**EAST WEST UNIVERSITY**

**CSE438**

**Section: 01**

**Lab: 02 Report**

**Topic: Contrast stretching,** **bit plane slicing,** **using Logarithmic Transformation and Power-law Transformation.**

**Submitted By:**

**Name: Md Sifat Ullah Sheikh ID: 2022-1-60-029**

**Submitted To:**

**K.M. Safin Kamal**

**Lecturer**

**Department of Computer Science & Engineering**

**Date: 7 July 2025**

**Q1. Use contrast stretching on the image from Figure 1.**

**CODE:**

import cv2

import numpy as np

import matplotlib.pyplot as plt

fig\_01\_path = '/kaggle/input/lab-02/Lab\_02/Picture1.png'

fig\_01 = cv2.imread(fig\_01\_path,cv2.IMREAD\_GRAYSCALE)

def show\_image(fig\_01\_path, title = "", cmap='gray'):

plt.imshow(fig\_01\_path, cmap = cmap)

plt.title (title)

plt.axis ('off')

plt.show

show\_image(fig\_01,"Figure\_01")

import os

output\_dir = '/kaggle/working/img'

output\_path = os.path.join(output\_dir, 'comparison\_contrast\_stretching.png')

os.makedirs(output\_dir, exist\_ok=True)

def contrast\_stretching(fig\_01):

min\_val = np.min(fig\_01)

max\_val = np.max(fig\_01)

stretched = ((fig\_01 - min\_val) / (max\_val - min\_val)) \* 255

return stretched.astype(np.uint8)

stretched\_image = contrast\_stretching(fig\_01)

fig, axs = plt.subplots(1, 2, figsize=(10, 5), dpi=300)

axs[0].imshow(fig\_01, cmap='gray')

axs[0].set\_title('Original Image')

axs[0].axis('off')

axs[1].imshow(stretched\_image, cmap='gray')

axs[1].set\_title('Contrast Stretched')

axs[1].axis('off')

plt.tight\_layout()

plt.savefig(output\_path, dpi=300, bbox\_inches='tight', pad\_inches=0)

plt.show()

plt.close()

print(f"Comparison image saved successfully to: {output\_path}")



output\_dir = '/kaggle/working/img'

output\_path = os.path.join(output\_dir, 'histogram\_contrast\_stretching.png')

def show\_image\_and\_hist(img, title):

fig, axs = plt.subplots(1, 2, figsize=(10, 4))

# Display image

axs[0].imshow(img, cmap='gray')

axs[0].set\_title(title)

axs[0].axis('off')

# Display histogram

axs[1].hist(img.ravel(), bins=256, range=(0, 256), color='blue')

axs[1].set\_title('Histogram')

plt.tight\_layout()

