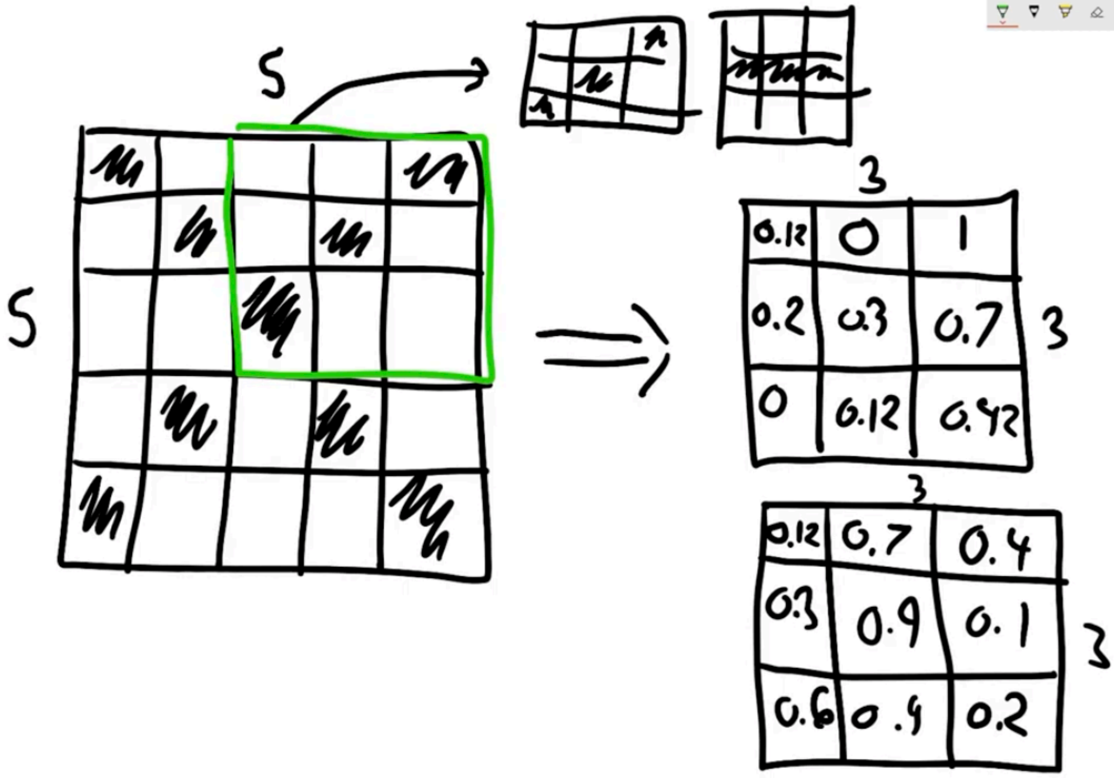
- ***How convolutional networks / layers work***
    - -> in this example there is an image
    - -> the image is made of pixels which can be divided into a grid
    - ***properties of convolutional maps***
        - -> a feature map / filter
        - -> properties of the convolutional map
            - the imput size <- the number of filters and the same size of the fileters
            - -> each of the filters is looking for a pattern in the image (e.g the presence of paws)
        - -> the filters are found by the convolutional neural network
    - ***how we look for filters in the image***
        - -> one filter might look for a diagonal line, a horizontal line or a straight line
        - -> there are 32 different ones in this example
        - -> there is a filter size
        - -> 3x3 spots in the image
        - -> defining how closely those spots match
        - **-> doing the dot product of the filter with the image we have**
            - **-> the component of that feature in the direction of this image**
            - **-> and each filter is looking for the presence of a different feature in the image**
            - -> you can store the outputs of the filters (different dot products with the image we have) in a matrix
            - -> this is a response map
            - -> there are 3x3 samples -> r, g, b

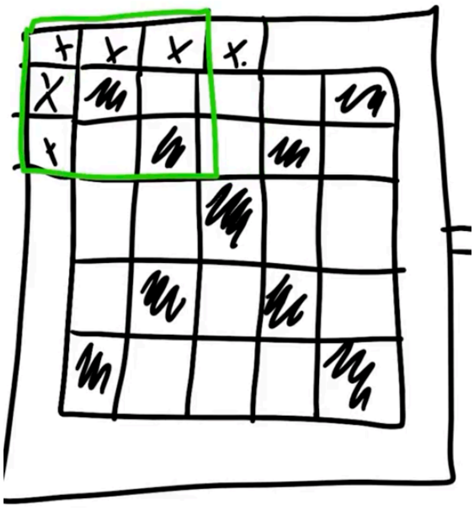


0.12	0	1
0.2	0.3	0.7
0	0.12	0.42

0.12	0.7	0.4
0.3	0.9	0.1
0.6	0.9	0.2

            - -> it's telling you about the presence of features in the image
            - -> this is done for many different features -> computationally heavy
- ***Pooling***
    - -> there is an output feature map
    - -> running the calculations is computationally heavy
    - -> looking at for example - the presence of different line types, to the combinations of different lines which we have -> pooling the information about the different lines which we have
        - -> we are building up to this, first from determining if lines are present - and then to finding correlations between those different lines which are present -> to identify the presence of e.g eyes in an image

- ***-> you can add padding around the image***
    - -> so that you can draw a square which makes the pixels on the edge in the centre
    - -> <u>this allows you to include the pixels which are on the edge of the image</u>



- ***-> stride***
    - -> this is how much the sample box is moved each time
    - -> for example 2px -> moving the sample box by 2px each time the calculations are ran
- ***-> a pooling operation***
    - -> min
    - -> max
    - -> average
    - -> taking specific values from a sample of the output feature
    - -> sampling 2x2 areas of the feature map
        - -> taking the min / max / average values of those points
    - ***-> pooling is taking an average from multiple filters***
        - -> the filter contains a matrix of dot products for a different feature with the pixels int he image
        - -> run that multiple times with different filters that check for the lines
        - -> then take the average of them and put them into another matrix -> this is pooling
        - -> this is done with a stride of 1 (in this example)
        - -> max pooling can be done
        - -> whether the feature exists
    - ***-> min / max / average tell you different things***
        - -> it's a way of combining the different matrices
        - -> the average presence of the feature, does the feature exist, does the feature not exist
        - <u>-> the three main properties of each convolutional layer are the input size, the number of filters, and the sample size of the filters</u>