1.a)

Equivariance applied to convolutional networks means that when the input image is transformed by a certain amount, the output is transformed in the same way.

Therefore, if we see a transformed version of an image input, we don’t need to do any new computation. We just need to find the same function repeated throughout the code and once we realize that we are seeing many copies of the same function, apply the same transformation in the output space. In this way, we remove the need for the network to learn transformed copies of each feature, resulting in a massive decrease of parameters that we need to train.

Summing up, the equivariance in convolutional neural networks is crucial to classify group of images transformed by translation, rotation, hue, scale or combination of those, with more efficiency.

2.a)

When we talk about filters in convolutional layers, we are talking about feature maps (aka activation map), that is the result of applying the filters through the pixel values of an input, such as the input image or another feature map.

The idea of visualizing a feature map for a specific input image would be to understand what features are detected or preserved. The expectation would be that the feature maps close to the input detect small or fine detail, whereas feature maps close to the output of the model capture more general features (less detail). This pattern is to be expected, as the model abstracts the features from the image into more general concepts that can be used to make a classification (higher layers learn more “complex” features based on the activations from the previous layers).

The number of output channels define the number of kernels, which convolve over our input to get feature maps. Since we have two different convolutional layers, in the first one we will have 16 learned kernels of 3x3. On the second we will have 32 of 3x3.