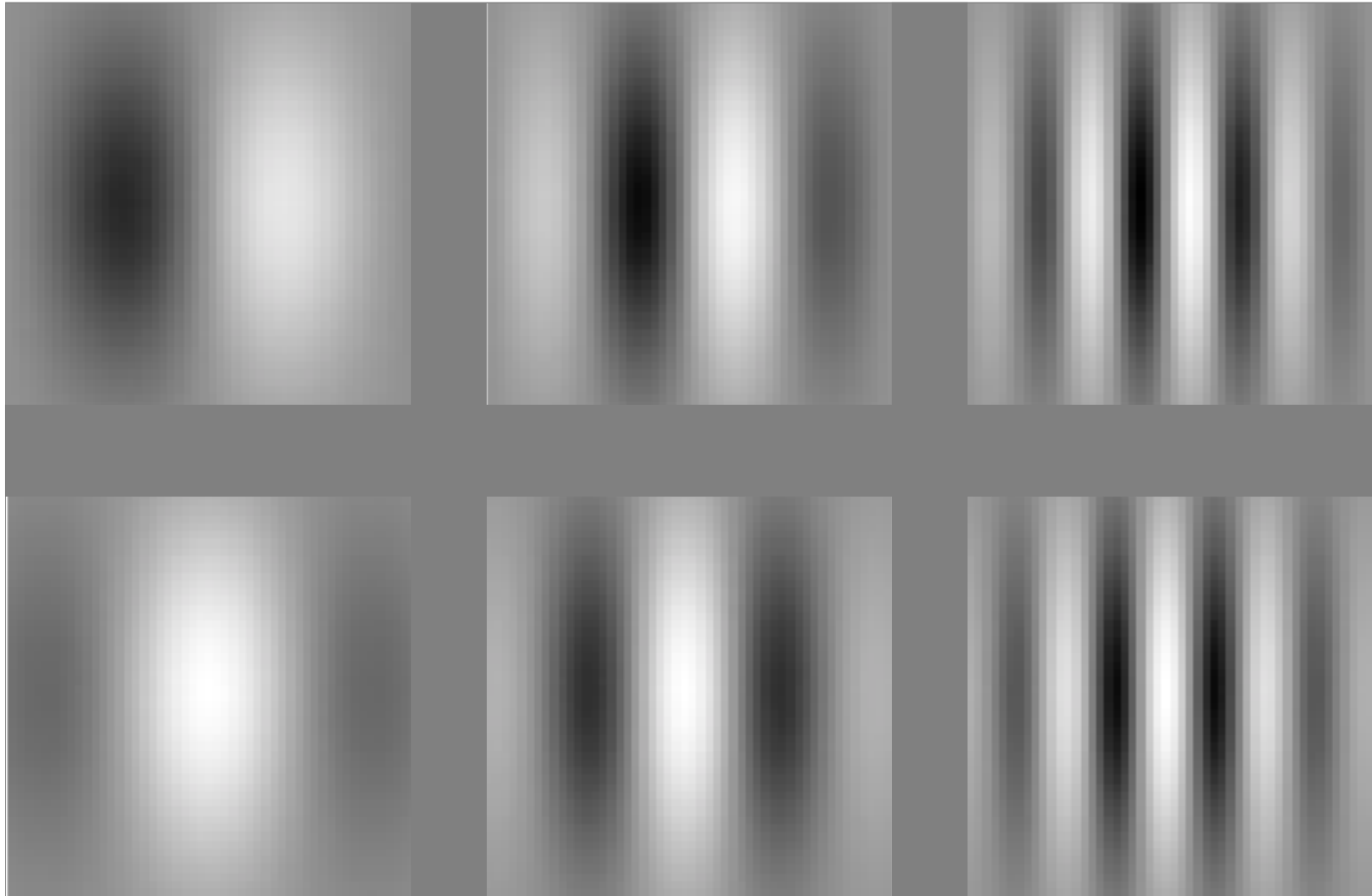
Gabor Responses
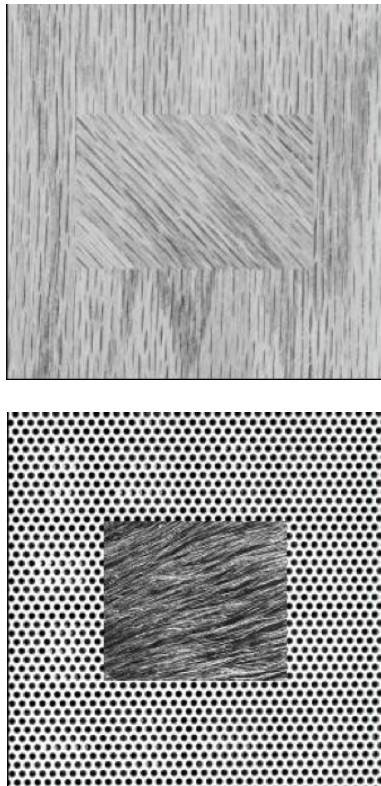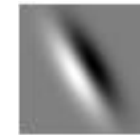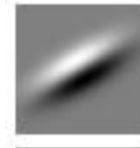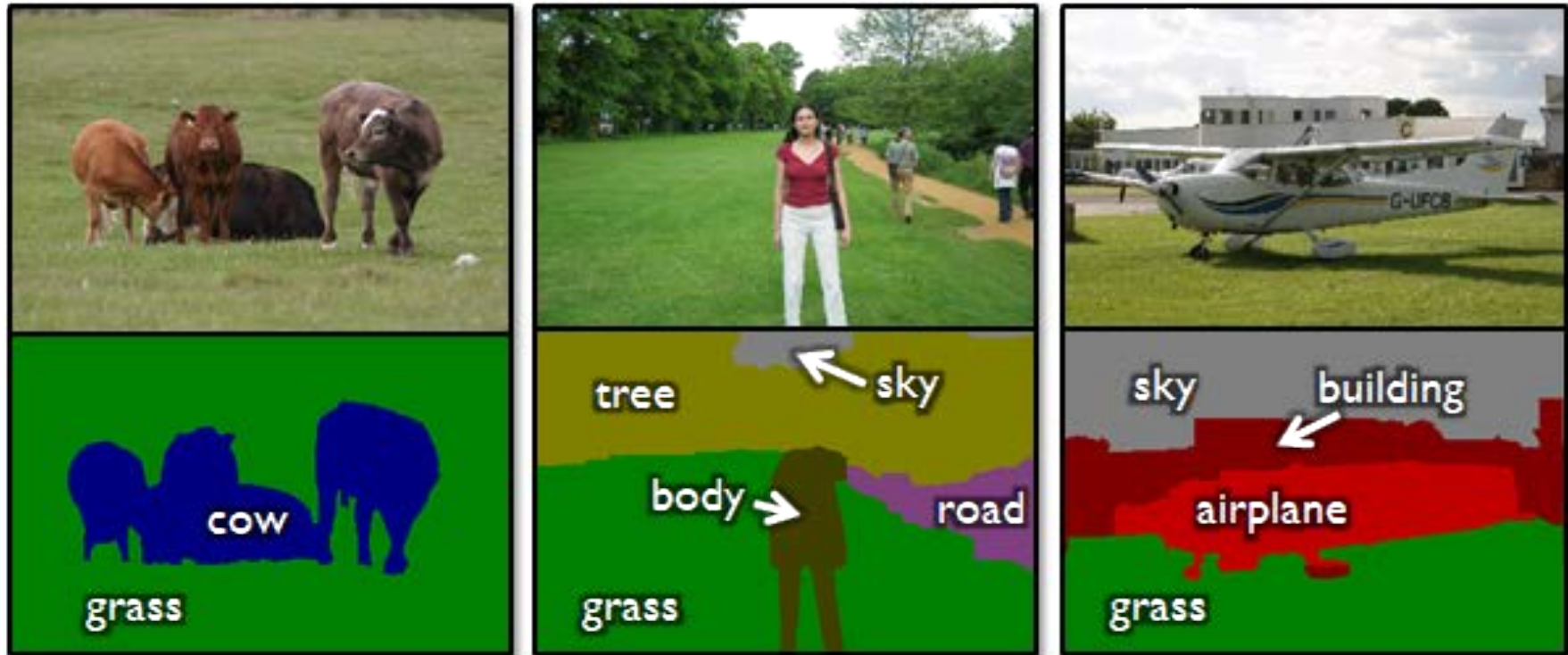
Filters:

