

# Image Analogy

In [ ]:

```
import matplotlib.pyplot as plt
import numpy as np
import imageio
from main import *
```

## Part 1. Basic Usage

## Part 2. Artistic Filters

### Example on paper

In [ ]:

```
a1_img_fn = 'images/rhone-src.jpg'
a1_img = np.float32(cv2.cvtColor(cv2.imread(a1_img_fn), cv2.COLOR_BGR2RGB)/255.0
)

a2_img_fn = 'images/rhone.jpg'
a2_img = np.float32(cv2.cvtColor(cv2.imread(a2_img_fn), cv2.COLOR_BGR2RGB)/255.0
)

b1_img_fn = 'images/jakarta.jpg'
b1_img = np.float32(cv2.cvtColor(cv2.imread(b1_img_fn), cv2.COLOR_BGR2RGB)/255.0
)

a1_img=resize_img(a1_img,0.8)
a2_img=resize_img(a2_img,0.8)
b1_img=resize_img(b1_img,0.8)

fig, axes = plt.subplots(1, 3)
axes[0].imshow(a1_img)
axes[0].set_title('Image A'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(a2_img)
axes[1].set_title("Image A'"), axes[1].set_xticks([]), axes[1].set_yticks([])
axes[2].imshow(b1_img)
axes[2].set_title("Image B"), axes[2].set_xticks([]), axes[2].set_yticks([])
```

Out[ ]:

```
(Text(0.5, 1.0, 'Image B'), [], [])
```

Image A



Image A'



Image B



In [ ]:

```
kappa = 0.5  
b2_img_list_1 = start(a1_img,b1_img,a2_img,kappa,False)
```

In [ ]:

```
kappa = 0.5  
b2_img_list_2 = start(a1_img,b1_img,a2_img,kappa,True)
```

In [ ]:

```
kappa = 5  
b2_img_list_3 = start(a1_img,b1_img,a2_img,kappa,True)
```

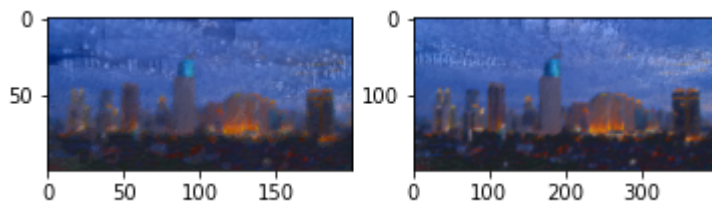
In [ ]:

```
kappa = 20  
b2_img_list_4 = start(a1_img,b1_img,a2_img,kappa,True)
```

In [ ]:

```
length = len(b2_img_list_1)  
  
fig, axes = plt.subplots(1, length)  
for i in range(length):  
    axes[i].imshow(b2_img_list_1[i])  
  
imageio.imsave('output/paint1.jpg', b2_img_list_1[-1])
```

Lossy conversion from float32 to uint8. Range [0, 1]. Convert image to uint8 prior to saving to suppress this warning.



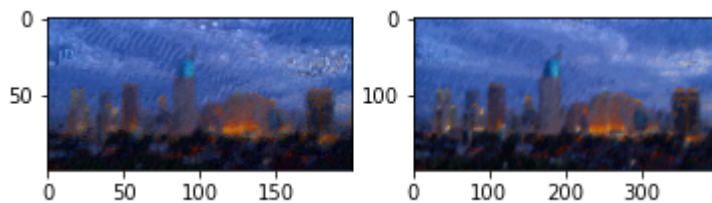
In [ ]:

```
length = len(b2_img_list_2)

fig, axes = plt.subplots(1, length)
for i in range(length):
    axes[i].imshow(b2_img_list_2[i])

imageio.imwrite('output/paint2.jpg', b2_img_list_2[-1])
```

Lossy conversion from float32 to uint8. Range [0, 1]. Convert image to uint8 prior to saving to suppress this warning.



