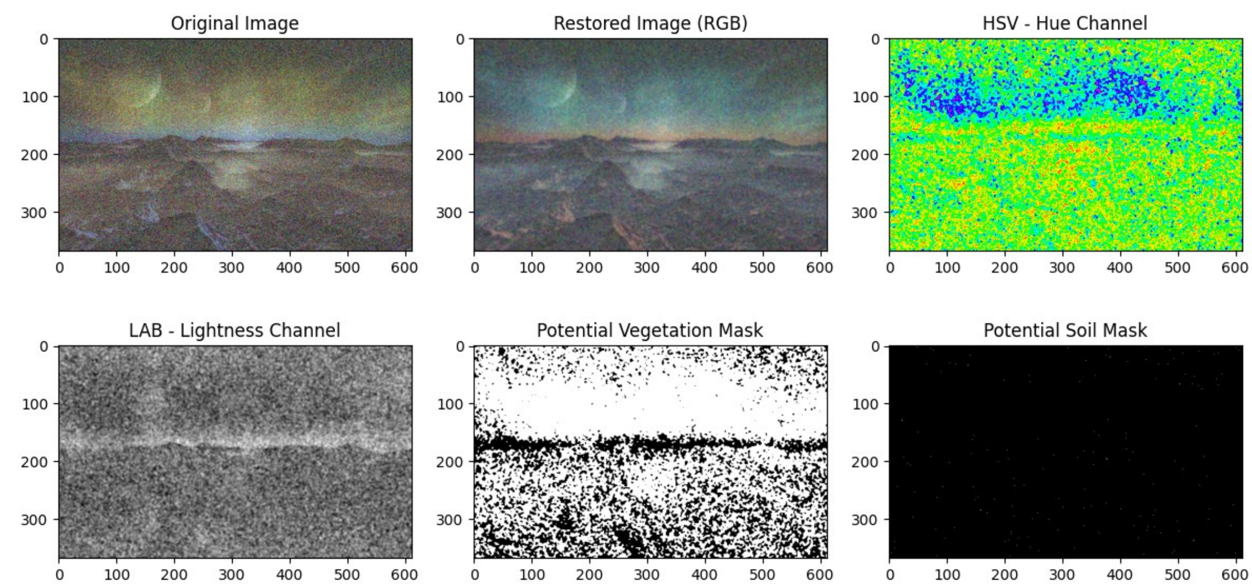
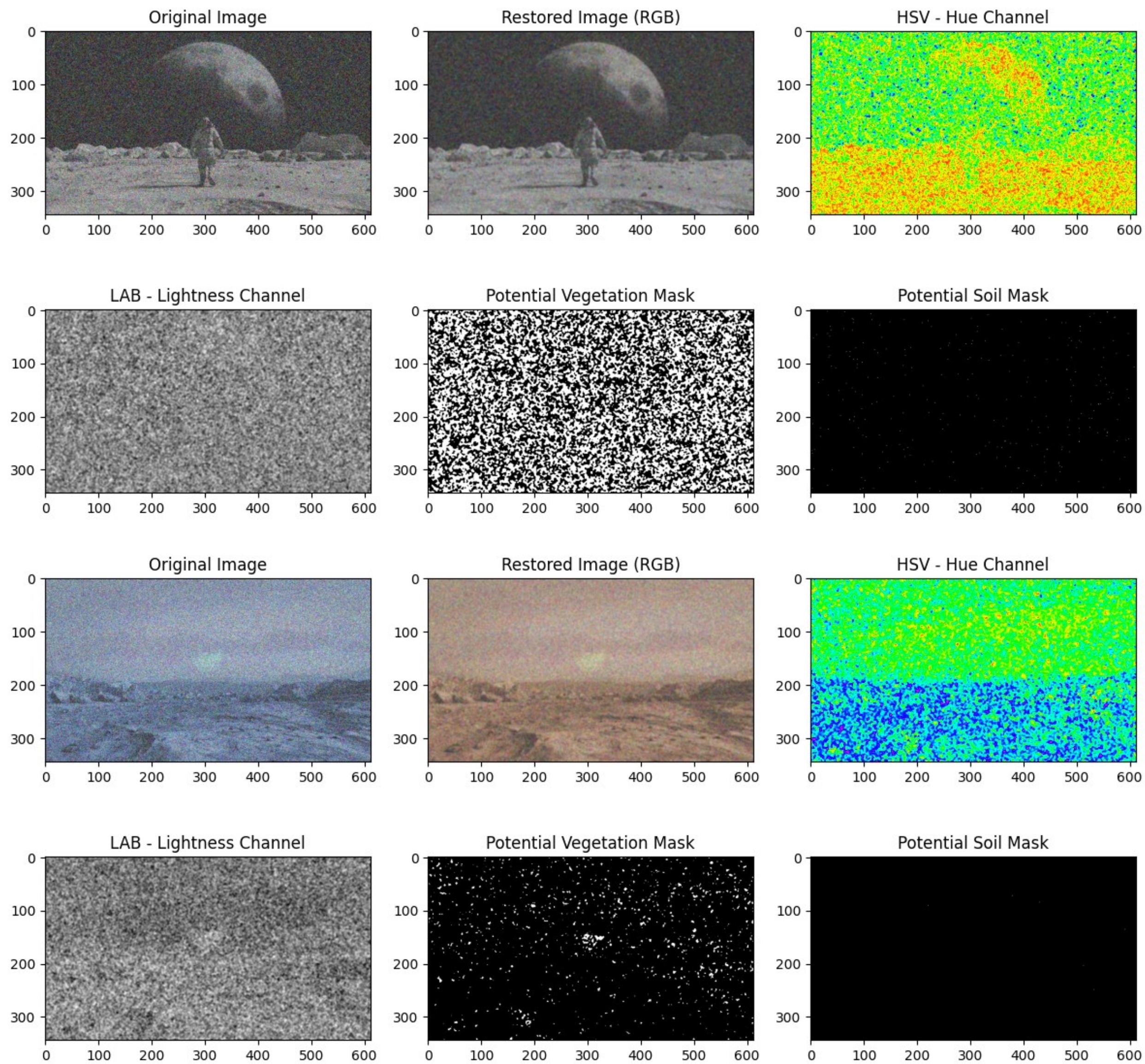
```
import cv2
import matplotlib.pyplot as plt

def restore_image(image_path):
  """
  Restores a corrupted image using noise reduction and inpainting.

  Args:
      image_path (str): Path to the corrupted image file.

  Returns:
      tuple: A tuple containing the original and restored images.
  """
  # Load the image
  img = cv2.imread(image_path)

  # Apply Gaussian blur for noise reduction
  blurred_img = cv2.GaussianBlur(img, (5, 5), 0)

  # Inpainting
  # Create a mask for inpainting
  mask = np.zeros(img.shape[:2], dtype="uint8")



  # Perform inpainting
  inpainted_img = cv2.inpaint(blurred_img, mask, 3, cv2.INPAINT_TELEA)

  return img, inpainted_img

# Define image paths (replace with your actual paths)
image_paths = [
  'images/1098413160-612x612.jpgnoisy.jpg',
  'images/1353996787-612x612.jpgnoisy.jpg',
  'images/399382166-612x612.jpgnoisy.jpg',
  'images/913058614-612x612.jpgnoisy.jpg'
]

# Restore and display images
for image_path in image_paths:
  original_img, restored_img = restore_image(image_path)
  cv2.imshow("Original Image", original_img)
  cv2.imshow("Restored Image", restored_img)
  cv2.waitKey(0)
  cv2.destroyAllWindows()

print("Image restoration completed for all images!")

#now part 2 of assignment
def restore_and_analyze_image(image_path):
```

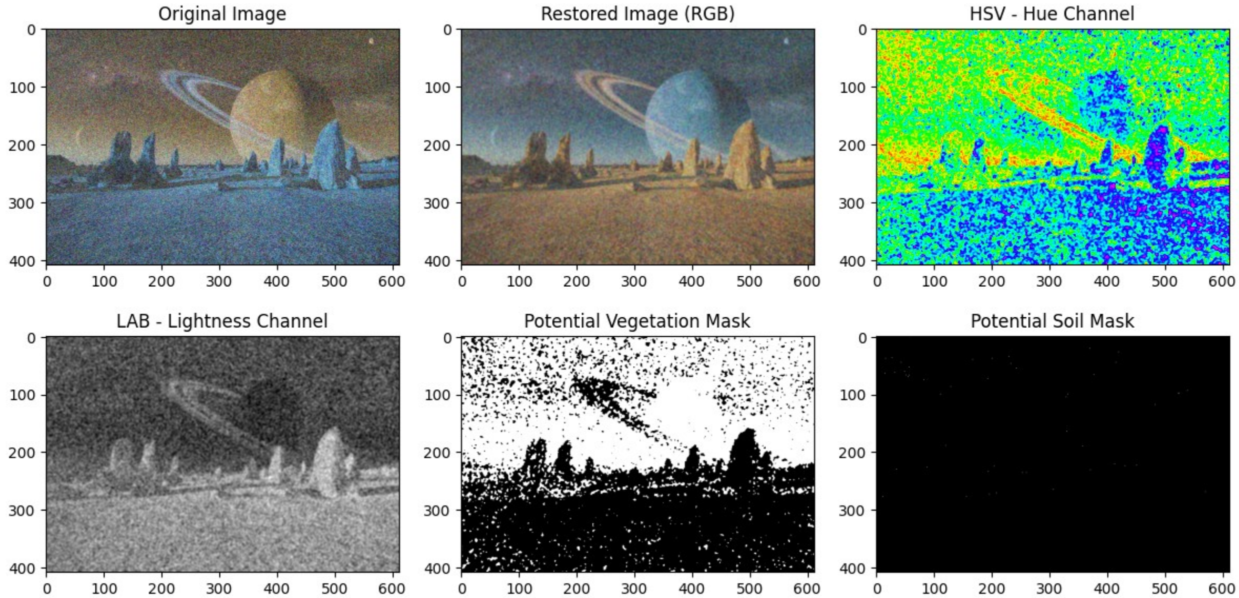
```
  """
  Restores, converts to color spaces, analyzes, and visualizes an
image.

  Args:
      image_path (str): Path to the corrupted image file.
  """
  # Restore image (same as previous implementation)
  original_img, restored_img = restore_image(image_path)

  # Convert to different color spaces
  bgr_img = restored_img  # Assuming restored image in BGR format
  hsv_img = cv2.cvtColor(bgr_img, cv2.COLOR_BGR2HSV)
  lab_img = cv2.cvtColor(bgr_img, cv2.COLOR_BGR2LAB)

  # Analyze color channels (**replace with your specific analysis**)

  # Example 1: Analyze vegetation using Green - Red ratio
  green_channel = bgr_img[:, :, 1]
  red_channel = bgr_img[:, :, 2]
  green_red_ratio = green_channel.astype(float) /
red_channel.astype(float)  # Avoid division by zero
  vegetation_mask = cv2.inRange(green_red_ratio, lower_veg_bound,
upper_veg_bound)  # Define thresholds

  # Example 2: Analyze soil using Saturation in HSV
  saturation = hsv_img[:, :, 1]
  soil_mask = cv2.inRange(saturation, lower_soil_bound,
upper_soil_bound)  # Define thresholds

  # Visualize specific color components
  plt.figure(figsize=(12, 6))
  plt.subplot(231), plt.imshow(original_img)
  plt.title("Original Image")
  plt.subplot(232), plt.imshow(cv2.cvtColor(bgr_img,
cv2.COLOR_BGR2RGB))
  plt.title("Restored Image (RGB)")
  plt.subplot(233), plt.imshow(hsv_img[:, :, 1], cmap="hsv")  # Plot
Hue channel
  plt.title("HSV - Hue Channel")
  plt.subplot(234), plt.imshow(lab_img[:, :, 2], cmap="gray")  # Plot
B channel (represents lightness in LAB)
  plt.title("LAB - Lightness Channel")
  plt.subplot(235), plt.imshow(vegetation_mask, cmap="gray")
  plt.title("Potential Vegetation Mask")
  plt.subplot(236), plt.imshow(soil_mask, cmap="gray")
  plt.title("Potential Soil Mask")
  plt.tight_layout()
  plt.show()
```

```python
# Define image paths (replace with your actual paths)
image_paths = [
 'images/1098413160-612x612.jpgnoisy.jpg',
  'images/1353996787-612x612.jpgnoisy.jpg',
  'images/399382166-612x612.jpgnoisy.jpg',
  'images/913058614-612x612.jpgnoisy.jpg'
]

# Color threshold definitions (
lower_veg_bound = 1.0  # Adjust for vegetation Green-Red ratio 
threshold
upper_veg_bound = 2.0  # Adjust for vegetation Green-Red ratio 
threshold
lower_soil_bound = 0.1  # Adjust for soil saturation threshold
upper_soil_bound = 0.3  # Adjust for soil saturation threshold

# Process each image
for image_path in image_paths:
  restore_and_analyze_image(image_path)

print("Image restoration, analysis, and visualization completed for 
all images!")
```

```
Image restoration completed for all images!
```




Original Image
Restored Image (RGB)
HSV - Hue Channel
LAB - Lightness Channel
Potential Vegetation Mask
Potential Soil Mask
Original Image
Restored Image (RGB)
HSV - Hue Channel
LAB - Lightness Channel
Potential Vegetation Mask
Potential Soil Mask
0 100 200 300
0 100 200 300 400 500 600




```
Image restoration, analysis, and visualization completed for all
images!
```