11_Clustering_Photos

July 10, 2025

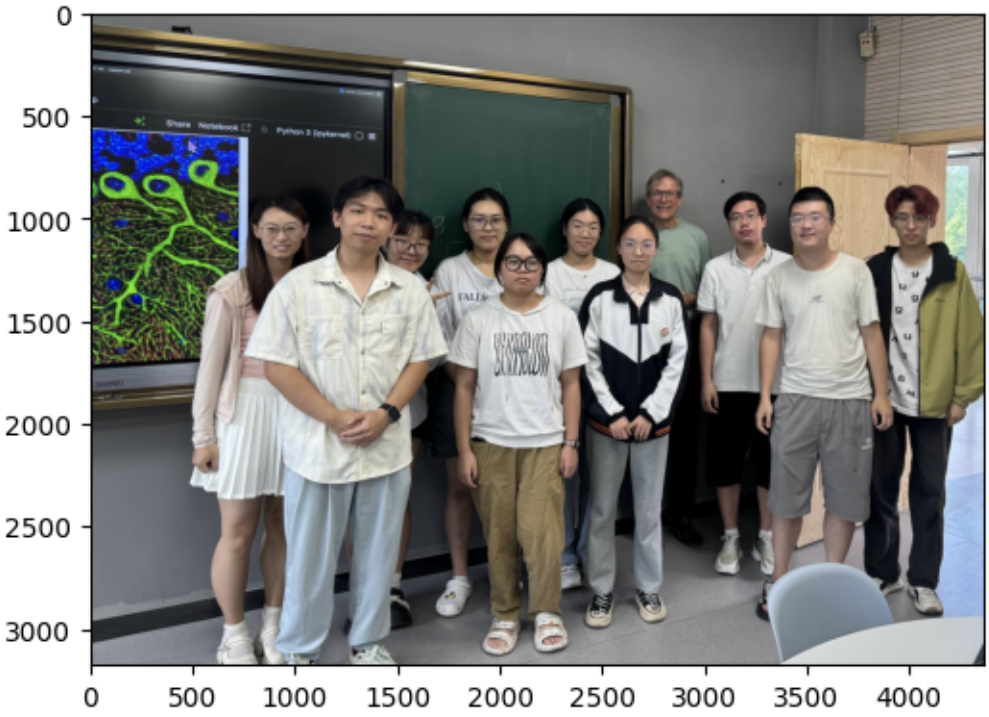
1 Clustering Images

[2]:
```
from numpy import concatenate, zeros, linspace

from matplotlib.pyplot import subplots

from matplotlib.image import imread
```

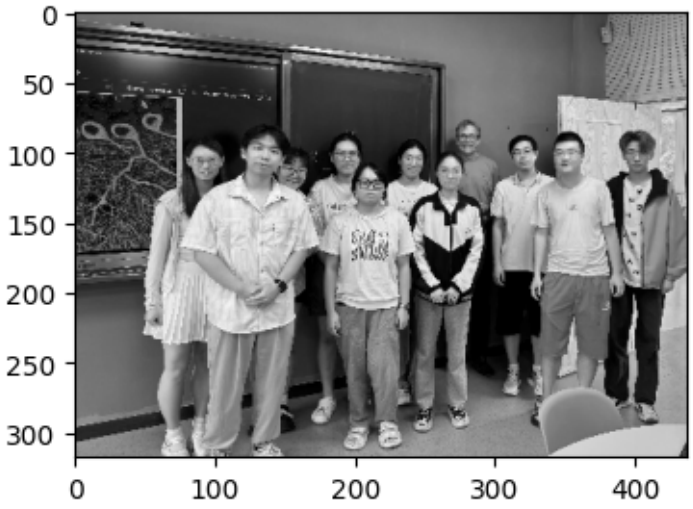
1.1 Load Image

[7]:
```
img = imread('IMG_9113.jpg')

fig, ax = subplots(figsize=(6, 5))

ax.imshow(img);

img.shape
```

[7]:
```
(3176, 4364, 3)
```



[13]:
```
downsample = 10

fig, ax = subplots(figsize=(5, 3))

ax.imshow(img[::downsample, ::downsample, 0], cmap='gray');
```



1.2 Downsample and Flatten RGB Layers

[17]:
```
downsample = 5

img_r = img[::downsample, ::downsample, 0].reshape(-1, 1)
img_g = img[::downsample, ::downsample, 1].reshape(-1, 1)
img_b = img[::downsample, ::downsample, 2].reshape(-1, 1)

img_reshaped = concatenate((img_r, img_g, img_b), axis = 1)

img_reshaped.shape
```