Images:

Responses:
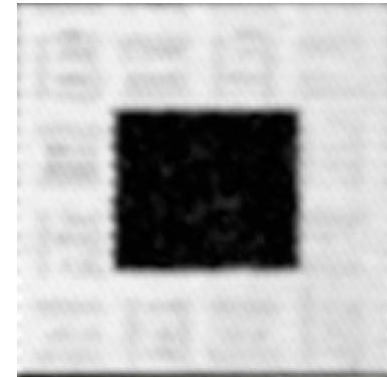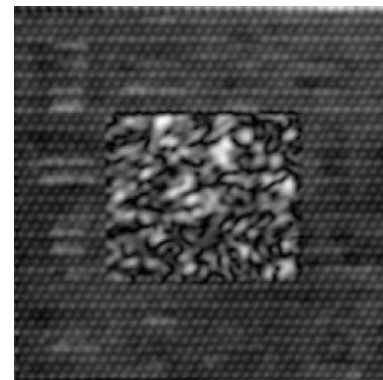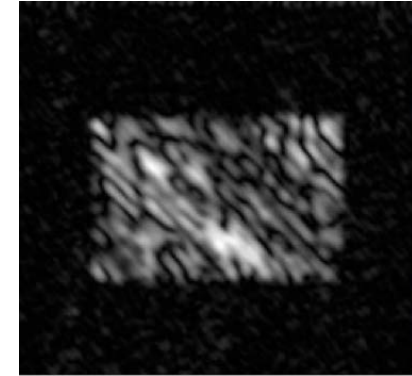
GABOR FILTER CHARACTERISTICS

- Respond strongly at points in an image where there are components that locally have a particular spatial frequency and orientation.

- In theory, by applying a very large number of Gabor filters at different scales, orientations and spatial frequencies, one can analyze an image into a detailed local description.

- In practice, it is not known how many filters, at what scale, frequencies, and orientations, to use. This tends to be application dependent.

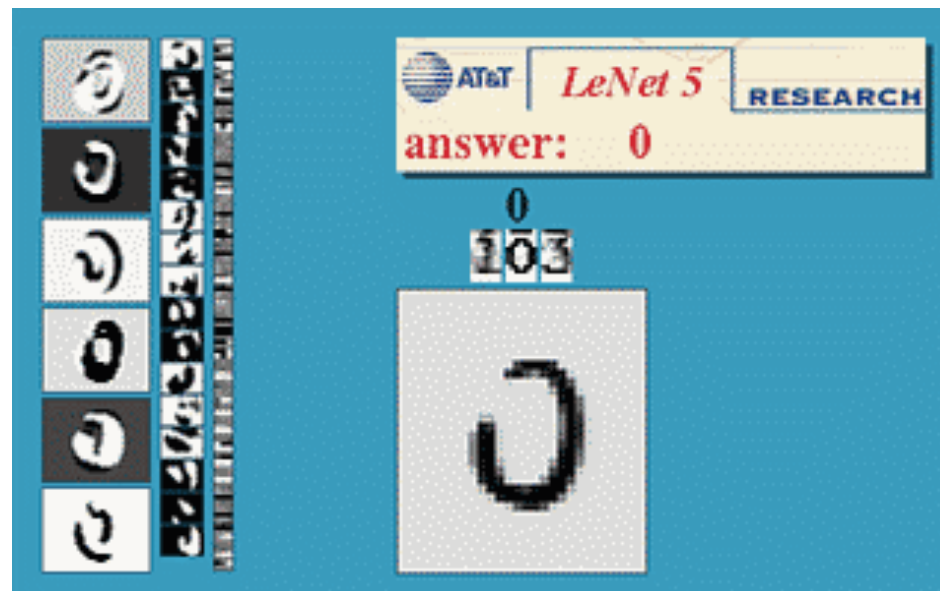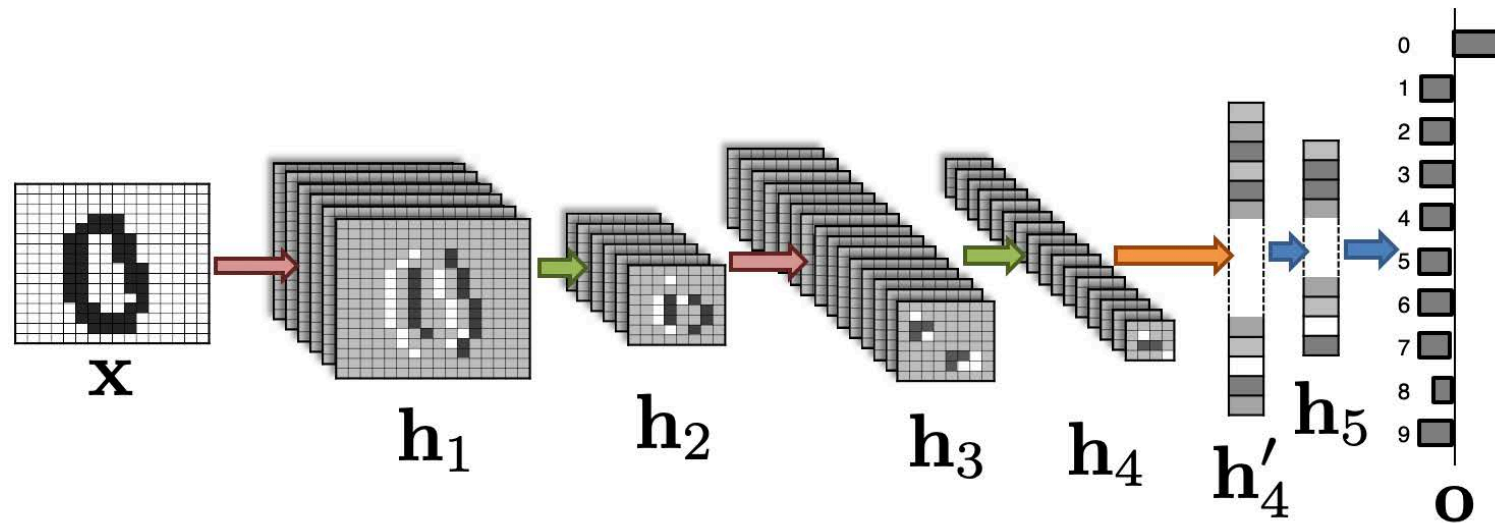# ML to the Rescue: Texton Boost

Use AdaBoost to perform classification on the output of Gabor filters.
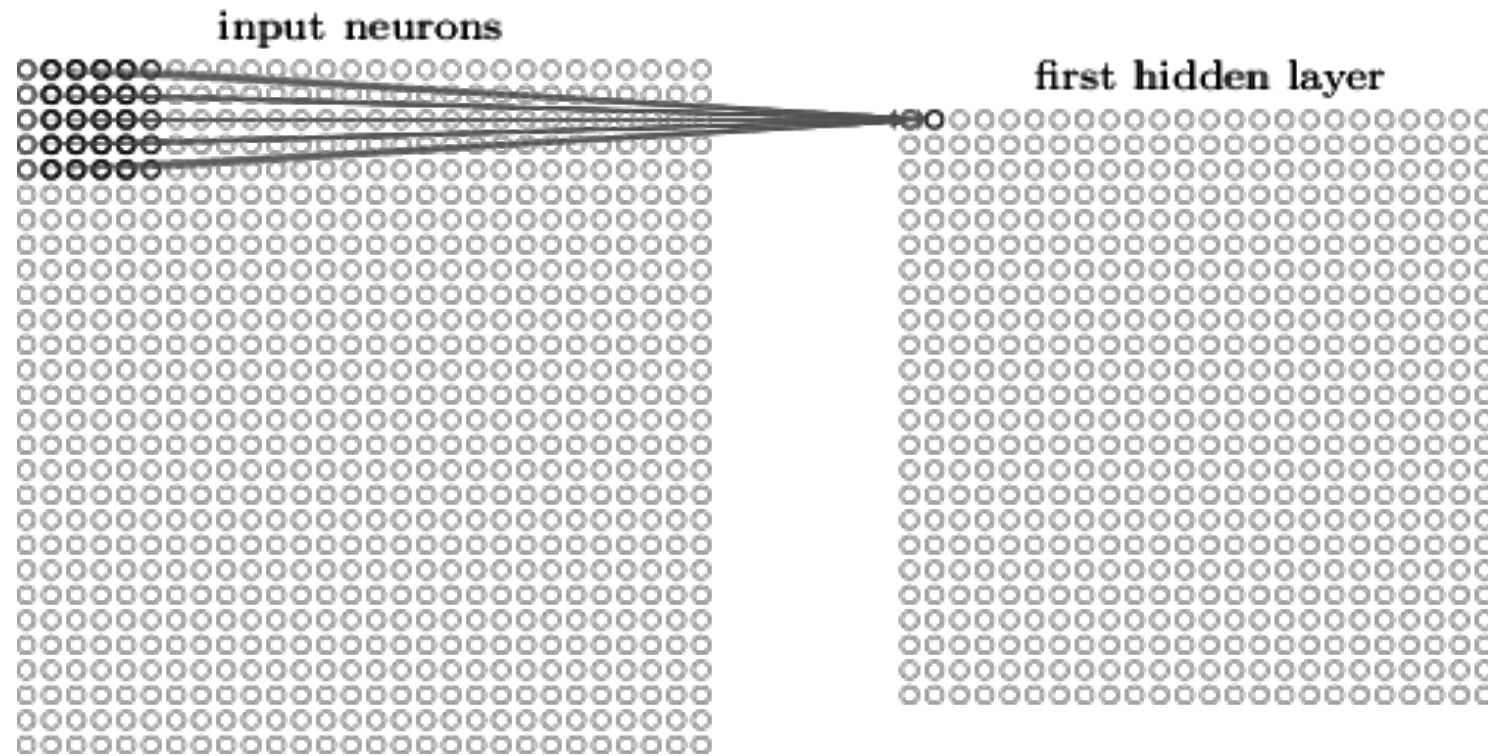
ML to the Rescue: Texton Forests
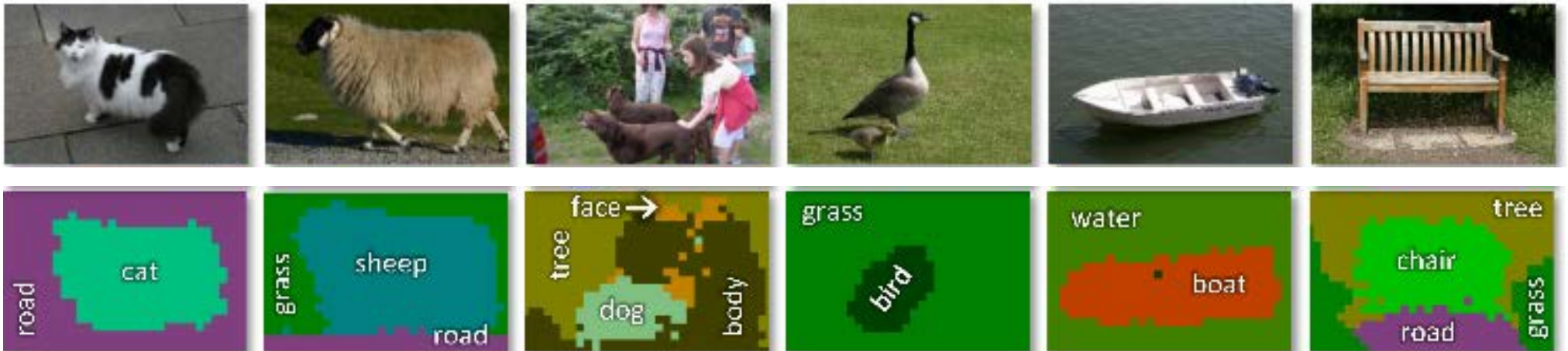
- Using Decision Forests to perform classification on the output of Gabor filters works better in this case.

