working with images

Edge Detection

In the computer vision paper of **Viola and Jones**, a **weak classifier** was used for **edge detection** in images by moving the filter around an image and subtracting what is in the white area from what is in the black area:

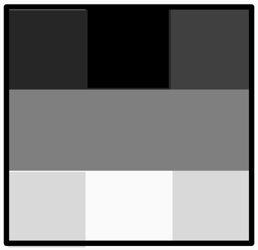
The weak classifier does not find much difference between the areas on the man's forehead, thus no edge is detected. The same result happens when the weak classifier is applied to the man's beard. However, when the weak classifier is applied to the man's eyes, the subtracted areas result in a large number because the man's eyes are much darker than the surrounding area of the man's cheeks. Thus, the weak classifier successfully detects an edge.

Sobel Edge Detector

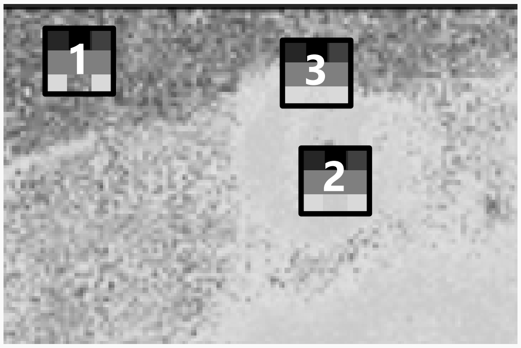
A more powerful **edge detector** is the **Sobel Edge Detector**, which is slightly different than the Viola and Jones weak classifier. The **Sobel Edge Detector** puts more weight on the center gradients of the image than on the outer edges.

Weak Classifier

Sobel Edge Detector

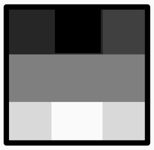


In the example below, a Sobel Edge Detector is applied to a section of a noisy image. The detector appropriately does not register anything in areas **1** and **2**; the value of the gradients are ≈ 0. The detector does, however, register a large gradient (change) when the pixel intensities are subtracted in area **3** and determines the presence of an edge.

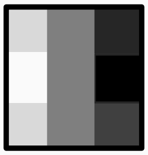


Translating the Sobel Edge Detector from greyscale into numerical value produces the resulting matrices:

$$S_{horizontal} = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$



$$S_{vertical} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$



In the above matrices, 0 represents neutral. The outer pixel are weighted accordingly, with the middle values of the outer pixels weighted most heavily. In the horizontal edge detector, the top intensities will be subtracted from the bottom. In the vertical edge detector, the right intensities will be subtracted from the left. Note that the inclusion of +1s and -1s on the matrix edges are for smoothing purposes. The model would be unstable in application if just the +2s and -2s were applied

To illustrate, the vertical edge detector matrix is decomposed to show that actual vertical edge detector and the corresponding smoother which is similar to a Gaussian (normalizes the edge):

$$S_{vertical} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & +1 \end{bmatrix}$$

y axis smoothing (Gaussian-like)

x axis derivative (vertical detector)

$G_{horizontal} = S_{horizontal} \times A$ → Image with edges detected in the **horizontal** direction.

$G_{vertical} = S_{vertical} \times A$ → Image with edges detected in the **vertical** direction.

$$G = \sqrt{G_{vertical}^2 + G_{horizontal}^2} \qquad \theta = \operatorname{atan}\left(\frac{G_{horizontal}}{G_{vertical}}\right)$$

The above notation illustrates the convolving of an edge detector with an image. The **Sobel Edge Detector** will register edges in both the **horizontal** and **vertical** directions. The **Pythagorean Theorem** is applied to compute the total magnitude of the edge. The angle of the edge is obtained through the **tangent** of the angle being the change in $y \rightarrow \Delta y$ over the change in $x \rightarrow \Delta x$. As seen in the above notation: $\text{edge angle} = \frac{\Delta y = \text{horizontal detector gradient}}{\Delta x = \text{vertical detector gradient}}$. The results are plotted below: