Instructor: Andy Mina
Grade Level and Subject: 12th Grade - Introduction to Computer Vision
Topic: Introduction to Convolutions
Lesson: 09_project

NYS Computer Science and Digital Fluency Learning Standards	9-12.CT.4 - Implement a program using a combination of student-defined and third-party functions to organize the computation. 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. 9-12.CT.8 - Develop a program that effectively uses control structures in order to create a computer program for practical intent, personal expression, or to address a societal issue. 9-12.CT.9 Systematically test and refine programs using a range of test cases, based on anticipating common errors and user behavior.
Content Objective	Students will be able to: ● Create a line art image from a given image by identifying edges with convolution ● Use different edge detector kernels to find edges (Prewitt, Sobel, and Scharr)
Scaffolding Needed	Students should be able to: ● Code basic edge detection using convolution with the Prewitt and Sobel kernels ● Loop through 2D arrays
Key Vocabulary	n/a
Assessments	Line Art Generator (Summative) The Line Art Generator is a short-term project designed as a summative assessment. In this project, students read an image from a file path provided via \`input()\`, apply edge detection using the Prewitt, Sobel, and Scharr kernels, and generate a line-art image where the background color and foreground color are set by the user. Roll for Confidence (Formative)

	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: 1. **Not confident.** Needs a re-explanation or summary of the lesson with emphasis on key points. 2. **Pretty shaky.** Needs a brief recap and some teacher-guided practice to solidify concepts and understanding. 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. 5. **Extremely confident.** Needs little to no guidance and can tackle problems of exceptional difficulty with relative ease. Indicative of an under-challenged student. These checks shouldn’t take any longer than one minute.
Materials	09_slides

Lesson Component	**Description or Execution of Lesson Component (w/ scripting when appropriate)**
Essential Question	How can we apply edge detection in real life?
Do Now	**S1-2, 5m** Give students time to plan their work for the day. Emphasize that the planning process is just as important as the code, sometimes even more so. Have students roll for confidence in how they’re feeling about the project.
Presentation of Content	**S3-4, 4m** Explain that the goal is for students to submit this project by Monday. If groups are having difficulty, they can reference these slides for places where we’ve tackled the same or similar problems. In addition, the kernels you need have been pasted for reference.

	For the rest of the period, let students work in pairs and drift around answering questions and offering programming advice. For students who finish early, they should complete the Laplacian extra credit on S6.
Homework	n/a