# Question 1:

Code (python notebook) and results csv file attached in zip file.

# Question 2

Code (python notebook) attached in zip file.

# Question 3

1. Let denote an indicator for the presence of image in the row and column of a very long and wide canvas (for simplicity we assume a black and white image).

Let denote a matrix , which will serve as our filter.

Let denote a measure by which we shift X, i.e., a shift transformation to X, c pixels to the right will be denoted by

Next, we apply a 2-d convolution function (denoted by f) with the filter H, to X after being shifted c pixels to the right and c pixels up (without loss of generality) –

Alternatively, we could first apply f onto x and only then apply the shift transformation, resulting in the same formula above (this is easily seen as is not affected by the transformation, only ).

1. A shift transformation will be equivariant to convolution if the image is not shifted outside the boundaries of the canvas, meaning the filter will be able to perform on all the pixels, i.e., the content of the image remains the same and no information is changed in the image.

This is true for any filter size.

1. Max-pooling’s results are invariant to a one-pixel shift only in case the windows are NOT disjoint (locally true).

Given a 3x3 filter with stride 3, windows are disjoint hence resulting in not high enough frequency for sampling each portion of the feature map – in violation of Nyquist–Shannon sampling theorem.

1. Shared weights across all inputs are a sufficient condition.

Let denote the vector of inputs. And let denote the node in the hidden layer. Given , our activation function, is monotonous, each node’s output is ). Thanks to the property, for each node, the output is permutation invariant.

Note that under these conditions you lose the spatial relationship, which is needed for computer vision.

# Question 4

Code (python notebook) attached in zip file.