



# 컴퓨터 비전

## 과제 #6

담당교수: 김낙현 교수님

제출일자: 21.06.02

학 과: 전자공학과

학 년: 4학년

이 름: 배준성






학 번: 201501487



**한국외국어대학교**  
HANKUK UNIVERSITY OF FOREIGN STUDIES

1. 강의자료 8-F의 13쪽에 있는 자료(뒤 15쪽)에서는 'elephant.jpg' 영상에 대한 실험을 수행하였다. 이와 유사하게 자전거, 자동차, 동물 등 다양한 영상을 5장 수집하여 실험을 수행하여 나오는 결과를 제시하라.

사용한 이미지

iguana	car	lion	german_shepherd	bicycle
				

iguana 실행 결과

```
[('n01677366', 'common_iguana', 0.49849588),
 ('n01675722', 'banded_gecko', 0.17840719),
 ('n01688243', 'frilled_lizard', 0.15797351)]
```

car 실행결과

```
[('n04285008', 'sports_car', 0.50676185),
 ('n03459775', 'grille', 0.37127006),
 ('n03100240', 'convertible', 0.04808751)]
```

lion 실행결과

```
[('n02129165', 'lion', 1.0),
 ('n02403003', 'ox', 3.3208785e-08),
 ('n02130308', 'cheetah', 2.4460903e-08)]
```

german\_shepherd 실행결과

```
[('n02106662', 'German_shepherd', 0.9559662),
 ('n02105162', 'malinois', 0.043974064),
 ('n03803284', 'muzzle', 2.4988598e-05)]
```

bicycle 실행결과

```
[('n03792782', 'mountain_bike', 0.5761102),
 ('n04509417', 'unicycle', 0.104438886),
 ('n02835271', 'bicycle-built-for-two', 0.10287503)]
```

모든 실험 결과에 상당히 높은 수준의 정확도를 보였다 특히 lion과 german\_shepherd의 경우 아주 높은 정확도를 보였다. car의 경우 sports\_car에 해당하는 모델은 아니었지만 car의 범주안에 들어가는 정확도를 보였다.

2. 앞의 프로그램에서 'VGG16' 대신 'ResNet50'을 적용한 결과를 제시하라.

사용한 이미지는 전과 같다.

iguana 실행 결과

```
[('n01677366', 'common_iguana', 0.9770001),  
 ('n01688243', 'frilled_lizard', 0.0071277292),  
 ('n01687978', 'agama', 0.005868777)]
```

car 실행결과

```
[('n03459775', 'grille', 0.6996278),  
 ('n03770679', 'minivan', 0.07003792),  
 ('n02930766', 'cab', 0.06276045)]
```

lion 실행결과

```
[('n02129165', 'lion', 0.99927574),  
 ('n02129604', 'tiger', 0.00020777392),  
 ('n02127052', 'lynx', 0.00012629197)]
```

german\_shepherd 실행결과

```
[('n02106662', 'German_shepherd', 0.9333317),  
 ('n02105162', 'malinois', 0.066593364),  
 ('n02114712', 'red_wolf', 2.852003e-05)]
```

bicycle 실행결과

```
[('n03792782', 'mountain_bike', 0.17319182),  
 ('n02835271', 'bicycle-built-for-two', 0.15606861),  
 ('n03930313', 'picket_fence', 0.13590068)]
```

동물 사진들에 대한 정확도는 높아진 반면 car와 bicycle에 대한 정확도는 심하게 떨어졌으며 car 이미지의 경우 grille 이라고 인식하였다.

3. 이 자료 8~13쪽에서 수행된 과정을 'VGG16' 대신 'ResNet50'을 적용한 결과를 제시하라.  
이 경우 10쪽의 features 관련 부분이 어떻게 수정되어야 하는지 검토해야 한다.

```
4. import keras
5.
6. from keras.applications.resnet50 import ResNet50
7.
8. conv_base = ResNet50(weights='imagenet',
9.                        include_top=False,
10.                       input_shape=(150, 150, 3))
11.
12. conv_base.summary()
13. import os
14. import numpy as np
15. from keras.preprocessing.image import ImageDataGenerator
16.
17. base_dir = '/content/gdrive/My Drive/cats_and_dogs_small'
18.
19. train_dir = os.path.join(base_dir, 'train')
20. validation_dir = os.path.join(base_dir, 'validation')
21. test_dir = os.path.join(base_dir, 'test')
22.
23. datagen = ImageDataGenerator(rescale=1./255)
24. batch_size = 20
25. def extract_features(directory, sample_count):
26.     features = np.zeros(shape=(sample_count, 5, 5, 2048))
27.     labels = np.zeros(shape=(sample_count))
28.     generator = datagen.flow_from_directory(# 데이터를 읽음
29.     directory,
30.     target_size=(150, 150),
31.     batch_size=batch_size,
32.     class_mode='binary')
33.     i = 0
34.     for inputs_batch, labels_batch in generator:
35.         features_batch = conv_base.predict(inputs_batch) # conv_
base 사용
36.         features[i * batch_size : (i + 1) * batch_size] = feature
s_batch
37.         labels[i * batch_size : (i + 1) * batch_size] = labels_ba
tch
38.         i += 1
39.         if i * batch_size >= sample_count:
40.             break
41.     return features, labels
42.
43. train_features, train_labels = extract_features(train_dir, 2000)
```

```
44.validation_features, validation_labels = extract_features(validat
    ion_dir, 1000)
45.test_features, test_labels = extract_features(test_dir, 1000)
46.train_features = np.reshape(train_features, (2000, 5 * 5 * 2048))
47.validation_features = np.reshape(validation_features, (1000, 5 *
    5 * 2048))
48.test_features = np.reshape(test_features, (1000, 5 * 5 * 2048))
49.from keras import models
50.from keras import layers
51.from keras import optimizers
52.
53.model = models.Sequential()
54.model.add(layers.Dense(256, activation='relu', input_dim=5 * 5 *
    2048))
55.model.add(layers.Dropout(0.5))
56.model.add(layers.Dense(1, activation='sigmoid'))
57.
58.model.compile(optimizer=optimizers.RMSprop(lr=2e-5),
59.               loss='binary_crossentropy',
60.               metrics=['acc'])
61.
62.history = model.fit(train_features, train_labels,
63.                    epochs=20,
64.                    batch_size=20,
65.                    validation_data=(validation_features, validat
    ion_labels))
66.import matplotlib.pyplot as plt
67.
68.acc = history.history['acc']
69.val_acc = history.history['val_acc']
70.loss = history.history['loss']
71.val_loss = history.history['val_loss']
72.
73.epochs = range(len(acc))
74.
75.plt.plot(epochs, acc, 'bo', label='Training acc')
76.plt.plot(epochs, val_acc, 'b', label='Validation acc')
77.plt.title('Training and validation accuracy')
78.plt.legend()
79.
80.plt.figure()
81.
82.plt.plot(epochs, loss, 'bo', label='Training loss')
83.plt.plot(epochs, val_loss, 'b', label='Validation loss')
84.plt.title('Training and validation loss')
85.plt.legend()
86.
87.plt.show()
```

위와 같은 코드로 수정해 주었는데 ResNet50을 사용하고 features 부분을  $4 * 4 * 512$ 에서  $5 * 5 * 2048$ 로 수정해 주었다.