[17]:
```
(555228, 3)
```

[19]:
```
img[0, 1000:1010, 0]
```

[19]:
```
array([127, 127, 127, 127, 127, 127, 127, 127, 127, 127], dtype=uint8)
```

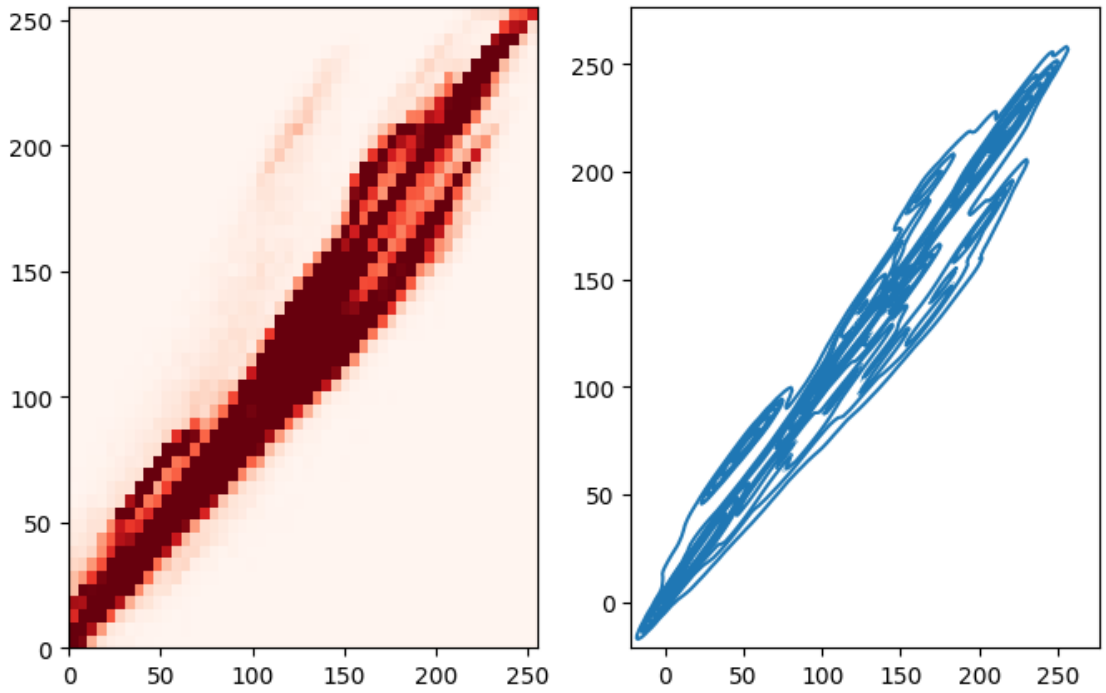
1.3 Visualise State Space

Seaborn: https://seaborn.pydata.org

c.f. pair grid example https://seaborn.pydata.org/examples/pair_grid_with_kde.html

kdeplot documentation https://seaborn.pydata.org/generated/seaborn.kdeplot.html

[22]:
```
fig, ax = subplots(ncols=2, figsize=(8, 5))

# 2D Histogram
ax[0].hist2d(img_reshaped[:, 0], img_reshaped[:, 1], bins=50, vmax=500,␣
 ↪cmap='Reds');

from seaborn import kdeplot

# Density Plot
kdeplot(x=img_reshaped[:, 0], y=img_reshaped[:, 1], ax=ax[1]);
```



