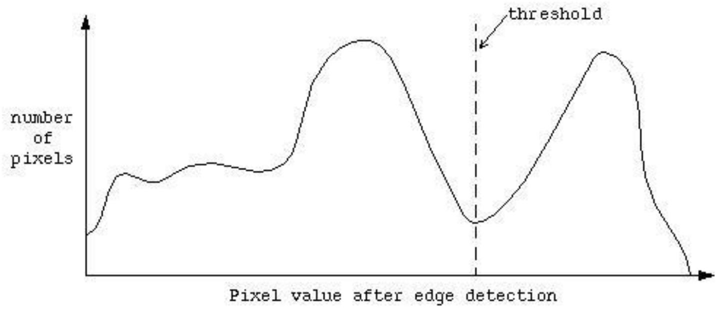
✓**Boundary Detection**

- It is performed by finding the boundaries between objects, thus indirectly defining the objects.
- This method is usually begin by marking points that may be a part of an edge
- These points are then merged into line segments, and the line segments are then merged into object boundaries

- After the edge detection operation, the next step is to threshold the results
- One method to do this is to consider the histogram of the edge detection results, selecting the best valley
- With a bimodal histogram (a histogram with two major peaks), an analytical solution is available to find a good threshold value

Figure 4.3-11: EDGE DETECTION THRESHOLD



This method works best with a bimodal (two peaks) histogram.

- A bimodal histogram is typical for computer applications where we have one object against a background of high contrast
- This method provides a theoretically good solution based on the assumption that each peak has a Gaussian shape and the peaks are fairly well separated
- This method is called *minimizing within group variance*, or the *Otsu method*

- **Otsu Method:**

  - Let *P(g)* be the histogram probability for gray level *g*, which is simply the count of the number of pixels at gray level *g* normalized by the total number of pixels in the image, and is given by:

$$P(g) = \frac{1}{(\#\,Rows)(\#\,Columns)} \sum_{I(r,c)=g} \frac{I(r,c)}{g};$$

where (# Rows)(# Columns) is the total number of pixels

Let $\sigma^2_w(t)$ be the within group variance, which is a weighted sum of the variance of the two groups, as a function of the threshold t, defined as follows:

$$\sigma_w^2(t) = P_1(t)\sigma_1^2(t) + P_2(t)\sigma_2^2(t)$$

where

$$P_1(t) = \sum_{g=1}^{t} P(g)$$

$$P_2(t) = \sum_{g=t+1}^{Maxgray} P(g)$$

$$\mu_1(t) = \sum_{g=1}^{t} g \times P(g) / P_1(t);$$

$$\mu_2(t) = \sum_{g=t+1}^{Maxgray} g \times P(g) / P_2(t)$$

$$\sigma_1^2(t) = \sum_{g=1}^{t} [g - u_1(t)]^2 P(g) / P_1(t);$$

$$\sigma_2^2(t) = \sum_{g=t+1}^{Maxgray} [g - u_2(t)]^2 P(g) / P_2(t)$$

Where *Maxgray* is the maximum gray level value

- We simply then find the value of the threshold *t* that will minimize the within group variance, $\sigma^2_w(t)$
- This can be done calculating the values for $\sigma^2_w(t)$ for each possible gray level value and selecting the one that provides the smallest $\sigma^2_w(t)$
- We can usually streamline this search by limiting the possible threshold values to those between the modes, the two peaks, in the histogram

---

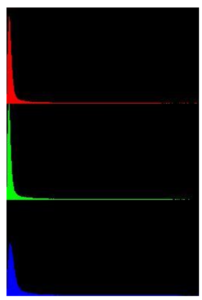
- Often, the histogram of an image that has been operated on by an edge detector is unimodal (one peak), so it may be difficult to find a good valley
- A method that provides reasonable results for unimodal histograms is to use the *average value* for the threshold
- With very noisy images and a unimodal histogram, a good rule of thumb is to use 10% to 20% of the maximum value as a threshold

Figure 4.3-12: Average Value Thresholding

Original image

Image after Sobel edge detector

Unimodal histogram of image after Sobel



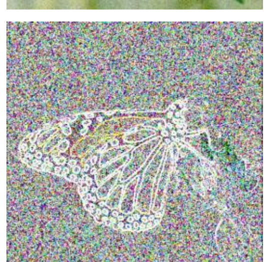
Sobel image after thresholding with average value

Figure 4.3-13: Thresholding Noisy Images

Original image with Gaussian noise added (zero mean, variance = 800)

Sobel edge detector results

Threshold on Sobel at 10% of maximum



Threshold on Sobel at 20% of maximum

- After thresholding*, edge linking* is used to merge the existing edge segments into boundaries
- The simplest approach connects each point that has passed the threshold test to all other such points that are within a maximum distance
- This method tends to connect many points and is not useful for images where too many points have been marked

- Alternately, we can perform edge linking on the edge detected image before we threshold it
- If this approach is used, we look at small neighborhoods (3x3 or 5x5) and link points similar in both magnitude and direction
- The entire image undergoes this process, while keeping a list of the linked points, which determines boundaries

- The *Hough transform* combined with the *snake eating edge linking algorithm* is one method to use the Hough transform for segmentation via boundary detection

- The Hough transform can be extended to search for any geometric shape that can be described by mathematical equations, such as circles, ellipses or parabolas

- To extend this concept we simply define a parameter vector and apply the Hough algorithm to this new parameter space

✓**Deep Learning Segmentation Methods**

- Use of many layered CNNs that are trained to take the original image as an input and output a segmented image

- **Semantic Segmentation**: Classify each pixel with a particular class, or object type, by assigning it the class label or color

- **Instance-based Segmentation**: Classify each object with its own individual label, such as when we are trying to identify a person's face, e.g. "John"

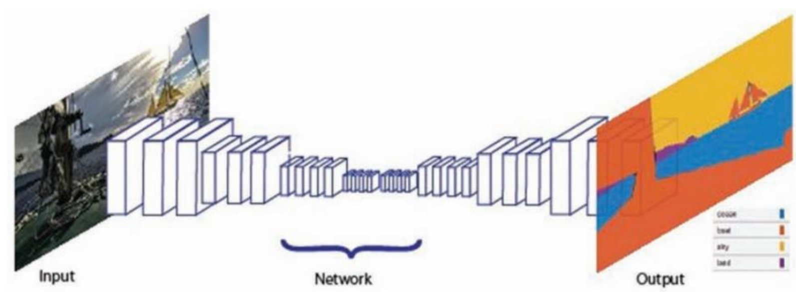


FIGURE 5.5-1
Semantic segmentation process. The goal for the network with semantic segmentation is to label each pixel as belonging to a specific class.

- **Convolutional Neural Network:**

  - A specific type of ANN
  - With semantic Segmentation, the input is the image and the outputs are the classes to which the pixels are assigned
    - Circles are processing elements: neurons/nodes
    - Typical NN will perform summation process, but for a CNN will perform convolution
    - Weighted sum of input image, weights are the convolution mask or filters

- **CNN (cont…) :**
  - After summation output of neuron is controlled by the activation function (ReLU)
  - Without ReLU, deep network will learn very slowly and may not ever reach a point of efficacy
  - Final output layer typically uses a softmax or sigmoid activation function

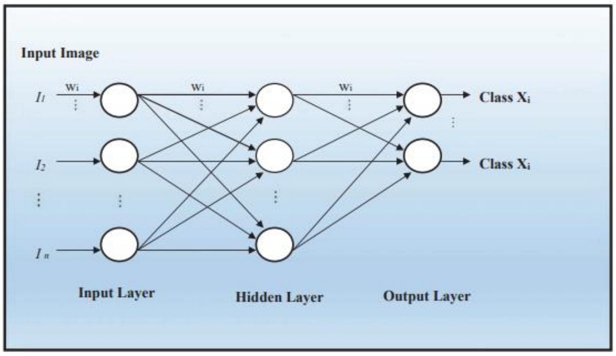


FIGURE 5.5-2
Artificial neural network. Here, we show a neural network architecture with an input layer, where the image is the input, one hidden layer and the output layer. In this case, the output layer corresponds to the classes. The circles are the processing elements, neurons, and the arrows represent the connections. Associated with each connection is a weight (w_i, shown only across the top for clarity) by which the signal is multiplied. These weights are adjusted during the training (learning) phase.

- **CNN (cont…) :**
  - A standard CNN has many layers, and a number of these are convolution layers
  - A convolution layer in a CNN is specified by the number and size of the filters
  - Called Deep learning due to convolution layers are numerous, so the network has many layers, the more deeper we get into the network the more specific and complex the filters are
  - The Learning aspect of the process requires the network to be exposed to many many sample images along with the desired output image

- **CNN (cont…) :**
  - During Learning/training many samples and the correct results are used via various mathematical processes that search in the very high-level feature space to minimize some error function or maximize a success function
  - During the training the weights of the filters are adjusted to achieve optimal results

•CNN (cont…) :

- Generally Network used for Segmentation uses the Encoder/Decoder structure
- Encoding: Image is downsampled (averaging, max pooling for CNN) and the lower resolution help find features to discriminate between classes
- Decoding: the image is upsampled back to its original size – transposed convolutional layer – stretching the image and zero-padding – adding rows and columns of zeros and then performing convolution

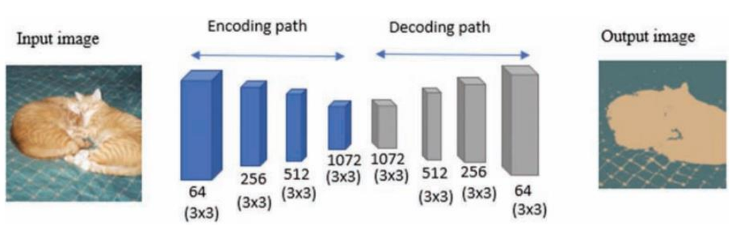


FIGURE 5.5-3
An encoder/decoder structure is commonly used for image segmentation with a CNN. The image is downsampled during encoding, which reduces image size, and is upsampled during decoding to generate the final image. The numbers under each layer represent the number of filters in each layer, while the numbers in parenthesis represent the filter size.

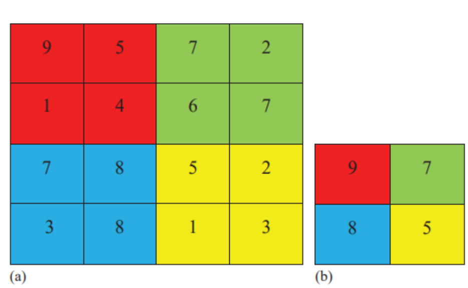


FIGURE 5.5-4
Max pooling. This illustrates how max pooling works to downsample an image in a CNN by selecting the maximum value in a small region. (a) The image and (b) the result of max pooling with a filter size = 2 × 2.

- **CNN (cont…) :**
  - In practice, the developer who wants to use a CNN segmentation can starts with one of the standard networks that has already been trained with millions of images, and thousand of classes, and develop an output layer that is specific to the application
  - The filters developed in these networks – first layers generate filters that extract primitive features – edges
  - Deeper in the network – we see filters for corners, shapes, textures etc.

✓**Combined Segmentation Approaches**

- Image segmentation methods may actually be a combination of region growing methods, clustering methods and boundary detection
- In boundary detection, heuristics applicable to the specific domain must be employed in order to find the true object boundaries

- Finding boundaries of different features, such as texture, brightness, or color, and applying artificial intelligence techniques at a higher level to correlate the feature boundaries found to the specific domain may give the best results

- Optimal image segmentation is likely to be achieved by focusing on the application, and on how the different methods can be used, singly or in combination, to achieve the desired results