Homework #1 Report

Part 1 - Image Representation

Results

Original RGB Image

Visual Observation: The original RGB image displays the full color spectrum with all three color channels (Red, Green, and Blue) intact. The image retains complete chromatic information, allowing for differentiation of objects based on their color properties. The spatial dimensions and resolution are preserved in the original format.

Grayscale Image

Visual Observation: The grayscale conversion transforms the color image into a monochromatic representation where intensity values range from black (0) to white (255). Objects that were previously distinguishable by color may now appear similar if they share comparable luminance values. The conversion uses a weighted formula that prioritizes the green channel due to human eye sensitivity, resulting in a perceptually accurate brightness representation.

5×5 Pixel Region Values:

(9, 12, 27)	(9, 12, 27)	(9, 12, 27)	(9, 12, 27)	(9, 12, 27)
(9, 12, 27)	(9, 12, 27)	(9, 12, 27)	(9, 12, 27)	(9, 12, 27)
(9, 12, 27)	(9, 12, 27)	(9, 12, 27)	(9, 12, 27)	(9, 12, 27)
(9, 12, 27)	(9, 12, 27)	(9, 12, 27)	(9, 12, 27)	(9, 12, 27)
(9, 12, 27)	(9, 12, 27)	(9, 12, 27)	(9, 12, 27)	(9, 12, 27)

Grayscale Pixel Values:

13	13	13	13	13
13	13	13	13	13
13	13	13	13	13
13	13	13	13	13
13	13	13	13	13

Analysis of Pixel Region: The 5×5 top-left region shows uniform dark pixels with a bluish tone — Blue (27) is highest, followed by Green (12) and Red (9). After grayscale conversion using the formula `0.299R + 0.587G + 0.114B`, all pixels become intensity 13, demonstrating how the weighted formula prioritizes green while preserving perceptual accuracy.

What information is stored in each pixel?

In an RGB color image, each pixel stores three intensity values (Red, Green, Blue) ranging from 0 to 255 (8-bit), allowing 16,777,216 color combinations. In a grayscale image, each pixel stores a single intensity value (0-255) representing luminance.

What information is lost when converting to grayscale?

Color information is lost while luminance is preserved. The three RGB channels are combined using a weighted formula `(0.299R + 0.587G + 0.114B)`. Different RGB combinations with the same luminance become indistinguishable in grayscale, eliminating color-based object differentiation.

Part 2 - Basic Image Operations

Results

Brightness Adjustment

Visual Observation: The image appears noticeably lighter with previously dark regions becoming visible, revealing obscured details while maintaining color relationships.

Explanation: Adds constant value (50) to all pixels: `I_new = I_original + 50`. This uniformly increases luminance, shifting the histogram right while preserving relative intensity differences and edge information.

Contrast Adjustment

Visual Observation: Enhanced definition with pronounced separation between light and dark regions. Colors appear more saturated, edges sharper, creating a more dramatic appearance.

Explanation: Multiplies pixels by scaling factor (1.5): `I_new = 1.5 × I_original`. This stretches the histogram, amplifying differences between features. May cause saturation at extremes (0 or 255).

Negative Transformation

Visual Observation: Creates a photographic negative effect with inverted colors. Light becomes dark and vice versa. All structural elements, edges, and spatial relationships remain perfectly preserved.

Explanation: Inverts all pixels: `I_new = 255 - I_original`. Maps each intensity to its complement while preserving structural information and relative intensity differences. Useful for medical imaging and revealing features more visible in inverted form.

Part 3 - Mathematical Operations on Images

Results

Image Addition (Blending)

Visual Observation: A semi-transparent overlay where features from both images merge together. Common content appears prominent while distinct features remain visible at reduced intensity.

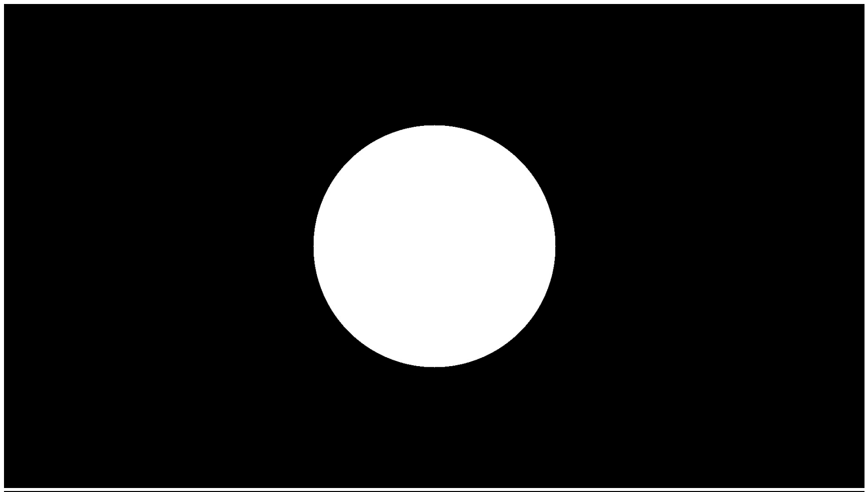
Explanation: Weighted average of pixels: `I_result = 0.5 × I_img1 + 0.5 × I_img2`. Useful for overlay effects, comparing similarities, and merging complementary information. Requires identical image dimensions.

