



ML Day16 (Matplotlib)



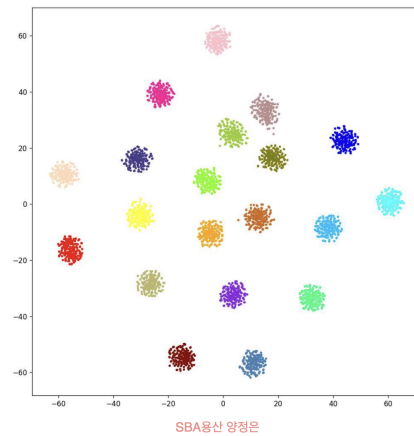
Few-Shot Image Recognition with Knowledge Transfer

Zhimao Peng¹, Zechao Li¹, Junge Zhang¹, Yan Li¹, Guo-Jun Qi², Jinhui Tang¹

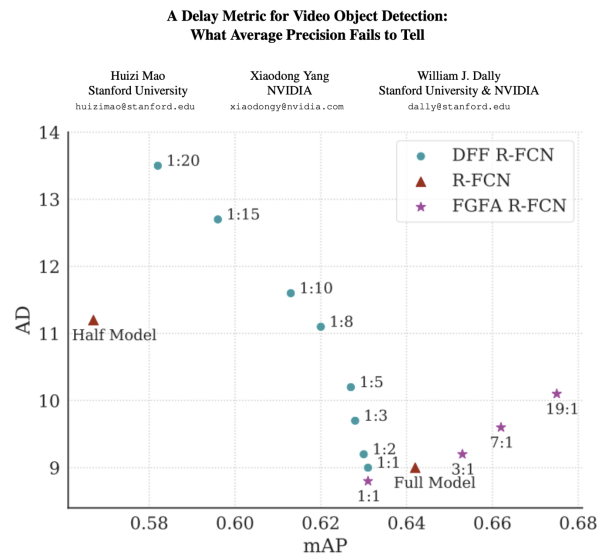
¹School of Computer Science and Engineering, Nanjing University of Science and Technology

²Institute of Automation, Chinese Academy of Sciences ³Huawei Cloud

{zhimaopeng, guojunqi}@gmail.com, {zechao.li, jinhuitang}@njust.edu.cn
jgzhang@nlpr.ia.ac.cn, yan.li@cripac.ia.ac.cn



2



4

▼ 시각화 실습 (2)

```
np.random.seed(0)

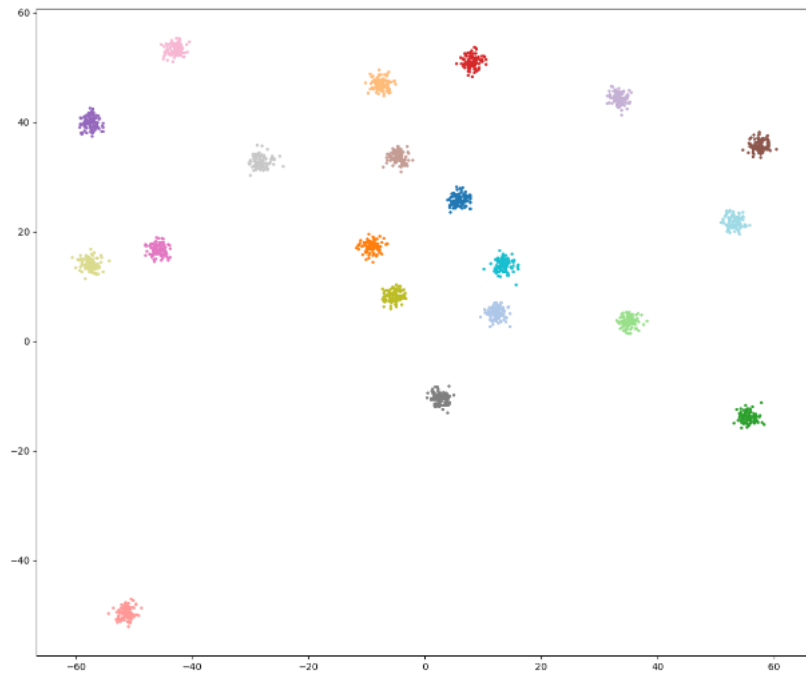
n_class = 20
n_data = 100

center_pt = np.random.uniform(-60, 60, (n_class, 2)) # 중심이 될 point를 uniform을 통해 -60 ~ 60 사이 동일한 확률을 가진 수를 (n_class,
cmap = cm.get_cmap('tab20')
colors = [cmap(i) for i in range(n_class)]

fig, ax = plt.subplots(figsize=(12, 10))

for class_idx in range(n_class):
    center = center_pt[class_idx]
    x_data = center[0] + np.random.normal(0, 1, (n_data,))
    y_data = center[1] + np.random.normal(0, 1, (n_data,))
    ax.scatter(x_data, y_data,
               s=5,
               c=colors[class_idx])

fig.tight_layout()
plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.cm as cm

names = ['DFF R-FCN', 'R-FCN', 'FGFA R-FCN']
dff_data = np.array([(0.581, 13.5), (0.598, 12.8), (0.618, 11.7),
                    (0.62, 11.3), (0.624, 10.2), (0.627, 9.8),
                    (0.629, 9.2), (0.63, 9)])
r_data = np.array([(0.565, 11.2), (0.645, 9)])
fgfa_data = np.array([(0.63, 8.8), (0.653, 9.3), (0.664, 9.6),
                    (0.676, 10.1)])
dff_text = ['1:20', '1:15', '1:10', '1:8', '1:5', '1:3', '1:2', '1:1']
r_text = ['Half Model', 'Full Model']
fgfa_text = ['1:1', '3:1', '7:1', '19:1']

colors_li=['skyblue', 'red', 'purple']
marker_li=['o', '^', '*']

fig, ax = plt.subplots(figsize=(7, 5))

for i in range(len(dff_data)):
    # dff_data의 길이 만큼 for문으로
    ax.scatter(dff_data[i][0], dff_data[i][1], # dff_data[i]의 0번 원소를 x축 데이터로, dff_data[i]의 1번 원소를 y축 데이터
               color='skyblue',
               s=50,
               marker='o')
    ax.text(x=dff_data[i][0], y=dff_data[i][1], # 위 코드에서 scatter로 구현될 각 점들에 해당하는 text를 지정
            s=dff_text[i],
            fontsize=10,
            va='bottom',
            ha='left')
    # 문자의 수평부분이 어디에 위치할 지 지정
    # 문자의 수직부분이 어디에 위치할 지 지정

for j in range(len(fgfa_data)):
    ax.scatter(fgfa_data[j][0], fgfa_data[j][1],
               color='purple',
               s=50,
               marker='*')
    ax.text(x=fgfa_data[j][0], y=fgfa_data[j][1],
            s=fgfa_text[j],
            fontsize=10,
            va='bottom',
            ha='left')

for k in range(len(r_data)):
    ax.scatter(r_data[k][0], r_data[k][1],
               color='r',
               s=50,
               marker='^')
    ax.text(x=r_data[k][0], y=r_data[k][1],
            s=r_text[k],
            fontsize=10,
            va='bottom',
            ha='left')

ax.set_xlabel('mAP',
```

```

        fontsize=10)
ax.set_ylabel('AD',
              fontsize=10)

ax2 = ax.twinx()                                # 위에서 그린 scatter의 ax의 twinx를 ax2로 지정

for f in range(len(names)):                      # for문으로 names의 길이만큼(3) 빈 scatter를 그리고
    ax2.scatter([], [],                          # 이후에 ax2를 활용하여 legend를 그리기 위해 size와 color, marker, label을 지정해준다.
                 s=20,
                 c=colors_li[f],
                 marker=marker_li[f],
                 label=names[f])

ax2.legend(loc='upper right',                   # 비어있는 scatter를 활용하여 legend 생성
          bbox_to_anchor=(1, 1),
          fontsize=10,
          ncol=1)
ax2.tick_params(axis='y',                      # 빈 scatter로 인해 그려진 우측 tick과 ticklabel을 삭제
               right=False, labelright=False)

ax.set_xlim([0.55, 0.7])
ax.set_ylim([8.5, 14])

ax.grid(linestyle='--')                       # grid의 linestyle을 지정(점선)
fig.tight_layout()
plt.show()

```