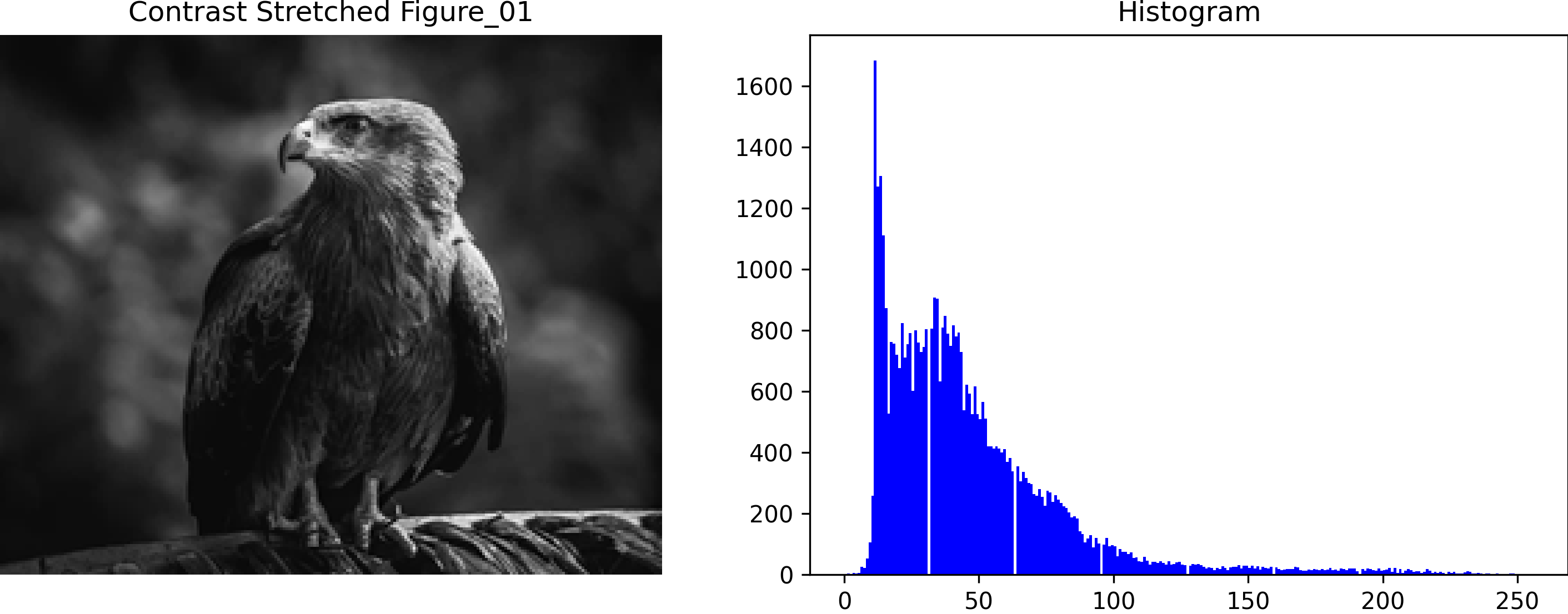
plt.savefig(output\_path, dpi=300, bbox\_inches='tight', pad\_inches=0)

plt.show()

stretched\_img = contrast\_stretching(fig\_01)

show\_image\_and\_hist(fig\_01,"Orginal Figure\_01")

show\_image\_and\_hist(stretched\_img,"Contrast Stretched Figure\_01")



**Q2. Apply bit plane slicing on the image 1**

**CODE**

output\_dir = '/kaggle/working/img'

output\_path = os.path.join(output\_dir, 'bit plane slicing on the image.png')

# Extract bit planes

bit\_planes = []

for i in range(8):

bit\_plane = (fig\_01 >> i) & 1

bit\_planes.append(bit\_plane \* 255)

fig, axs = plt.subplots(2, 4, figsize=(12, 6), dpi=300)

fig.suptitle('Bit Plane Slicing', fontsize=16)

for i in range(8):

row = i // 4

col = i % 4

axs[row, col].imshow(bit\_planes[i], cmap='gray')

axs[row, col].set\_title(f'Bit Plane {i}')

axs[row, col].axis('off')

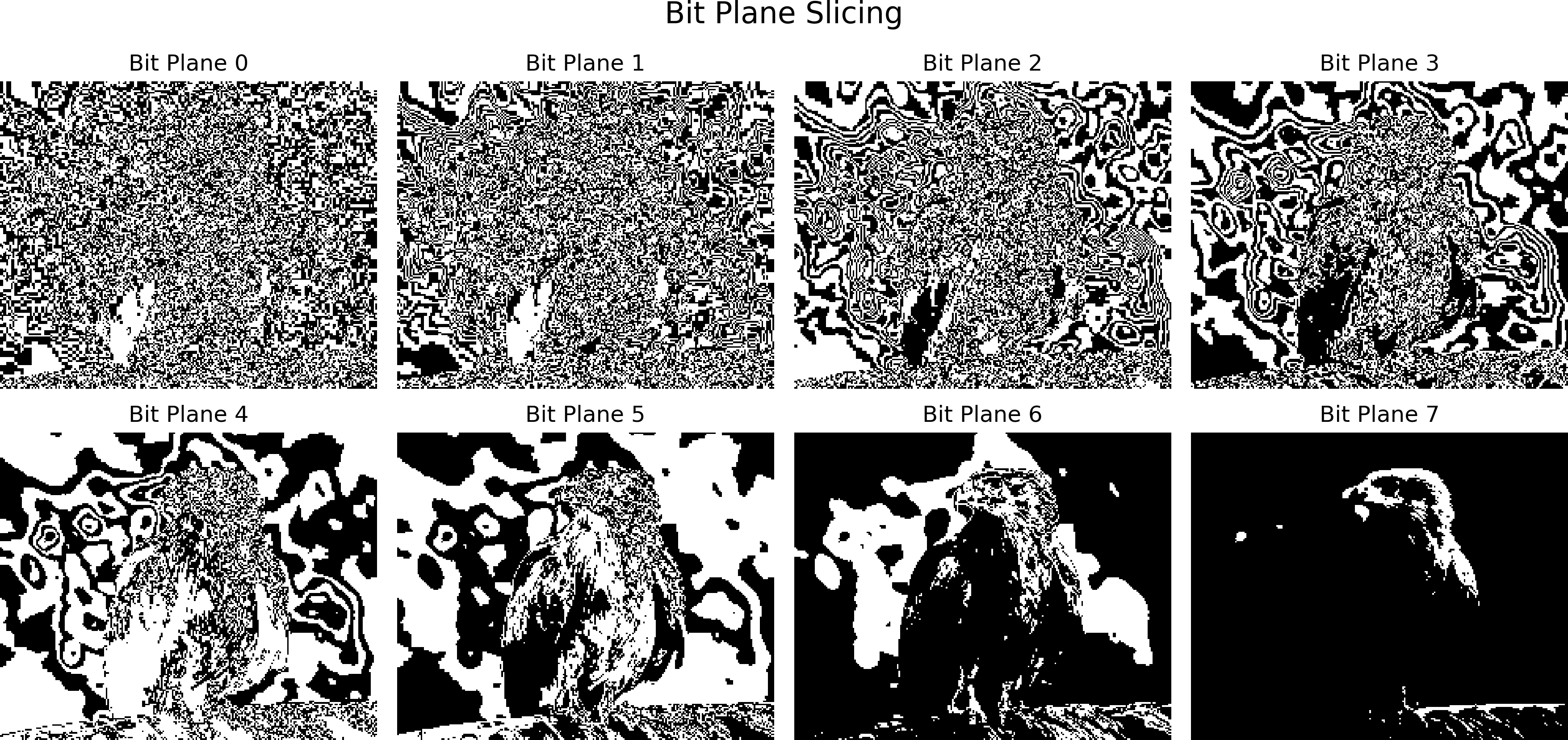
plt.tight\_layout(rect=[0, 0, 1, 1])

plt.savefig(output\_path, dpi=300, bbox\_inches='tight', pad\_inches=0)

plt.show()

plt.close()

print(f"Bit plane slicing image saved to: {output\_path}")



**Q3. Change the contrast of the image using Logarithmic Transformation and Power-law Transformation.**

**CODE**

def log\_transform(img):

img\_float = img.astype(np.float32)

c = 255 / np.log(1 + np.max(img\_float))

log\_image = c \* np.log(1 + img\_float)

return np.uint8(np.clip(log\_image, 0, 255))

def gamma\_transform(img, gamma=0.5):

img\_normalized = img / 255.0

gamma\_corrected = np.power(img\_normalized, gamma)

return np.uint8(np.clip(gamma\_corrected \* 255, 0, 255))

log\_img = log\_transform(fig\_02)

gamma\_img = gamma\_transform(fig\_02, gamma=0.5)

titles = ['Original Image', 'Log Transform', 'Gamma Transform (γ=0.5)']

images = [fig\_02, log\_img, gamma\_img]

plt.figure(figsize=(15, 5), dpi=300)

for i in range(3):

plt.subplot(1, 3, i+1)

plt.imshow(images[i], cmap='gray')

plt.title(titles[i])

plt.axis('off')

plt.tight\_layout()

output\_dir = '/kaggle/working/img'