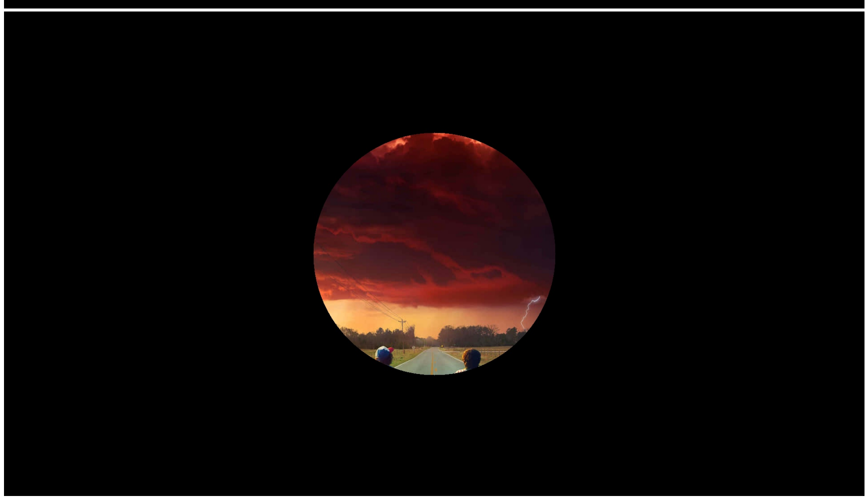
Image Subtraction

Visual Observation: Emphasizes differences between images. Similar regions appear dark while different areas display brighter intensities, creating a difference map.

Explanation: Absolute difference: `I_result = |I_img1 - I_img2|`. Identical pixels yield black, maximum differences yield bright values. Used for motion detection, change detection, and background subtraction.

Masking





Visual Observation: Binary mask shows white circular region on black background. Masked result displays only the circular region while pixels outside are zeroed out, isolating the central portion.

Explanation: Binary mask with values 0 or 255 using cv2.bitwise_and(): `I_result = I_original & mask`. White regions (255) retain original values, black regions (0) become zero. Enables ROI extraction, background removal, and selective filtering. Masks can be any shape from segmentation or thresholding.

Questions

1. Which operation helped most in improving image visibility?

Brightness adjustment proved most effective for underexposed images by uniformly increasing pixel intensities, revealing previously imperceptible details in dark regions without altering relative relationships between elements.

However, the optimal choice depends on image characteristics. Contrast adjustment is superior when brightness is adequate but definition is lacking, as it amplifies intensity differences to make edges and textures more pronounced.

For this assignment, brightness adjustment demonstrated clear improvement by lifting the entire intensity range while preserving relative contrasts.

2. Which operation only changed appearance but not structure?

Negative transformation changes only appearance without affecting structural information. It preserves all spatial relationships, edges, and gradients while inverting intensities (255 - I). Despite the dramatic visual reversal, object boundaries, textures, and edge magnitudes remain intact.

Structure in image processing refers to spatial arrangement and relative relationships, not absolute intensity values. Since negative transformation applies a one-to-one invertible mapping, no structural information is lost. Edge detection would produce identical results on both original and negative images.

Brightness adjustment also preserves structure within valid range (0-255) through uniform addition. However, clipping at boundaries may collapse distinct intensities. Contrast adjustment can affect structure through saturation at extremes (0 or 255), potentially merging distinct features.

3. Why is preprocessing important before computer vision?

Preprocessing is crucial for several fundamental reasons:

- **Normalization of Imaging Variations:** Real-world images are captured under variable conditions (lighting, exposure, sensors). Preprocessing normalizes these variations, ensuring consistent input to algorithms. Without it, models encounter distributions different from training conditions, producing unreliable results.
- **Feature Enhancement:** Preprocessing enhances relevant features while suppressing noise. Contrast enhancement improves edge visibility for robust detection. Masking enables focused analysis on regions of interest, reducing computational complexity.
- **Addressing Quality Issues:** Images often suffer from noise, blur, and artifacts. Preprocessing improves signal-to-noise ratio and quality. For example, brightness adjustment reveals subtle variations in medical imaging; contrast enhancement improves visibility in autonomous vehicles.
- **Domain Adaptation:** Preprocessing transforms images from various sources into a common representation space, essential for training generalizable models. Without it, a model trained on indoor images would fail on outdoor scenes.

- **Computational Efficiency:** Operations like masking reduce computational requirements by limiting processing to relevant areas, critical for real-time applications.

In summary, preprocessing bridges the gap between imperfect real-world acquisitions and the idealized inputs algorithms expect, determining the success of the entire pipeline.