

# Assignment4 Report

2019311851 박태희

## 1. extract features (e.g., SIFT) from images



## 2. Learn visual dictionary (e.g., K-means clustering)



1. 과제에서 주어진 sift 파일에는 각 이미지마다 추출한 feature들이  $n \times 128$ 차원으로 저장되어 있다. 먼저 이 모든 feature들을 가지고 k-means clustering 과정을 통해 feature들을  $m$ 개의 그룹으로 나눌 수 있는 중심점을 찾아야하기 때문에 sklearn에 있는 kmeans clustering 을 활용하여 중심점을 찾아내었다.

```
def kmeans(sifts):  
    model = KMeans(n_clusters=128, random_state=10, max_iter=1000)  
  
    # 학습 코드  
    model.fit(sifts)  
    print(model.cluster_centers_)  
    np.save('cluster5', model.cluster_centers_)  
    np.save('predict', model.predict(sifts))  
  
    # 사전학습 load  
    centers = np.load('cluster5.npy')  
    predict = np.load('predict.npy')  
    return centers, predict
```

이때 센터의 개수에 해당하는  $n\_clusters$ 같은 경우 8부터 1024까지 다양한 숫자를 적용해보았고, 이때 가장 높은 정확도를 나타내는 128개의  $n\_cluster$ 를 최종 결과값을 내는데 사용하게 되었다.

2. 다음 과정에서는 1에서 구한 center들을 활용해 boww를 만들게 된다. 이 과정에서 센터와 각 이미지 feature들의 거리를 구해야 하는데, 이때에는 l2 normalization(유클리디안 거리)을 활용하여 거리를 계산하였다. L2 거리를 구하는데에는 numpy에서 제공하는 함수인 linalg.norm을 활용하였고, 이미지의 각 feature와 센터 사이의 거리를 계산하고 가장 거리가 가까운 센터에 해당하는 인덱스에 +1을 해주는 방식으로 boww를 구성하였다. 다만, 여기서 2번째로 가까운 센터에 대해서 가장 작은 거리의 1.5배 이하의 값이 나온다면 그 센터에 대해서만 추가적으로 +1을 해주는 과정을 거쳤다(실제 성능 향상을 보임)

이렇게 되면 boww는 각 이미지의 feature들에 대해 center와의 유사도를 계산한 일종의 histogram과 유사한 1000 x center\_num 의 이중리스트가 생성되게 된다.

```
def histo(sifts, centers):
    hist_arr = []
    center_n = len(centers)
    for sift in sifts:
        l2_norm = [0 for _ in range(center_n)]
        for feature in sift:
            tmp = np.zeros(len(centers))
            for i, center in enumerate(centers):
                tmp[i] += np.linalg.norm(feature - center, ord=2)
            n = np.argmin(tmp)
            l2_norm[n] += 1
            number = tmp[n]
            tmp[n] = np.inf
            n = np.argmin(tmp)
            if number * 1.5 > tmp[n]:
                l2_norm[n] += 1
        hist_arr.append(l2_norm)
    return np.array(hist_arr, dtype="float32")
```

3. 마지막으로는 과제 설명에 나온 형식에 맞추어 .des파일에 입력해주었다.

```
with open('A4_t3.des', 'wb') as fp:
    nd = np.array([1000, len(k_center)], dtype=int)
    fp.write(nd.tobytes())
    fp.write(boww.tobytes())
```