EVA-5 Week 1 Assignment

# 1.What are Channels and Kernels (according to EVA)?

Channels: Contextually specific information that are grouped and held together in a container is called channel. The channel contains collection of specific features, colors, frequency spectrum, Audio track etc.

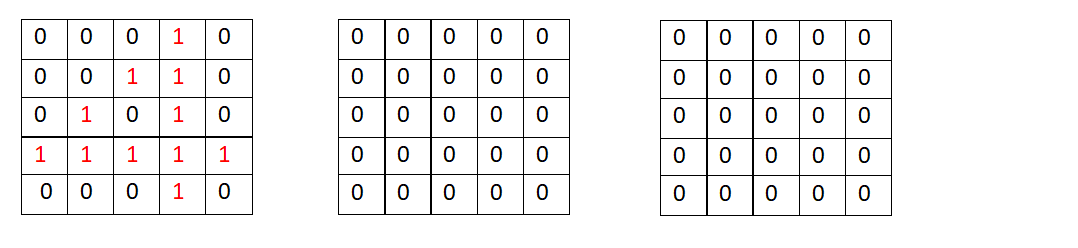
For example:

1. RadioCity is a radio station whose frequency modulation is 91.1 Hz. This frequency is allocated to RadioCity Channel.
2. When recording a music, the sounds of different musical instruments are collected in a separate container called track and later mixed to combine them to form a music. The track which contains specific sounds of musical instrument are called Channel.

Coming to an Image, the image is a combination of light intensity and color information. So, the colors are separated out and grouped with their intensity in a container called channel.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 |

5X5 Image of number 4



Red Channel Green Channel Blue Channel

Kernel: A filter that is used to extract specific information from given input is called Kernel.

 The tea filter shown in above figure extracts the tea essence while separating out the tea residues. Same way to extracts features from image we use filters called Kernels.

Image will have features for example like thin eyebrow, blue pupil, silky hair etc made up of edges, gradient, texture etc. Kernels are used to extract these features from given image and collection of specific features are assigned to specific channels.

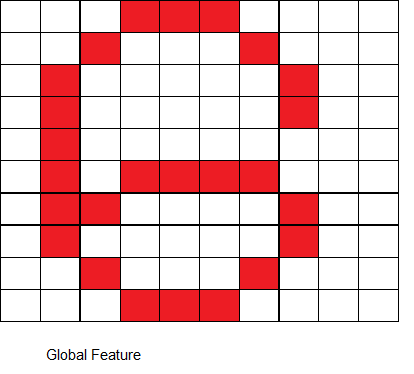
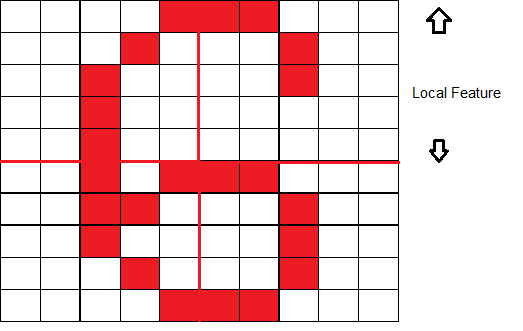
<from Pavan & Varsha>

# 2. Why should we (nearly) always use 3x3 kernels?

 or or  or 

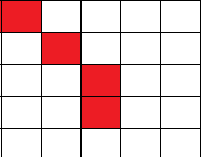
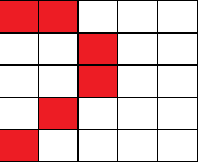
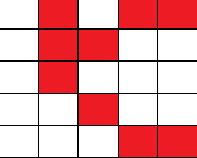
Images are made up of edges, gradients, texture, patterns and object. The edges are the basic building blocks of an image. The small portion of these edges put together is called **local feature** in a image.

Example



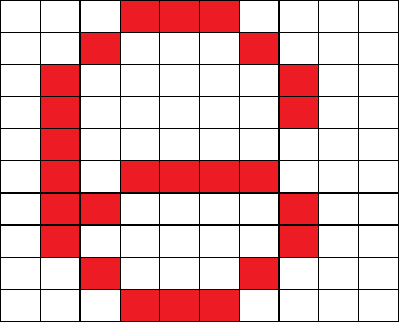
From the above example we can understand that local features are multiple feature points on the image and global feature is mostly an entire image.

With local feature extraction we can identify granular feature which makes easier to extract similar patterns of features. For example with local feature

we could extract features of 9,8,0 etc. With global feature “6” we can only extract “6” present in any other image. Hence extracting local feature makes the architecture more robust.

To extract local features, we look at small group pf pixels which is called “receptive field”. With smaller receptive field the amount of information of feature extracted is more. These features are useful in later layers when forming an object. Hence, we try to keep kernel size as low as possible to extract local features.



|  |  |  |
| --- | --- | --- |
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

We need to arrive at this weight for 3 X 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -1 | -1 | - | - | - |
| - | - | - | - | - |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

We need to arrive at this weight for 5 X 5

When we initialize 3X3 and 5X5 matrix to a random value at the begin of training say,

|  |  |  |
| --- | --- | --- |
| 2 | 3 | 4 |
| 8 | 8 | 8 |
| 6 | 7 | 2 |

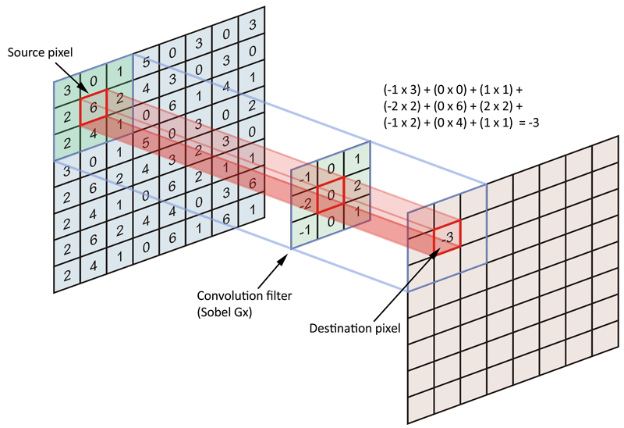
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 5 | 4 | 3 | 2 | 1 |
| 6 | 6 | 6 | 6 | 6 |
| 9 | 8 | 3 | 2 | 4 |
| 3 | 2 | 9 | 7 | 7 |

After back propagation and the weight adjustments, we can see that smaller matrix will have better weight shared among them compared to a bigger kernel. Hence lower kernel size is computationally efficient. Because of low kernel size, multiple kernels must be used to learn much more features, thus more layers are created. With more number of layers we can learn more complex features.

So why not 1 X 1 ?

Kernel size of 1X1 will basically extract each pixel feature, which is fine grained with no information from neighboring pixel. So 1X1 is not used extensively.

So why not 2 X 2 ?



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |

|  |  |
| --- | --- |
| -1 | 0 |
| -1 | 0 |

|  |  |  |
| --- | --- | --- |
| -1 | 0 | 1 |
| -1 | 0 | 1 |
| -1 | 0 | 1 |

|  |  |  |
| --- | --- | --- |
| 3 | 0 | -3 |
| 3 | 0 | -3 |
| 3 | 0 | -3 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | -2 | 0 |
| 0 | 0 | -2 | 0 |
| 0 | 0 | -2 | 0 |
| 0 | 0 | -2 | 0 |

In the above diagram we can see the output odd Kernel size of 3X3 is centered along the red line and other values are symmetric across it. If we choose even size filter like 2X2 the symmetricity will be lost, and there will be distortion in the image.

Hence 3X3 is always chosen to be working perfect in most cases.