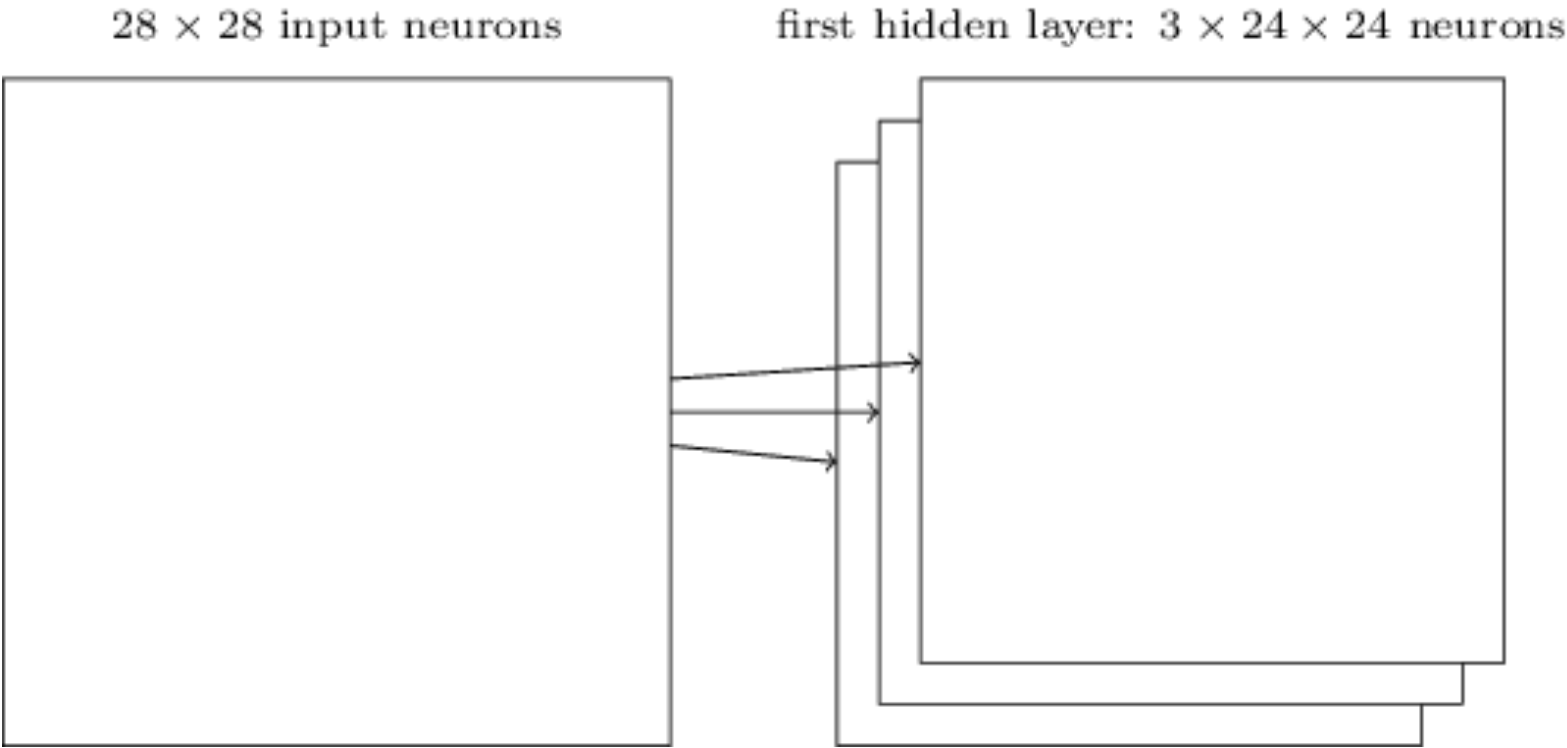- But what works even better, is ……

Reminder: ConvNets
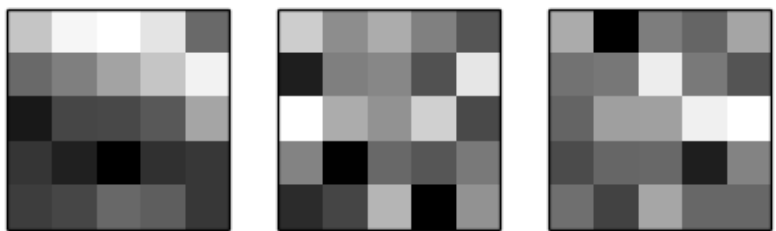
Reminder: Convolutional Layer

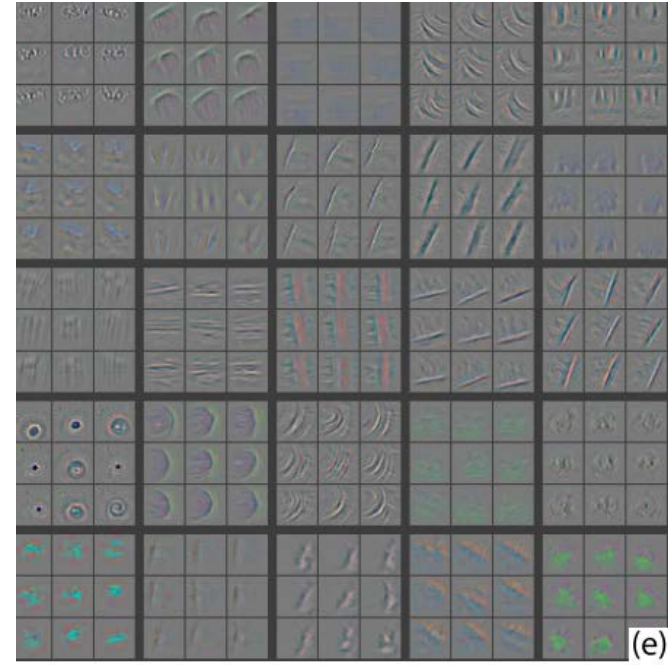$$\sigma\left(b + \sum_{x=0}^{n_x}\sum_{y=0}^{n_y} w_{x,y} a_{i+x,j+y}\right)$$

