Figure 1: Headphones and carrots.

1 Procedure

We test our edge detector on the two images shown in Figure 1. The mic and headphones image was obtained from a wallpaper website[1] and the carrots image from a stock image website.[2]

We used the Python Imaging Library to import each image and create an array of RGB tuples. Each component in the tuple was added to get the brightness of each pixel. We computed the magnitude of the gradient of this 2D array by using a Gaussian first derivative stencil provided by `scipy.ndimage`. This stencil requires a standard deviation σ to compute how wide the distribution is, i.e. how many neighbors we take into account when computing the gradient at a pixel. Results are shown in figures 2 and 3 for $\sigma = 0.5, 1, 3$, and 5. A smaller choice of $\sigma = 0.2$ yielded no visible edges for both test images.

2 Comments

2.1 Color limitations

The magnitude of the gradient shows where the image substantially changes brightness, indicating an edge. Notably, we only use the brightness instead of each RGB color component. This works for our images because real-world edges in photos correlate with large brightness changes. However, if we were to edge-detect a flag where one band is pure blue $(0, 0, 256)$ and another is pure red $(256, 0, 0)$, the program would fail to find the edge between these bands.

2.2 CAPTCHAs

While edge-detection would be an important step in beating CAPTCHAs, it is probably not the most difficult step. Some CAPTCHA-like images simply overlay colored but nicely typed letters on a noisy background. Edge detection would be the main step in identifying letters in an image like this. The edges could be passed to OCR.

However, real CAPTCHAs often don't rely on a noisy background. Instead, they rely on warping the letters, requiring some interpretation of the edge curvatures and some way to unwarp the image before comparing it to standard fonts. Edge detection would be required to isolate the letters from the background, but the unwarping would be the difficult step.

[1] `http://www.pickywallpapers.com/1920x1080/music/headphones/studio-mic-with-headphones-wallpaper/download/`
[2] `http://pngimg.com/upload/carrot_PNG4985.png`

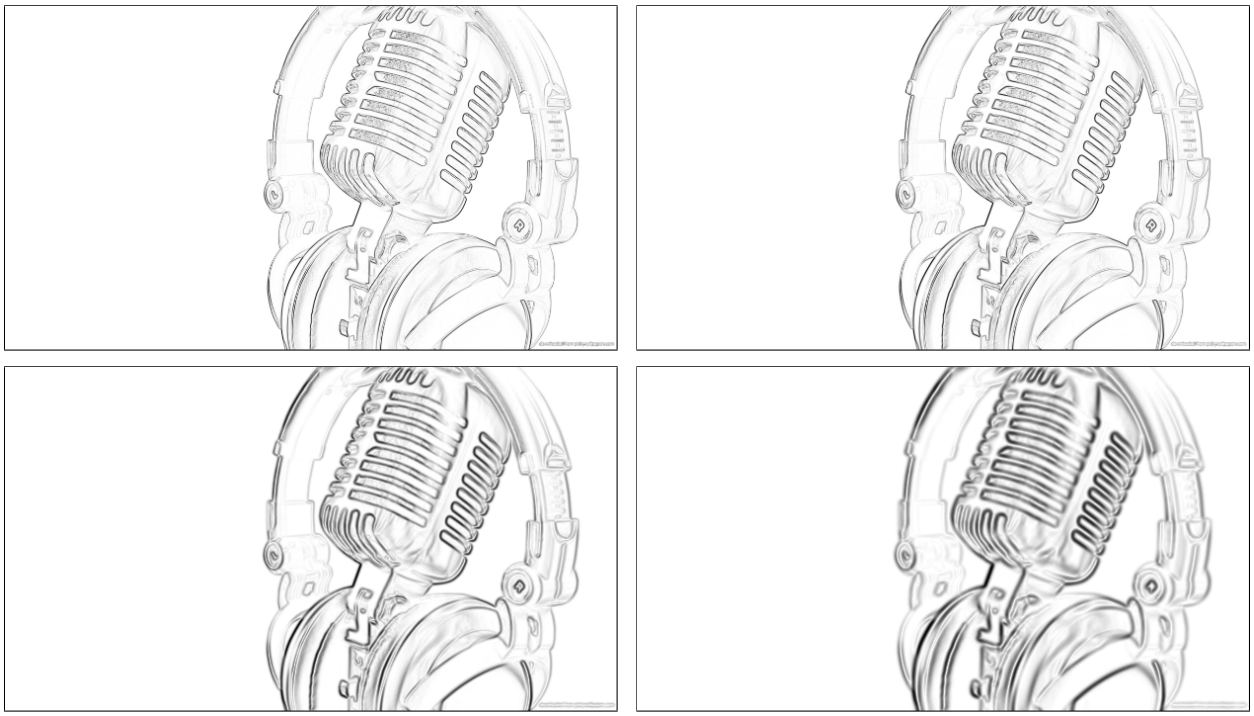

Figure 2: From top left to bottom right: edge detection with $\sigma = 0.5, 1, 3, 5$.

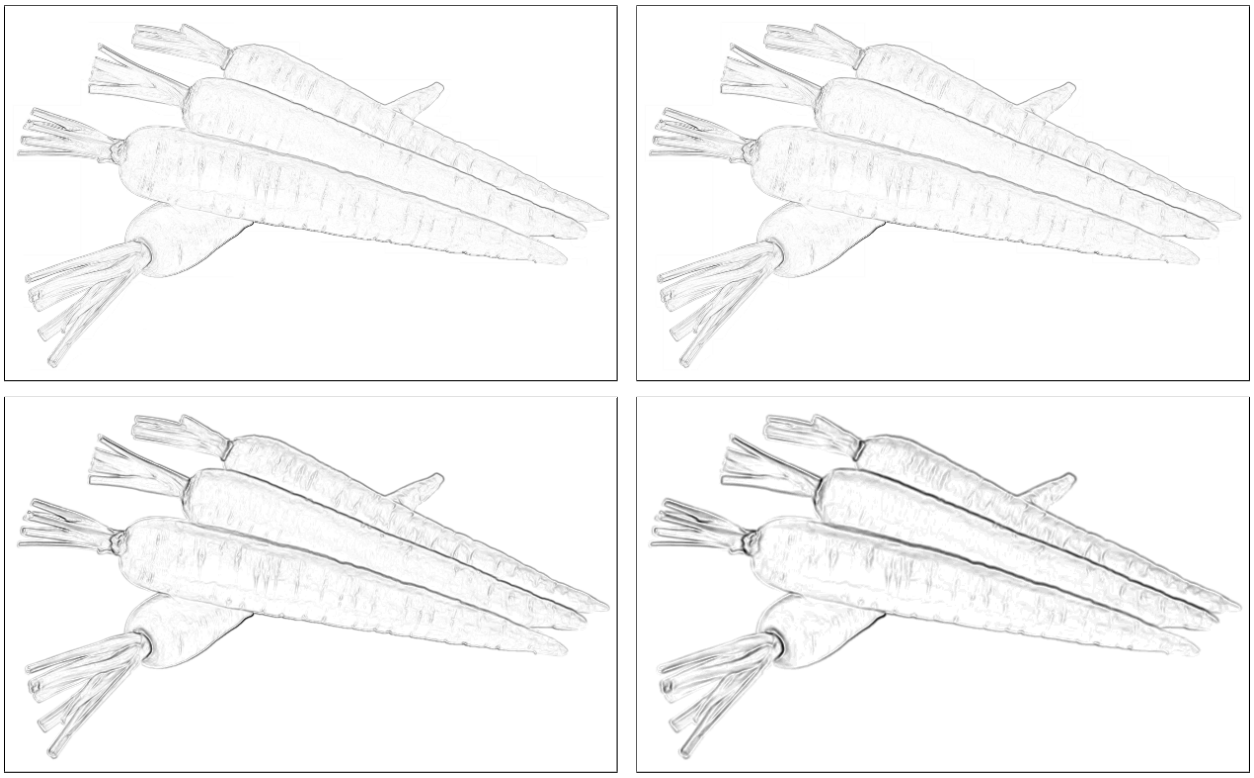

Figure 3: From top left to bottom right: edge detection with $\sigma = 0.5, 1, 3, 5$.