

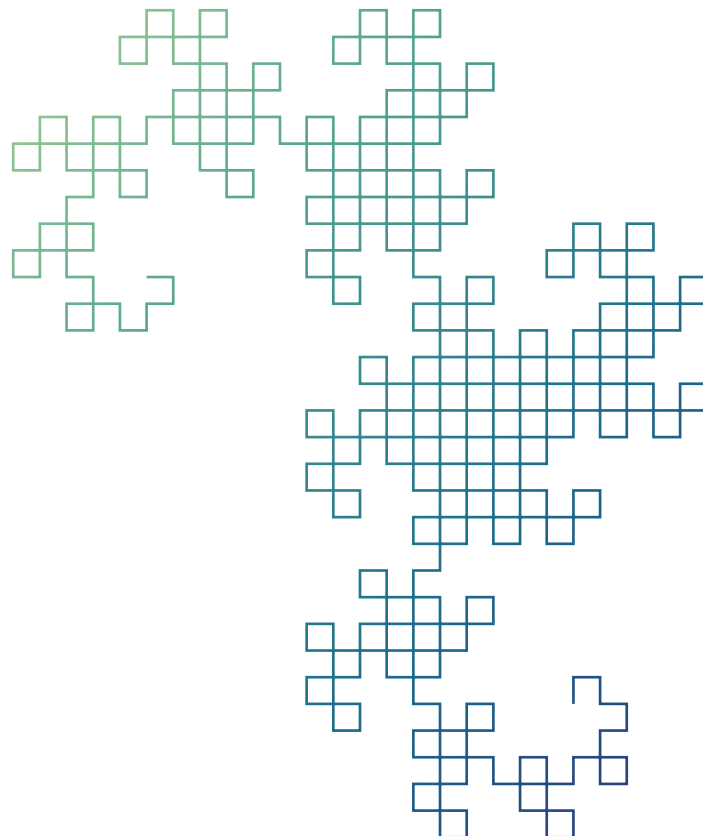
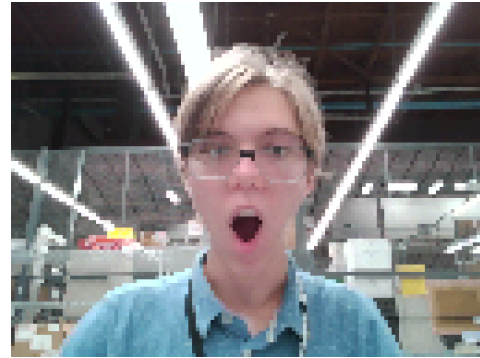
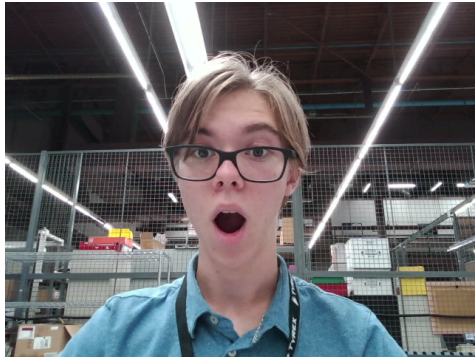
# Using Tensors for Image Manipulation

Jackson Collins

2024-08-02

## I. Pixelating Images with Maxpooling

Maxpooling is an operation that is meant to be used to extract features from a provided dataset. This program chooses to use `max_pooling2d` to perform down sampling on the image. The resolution can be adjusted by changing the kernel size. This version of max pooling returns the highest value in the kernel, but there are other types of max pooling depending on what your desired outcome is. If you want a pixelated image that is more true to life `avg_pool2d` might be a better option.