In [ ]:

```
length = len(b2_img_list_3)

fig, axes = plt.subplots(1, length)
for i in range(length):
    axes[i].imshow(b2_img_list_3[i])

imageio.imwrite('output/paint3.jpg', b2_img_list_3[-1])
```

In [ ]:

```
length = len(b2_img_list_4)

fig, axes = plt.subplots(1, length)
for i in range(length):
    axes[i].imshow(b2_img_list_4[i])

imageio.imwrite('output/paint4.jpg', b2_img_list_4[-1])
```

**My images**

In [ ]:

```
a1_img_fn = 'images/beachA1.jpg'
a1_img = np.float32(cv2.cvtColor(cv2.imread(a1_img_fn), cv2.COLOR_BGR2RGB)/255.0
)

a2_img_fn = 'images/beachB1.jpg'
a2_img = np.float32(cv2.cvtColor(cv2.imread(a2_img_fn), cv2.COLOR_BGR2RGB)/255.0
)

b1_img_fn = 'images/flowerB1.jpg'
b1_img = np.float32(cv2.cvtColor(cv2.imread(b1_img_fn), cv2.COLOR_BGR2RGB)/255.0
)

a1_img=resize_img(a1_img,1)
a2_img=resize_img(a2_img,1)
b1_img=resize_img(b1_img,1)

fig, axes = plt.subplots(1, 3)
axes[0].imshow(a1_img)
axes[0].set_title('Image A'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(a2_img)
axes[1].set_title("Image A'"), axes[1].set_xticks([]), axes[1].set_yticks([])
axes[2].imshow(b1_img)
axes[2].set_title("Image B"), axes[2].set_xticks([]), axes[2].set_yticks([])
```

Out[ ]:

```
(Text(0.5, 1.0, 'Image B'), [], [])
```



In [ ]:

```
kappa = 1
b2_img_list_flower = start(a1_img,b1_img,a2_img,kappa,True)
```

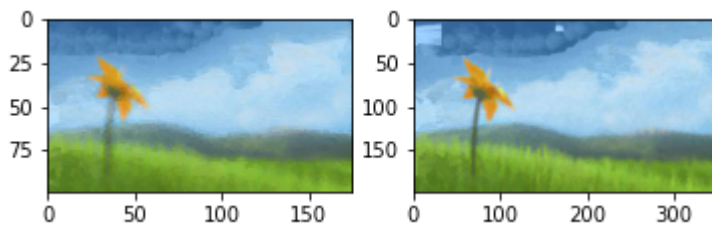
In [ ]:

```
length = len(b2_img_list_flower)

fig, axes = plt.subplots(1, length)
for i in range(length):
    axes[i].imshow(b2_img_list_flower[i])

imageio.imsave('output/flower_oil.jpg', b2_img_list_flower[-1])
```

Lossy conversion from float32 to uint8. Range [0, 1]. Convert image to uint8 prior to saving to suppress this warning.



In [ ]:

```
a1_img_fn = 'images/beachA1.jpg'
a1_img = np.float32(cv2.cvtColor(cv2.imread(a1_img_fn), cv2.COLOR_BGR2RGB)/255.0
)

a2_img_fn = 'images/beachA2.jpg'
a2_img = np.float32(cv2.cvtColor(cv2.imread(a2_img_fn), cv2.COLOR_BGR2RGB)/255.0
)

b1_img_fn = 'images/leavesC1.jpg'
b1_img = np.float32(cv2.cvtColor(cv2.imread(b1_img_fn), cv2.COLOR_BGR2RGB)/255.0
)

a1_img=resize_img(a1_img,1)
a2_img=resize_img(a2_img,1)
b1_img=resize_img(b1_img,1)

fig, axes = plt.subplots(1, 3)
axes[0].imshow(a1_img)
axes[0].set_title('Image A'), axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(a2_img)
axes[1].set_title("Image A'"), axes[1].set_xticks([]), axes[1].set_yticks([])
axes[2].imshow(b1_img)
axes[2].set_title("Image B"), axes[2].set_xticks([]), axes[2].set_yticks([])
```

Out[ ]:

(Text(0.5, 1.0, 'Image B'), [], [])



In [ ]:

```
kappa = 1.3  
b2_img_list_leaves = start(a1_img,b1_img,a2_img,kappa,True)
```

In [ ]:

```
length = len(b2_img_list_leaves)  
  
fig, axes = plt.subplots(1, length)  
for i in range(length):  
    axes[i].imshow(b2_img_list_leaves[i])  
  
imageio.imwrite('output/leaves_oil.jpg', b2_img_list_leaves[-1])
```

Lossy conversion from float32 to uint8. Range [0, 1]. Convert image to uint8 prior to saving to suppress this warning.

