Homework 1 - Image Manipulation

Basic image manipulation functions using Python

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from skimage import color, io

%matplotlib inline
plt.rcParams['figure.figsize'] = (10.0, 8.0)
plt.rcParams['image.interpolation'] = 'nearest'
plt.rcParams['image.cmap'] = 'gray'
```

In [2]:
```
def load(image_path):
    out = io.imread(image_path)
    out = out.astype(np.float64) / 255
    return out

def display(img):
    plt.figure(figsize=(5,5))
    plt.imshow(img)
    plt.axis('off')
    plt.show()

def resize_images(img1, img2):
    from skimage.transform import resize
    target_height = 1440
    target_width = 1920
    img1_resized = resize(img1, (target_height, target_width), anti_aliasing
    img2_resized = resize(img2, (target_height, target_width), anti_aliasing
    return img1_resized, img2_resized
```

In [3]:
```
def dim_image(image):
    return 0.5 * image ** 2

def convert_to_grey_scale(image):
    return color.rgb2gray(image)

def rgb_exclusion(image, channel):
    out = image.copy()
    if channel == 'R':
        out[:, :, 0] = 0
    elif channel == 'G':
        out[:, :, 1] = 0
    elif channel == 'B':
        out[:, :, 2] = 0
    return out

def lab_decomposition(image, channel):
    lab = color.rgb2lab(image)
```

```
    if channel == 'L':
        return lab[:, :, 0]
    elif channel == 'A':
        return lab[:, :, 1]
    elif channel == 'B':
        return lab[:, :, 2]

def hsv_decomposition(image, channel='H'):
    hsv = color.rgb2hsv(image)
    if channel == 'H':
        return hsv[:, :, 0]
    elif channel == 'S':
        return hsv[:, :, 1]
    elif channel == 'V':
        return hsv[:, :, 2]

def mix_images(image1, image2, channel1, channel2):
    out = rgb_exclusion(image1, channel1)
    out[:, out.shape[1] // 2:, :] = rgb_exclusion(image2[:, out.shape[1] //
    return out

def mix_quadrants(image):
    h, w, _ = image.shape
    h //= 2
    w //= 2
    out = np.empty_like(image)
    out[:h, :w, :] = rgb_exclusion(image[:h, :w, :], 'R')
    out[:h, w:, :] = dim_image(image[:h, w:, :])
    out[h:, :w, :] = image[h:, :w, :] ** 0.5
    out[h:, w:, :] = rgb_exclusion(image[h:, w:, :], 'R')
    return out
```

Load Images

In [4]:
```
image1_orig = load('images/zach.jpg')
image2_orig = load('images/icarus.jpg')

image1, image2 = resize_images(image1_orig, image2_orig)

print("Original Images:")
print("Zach (Human):")
display(image1)
print("Icarus (Dog):")
display(image2)
```

```
Original Images:
Zach (Human):
```

Icarus (Dog):

Dim Image

In [5]:
```
dimmed = dim_image(image1)
print("Dimmed Image:")
display(dimmed)
```

Dimmed Image:

Grayscale Conversion

In [6]:
```
grey = convert_to_grey_scale(image1)
print("Grayscale Image:")
display(grey)
```

```
Grayscale Image:
```

RGB Channel Exclusion

In [7]:
```
no_red = rgb_exclusion(image1, 'R')
no_green = rgb_exclusion(image1, 'G')
no_blue = rgb_exclusion(image1, 'B')

print("Without Red:")
display(no_red)
```

```
print("Without Green:")
display(no_green)
print("Without Blue:")
display(no_blue)
```

```
Without Red:
```

```
Without Green:
```

```
Without Blue:
```

LAB Decomposition

In [8]:
```
lab_l = lab_decomposition(image1, 'L')
lab_a = lab_decomposition(image1, 'A')
lab_b = lab_decomposition(image1, 'B')

print("L Channel:")
display(lab_l)
print("A Channel:")
display(lab_a)
print("B Channel:")
display(lab_b)
```

```
L Channel:
```

```
A Channel:
```

B Channel:

HSV Decomposition

In [9]:
```
hsv_h = hsv_decomposition(image1, 'H')
hsv_s = hsv_decomposition(image1, 'S')
hsv_v = hsv_decomposition(image1, 'V')

print("H Channel:")
display(hsv_h)
print("S Channel:")
display(hsv_s)
print("V Channel:")
display(hsv_v)
```

H Channel:

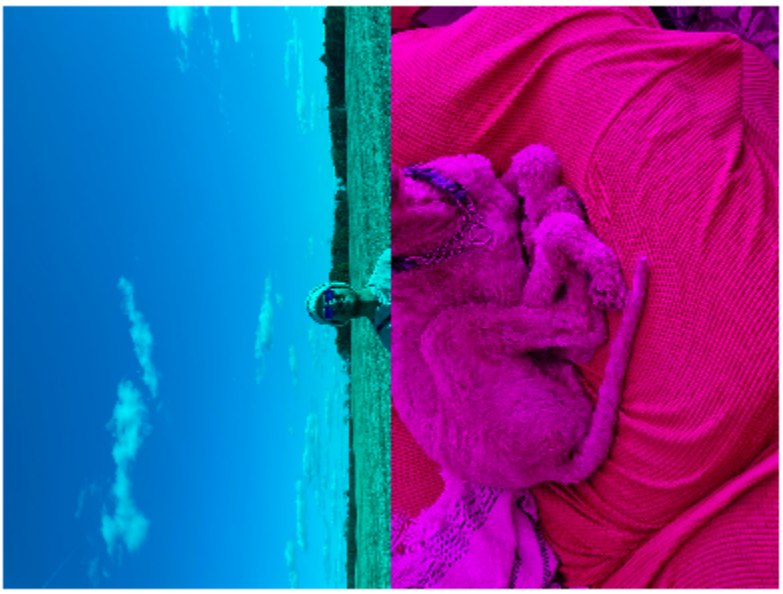
S Channel:

V Channel:

Mix Images

In [10]:
```
mixed = mix_images(image1, image2, 'R', 'G')
print("Mixed Image (left half no R, right half no G):")
display(mixed)
```

```
Mixed Image (left half no R, right half no G):
```



Mix Quadrants (Extra Credit)

In [11]:
```
quad_mixed = mix_quadrants(image1)
print("Mixed Quadrants:")
display(quad_mixed)
```

```
Mixed Quadrants:
```