**Instructor:** Andy Mina
**Grade Level and Subject:** 12th Grade - Introduction to Computer Vision
**Topic:** Introduction to Convolutions
**Lesson:** 07_comparing_edge_detectors

| | |
|---|---|
| **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.3 - Refine and visualize complex data sets to tell different stories with the same data set<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.<br>9-12.CT.6 - Demonstrate how at least two classic algorithms work, and analyze the trade-offs related to two or more algorithms for completing the same task. |
| **Content Objective** | Students will be able to:<br>● Construct a kernel that takes the derivative in the x/y<br>● Compare different edge detection kernels<br>● Code edge detection using convolution |
| **Scaffolding Needed** | Students should be able to:<br>● Understand how we use derivatives to detect edges |
| **Key Vocabulary** | n/a |
| **Assessments** | <u>Roll for Confidence (Formative)</u><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. |

| | |
|---|---|
| | 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** | 07_slides |

| **Lesson Component** | **Description or Execution of Lesson Component (w/ scripting when appropriate)** |
|---|---|
| Essential Question | What kernels can/should we use to detect edges? |
| Do Now | **S1, 8m**<br>Give students a lot of time to try and come up with a new edge detection kernel with or without their partners. Remind them that we want to calculate the change surrounding the center pixel but not include it.<br><br>"Reminder that so far we've mainly seen kernels that are 3x3. However, we can work with bigger kernels like 5x5 and 7x7. If you're having difficulty trying to discover a new kernel, try:<br>● expanding the Prewitt kernels to be bigger<br>OR<br>● considering changing the weights of the kernel"<br><br>Have students share any derivations or kernels they've discovered with the class. |

| | |
|---|---|
| Presentation of Content | **S2, 5m**<br>Read through the slides for students. Explain that good localization means that there's little to no influence of far pixels incorrectly labeling this as an edge. Larger kernels like 5x5 or 7x7 have poor localization meaning it's easy to get false positives since far pixel contributes the derivative *too* much.<br><br>**S3-4, 4m**<br>Walk students through discovering the Sobel kernel. If students have already come up with this in the Do Now, prompt them to explain why they changed the weights instead. Explain that the most common Sobel operator uses n=2.<br><br>"The Sobel kernel usually provides somewhat better results than the Prewitt kernel because it makes adjacent pixels weigh more. This means comparatively either diagonal edges are weaker or horizontal/vertical edges are stronger. Let's try deriving the Sobel another way."<br><br>**S5, 4m**<br>Read out the slide for students and let them convolve these two kernels. "For those of you that pursue higher math, this two 1D convolution is similar to vector multiplication in Matrix Algebra."<br><br>**S6, 3m**<br>Review this Sobel derivation with students. Note that the n=2 Sobel already includes a small Gaussian which is why it's generally the most popular, larger n's sometimes put too much weight and give a small blur.<br><br>**S7–8, 4m**<br>Go through the pros and cons of the Sobel kernel with students and emphasize that the Sobel includes a Gaussian, but usually, we want to do another small Gaussian to make sure there isn't any lingering noise.<br><br>"Note the first con of the Sobel kernel poses a new problem. Sometimes it's hard to find diagonal edges. Can you come up with a kernel to get around this?" |

Roll for confidence in understanding:

- Why the two kernels (Prewitt and Sobel) are different
- What problem they solve
- Pros and cons of each

**S9-10, 4m**

"We've talked about the different kernels you can use for edge detection. When we convolve the image with the x kernel, we get only the edges in the x direction. When we convolve the image with the y kernel, we get only the edges in the y direction.

There are two main ways to combine the x edges and the y edges to get ALL edges:

1. We take the absolutely value of both and average the result
2. We use the Pythagorean Theorem to find the magnitude (or hypotenuse) of the components we split out before

The Pythagorean Theorem leads to better results, but the average is quicker. For this class, we'll be using the Pythagorean Theorem.

**S11-15, 6m**

Walk students through the process of edge detection step-by-step.
"First we'll take our image and convolve it with the Prewitt x derivative kernel. This will gives us the edges in the x direction. Then we convolve our image with the Prewitt y derivative kernel to get the edges in the y direction. Finally, we apply the Pythagorean Theorem to find the combined edges."

Ask students why some edges are missing. For example, in S12 the top and bottom edges of the rectangle are missing and in S13, the left and right edges are missing.

Note to students that we'll only normalize after we combine the edges. We want to be as accurate as possible for as long as possible.

Roll for confidence in how students feel about doing this process on their own. Remind students that they know

| | |
|---|---|
| | how to:<br>● Create kernels (Prewitt and Sobel)<br>● Convolve images with kernels<br>● Use numpy to square arrays and find the square root<br>● Normalize values<br><br>**S16-18, 6m**<br>Calculate the edges in the shapes image using the Sobel operator now. This should be almost entirely student led.<br><br>**S19, 3m**<br>Remind students that up until we've been talking about the strength of the edge and ignoring the direction. "We will still mostly focus on the strength, but it's also good to see what edge detection with direction might look like." |
| Homework | n/a |