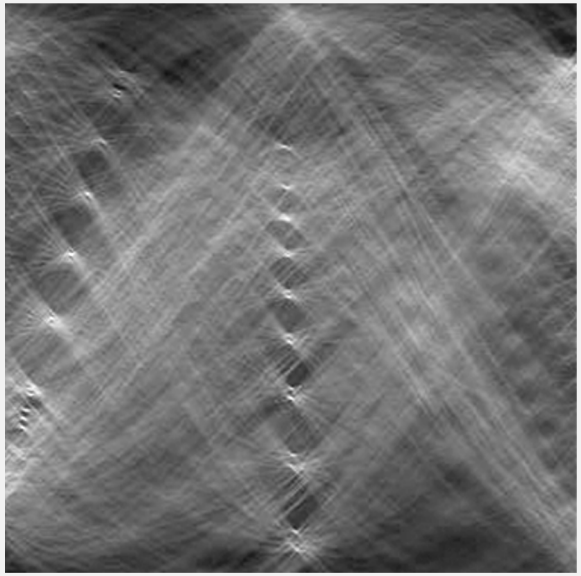
Department of Computer Science
University of Bristol

COMS30030
Image Processing and Computer Vision



Week 03

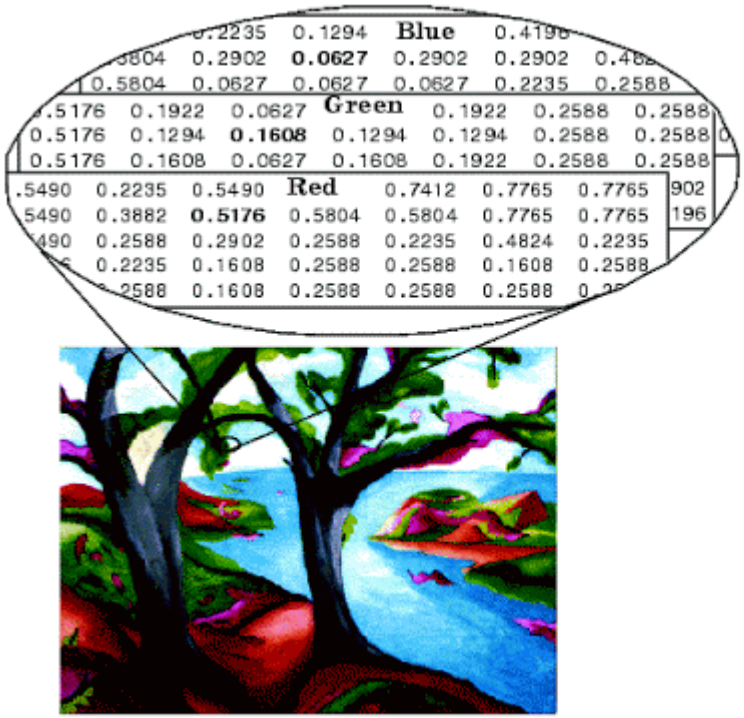
# Edge Detection

Alessandro Masullo

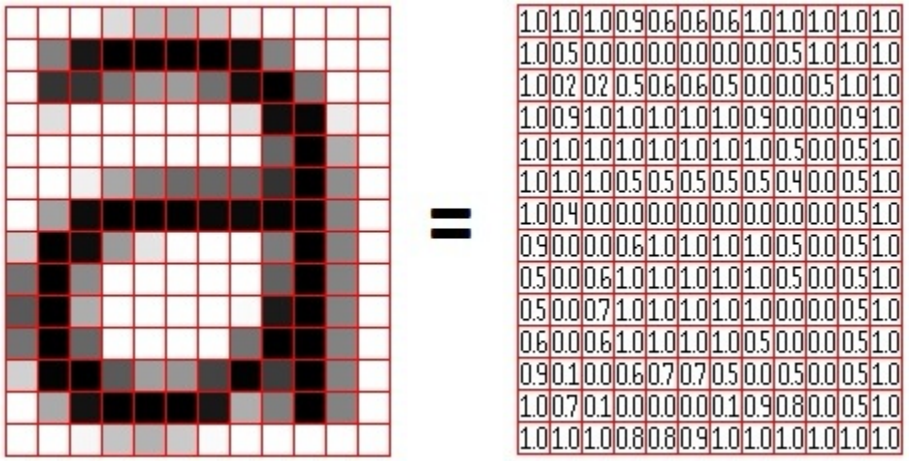
Original slides from Majid Mirmehdi and Tilo Burghardt

# Beyond the Matrix

- Images are matrices of numbers
- Shapes are more informative than individual pixel values




Source: http://www.ece.northwestern.edu/local-apps/matlabhelp/toolbox/images/intro8.html



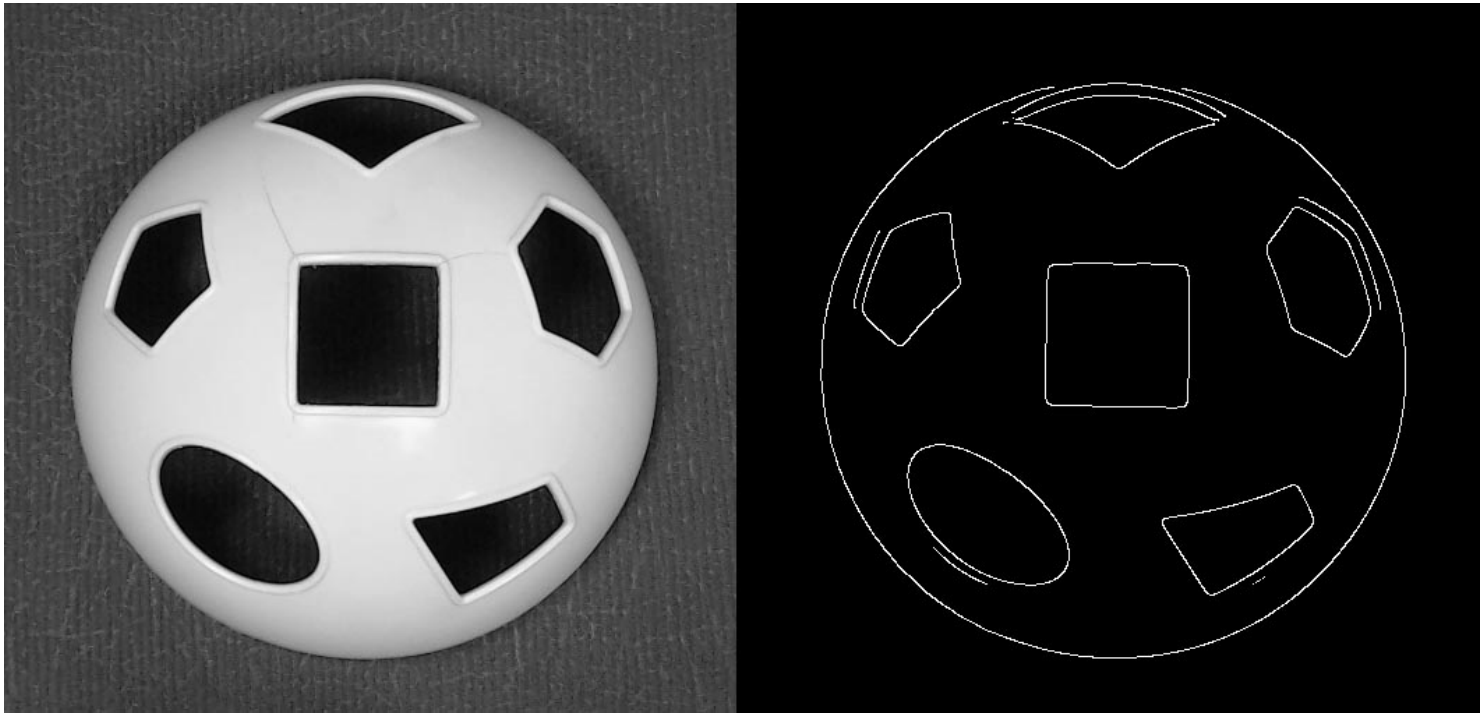
Source: https://www.datasciencecentral.com/profiles/blogs/image-classification-with-hsv-color-model-processing

