

Assignment 2

February 10, 2025

1 Pixel-wise contrast enhancement

1.1 Gray scale Image

```
1 def gamma_transform(image, gamma):
2     # Normalize to [0,1]
3     image = image.astype(np.float32) / 255.0
4     # Apply gamma correction
5     corrected = np.power(image, gamma)
6     # Convert back to [0,255]
7     return (corrected * 255).astype(np.uint8)
```

A gamma=1 is equals to the original gray-scale picture, a gamma=0.5 makes dark regions look brighter, and a gamma=2.0 darkens bright regions. See figure 1.



Figure 1: Pixel-wise contrast enhancement on a gray scale picture

1.2 Color Image - RGB correction

See figure 2.

```
1 def gamma_correct_hsv(image, gamma):
2     hsv_image = color.rgb2hsv(image)
3     hsv_image[..., 2] =
4         gamma_transform(
5             (hsv_image[..., 2] * 255).astype(np.uint8), gamma) / 255.0
6     return (color.hsv2rgb(hsv_image) * 255).astype(np.uint8)
```

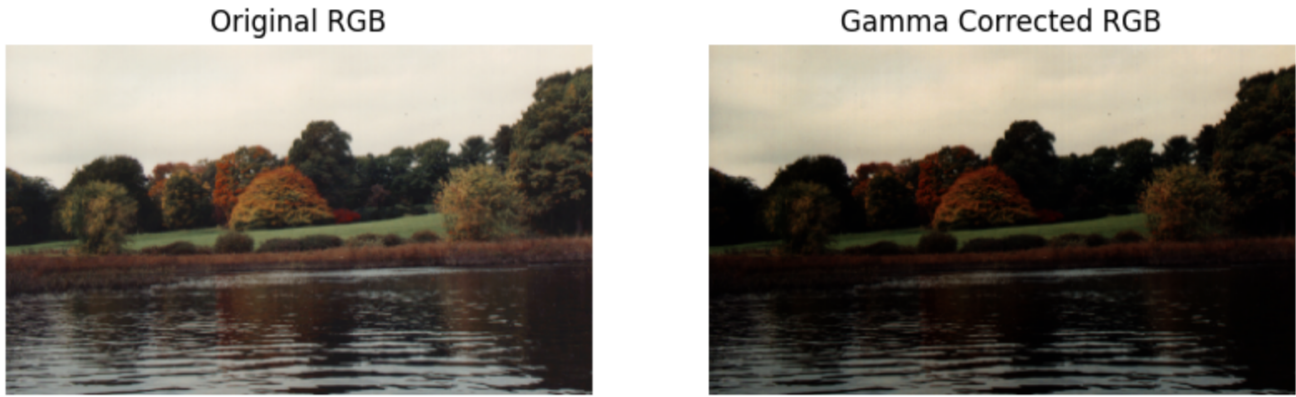


Figure 2: Pixel-wise contrast enhancement on a gray scale picture

1.3 Color Image - HSV color representation

```

1 def gamma_correct_hsv(image, gamma):
2     hsv_image = color.rgb2hsv(image)
3     hsv_image[..., 2] =
4         gamma_transform((hsv_image[..., 2] * 255).astype(np.uint8), gamma) /
5         255.0
6     return (color.hsv2rgb(hsv_image) * 255).astype(np.uint8)

```

See figure 3. The result of the RGB Correction is a little more brighter on the bright part, and thus look less natural. This may because the RGB Correction alters all channels independently, and the colors may be distorted. The HSV Correction modifies only the brightness while preserving colors, so it looks better.



Figure 3: Pixel-wise contrast enhancement on a gray scale picture

2 Reverb Convolution

3 Image filtering and enhancement

4 Histogram-based processing