Instructor: Andy Mina
Grade Level and Subject: 12th Grade - Introduction to Computer Vision
Topic: Introduction to Convolutions
Lesson: 08_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	**Alpha-blend**: to blend a background color and foreground color to give the illusion of transparency; the alpha term in the formula refers to the intensity of foreground color
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.
Materials	08_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, 8m** Give students ample time to research applications of edge detection that they find interesting. Spend the last three minutes of this section sharing student research. **S2, 2m** “Some simple applications of edge detection are boundary detection in video games like GTA V, object identification (finding coins in this example), and identifying text. For identifying text, we can extract the edges and apply a convolution match to see how similar the edges match the edges of known 8s.” **S3, 4m** “An advanced application of edge detection would be self-driving cars as others have mentioned, but my favorite is something called ‘seam-carving’ which does content-aware image resizing.”
Presentation of Content	**S4-7, 8m** Walk through the project introduction with students. Heavily encourage them to refer back to old slides, homework, and code when completing this project. **S8, 4m** Share with students that they should work on this project in three stages: 1. After reading the project description, work on the starter code like normalizing an array, reading and displaying an image, and convolution. 2. Write your edge detection code. Hint: you want to make a function that convolves edges with given kernels so it’s easy to apply multiple kernels (Prewitt vs. Sobel vs. others) 3. Recolor the image and blend the background and foreground properly.

	For the rest of the period, let students work in pairs and drift around answering questions and offering programming advice.
Homework	Continue working on Line Art Generator