# Edges in Artistic Drawings

Source: http://www.ateliermagique.com

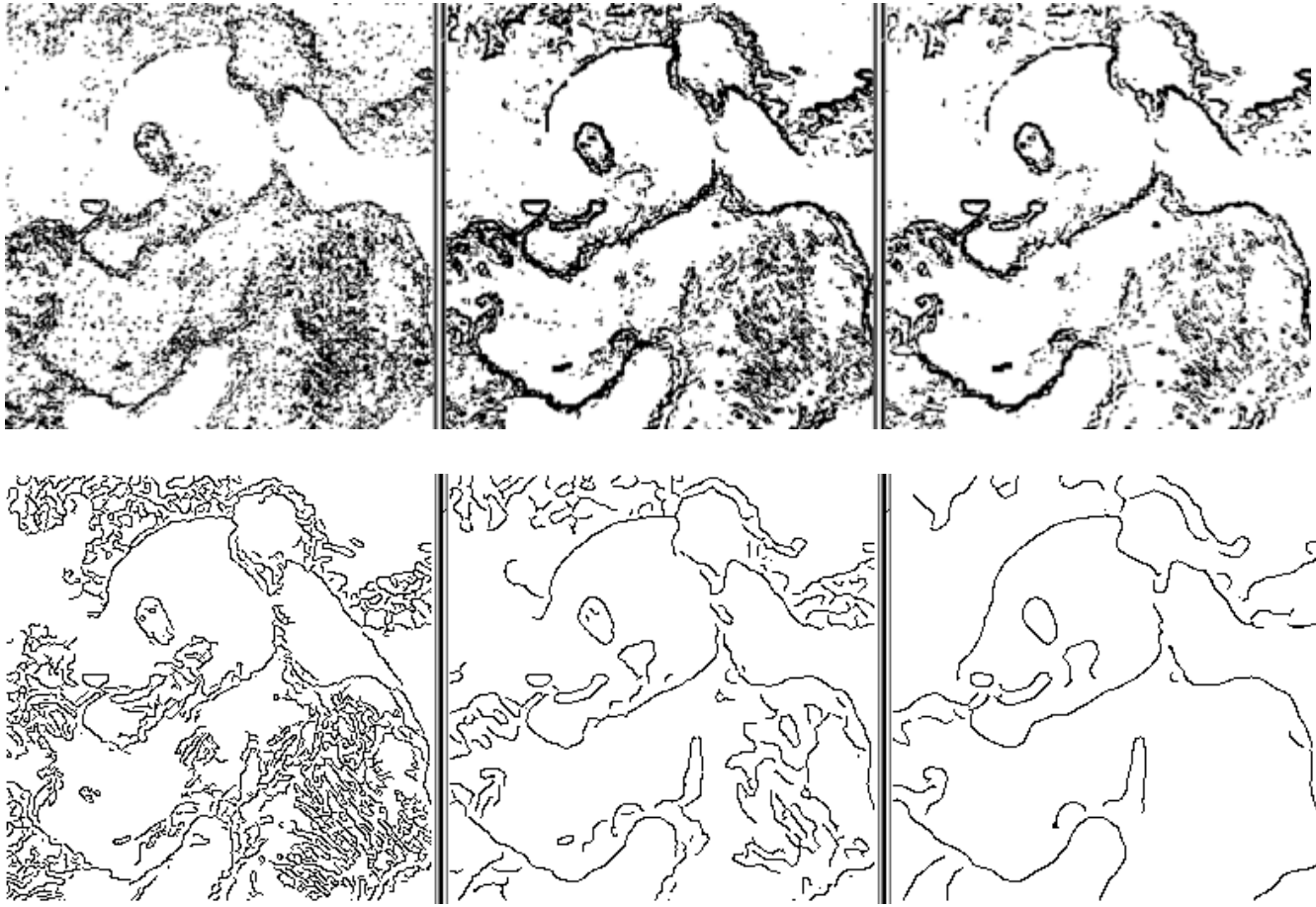
# What are edges?

- Edges highlight the contour of shapes
- They can be used to identify objects



Source: http://www.walrusvision.com/wordpress/introduction-to-edge-detection/

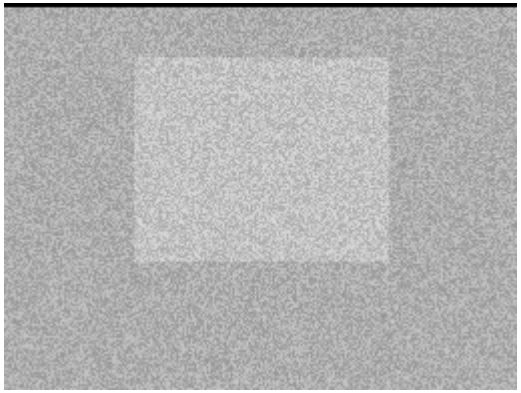
# Difficult edges

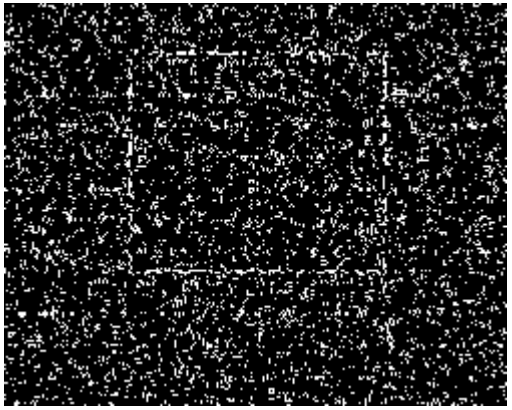


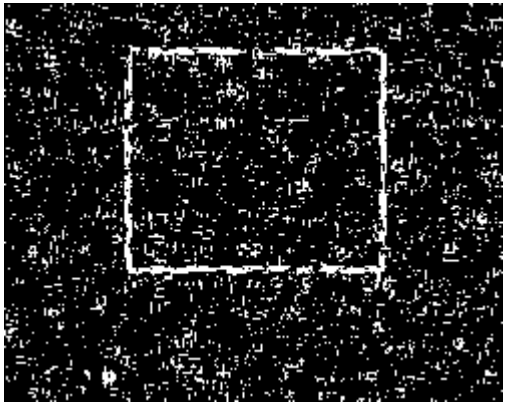
Source: http://fourier.eng.hmc.edu/e161/lectures/canny/node1.html

