**Instructor:** Andy Mina
**Grade Level and Subject:** 12th Grade - Introduction to Computer Vision
**Topic:** Introduction to Convolutions
**Lesson:** 04_simple_kernels

| | |
|---|---|
| **NYS Computer Science and Digital Fluency Learning Standards** | 9-12.CT.2 - Collect and evaluate data from multiple sources for use in a computational artifact.<br>9-12.CT.5 - Modify a function or procedure in a program to perform its computation in a different way over the same inputs, while preserving the result of the overall program. |
| **Content Objective** | Students will be able to:<br>● Visualize convolution kernels as images<br>● Reasonably infer how different convolution kernels will produce different outputs<br>● Understand how convolution is a form of image editing |
| **Scaffolding Needed** | Students should be able to:<br>● Understand how 2D convolution can modify input data<br>● Connect 2D arrays to images |
| **Key Vocabulary** | **Identity kernel**: a kernel that results in the original image; equivalent to multiplying a number by 1<br>**Shift kernel**: a kernel that shifts the image in the direction of the weighted pixel<br>**Block blur kernel**: a kernel that averages the colors of surrounding pixels to create a blur artifact |
| **Assessments** | Roll for Confidence (Formative)<br>Students will be asked to “roll for confidence” and respond by showing the instructor a number from 1 to 5 on one of their hands. Their confidence is representative of how comfortable they feel in continuing to explore and compare other sorting algorithms on their own. Scores represent the following:<br>1. **Not confident.** Needs a re-explanation or summary of the lesson with emphasis on key points.<br>2. **Pretty shaky.** Needs a brief recap and some teacher-guided practice to solidify concepts and understanding.<br>3. **Okay.** Needs some peer-guided practice and some more time to let things sink in. Ideal rating after |

| | |
|---|---|
| | the lesson.<br>4. **Pretty confident.** Needs some peer-guided practice for more challenging algorithms, but is self-sufficient for what's covered in class. Ideal rating before a unit test.<br>5. **Extremely confident.** Needs little to no guidance and can tackle problems of exceptional difficulty with relative ease. Indicative of an under-challenged student.<br>These checks shouldn't take any longer than one minute. |
| **Materials** | 04_simple_kernels slides, 04_simple_kernels homework |

| **Lesson Component** | **Description or Execution of Lesson Component (w/ scripting when appropriate)** |
|---|---|
| Essential Question | How can we edit images with convolution? |
| Do Now | **S1, 5m**<br>Read the Do Now out for students and prompt them to discuss. They should come up with an answer for at least one of these questions. Refer students back to L0, S14 to see when we used different convolution weights in 1D convolution. After students have had time to discuss, share answers with the class<br><br>**S2, 3m**<br>Go through the slide with students. Remind students that image pixels are in the range 0-255 so if we have weights that sum to greater than 1, they must be normalized to prevent "wonkiness and color clipping." |
| Presentation of Content | **S3, 3m**<br>Go through this a little slower since the math looks confusing. "The formula on this slide is for normalizing between any two ranges. However, since most of the time we'll be normalizing to be in the 0-255 range, a lot of simplifies down. \`new_min\` is 0 so the equation boils down to 'divide and multiply by 255.'"<br><br>**S4-6, 5m**<br>Convolution kernels can be viewed as images since they're also 2D arrays. The only difference is that the |

| | |
|---|---|
| | brightness at a certain pixel represents that pixel's weight in the result.<br>“Since kernels are just 2D arrays, how do we handle negative numbers?”<br><br>Introduce to students that we divide the 0-255 into two to get positive and negative weights. Hint that this particular kernel will reappear and they should consider that it’s negative. Roll for confidence.<br><br>**S8, 1m**<br>“Hopefully you guys know what the ‘Who’s that Pokemon game’ is because we’ll be playing a convolution version today. I’ll give you an input image, the cat onscreen, and a mystery kernel that it's convolved with. Your goal is to predict what the outcome image will be by looking at the kernel weights. Unless I say so, the kernels don’t have negative weights.”<br><br>**S9-10, 6m**<br>Give students time to think about how this kernel might result in a new convolved image. If students are struggling, walk through what might happen when you convolve at one pixel, then extrapolate what would happen for other pixels.<br><br>“This is the identity kernel, equivalent when you multiply a number by 1.”<br><br>**S11-12, 6m**<br>Now that students have seen the identity kernel, prompt them to consider what happens when we put more weight on the outside pixels.<br><br>“With the shift kernel, when we convolve at a pixel we don’t put any weight on the center. Instead we only consider one of the outside pixels meaning the image will shift in the direction of the weighted pixel.”<br><br>**S13-14, 6m**<br>This should be almost entirely student-led. Point out that all pixels in the kernel have the same weight and this is similar to a kernel we’ve used many times now.<br><br>“We call this one the ‘Block Blur’ kernel since, in larger images and kernels, it leaves a blocky trace. We’ll talk |

| | |
|---|---|
| | more about how to get rid of this blockiness on Monday. Since all pixels are weighted evenly, we're taking the average of the target pixel and surrounding pixels."<br><br>**S15-16, 3m**<br>Conversely, these slides should almost be entirely instructor-led since it's the first kernel with negative weights we've seen.<br><br>"This kernel uses negative weight and places different emphasis on the surrounding pixels. The center pixel is white so it has a strong positive weight. The surrounding pixels are black so it has a strong negative weight. The corner pixels are medium gray so they have no weight in the result."<br><br>**S17-18, 6m**<br>Remind students that sometimes it's easier to view kernels as images than numbers, but ultimately when we code it, they will have to be numbers. These slides have the number version of the kernels we've seen today. Each of these kernels can be scaled up to be bigger than 3x3 kernels, but these are the basic ones. |
| Homework | 04_simple_kernels homework |