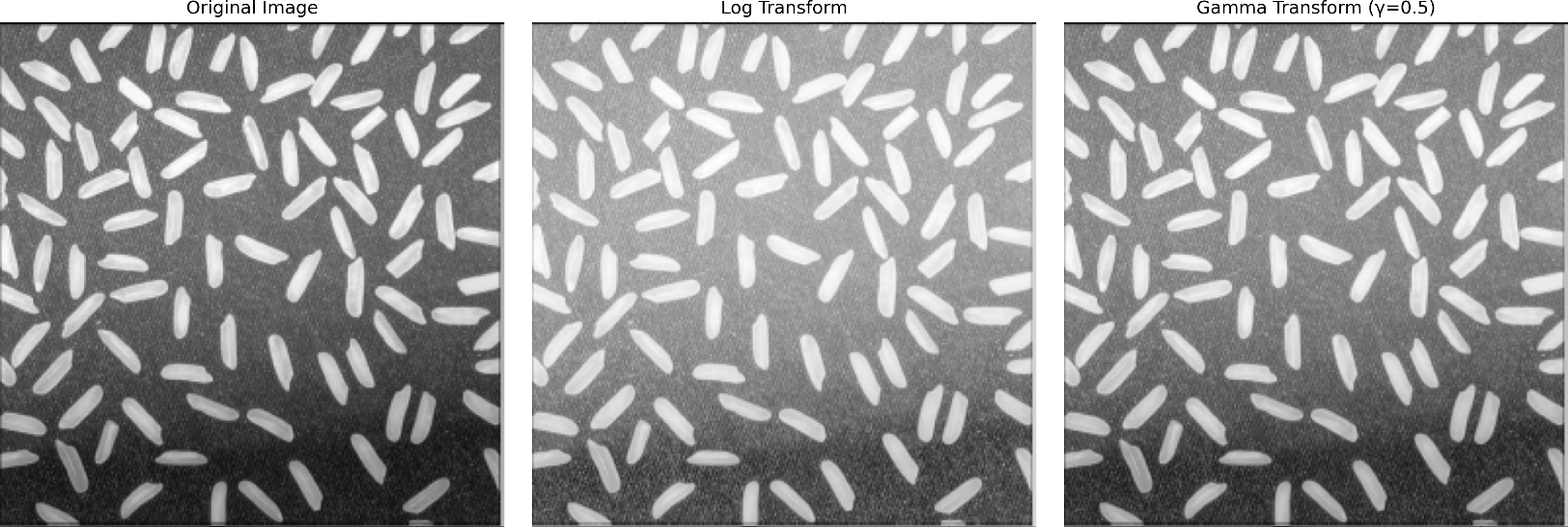
os.makedirs(output\_dir, exist\_ok=True)

output\_path = os.path.join(output\_dir, 'figure02\_log\_gamma\_transform.png')

plt.savefig(output\_path, dpi=300, bbox\_inches='tight', pad\_inches=0)

plt.show()

print(f"Transformed image saved to: {output\_path}")



fig, axs = plt.subplots(3, 2, figsize=(12, 12), dpi=300)

fig.suptitle('Image and Histogram Comparison', fontsize=16)

for i in range(3):

axs[i, 0].imshow(images[i], cmap='blue')

axs[i, 0].set\_title(titles[i])

axs[i, 0].axis('off')

axs[i, 1].hist(images[i].ravel(), bins=256, range=(0, 256), color='gray')

axs[i, 1].set\_title(f'Histogram of {titles[i]}')

axs[i, 1].set\_xlim([0, 255])

plt.tight\_layout(rect=[0, 0, 1, 0.96])

output\_dir = '/kaggle/working/img'

os.makedirs(output\_dir, exist\_ok=True)

output\_path = os.path.join(output\_dir, 'figure02\_log\_gamma\_hist\_comparison.png')

plt.savefig(output\_path, dpi=300, bbox\_inches='tight', pad\_inches=0)

plt.show()

print(f"Image and histogram comparison saved to: {output\_path}")

A screenshot of a graph

AI-generated content may be incorrect.