Features ?
To explain the Convolutional NNs, we will look at the edge detection example first.
Edges

- Vertical edges
- Horizontal edges

Image credit: Andrew Ng
Discontinuities in images are features that are often useful for initializing an image analysis procedure.

Edges are important information for understanding an image; by removing “non-edge” data we also simplify the data.
Edge Models
Characterizing Edges

- An edge is a place of rapid change in the image intensity function

![Image of an edge in an image, intensity function along a horizontal scanline, and the first derivative with a peak indicating an edge location.]

edges correspond to extrema of derivative
Derivatives and Average

- **Derivative**: rate of change
- **Examples**:
  - Speed is a rate of change of a distance, \( X = V \cdot t \)
  - Acceleration is a rate of change of speed, \( V = a \cdot t \)
Derivative

\[
\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x) = f_x
\]

Example: \( y = x^2 + x^4 \)

\[
\frac{dy}{dx} = 2x + 4x^3
\]
Discrete Derivative

\[
\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x)
\]

\[
\frac{df}{dx} = \frac{f(x) - f(x-1)}{1} = f'(x)
\]

\[
\frac{df}{dx} = f'(x) - f(x - 1) = f''(x)
\]
Discrete Derivative / Finite Difference

\[
\frac{df}{dx} = f(x) - f(x - 1) = f'(x) \quad \text{Backward difference}
\]

\[
\frac{df}{dx} = f(x) - f(x + 1) = f'(x) \quad \text{Forward difference}
\]

\[
\frac{df}{dx} = f(x + 1) - f(x - 1) = f'(x) \quad \text{Central difference}
\]
Example: Finite Difference

\[ f(x) = \begin{bmatrix} 10 & 15 & 10 & 10 & 25 & 20 & 20 & 20 \end{bmatrix} \]

\[ f'(x) = \begin{bmatrix} 0 & 5 & -5 & 0 & 15 & -5 & 0 & 0 \end{bmatrix} \]

Derivative Masks

- Backward difference: \([-1 \quad 1]\)
- Forward difference: \([1 \quad -1]\)
- Central difference: \([-1 \quad 0 \quad 1]\)
Derivative in 2-D

Given function:

\[ f(x, y) \]

Gradient vector:

\[ \nabla f(x, y) = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} \]

Gradient magnitude:

\[ |\nabla f(x, y)| = \sqrt{f_x^2 + f_y^2} \]

Gradient direction:

\[ \theta = \tan^{-1} \frac{f_x}{f_y} \]
Derivative of Images

Derivative masks (filters):

\[
\begin{align*}
 f_x &\Rightarrow \frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \\
 f_y &\Rightarrow \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}
\end{align*}
\]

\[
I = \begin{bmatrix}
10 & 10 & 20 & \frame{20} & 20 & 20 \\
10 & 10 & 20 & \frame{20} & 20 & 20 \\
10 & 10 & 20 & \frame{20} & 20 & 20 \\
10 & 10 & 20 & \frame{20} & 20 & 20 \\
\end{bmatrix}
\]

\[
I_x = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frame{10} & 10 & 0 & 0 & 0 \\
0 & 10 & 10 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]
Both zebras and dalmatians have black and white pixels in similar numbers.

The difference between the two is the characteristic appearance of small groups of pixels rather than individual pixel values.

Adapted from Pinar Duygulu, Bilkent University
We will discuss neighborhood operations that work with the values of the image pixels in the neighborhood.

- Spatial domain filtering
- Frequency domain filtering
- Image enhancement
- Finding patterns
Spatial domain filtering

- Some neighborhood operations work with
  - the values of the image pixels in the neighborhood, and
  - the corresponding values of a subimage that has the same dimensions as the neighborhood.
- The subimage is called a **filter** (or mask, kernel, template, window).
- The values in a filter subimage are referred to as **coefficients or weights**, rather than pixels.
Spatial domain filtering

- Operation: modify the pixels in an image based on some function of the pixels in their neighborhood.
- Simplest: linear filtering (replace each pixel by a linear combination of its neighbors).
- Linear spatial filtering is often referred to as “convolving an image with a filter”.

Linear filtering

For a linear spatially invariant system

\[ f[m, n] = g \otimes h = \sum_{k,l} g[m - k, n - l]h[k, l] \]

\[
\begin{array}{cccccccc}
111 & 115 & 113 & 111 & 112 & 111 & 111 & 111 \\
135 & 138 & 137 & 139 & 145 & 146 & 149 & 147 \\
163 & 168 & 188 & 196 & 206 & 202 & 206 & 207 \\
180 & 184 & 206 & 219 & 202 & 200 & 195 & 193 \\
189 & 193 & 214 & 216 & 104 & 79 & 83 & 77 \\
191 & 201 & 217 & 220 & 104 & 60 & 68 & 68 \\
195 & 205 & 216 & 222 & 113 & 68 & 69 & 83 \\
199 & 203 & 223 & 228 & 108 & 68 & 71 & 77 \\
\end{array}
\]

\[
\begin{array}{cccc}
-1 & 2 & -1 \\
-1 & 2 & -1 \\
-1 & 2 & -1 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
? & -5 & 9 & -9 & 21 & -12 & 10 & ? \\
? & -29 & 18 & 24 & 4 & -7 & 5 & ? \\
? & -50 & 40 & 142 & -88 & -34 & 10 & ? \\
? & -41 & 41 & 264 & -175 & -71 & 0 & ? \\
? & -23 & 33 & 360 & -217 & -134 & -23 & ? \\
\end{array}
\]

### g[m,n] h[m,n] f[m,n]
Be careful about indices, image borders and padding during implementation.

- zero
- fixed/clamp
- periodic/wrap
- reflected/mirror

Border padding examples.

Adapted from CSE 455, U of Washington
Often, an image is composed of
- some underlying ideal structure, which we want to detect and describe,
- together with some random noise or artifact, which we would like to remove.

Smoothing filters are used for blurring and for noise reduction.

Linear smoothing filters are also called averaging filters.
Smoothing spatial filters

\[
\frac{1}{9} \times (10 \times 1 + 11 \times 1 + 10 \times 1 + 9 \times 1 + 10 \times 1 + 11 \times 1 + 10 \times 1 + 9 \times 1 + 10 \times 1) = \frac{1}{9} \times 90 = 10
\]

Adapted from Octavia Camps, Penn State
Smoothing spatial filters

\[
\frac{1}{9} \left( 10 \times 1 + 9 \times 1 + 11 \times 1 + 9 \times 1 + 99 \times 1 + 11 \times 1 + 11 \times 1 + 10 \times 1 + 10 \times 1 \right) = \frac{1}{9} \left( 180 \right) = 20
\]

Adapted from Octavia Camps, Penn State
Smoothing spatial filters

Common types of noise:

- **Salt-and-pepper noise**: contains random occurrences of black and white pixels.
- **Impulse noise**: contains random occurrences of white pixels.
- **Gaussian noise**: variations in intensity drawn from a Gaussian normal distribution.

Adapted from Linda Shapiro, U of Washington.
Smoothing spatial filters

**FIGURE 3.36** (a) Image from the Hubble Space Telescope. (b) Image processed by a $15 \times 15$ averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)
Smoothing spatial filters

A weighted average that weighs pixels at its center much more strongly than its boundaries.

\[ G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp \left( -\frac{x^2 + y^2}{2\sigma^2} \right) \]

2D Gaussian filter

Adapted from Martial Hebert, CMU
Smoothing spatial filters

- If $\sigma$ is small: smoothing will have little effect.

- If $\sigma$ is larger: neighboring pixels will have larger weights resulting in consensus of the neighbors.

- If $\sigma$ is very large: details will disappear along with the noise.

Adapted from Martial Hebert, CMU
Smoothing spatial filters

Width of the Gaussian kernel controls the amount of smoothing.

Adapted from K. Grauman
Smoothing spatial filters

Result of blurring using a uniform local model.

Produces a set of narrow horizontal and vertical bars – ringing effect.

Result of blurring using a Gaussian filter.

Adapted from David Forsyth, UC Berkeley
Smoothing spatial filters

Gaussian versus mean filters

Adapted from CSE 455, U of Washington
Order-statistic filters

- Order-statistic filters are **nonlinear spatial filters** whose response is based on:
  - ordering (ranking) the pixels contained in the image area encompassed by the filter, and then
  - replacing the value of the center pixel with the value determined by the ranking result.

- The best-known example is the **median filter**.

- It is particularly effective in the presence of impulse or salt-and-pepper noise, with considerably less blurring than linear smoothing filters.
Order-statistic filters

Adapted from Octavia Camps, Penn State
### Order-statistic filters

Adapted from Octavia Camps, Penn State
Salt-and-pepper noise

Adapted from Linda Shapiro, U of Washington
Gaussian noise

3x3

5x5

7x7

Adapted from Linda Shapiro, U of Washington
Spatially varying filters

Bilateral filter: kernel depends on the local image content.
See the Szeliski book for the math.

Adapted from Sylvian Paris
Spatially varying filters

Compare to the result of using the same Gaussian kernel everywhere

Adapted from Sylvian Paris
Objective of sharpening is to highlight or enhance fine detail in an image.

Since smoothing (averaging) is analogous to integration, sharpening can be accomplished by spatial differentiation.

First-order derivative of 1D function $f(x)$
$$f(x+1) - f(x).$$

Second-order derivative of 1D function $f(x)$
$$f(x+1) - 2f(x) + f(x-1).$$
Sharpening spatial filters

**For a function** $f(x, y)$, the **gradient** at $(x, y)$ is defined as

$$\nabla f = \left[ \frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right]^T$$

where its magnitude can be used to implement first-order derivatives.

<table>
<thead>
<tr>
<th>-1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Robert’s cross-gradient operators

<table>
<thead>
<tr>
<th>-1</th>
<th>-2</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Sobel gradient operators

<table>
<thead>
<tr>
<th>-1</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Sharpening spatial filters

- **Laplacian** of a function (image) $f(x, y)$ of two variables $x$ and $y$

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

is a second-order derivative operator.

**FIGURE 3.39**

(a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4).
(b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.
Sharpening spatial filters

FIGURE 3.40
(a) Image of the North Pole of the moon.
(b) Laplacian-filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)
Sharpening spatial filters

High-boost filtering

Adapted from Darrell and Freeman, MIT
Sharpening spatial filters

before

after

Adapted from Darrell and Freeman, MIT