Extracting Foreground Masks

September 2, 2019

1 Description

The basic idea is to define an spherical decision boundary around the color of the magenta petals and the yellow core of the flower. Using some threshold, we can extract pixels with intensities within said threshold from our defined colors.

We create a new image d from a given image f such that,

$$d(x, y) = \|f(x, y) - v_{color}\|^2$$

where v_{color} is the template color vector for matching.

[1]:

```
import numpy as np
from matplotlib import pyplot as plt
from PIL import Image, ImageFilter
```

This demonstration is done on `flower.jpg`. We use an in-built blur function to smoothen the image, which avoids scatter artifacts.

The same algorithm was used to generate mask of `bird.jpg`

[2]:

```
a = Image.open('./1/data/flower.jpg')
a = a.filter(ImageFilter.BLUR)
a = np.array(a)
h, w = a.shape[:2]
```

2 Define colors and thresholds

[3]:

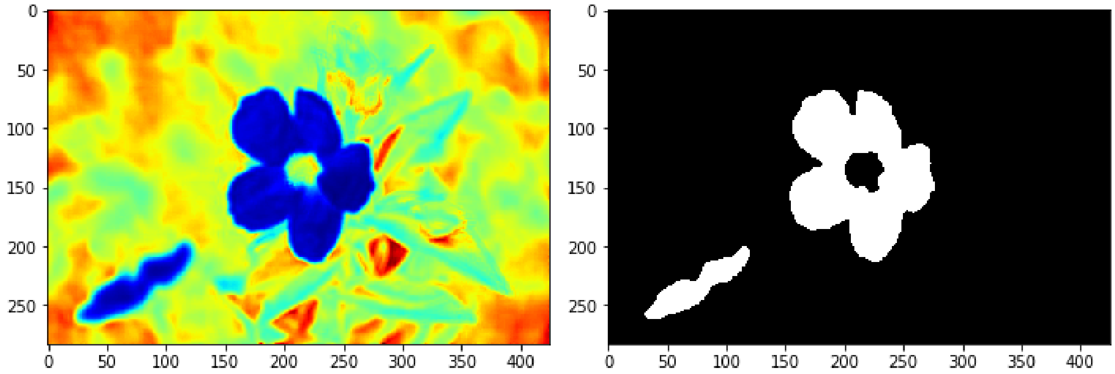
```
purple = [255, 0, 255]
t_purple = 0.25
yellow = [220, 100, 20]
t_yellow = 0.1
```

3 Generating sub-mask

Let us create a jet-plot of the distance matrix and threshold to extract our masks

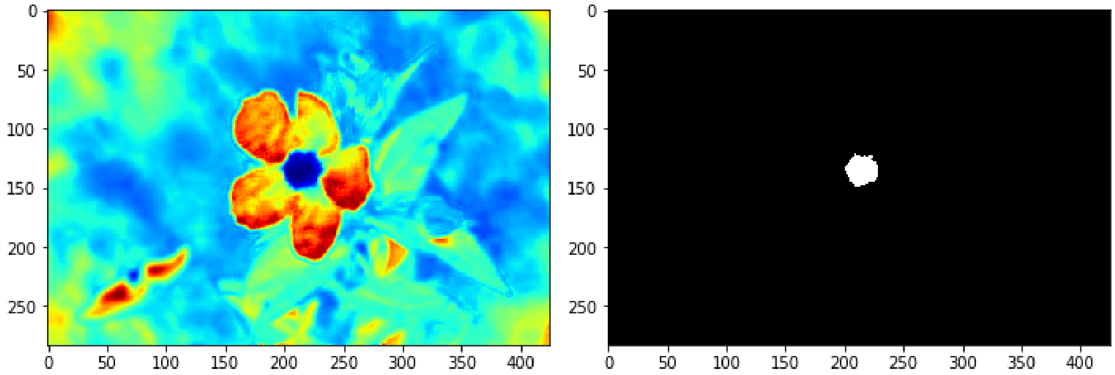
[4]:

```
mask1 = a.copy()
mask1 = np.square(mask1 - purple)
mask1 = np.sum(mask1, 2)
mask1 = mask1 / np.max(mask1)
plt.figure(figsize=(10,4))
plt.subplot(121)
plt.imshow(mask1, cmap='jet')
mask1 = mask1 <= t_purple
plt.subplot(122)
plt.imshow(mask1, cmap='gray')
plt.tight_layout()
plt.show()
```



[5]:

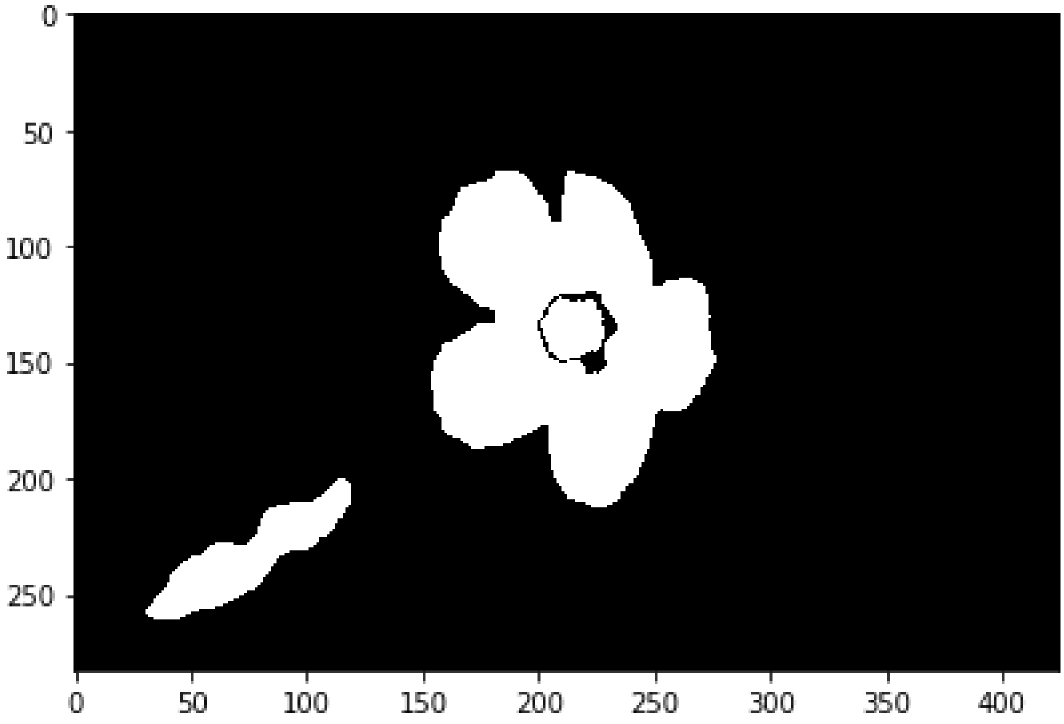
```
mask2 = a.copy()
mask2 = np.square(mask2 - yellow)
mask2 = np.sum(mask2, 2)
mask2 = mask2 / np.max(mask2)
plt.figure(figsize=(10,4))
plt.subplot(121)
plt.imshow(mask2, cmap='jet')
mask2 = mask2 <= t_yellow
plt.subplot(122)
plt.imshow(mask2, cmap='gray')
plt.tight_layout()
plt.show()
```



4 Merging the masks

[6]:

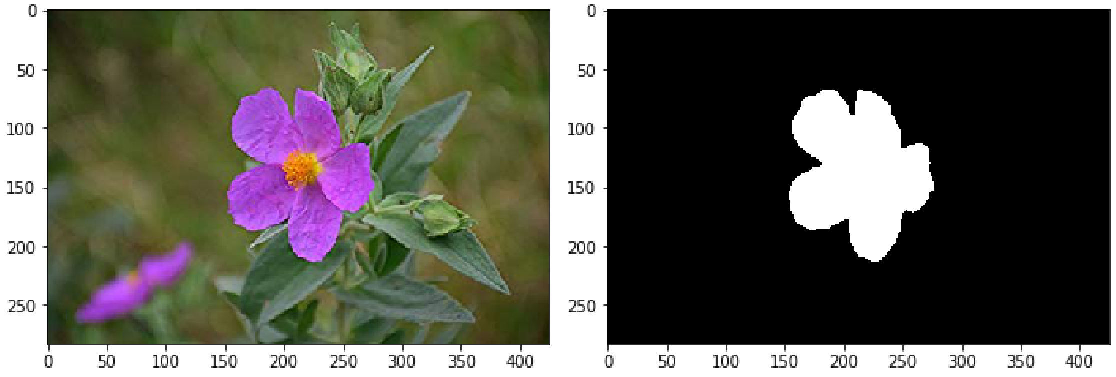
```
mask = mask1 + mask2
plt.imshow(mask, cmap='gray')
plt.tight_layout()
plt.show()
```



5 Post-processing

As we can see, the mask generated above is not quite what we wanted, but using a basic drawing software (MS Paint), we were able to remove the blob on the bottom left as well as fill in the hole inside the flower.

[7]:
```
mask_final = Image.open('./1/data/mask1.png')
img = Image.open('./1/data/flower.jpg')
plt.figure(figsize=(10,4))
plt.subplot(121)
plt.imshow(img)
plt.subplot(122)
plt.imshow(mask_final, cmap='gray')
plt.tight_layout()
plt.show()
```

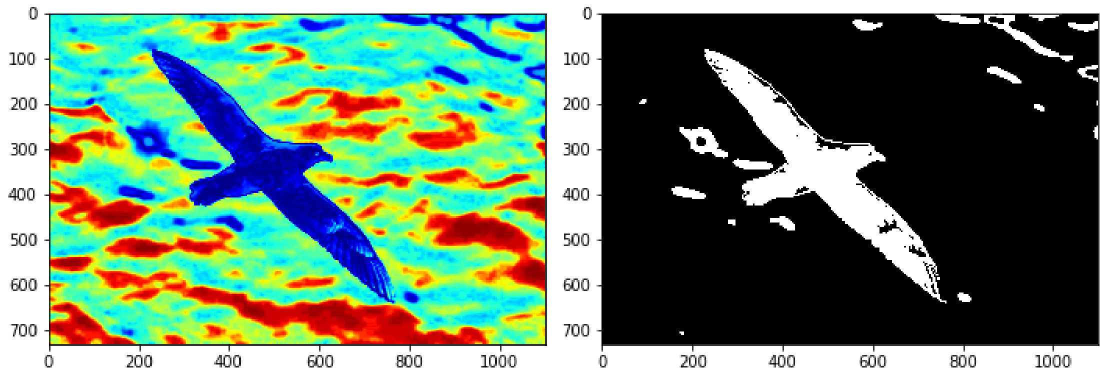


6 Bird image

Similarly, for `bird.jpg` the mask was extracted in similar fashion by multiple hit-n-trials.

[8]:
```
bird_plot = Image.open('./fig.png')
bird_plot
```

[8]:



After some manual correction in MS Paint, we get the following mask

[9]:
```
mask_final = Image.open('./1/data/mask2.png')
img = Image.open('./1/data/bird.jpg')
plt.figure(figsize=(10,4))
plt.subplot(121)
plt.imshow(img)
plt.subplot(122)
plt.imshow(mask_final, cmap='gray')
plt.tight_layout()
plt.show()
```