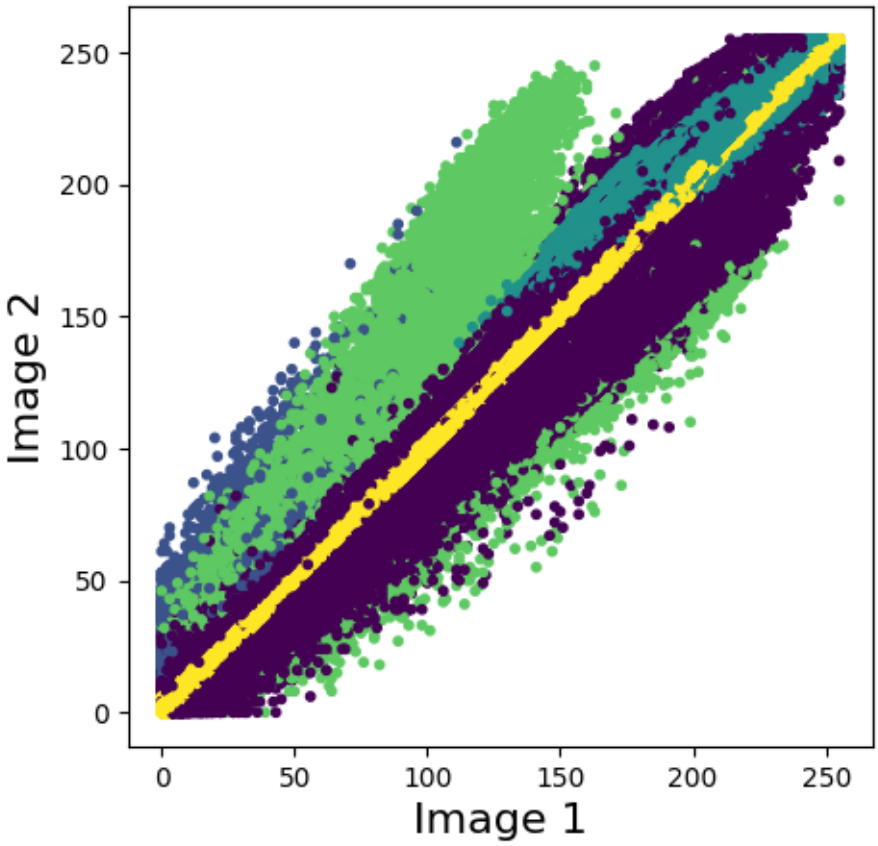
1.4 GMM Clustering

[24]:
```
from sklearn.mixture import GaussianMixture
```

[25]:
```
n_components = 5

SEED = 12345890

gmm = GaussianMixture(n_components=n_components, random_state=SEED)

all_img_labels = gmm.fit_predict(img_reshaped)

all_img_labels[0]
```

[25]:
```
np.int64(4)
```

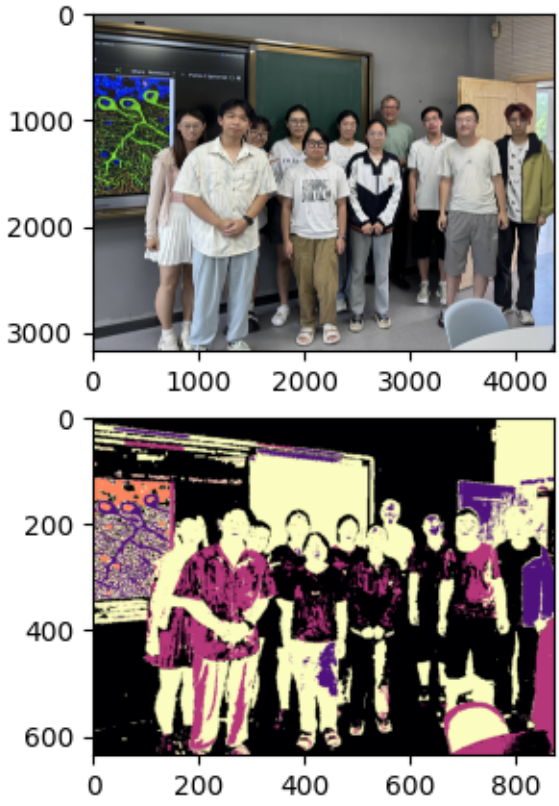
[28]:
```
fig, ax = subplots(figsize=(5, 5))

ax.scatter(img_reshaped[:, 0], img_reshaped[:, 1], c=all_img_labels, s=10)

ax.set_xlabel('Image 1', fontsize=16)
ax.set_ylabel('Image 2', fontsize=16);
```



1.5 Re-map Labels to Image

[30]:
```
all_img_labels_mapped = zeros(img[::downsample, ::downsample, 0].shape)

mask = all_img_labels_mapped>-1

all_img_labels_mapped[mask] = all_img_labels
```

[32]:
```
fig, ax = subplots(nrows=2, figsize=(5, 5))

ax[0].imshow(img, cmap='gray');
ax[1].imshow(all_img_labels_mapped, cmap='magma_r');
```


0
1000
2000
3000
0
1000
2000
3000
4000
0
200
400
600
0
200
400
600
800

[ ]: