Sightech Vision Systems, Inc.

Neuro-RAM Execution

It takes Billions of Pixels per second to evaluate the Neuro-RAM Hierarchies

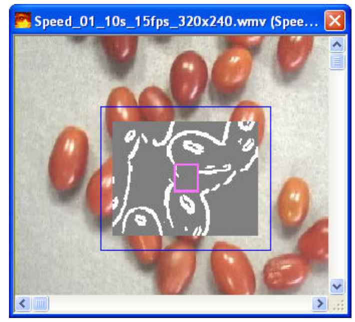
The Convolutions, Features, and Regions represent 3 levels of the multi-level hierarchy employed by Neuro-RAM. A large number of pixels makeup a Region-level decision, so we will discuss details regarding this large group of pixels and how we arrive at Region-level hierarchical decisions. We also show that we operate at the huge rate of more than one billion pixels per second for any given Area!

Published 2005

Sightech Vision Systems, Inc.
6580 Via del Oro
San Jose, CA 95126
Tel: 408.282.3770 Fax: 408.413-2600
Email: sales@Sightech.com
Web: http://www.sightech.com/

Table of Contents:

The VIEW mode shows how many pixels are used to arrive at a Region-level decision.



In the figure on the right, you can see the Area bounded by thin blue lines. No pixels are used outside of the blue lines to make decisions in this Area.

A simple Sobel edge-detection operator uses a 3x3 matrix of pixels of input to produce one BW bit (1 or 0) of reduced information.

The convolution-level processing inputs a tight matrix of pixels and produces a binary (BW) value from a convolution algorithm performed on that matrix of pixels. The output of each convolution is one value – centered in the middle of the matrix.

We now show a simple Sobel operator where two weighting matrices are used to caluculate Gx and Gy by multiplying the corresponding weights with the corresponding pixels.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Gx= (-1*p00) + (0*p01) + (1*p02) +
(-2*p10) + (0*p11) + (2*p12) +
(-1*p20) + (0*p21) + (1*p22)

Gy= (1*p00) + (2*p01) + (1*p02) +
(0*p10) + (0*p11) + (0*p12) +
(-1*p20) + (-2*p21) + (-1*p22)

M= abs(Gx) + abs(Gy), If (M >= Threshold) B=0, If (M < Threshold) B=1

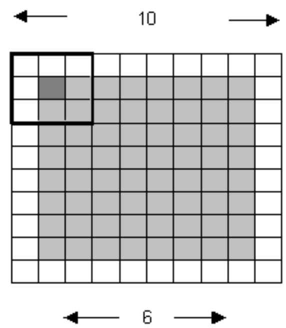
The absolute values of Gx and Gy are added together and compared to a user settable threshold. If the magnitude is equal or greater than the threshold, then the resulting BW pixel is white (1) - otherwise it is black (0).

A 3x3 operator reduces the resulting image size.

You can see that a 3x3 matrix of pixels of image data are highly reduced to produce one bit of BW information which is placed at the center of the matrix. This convolution performs an "edge detection" rendition of the original image. Lowering the user-settable threshold makes the convolution more sensitive and will result in detecting finer edges.

If the original image was 10x10 (100) pixels, then the convolved BW image will have

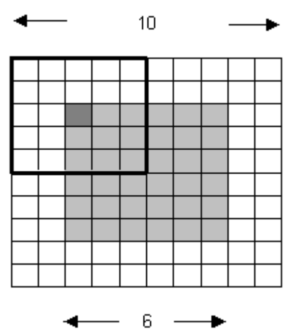
8x8 (64) pixels. All 100 pixels of the original image were used to produce this smaller image.



A 5x5 operator reduces the resulting image size even more.

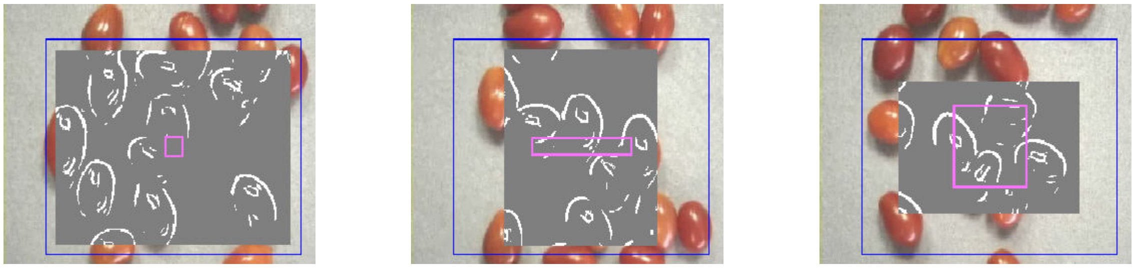
You can see that a 5x5 matrix of pixels of image data are highly reduced to produce one bit of BW information which is placed at the center of the matrix. This convolution performs an "edge detection" rendition of the original image. Lowering the user-settable threshold makes the convolution more sensitive and will result in detecting finer edges.

If the original image was 10x10 (100) pixels, then the convolved BW image will have 6x6 (36) pixels. All 100 pixels of the original image were used to produce this much smaller binary image.

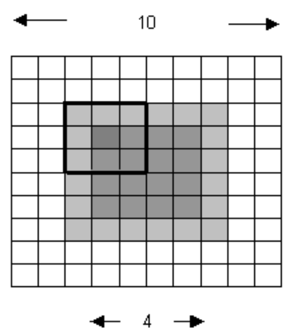


The next hierarchy level, the Feature reduces the data even further.

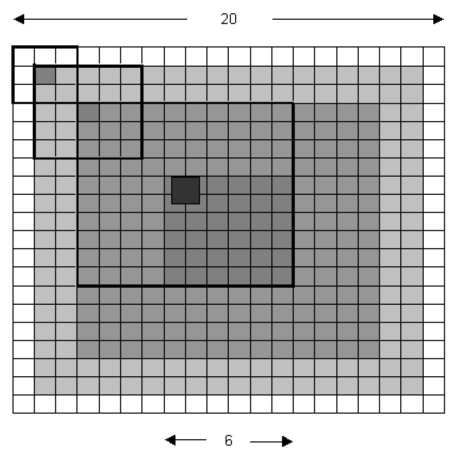
In the figure to the right, you can see a small pink rectangle in the center of the image. This is the outer boundary of the Feature. Features come in many sizes and shapes – see some examples below:



Notice the space between the outer blue border and the darker gray portion of the image in the center. This "empty" space contains pixels are used to create Feature-level decisions in the dark gray portion of the image. The width of this "empty" border is sum of the extra pixels needed by the image convolution and the extra pixels needed by the Feature to create a single binary decision that is centered in the middle of the Feature. In our example, if the Feature was a tiny 3x3, then after the 5x5 convolution, we would have a decision block of only 4x4 or 16 pixel positions.



The Region hierarchy level takes a block of 10x10 features and makes a decision for that block.



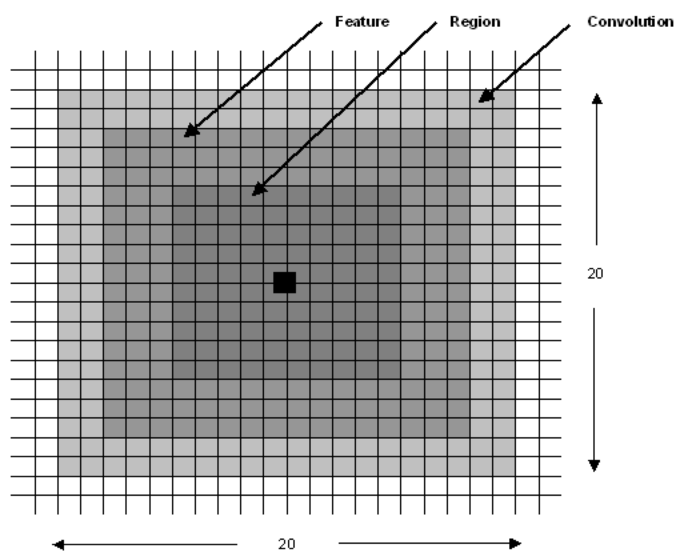
This is a large further reduction of the data. The convolution output fed into the Feature-level decisions. Now, the Feature-level decisions feed into the Region-level decisions. If we had a 20x20 image area, a 3x3 convolution, and a 5x5 feature, then we would have a

7.5 pixel "empty" border around the edge of the Area – to take into account for the Region-level decision.

The black dot is the center of the Region – this where we consider the decision happening. In the above example, each binary Region-level decision uses a 16 x 16 = 256 matrix of pixels. Since each pixel has 8-bits of graylevel information, this is a huge reduction from 256 x 8 = 2048 bits way down to 1 bit.

Often, more than one billion pixels per second evaluation rates are required to reach an Area-level decision.

This diagram to the right shows the 20x20=400 pixels involved with a 10x10 Region, 7x7 Feature, and a 5x5 Convolution. There is a huge amount of pixel-level information that goes into a single bit binary decision. The representative XY position of the binary Region-level decision is shown by the black dot in the center of the diagram.



This massive decision process executes for the Region center point shown, moves over 1 pixel position, and evaluates again. There are 400 pixels being evaluated for each Region-level pixel position! Depending on the size of the Area, we often evaluate more than 1 billion pixels per second!