# Convolution - A3 Writeup

Our project aims to visualize convolutions. To put this project into context, our final project will be about convolutional neural networks, so it made sense to start by visualizing and explaining convolutions in A3. Rather than exploring a particular dataset, in our project the viewer interactively explores the convolution algorithm (input, kernel and output) on a selected set of images from the CIFAR-10 dataset, using several commonly used kernels. Our visualization revolves around three main elements: the input image, kernel and output image. We ordered those elements from left to right in the visualization to convey the ordering of first having the input image, then applying the kernel, which generates the output image. For convolution it is important to see which pixels are involved in a convolution and where the convolved value shows in the output image. Initially we only used highlighting through color for the involved pixels and the output pixel. But changing the pixel color was confusing, since the pixel color directly corresponds to the underlying value, so by changing it the underlying convolution calculation does not correspond to the visualization. Thus we decided to use outlines for highlighting. To further emphasize the “path” of a patch from the input image, we decided to draw lines from the input patch, to the kernel and to the output pixel. At first we only used MNIST images, but realized that they are not interesting enough and thus we started using images from the CIFAR-10 dataset. Originally, all of our images were also gray, because in order to do convolution with a color image, one actually needs a 3D kernel. This seemed difficult to visualize effectively. But we realized that having only gray-scale images made it difficult to recognize the objects in the images, since the images are small (32x32). Thus we decided to use color images and use the same 2D kernel for each color channel. This made the visualization much more interesting, since it added color. It also made the objects in the images easier to identify. Another solution to this would have been to use larger images. However, to fit larger images on the screen we needed to make the pixels smaller. This would mean the user has to do too much brushing to perform the convolution, making the task tedious. We found that using a size of 32x32 struck a balance between the amount of work by the user to convolve the whole image and the effective communication of the concepts. Still, the convolution by brushing can get tiring for the user, so we also added an auto convolve button, which runs an animation that convolves the whole image automatically. We found that this was effective in visualizing how the output image gets built up pixel by pixel through the convolution.

At first, the viewer had full control over everything and could select an image and a kernel. However, this provided too little guidance, so we decided to add narrative by introducing the user to different kernels step-by-step and explaining the convolution algorithm and pointing out interesting aspects. We decided to use a slide format, where the users use previous and next buttons to navigate between the slides. On the very first slide we decided to only display numbers rather than an image, to make it clear that convolution can be seen as an abstract operation on numbers. We deliberately chose simple numbers so the viewer can do the calculations in their head. In the second slide, we apply the same idea to images, where the numbers now represent color. In the following slides we explain different kernels by pointing out interesting features in the images and explaining what the kernel does, which the viewer can immediately see by performing the convolution. The final slide corresponds to our original visualization, where the user has full freedom over the image and the kernel. This way the viewer is introduced to the visualization and the concept in a guided manner, but later has the freedom to do whatever they want. On the final slide, selecting a kernel also shows a description of the kernel right above the kernel matrix. We added this since it is useful to have this information available without having to go back to the slide where the kernel was explained. We also added tooltips to the kernel selection and all buttons, so that users get extra information when hovering over the list of kernels or the buttons.

During the development process, we regularly met in person to discuss what our visualization should look like and to discuss who works on which aspect. Overall, this consumed approximately 3 hours. This estimate does not include time spent communicating through text. Splitting the work was more difficult than expected. We decided to split up the visualization as best as we could. Usually each person worked on some aspect of the visualization individually and then we created pull requests to merge our work together. For instance, one person worked on the image loading and parsing them to give us a 3D tensor of color values, while another person focused on displaying the images in grids and another person worked on implementing the basic interactions. But many features are very closely related and although everybody had a main area of focus, everyone worked a little bit on every feature. It was most difficult and time consuming to get the three main components (the input image, the kernel and the output image) to interact with each other properly, since they all related to each other and are complex visual elements consisting of many svg's, rectangles, text elements and line elements. It was also time consuming to figure out an effective but short narrative and to give the viewer as much freedom in exploring the visualization while still providing guidance to make sure the viewer understands the important concepts. In total, we estimate that we spent 32 hours (hours of each individual combined) in working on the visualization.