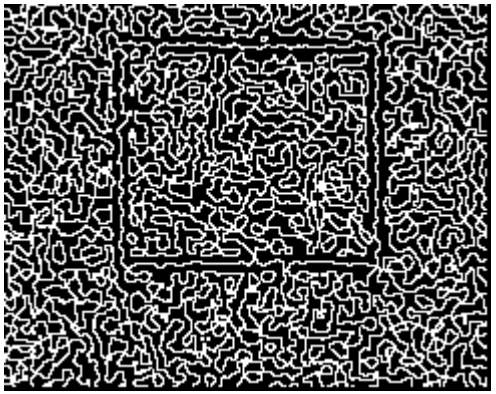
# Very difficult edges

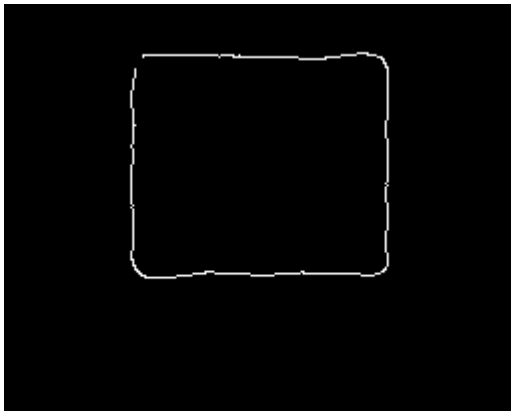
- Different techniques work for different cases





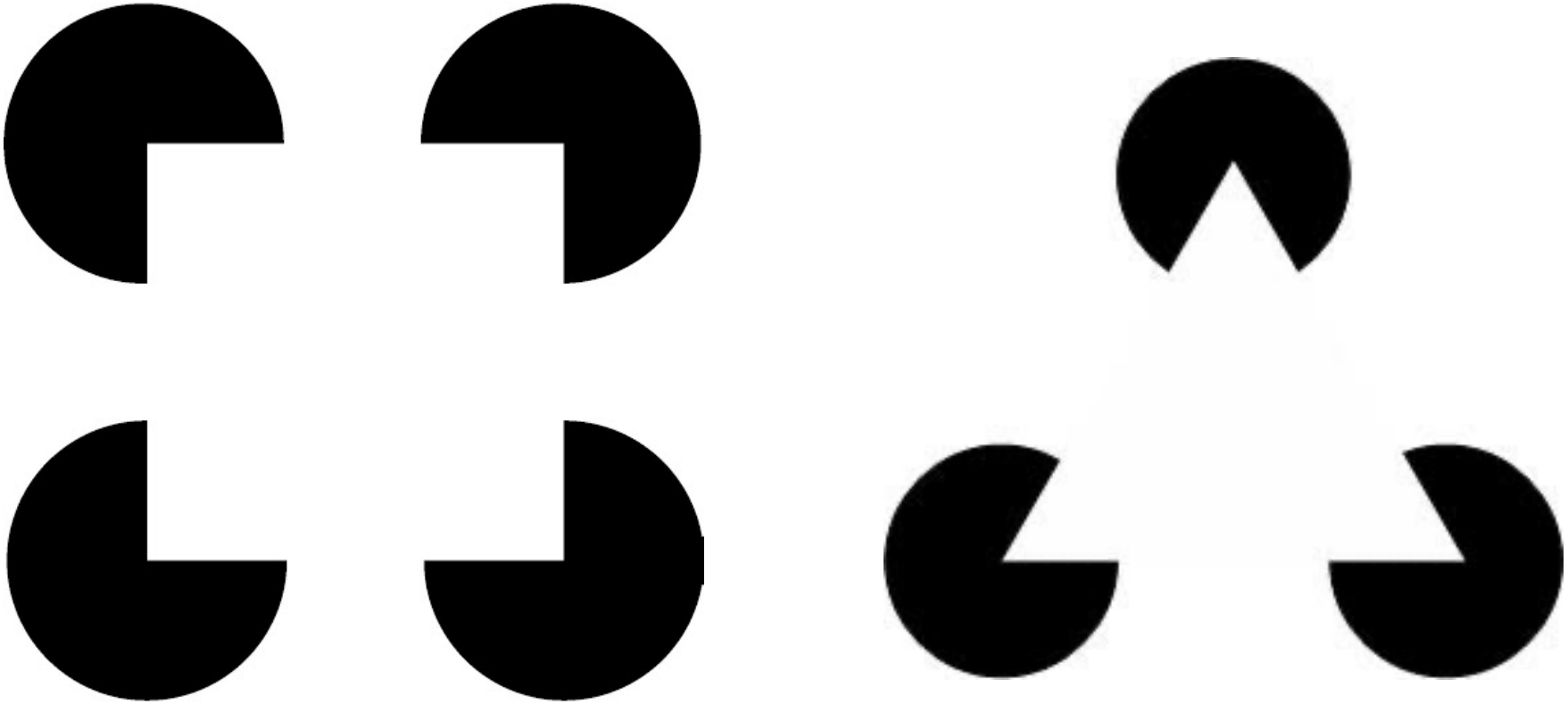






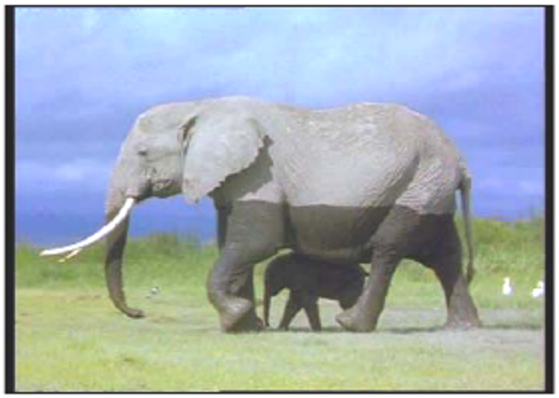
# Impossible edges

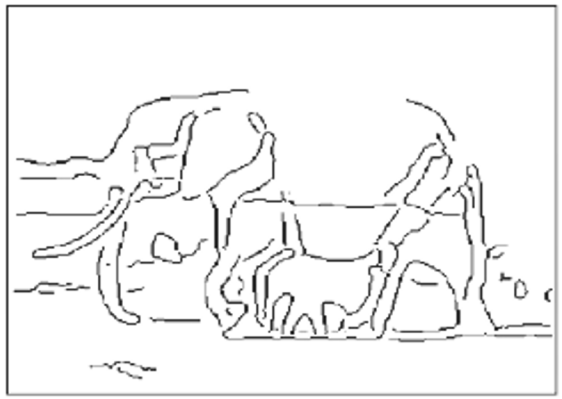
- Edges can also be produced by optical illusions



# Motivation: Why detecting Edges?

- **Edges:** sharp changes of image brightness
- **Sources:** Object boundaries, Patterns, Shadows, …
- Meaningful edges ← → Nuisance edges
- For Segmentation: finding object boundaries
- For Recognition: extracting patterns
- For Motion Analysis: reliable tracking regions





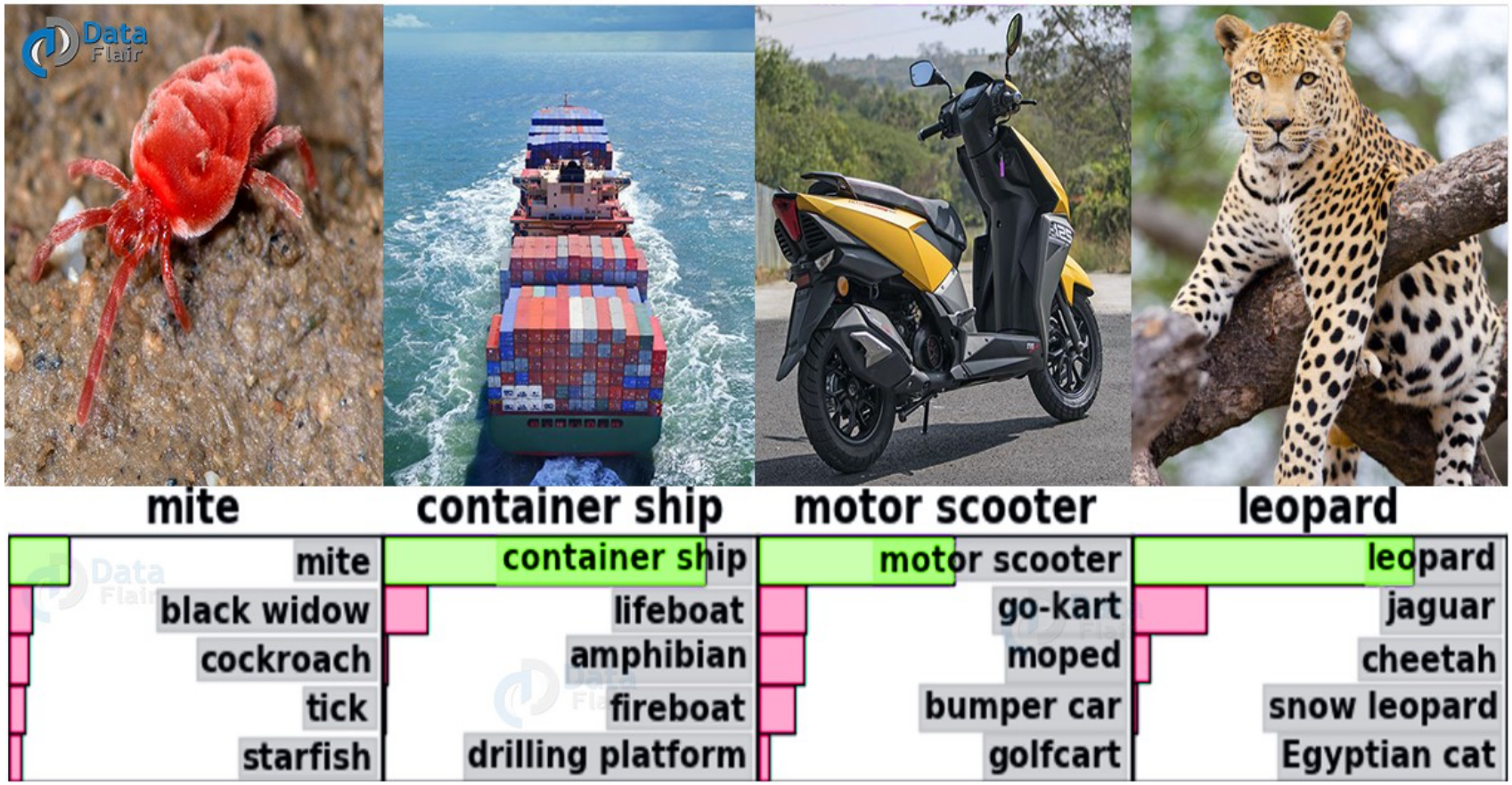
# Motivation: Why detecting Edges?

- Image segmentation


Source: https://theaisummer.com/Semantic_Segmentation/

# Motivation: Why detecting Edges?

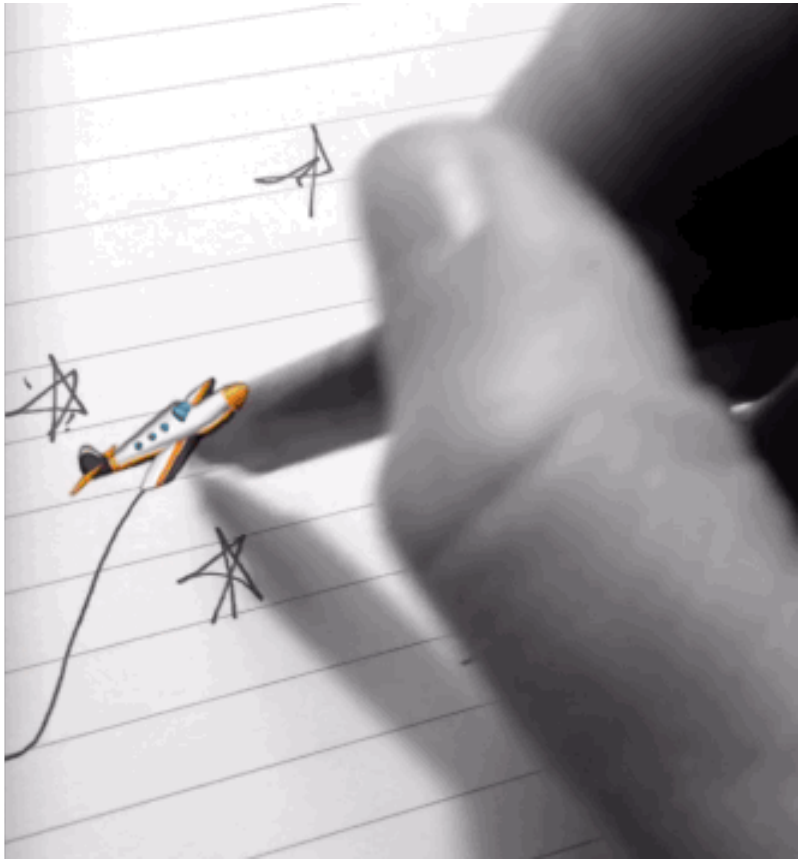
- Image recognition

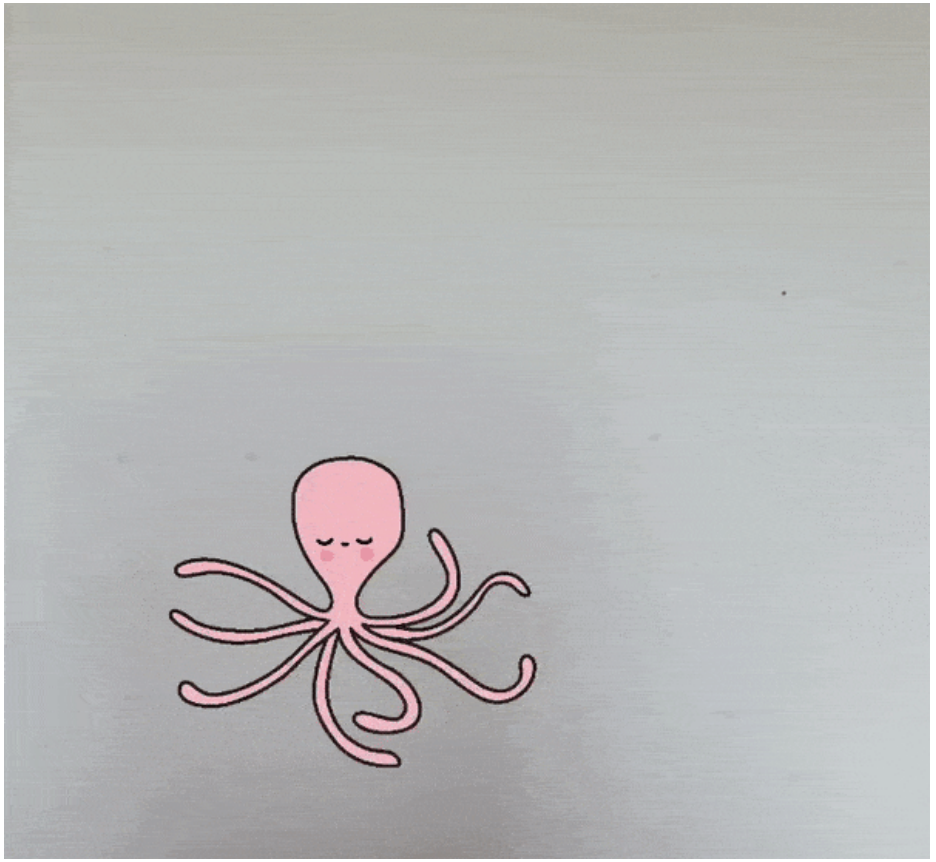



Source: https://medium.com/@rinu.gour123/tensorflow-image-recognition-using-python-c-df4d05e1ffec

# Motivation: Why detecting Edges?

- Motion analysis

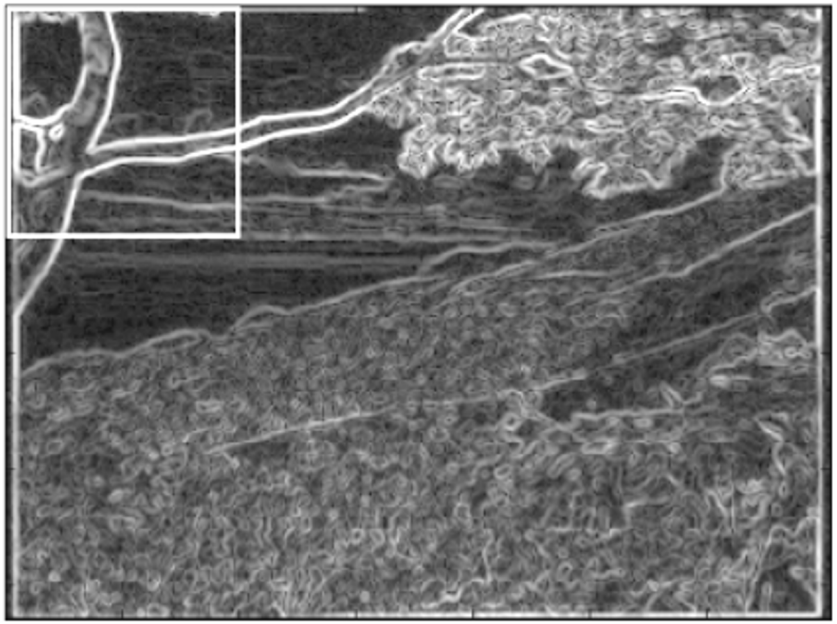




# Edge Detection Strategy

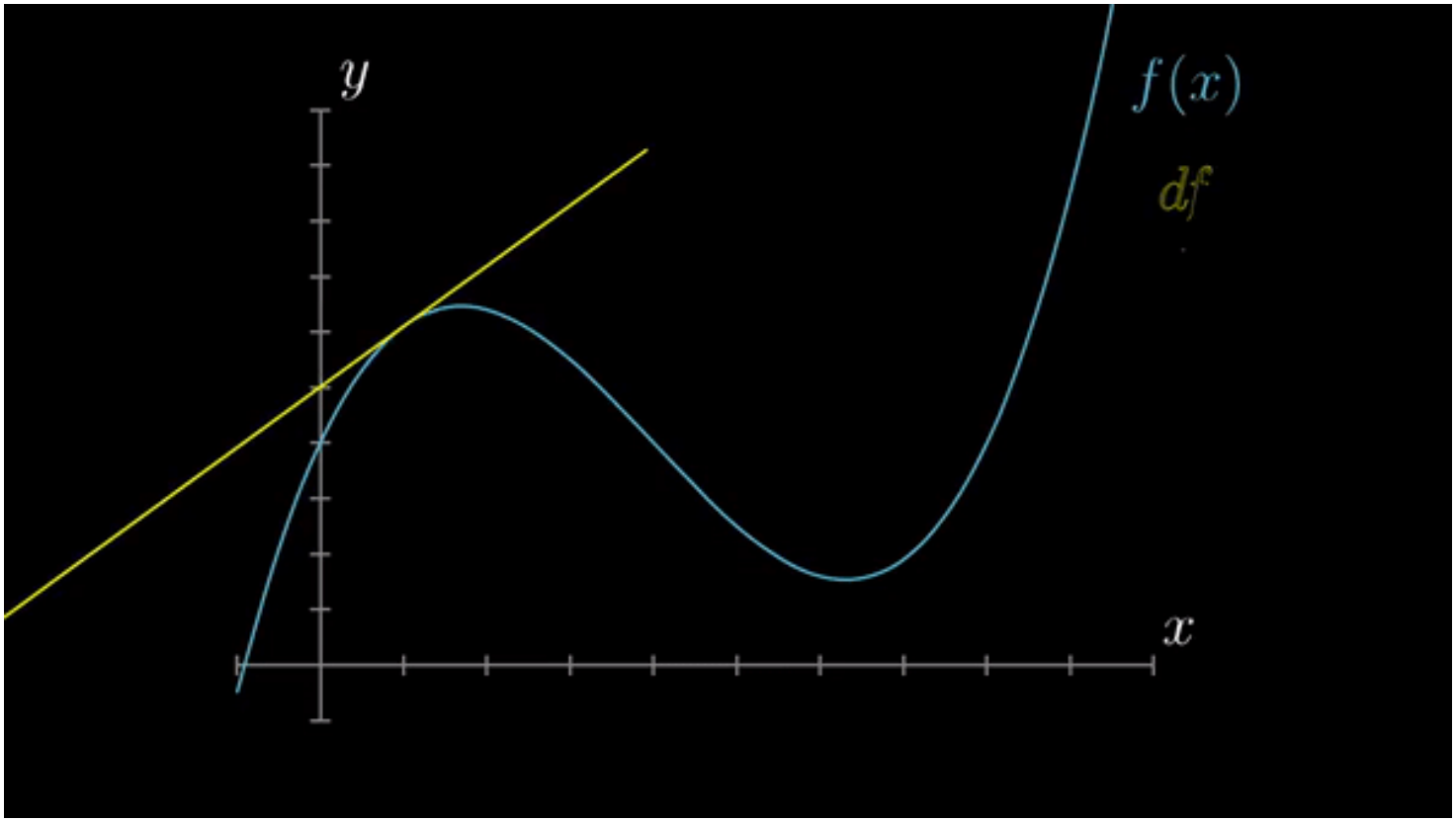
- **Recognition Strategy:**
  Determine a 'measure of change'
  in the pixel's neighbourhood

- First derivation in 2D space → Image Gradient



# Edge Detection Strategy

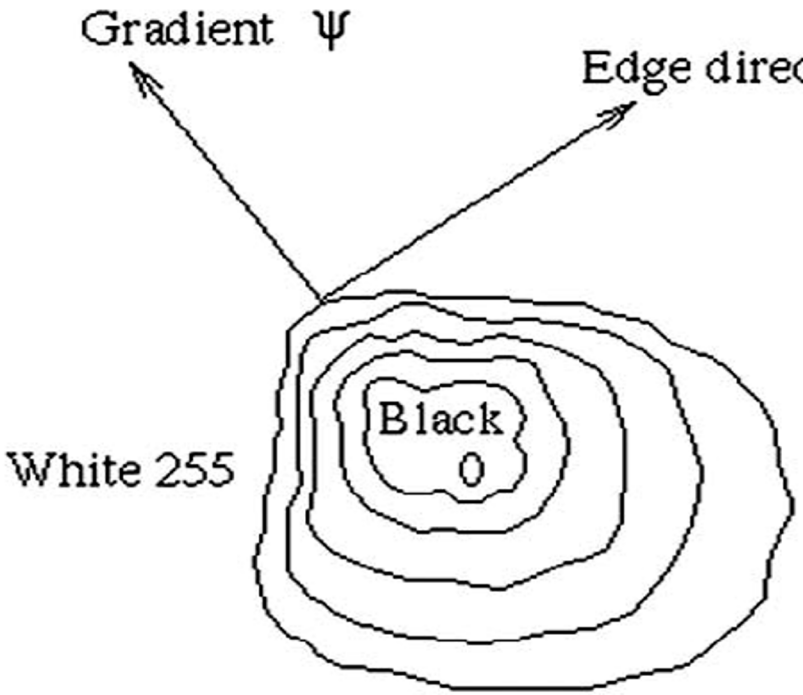
- Derivative to measure change




Source: https://www.youtube.com/watch?v=BLkz5LGWihw

# The Image Gradient

- A ***vector*** variable
  - Direction $\psi$ of the maximum growth of the function
  - Magnitude $|\nabla f(x,y)|$ of the growth
  - Perpendicular to the edge direction



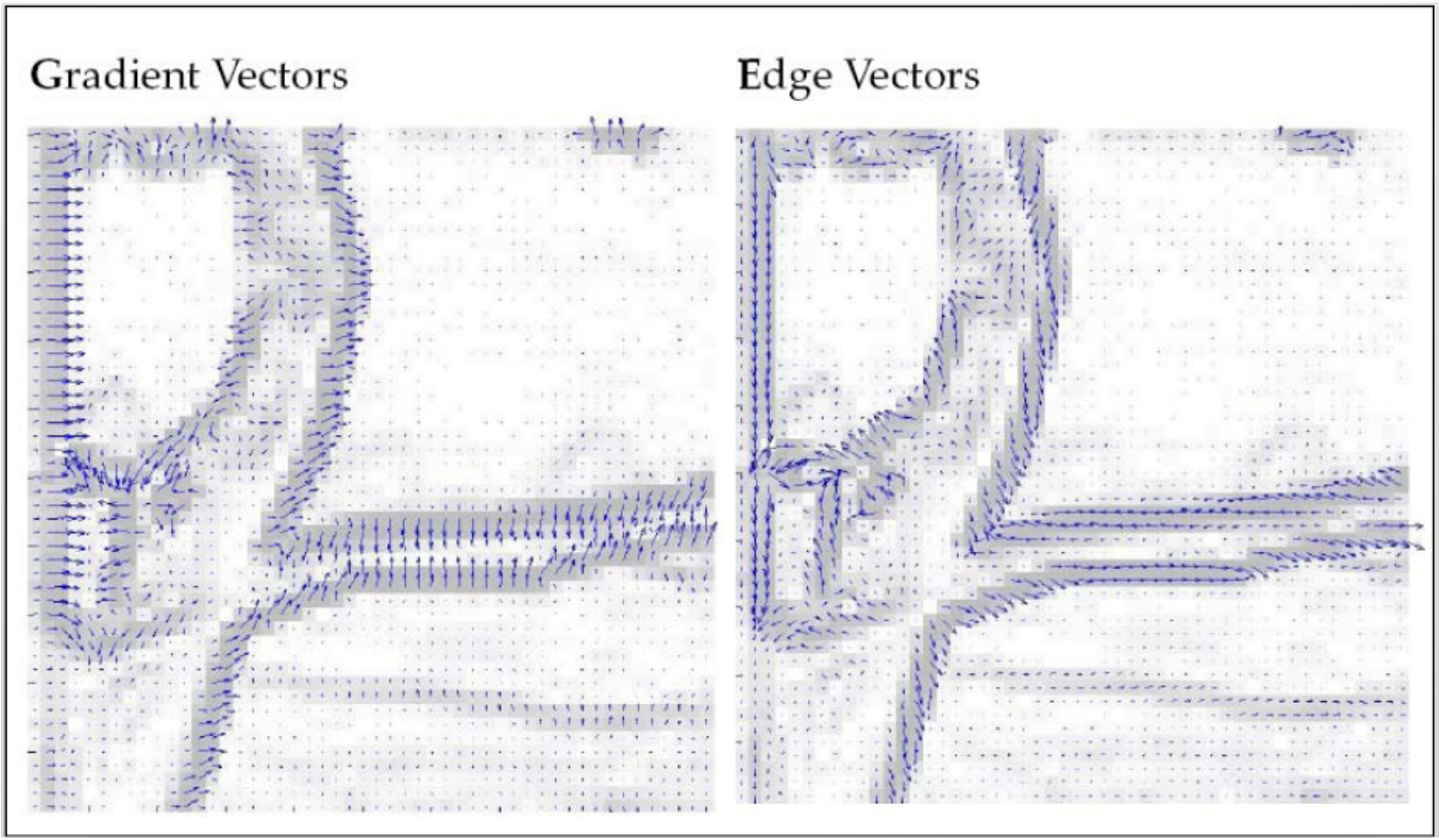

$$\nabla f(x,y) = \frac{\partial f}{\partial x}\hat{\mathbf{x}} + \frac{\partial f}{\partial y}\hat{\mathbf{y}}$$

$$|\nabla f(x,y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

$$\psi = \arctan\left(\frac{\partial f/\partial y}{\partial f/\partial x}\right)$$

$$\phi = \psi - \frac{\pi}{2}$$

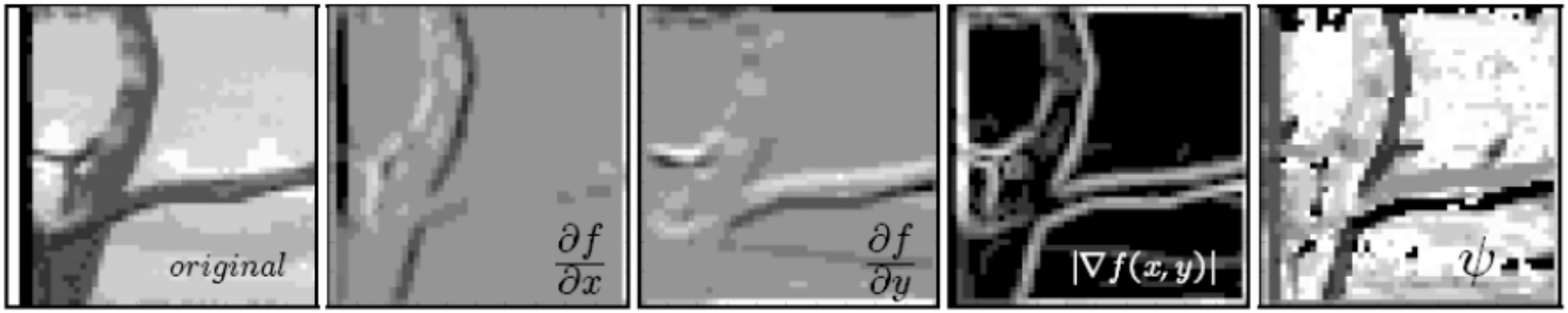
# Example: Gradient & Edge Vectors



# Gradient Extraction via Filtering

$$\frac{\partial f}{\partial x} \approx \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \frac{\partial f}{\partial y} \approx \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

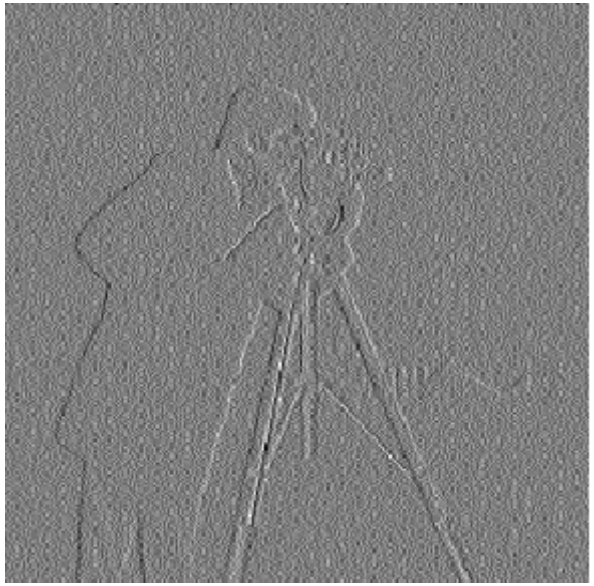
$$grad(f) = |\nabla f(x,y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \qquad \psi = \arctan\left(\frac{\partial f / \partial y}{\partial f / \partial x}\right)$$



