Grade Level	Middle School
Duration	4 Class Periods
CS Standards Addressed	8.1.8.NI.1: Model how information is broken down into smaller pieces, transmitted as addressed packets through multiple devices over networks and the Internet, and reassembled at the destination. 8.1.8.DA.1: Organize and transform data collected using computational tools to make it usable for a specific purpose. 8.1.8.DA.2: Explain the difference between how the computer stores data as bits and how the data is displayed 8.1.8.DA.5: Test, analyze, and refine computational models 8.1.8.AP.4: Decompose problems and sub-problems into parts to facilitate the design, implementation, and review of programs.

Introduction

In this lesson package, students explore how computers efficiently store and transmit information through compression. Over two class periods, students begin by investigating how repeated patterns in text can be identified and encoded to reduce file size without losing information. Through interactive and unplugged activities, they develop their own strategies for compressing text and discover that effective compression involves recognizing patterns and balancing tradeoffs between the encoded message and the information needed to decode it.

Building on this foundation, students then apply these ideas to digital images. They learn that images are composed of grids of pixels and use run-length encoding to compress pixel data by representing consecutive runs of the same color more efficiently. By comparing images that compress well with those that do not, students deepen their understanding that compression is not universally effective, but instead depends on the structure and patterns within the data.

Throughout the lesson, students make connections between abstract concepts and real-world applications, such as how computers store photos, stream videos, and transmit files. By the end of the lesson package, students will have a foundational understanding of how compression works across different types of data and why it is essential for managing memory and optimizing performance in modern computing systems.

This lesson package includes:

- A Student Worksheet with Video Questions
- Unplugged Text Compression Activity Instructions

Lesson Agenda

Class 1 – Text Compression

Objective: Students will explore how computers reduce file size through lossless text compression by identifying repeated patterns and encoding them more efficiently. They will understand that effective compression involves balancing the size of the encoded message with the information needed to decode it.

Time	Activity	Notes
10 mins	Warm-up: Video & Questions	1. Watch the code.org video on text compression and provide students with worksheet to answer questions as they watch. a. https://www.youtube.com/watch?v=LCGkcn1f-ms 2. Review answers with the class. *Student worksheet available separately in package.*
15 mins	Activity: Text Compression (Code.org)	In this activity, students will explore how computers reduce file size through lossless text compression. They will examine a block of repeated text and create a dictionary of patterns, assigning symbols to frequently repeated words or phrases. They then replace those patterns in the text with their symbols to produce a shorter encoded version. Through experimentation, students will discover that effective compression requires balancing the size of the compressed message with the size of the dictionary used to decode it, helping them understand how computers store and transmit information more efficiently.
20-25 mins	Activity: Unplugged Text Compression	Students practice unplugged text compression by identifying repeated patterns in a short poem or nursery rhyme. They draw boxes around repeated words or phrases and use arrows to reference earlier occurrences instead of rewriting the text. This demonstrates how computers compress files by storing repeated information once and pointing back to it when needed, introducing the basic idea behind real compression algorithms used in formats like ZIP and GIF. If time, allow students to complete the activity "Extra for Real Experts" at the end of the provided Unplugged guidance document (pg. 29). *Unplugged teacher information document available separately in package.*

Class 2 – Image Compression

Objective: Students will apply their understanding of compression to images by using run-length encoding to represent pixel data more efficiently. They will analyze when compression is effective and recognize that it depends on the presence of patterns within the data.

Time	Activity	Notes
5 mins	Warm-up: Question	"Two images are the same size. One is completely black. The other is a checkerboard pattern. Which one do you think would be easier for a computer to compress? Why?" • The solid color image compresses better because it has long runs of the same pixel.

		• The checkerboard has constant changes, which makes compression harder.

20 mins	Activity: Pixel Painter & Run-Length Encoding (RLE), *Easy Image*	Tie into last class: "Yesterday we compressed text by finding repeated words or phrases. Today we're going to see how computers compress images by finding repeated pixels."
		
