Trevon Woods
Computer Vision - ITAI - 1378
Patricia McManus
09/05/2025

Image Processing

What surprised me most was seeing how the kernel, used in most CNN convolutional layers, can detect vertical and horizontal edges in an image. I had a general idea of what kernels did before, but this exercise really helped me see what a CNN extracts in its early layers when it applies kernels to create feature maps. Another thing I found interesting was realizing that kernels are essentially the components that make up filters. Different kernels can completely change the effect of an image, producing results like blurring, sharpening, brightness adjustments, or edge detection. The most challenging part was wrapping my head around histogram equalization at first, since it involves analyzing the distribution of pixel intensities rather than just applying a kernel locally. On a larger scale, operations such as Histogram Equalization caught my attention as well. I discovered that it analyzes the distribution of pixel values across the entire image and redistributes them to improve contrast which is a fascinating concept.

The mathematical operations in this notebook directly produce the visual effects you see because every pixel in a digital image is represented by numbers (or sets of numbers, in the case of color images). When we apply mathematical operations to these numbers, we alter the pixel values. These changes translate into the way the image appears on screen. Every effect demonstrated in this lab, whether it's brightening an image, detecting edges, or applying an artistic filter, is simply the result of transforming the image's numerical representation. In other words, by changing the numbers, we change what we see.

These operations connect directly to AI tools like Nano Banana. Convolutions with kernels form the backbone of CNNs, which power many modern AI image-processing systems. While we manually applied kernels for tasks such as sharpening and edge detection, AI models learn thousands of these filters automatically from data during training, allowing them to capture increasingly complex features. Similarly, when we combine kernels to create filters like "soft glow" or "Instagram-style" effects, it mirrors how AI systems perform style transfer. The key difference is that AI learns not only the filters but also the best ways to combine them, producing more sophisticated and nuanced outputs. Tools like Nano Banana then allow users to control these outputs through natural language which makes these advanced image manipulation tools accessible and intuitive.

A clear real-world application of these techniques is photography and image editing. This is perhaps the most direct connection, since both traditional editing tools like Photoshop and smartphone camera filters rely on many of the same operations we explored: point operations (brightness, contrast, color correction), neighborhood operations (blurring, sharpening), histogram adjustments, and geometric transformations. Beyond editing, these techniques also power a wide range of computer vision tasks such as object detection and recognition, image segmentation, and image registration.

The medical field provides another powerful example. Histogram equalization can enhance the contrast of X-rays, CT scans, and MRIs, potentially making it easier for doctors to diagnose conditions. Edge detection may also assist in identifying tumors or other abnormalities. Outside healthcare, image-processing methods are widely applied in manufacturing and quality control (detecting defects, measuring dimensions), security and surveillance (facial recognition, activity monitoring), remote sensing and satellite imagery (enhancing details, analyzing land use), and even entertainment and special effects. These represent only a fraction of current applications, and with rapid progress in both traditional techniques and AI, the range of possibilities continues to expand.

Looking ahead, I see practical opportunities to integrate what we learned in this lab into future projects. For example, I could use traditional preprocessing techniques, such as sharpening kernels to enhance image clarity, before feeding data into an AI model, particularly if the input images are low-quality. Converting images to grayscale might also be beneficial for models optimized for black-and-white inputs. Additionally, the transformation techniques we practiced could be applied for data augmentation, making training sets more diverse and helping models generalize better to unseen data. In short, these foundational methods not only strengthen AI models but also provide flexible tools for a wide range of real-world applications.

For further exploration I'd like to dive deeper into the kernels and how they extract features in later layers of a CNN during training. I also would like to learn more about data augmentation techniques beyond the typical ones shown in this lab if there are any. Furthermore, my understanding of what happens under the hood of Digital Photography and Image Editing has improved as well. I always wondered how these programs help editors emphasize the warm colors of a sunset just by adjusting the white balance to warmer tones. I see now that it’s all done through convolutions using kernels.