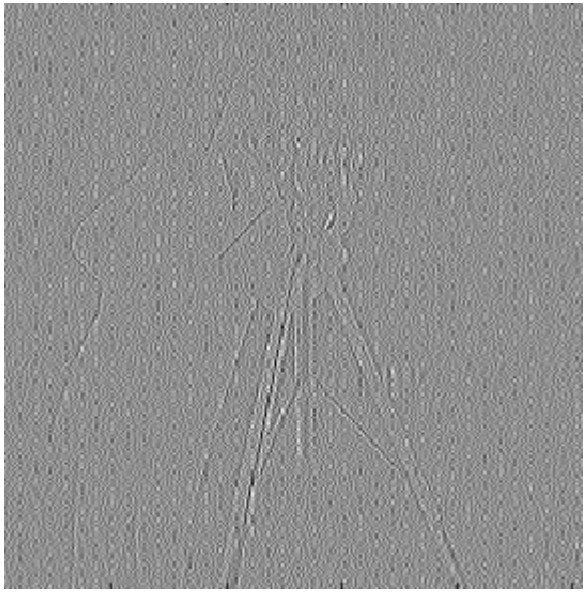
# Gradient Extraction via Filtering

- Example of differentiation with noise

Noisy cameraman



First derivative

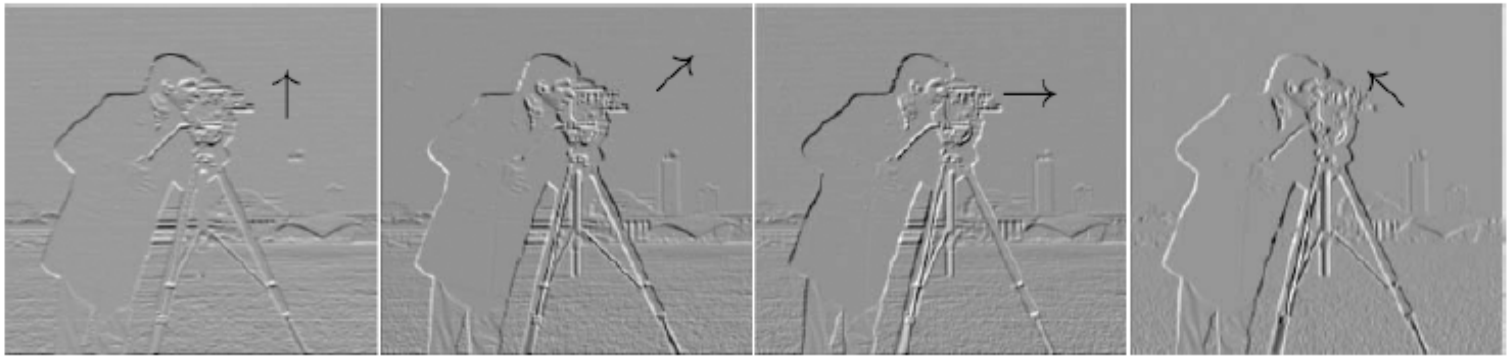


Second derivative

# Prewitt Operator

- Central difference $\frac{\delta f}{\delta x} \approx \frac{f(x+1)-f(x-1)}{2}$
- Mask [-1 0 1] is very sensitive to noise
- Smoothing in the perpendicular direction
- For 3x3 mask, $\nabla f$ estimated in 8 directions

$$h_{hor} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, h_{dia} = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}, \ldots$$

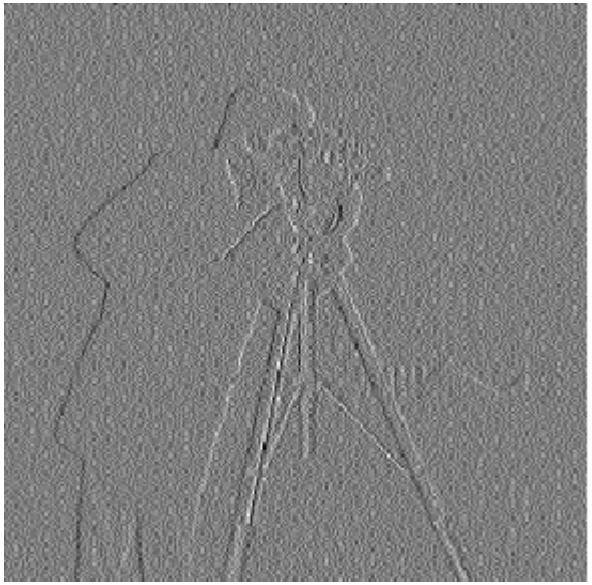


# Sobel Operator

- As Prewitt, relies on central differences
- Greater weight to the central pixels
- $\frac{\partial}{\partial x} \approx \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$, $\frac{\partial}{\partial y} \approx \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
- Can be approx as derivative of a Gaussian
- First Gaussian smoothing, then derivation
- $\frac{\partial}{\partial x}(I * G) = I * \frac{\partial G}{\partial x}$

# Sobel Operator

- Comparison with standard difference



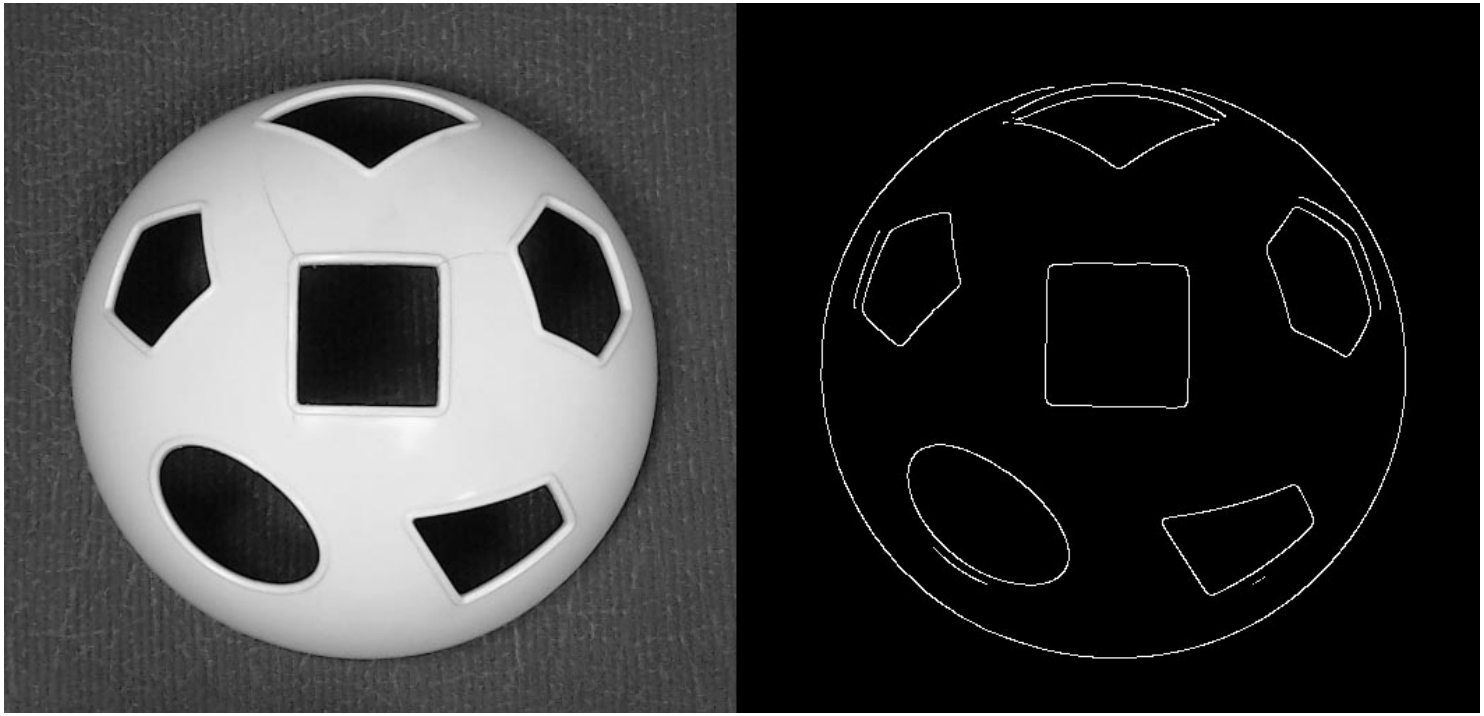
Simple difference

Noisy cameraman

Sobel difference

# Shape recognition

- From edges to shape detection
- Image processing is not image detection



# In the Next video

- Shape detection
- Line detection
- Representation of lines
- Hough transform
- Generalised Hough transform
- Algorithms for shape detection