Instructor: Andy Mina
Grade Level and Subject: 12th Grade - Introduction to Computer Vision
Topic: Introduction to Convolutions
Lesson: 03_code

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 9-12.CT.4 - Implement a program using a combination of student-defined and third-party functions to organize the computation. 9-12.DL.2 - Communicate and work collaboratively with others using digital tools to support individual learning and contribute to the learning of others.
Content Objective	Students will be able to: ● Perform 2D convolutions using scipy's convolve2d function and numpy arrays
Scaffolding Needed	Students should be able to: ● Perform 2D convolutions on pen and paper ● Identify and implement different convolution border solutions
Key Vocabulary	n/a
Assessments	<u>Roll for Confidence (Formative)</u> 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	03_slides, 03_homework, 03_instructor code-along, 03_student code-along

Lesson Component	**Description or Execution of Lesson Component (w/ scripting when appropriate)**
Essential Question	How do we perform 2D convolution in code?
Do Now	**S1, 3m** Give students time to fork the student code-along and read through the introduction section
Presentation of Content	**S2-5, 3m** Tell students that today is just the “code version” of yesterday’s lesson. We’re doing the same exact problems, just translating them to code. We’ll be using numpy to handle 2D arrays easier and scipy for to perform the actual convolution. **Code-Along Introduction, 2m** Briefly review the Introduction section with students. Note to students they got a `display_image` function in the starter so they should make use of it. They can ignore the internals of it but should know that it displays a 2D array as a grayscale image. **Class Practice Problem - 8m** Walk through the instructions in this section. Emphasize to students that we’re convolving two 2D arrays and that we can trust scipy to flip the kernels for us later on. When rounding to the nearest int, make it an example to look up how to do it with numpy. While we could do this with vanilla Python, numpy has a lot of these quality-of-life algorithms on array built in already; you just have to search for them. Stop for questions as needed and roll for confidence.

	Pair Practice Problem - 6m Start students off on this problem, but point out that this is still the same problem from last lesson so things aren’t brand new and there’s no reason to be nervous about it. While students work on this in pairs, walk around the room and catch potential errors early on. Review any potential pitfalls after students have finished. **Solo Practice Problem - 10m** Be strict that students should do this on their own (without the help of their partner), but are more than welcome to ask the instructor for help or refer back to old slides or problems. If they run into errors, they should debug on their own. The syntax and application are the same as the last two problems.
Homework	03_homework, convert 02_homework to code