Reminder: Feature Maps

28 × 28 input neurons

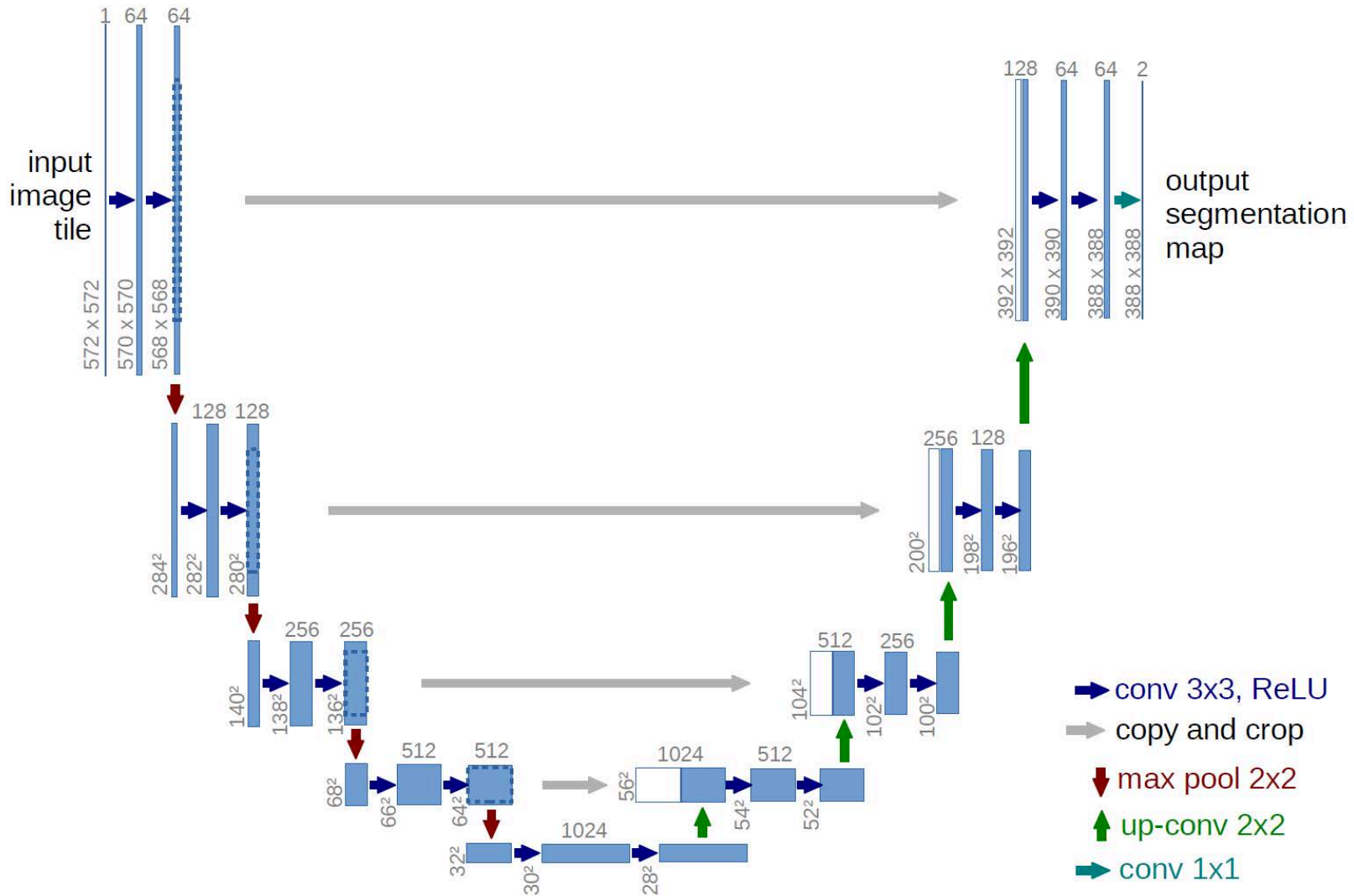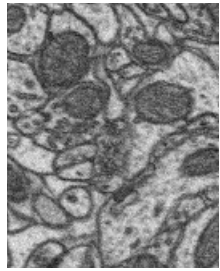first hidden layer: 3 × 24 × 24 neurons

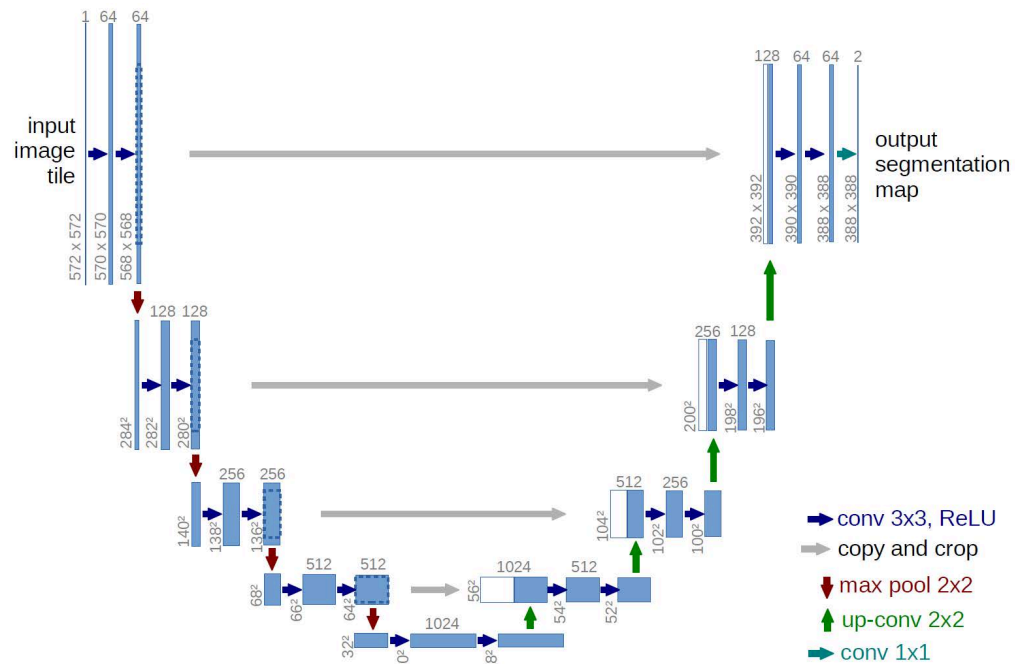Filters:

Learned Feature Maps

(c)

(e)
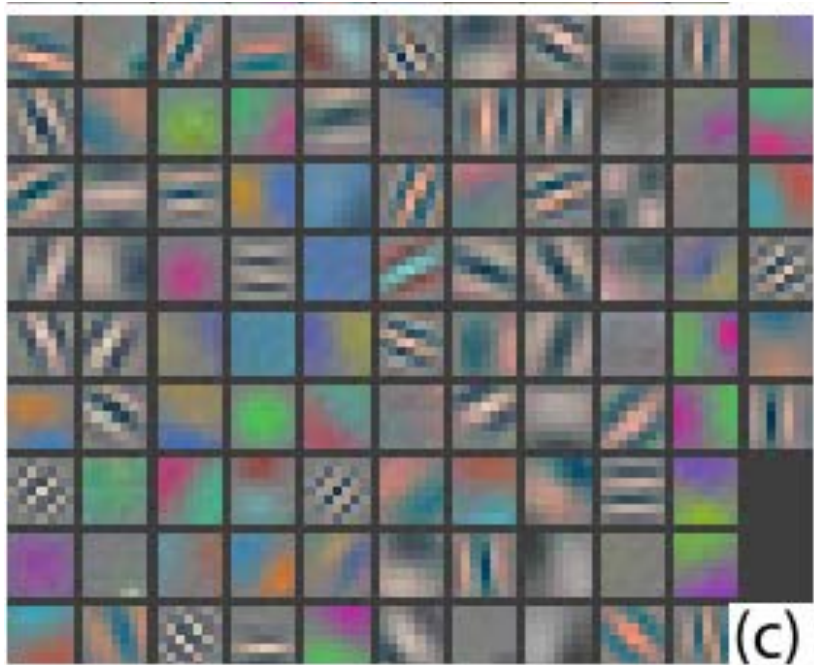
- Some of these convolutional filters look very Gabor like.

- The network requires a large training set to learn an effective filter bank.

- The older techniques still have their place in the absence of such training sets.
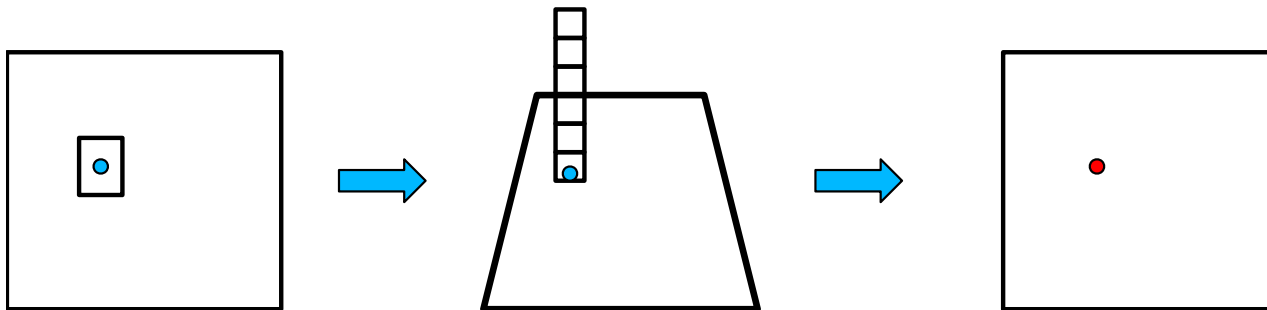
Reminder: U-Net Architecture

Reminder: Potential Interpretation

A key role of the ConvNet is to generate for every output pixel a feature vector containing the output of all the intermediate layers.

In Short

Texture is a key property of objects that is

- Non local
- Non trivial to measure
- Subject to deformations

➡ Hard to characterize formally and best used in conjunction with effective Machine Learning techniques.

➡ This seems to be exactly what Convolutional Neural Nets do.