

Performance Report

This is a brief project performance report for CS5330 Lab 1.

Image-to-ASCII process

The process of transforming an image into ASCII art involves 3 key steps (similar to what is suggested on Canvas), primarily focusing on image preparation, edge detection, and character mapping.

1. Step 1: Load, Resize, and Convert to Grayscale
 - Load the Image: Use OpenCV to read the image from the specified path.
 - Resize for ASCII Output: Calculate a new height while maintaining the aspect ratio and resize the image. This makes the ASCII output smaller and more manageable.
 - Convert to Grayscale: Transform the resized image into grayscale, which simplifies the mapping process by reducing color information.
2. Step 2: Apply Edge Detection (Canny vs. Sobel)
 - Canny Edge Detection: Detect edges in the grayscale image using the Canny algorithm. Adjust the thresholds to fine-tune the sensitivity of edge detection.
 - Sobel Edge Detection: An alternative method is Sobel edge detection, which calculates gradients in both horizontal and vertical directions, emphasizing areas of high contrast.
3. Step 3: ASCII Character Mapping
 - Define ASCII Characters: Create a list of ASCII characters to represent different levels of brightness.
 - Map Pixels to ASCII: Resize the edge-detected image to fit ASCII art dimensions. For each pixel in the resized image, determine the corresponding ASCII character based on its brightness level (scaled pixel value).
 - Generate ASCII Art: Combine rows of characters into a single string, where each row represents a line in the ASCII art.

Performance Metric

Option 1: SSIM

SSIM was our first choice used to measure the similarity between two images, SSIM focuses on structural information (luminance, contrast, and structure). It ranges from -1 (completely different) to 1 (perfectly identical). The function calculates the similarity based on pixel intensity patterns rather than pixel-by-pixel differences, making it more perceptually meaningful.

However, SSIM may not fully capture the perceptual qualities of ASCII art (since the art reduces image details and uses characters instead of pixels), therefore, we developed the following performance metric after consulted the website below.

Option 2: Customized criteria based on suggested material

5 out of 13 suggested attributes - [Wikipedia Link](#), were considered relevant and selected for this project: sharpness, noise dynamic range, contrast and artifacts.

The rest (color accuracy, distortion, vignetting, chromatic aberration, lens flare and moiré) were ignored mostly because they are less relevant to the project: e.g., since ASCII art typically does not involve color (or only does so minimally), color accuracy may not be crucial unless we are generating colored ASCII art, and the remaining 6 attributes are either lens-based or optical effects or more specific to camera-based photography which do not apply to the transformation of images into ASCII art.

For the 5 selected attributes:

1. Sharpness: This is key because edge detection and the preservation of fine details are central to ASCII art. We can assess sharpness by comparing the edges in the original image with those in the ASCII art. The existing SSIM score we calculated indirectly measures sharpness by evaluating structural similarity.
2. Noise: The introduction of noise (or reduction of details) in the ASCII art compared to the original image could reduce the quality. We have already been accounting for this through methods like edge retention, but it can be made explicit by measuring noise levels in both images.

3. **Dynamic Range:** Since ASCII art typically uses varying character densities to simulate brightness levels, we want to ensure the dynamic range of the original image is adequately captured in the ASCII conversion. This can be evaluated by comparing the tone mapping between the two images.
4. **Contrast:** ASCII art relies on contrast to differentiate areas of the image. we can compare the contrast (gamma) between the original and ASCII versions to see if the art correctly represents tonal shifts, especially around edges and regions of varying brightness.
5. **Artifacts:** Compression artifacts and other distortions can degrade image quality. For ASCII art, excessive simplifications or errors during the conversion could result in “artificing” in the form of misrepresentation of fine details. This could be measured as a form of pixel-to-character mapping error.

Performance Results (3 color images)

SSIM

	Graph 1	Graph 2	Graph 3
Canny Edge Detection	0.004610955016786405	0.005194058394808774	0.015629814158054058
Sobel Edge Detection	0.015299368983799353	0.062046289575641275	0.07725893569390421

Customized Method:

	Graph 1		Graph 2		Graph 3	
	Canny	Sobel	Canny	Sobel	Canny	Sobel
Sharpness	0.8093	0.8134	0.4554	0.4479	0.6522	0.390667
Noise Difference	5967.52	6517.47	2733.07	1442.95	3367.73	1624.84
Dynamic Range & Contrast	0.01122	-0.0102	-0.0861	-0.3392	-0.06187	-0.2081
Artifacts	36321.29	35428.27	18442.1	14237.4	23032.70	18200.24

Graph 1



Graph 2



Graph 3