		• Students will create **simple** pixel images using a grid-based drawing tool, and then compress those images using run-length encoding (RLE). They will examine rows of pixels and record consecutive runs of the same color as a count and value (for example, "5 black, 3 white") rather than listing each pixel individually.
		  ○ Pixel Painter: https://www.csunplugged.org/en/resources/pixel-painter
		  ○ RLE: https://www.csfieldguide.org.nz/en/interactives/run-length-encoding
		• By comparing the original image data with the encoded version, students observe how patterns in images allow computers to store visual information more efficiently and discover that compression works best when large areas contain repeated colors.
		
		See below for step-by-step instructions.

10-15 mins	Activity: Pixel Painter & Run-Length Encoding (RLE), *Difficult Image*	Students will now design a pixel image that is **difficult** to compress using run-length encoding. Building on the previous activity where students created images that compressed well due to large areas of repeated color, students should now create patterns with frequent color changes (such as checkerboards or alternating pixels). They then encode the image and compare the compressed size to their original. Through this comparison, students observe that while some images compress efficiently, others do not, reinforcing that compression depends on the presence of patterns in the data. *Use same instructions and links for this activity as above.*
5 mins	Wrap-up: Discussion (Whole Class)	Engage Students in a brief class discussion to reflect on their experience with image compression and connect it to the previous lesson on text compression. *See below for a discussion question bank.*

Pixel Painter and RLE Simulator Activity

Goal: Students explore run-length encoding (RLE) by creating simple pixel images and compressing them to see when RLE helps and when it doesn't.

Prep (before class):

- Open the Pixel Painter resource and the RLE simulator so you can demo both.
- Prepare one simple example image (stripes or a solid shape) and one noisy example (checkerboard or random pixels).
- Print or project the student instructions (1–2 short bullets) and set timer for activity segments.

Step-by-step instructions:

1. Introduce goal
    a. "We'll draw small pixel images, then compress them using run-length encoding to see how repetition affects file size."
2. Demonstrate (2–3 minutes)
    a. Show the Pixel Painter and draw a simple row (e.g., BBBBBWWWW).
    b. https://www.csunplugged.org/en/resources/pixel-painter/
    c. Enter that row into the RLE simulator and show output (e.g., 5B 4W). Point out compressed vs. original length.
3. Student image creation (5 minutes)
    a. Students work solo or in pairs. Ask each to create: one image designed to compress well and one designed to compress poorly. Keep grids small (8×8 or 10×6).
4. Compress with the simulator (6–8 minutes)
    a. https://www.csfieldguide.org.nz/en/interactives/run-length-encoding/
    b. Students copy rows from their image into the simulator and record the RLE output and the compressed size (including any simple dictionary overhead if you want to extend).
5. Quick comparison and tally (3 minutes)
    a. Each pair notes: original size (pixels), compressed representation length, and whether compression helped.
6. Whole class debrief (5 minutes)
    a. Ask pairs to share one "best compressor" and one "worst compressor" example. Emphasize tradeoffs and when RLE fails.

Discussion Bank (Wrap-up)

Use the questions below to guide a brief class discussion. Choose a few that best fit your students and the time available. The goal is to help students reflect on how compression works, when it is effective, and how it connects to real-world technology.

- Which image you created compressed the best? What made it effective?
- Which image compressed the worst? What was different about it?
- What patterns did you notice in the images that compressed well?
- Why does run-length encoding work better for some images than others?
- What happens when there are lots of color changes in an image?
- Can all images be compressed efficiently? Why or why not?
- How is compressing an image similar to what we did with text compression yesterday?
- What is the "pattern" in text vs. the "pattern" in images?
- Where do you think image compression is used in real life?
- Why is compression important for things like sending photos or streaming videos?
- What might happen if computers didn't use compression?
- Why might a detailed photo (like a forest or crowd) be harder to compress than a simple cartoon?
- Do you think computers always find the best way to compress something? Why or why not?