# Exercise Sheet 8

Docente: Miguel Tavares Coimbra

1. **Texture descriptors – Local Binary Patterns.** Create a 256 coefficient descriptor that consists of a histogram of texture binary patterns for each pixel.
   - Idea: Compare the *intensity* value of a pixel with its 8 neighbors.
   - Start on the upper left corner.
   - The result of the comparison is 1 if the value of the neighbor is larger or equal to the pixel value. Otherwise the result is 0.
   - Combine all the results into a single byte (using *bit-shifting*), thus creating a *binary pattern* for each pixel.
   - Create the histogram of the occurrences of each *binary pattern* for the whole image.

Binary Pattern: 11101001
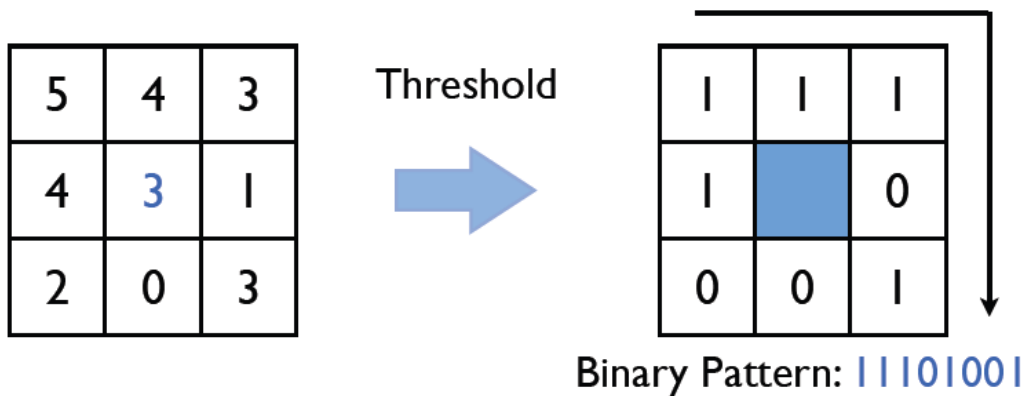
2. **Texture Descriptors – Local Edge Histograms.** Create an 80 coefficient descriptor that consists of a histogram of edge patterns for 16 sub-regions of the image.
   - Divide the image into 16 regions (4x4) with the same number of pixels.
   - For each region, apply 5 edge detectors using the digital filters represented in Figure 3.
   - The filter with the strongest response is the result for each pixel. If this response is weaker than a pre-defined threshold (ex: 11), the result for this pixel should be 'no edge' (hence, not accounted for in the histogram).
   - Create an edge histogram for each block, in which you count the occurrences of pixels with: vertical edge, horizontal edge, 45 degree edge, 135 degree edge, non-directional edge.
   - Concatenate the 16 histograms into a single 80 coefficient vector.
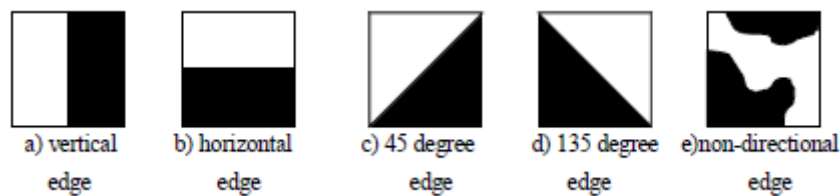   *Note: For more details read the support file: "VC_1415_P8_LEH.pdf"*

a) vertical edge   b) horizontal edge   c) 45 degree edge   d) 135 degree edge   e) non-directional edge

Figure 2 – Five types of edges

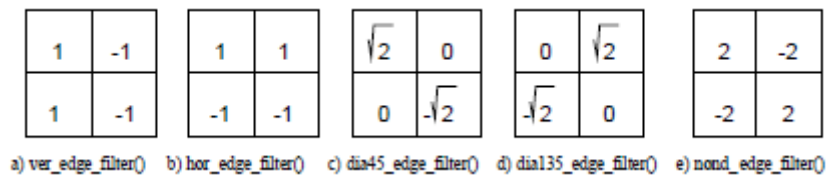a) ver_edge_filter()   b) hor_edge_filter()   c) dia45_edge_filter()   d) dia135_edge_filter()   e) nond_edge_filter()

Figure 3 – 2x2 filter masks for detecting edges