Dragon curve to fill empty space

## II. Image Preprocessing for Edge Detection

You can perform that functions needed to prepare an image for edge detection using convolutional layers. In this example we use conv2d with predefined weights that were found through trial and error.

```
r = torch.tensor([[[[ 0.7917, -2.4136, 0.7076, 3.0385, -1.4910],
                    [-1.8450, 2.1870, -1.4583, -1.8226, -1.2093],
                    [ 1.8683, 0.6509, -2.6384, 0.1825, 2.0432],
                    [ 0.0478, -0.2022, 1.2576, -1.5327, 1.5264],
                    [ 0.1937, -1.9253, 0.3620, -1.2662, 2.0145]],

                  [[ 1.2917, 0.0658, -0.9486, -0.9843, 1.1612],
                    [ 0.2130, -1.0223, -2.5381, 2.6797, 0.2830],
                    [-0.7533, -0.5361, -0.1855, -0.8605, 0.6685],
                    [ 1.2721, -1.9571, 0.8976, -2.7793, 1.5025],
                    [-0.2754, 0.0170, 1.9445, -1.1021, 2.3567]],

                  [[-1.0516, 0.6095, 0.1762, -1.8851, -3.3705],
                    [-0.3485, -3.8046, -0.5211, -0.0718, -1.5302],
                    [ 3.3458, 1.9384, 1.2905, -3.4550, 1.0836],
                    [-0.7728, -1.5056, -1.8409, -1.1651, -0.2800],
                    [ 1.0632, 0.2673, 1.0127, 2.3651, 1.9630]]]])

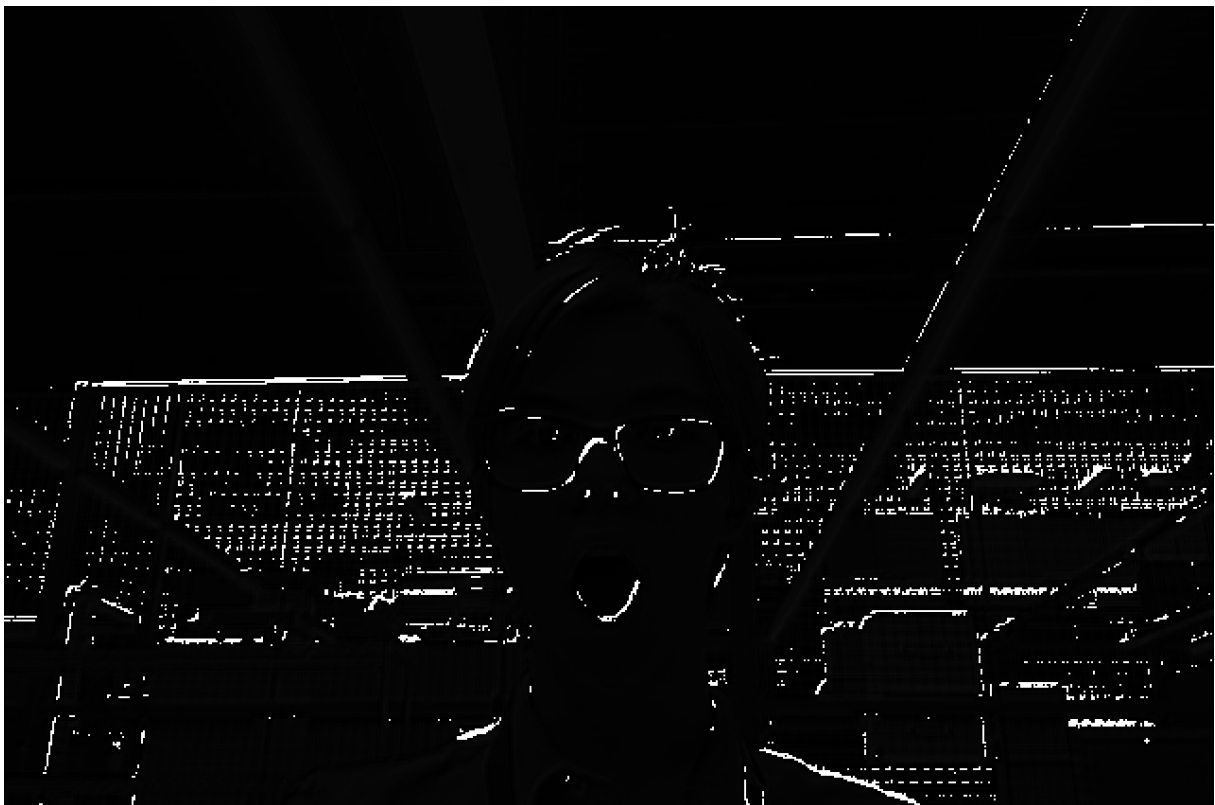
# ft is the frame after being converted into a tensor
ft = vF.rgb_to_grayscale(ft, 3)
ft = F.conv2d(ft, r)
ft = ft.round(decimals=0)

# before being displayed the image does have some
# preprocessing done to scale it back up to size
# it back up

ft = ToPILImage()(ft) # Next time use torchvision.transforms.functional.to_pil_image

frame = np.array(ft)

frame = cv2.resize(frame, (1280, 960), interpolation=0)
```



If you would like to play with the code from from this article go to <https://github.com/WagyuDEV/nn-edge-dtc>