CNN Project Presentation – Transportation Classification

김은비, 김지은, 성현민, 홍성민

Contents

1

Data Overview

Data Overview

Aim: 10개의 운송수단을 선택한 후,
이를 컴퓨터에게 학습시켜 각 이미지 구분할 수 있는 모델 생성


airplane_1.jpg


bike001.jpg

bus (41).jpg


motorcycle052.jpg


train_102.jpg


Helicopter_150.jpg


car_484.jpg


truck_500.jpg


hotairballoon_105.jpg


Ship_01.jpg

2

Methodology

Methodology I

Image Crawling: Bing & Google에서 이미지 크롤링 할 수 있는 코드 생성 후,
각 카테고리별로 최소 500개 씩 이미지 선택

```
[ ]  !pip install icrawler
```

```
[ ]  from icrawler.builtin import BingImageC
```

```
[ ]  bing_crawler = BingImageCrawler(
         downloader_threads=4,
         storage={'root_dir': 'C:/bus'} # va
     )
     # filter
```

```
bing_crawler.crawl(keyword='bus', filte
```

In [1]:

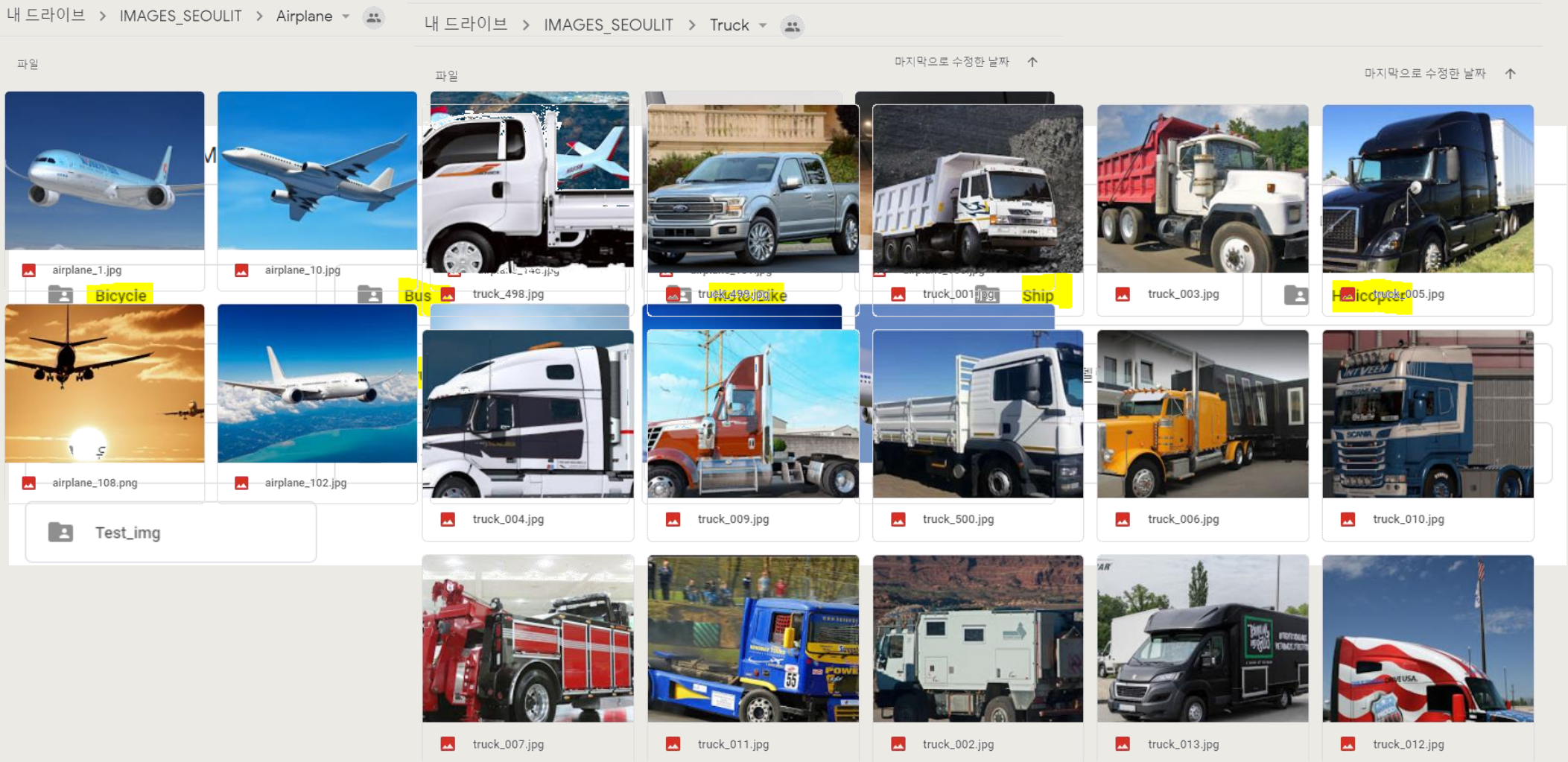
```
from google_images_download import google_images_download

import ssl  # ssl Error 발생 시
ssl._create_default_https_context = ssl._create_unverified_context


def imageCrawling(keyword, dir):
    response = google_images_download.googleimagesdownload()

    arguments = {"keywords":keyword,      # 검색 키워드
                 "limit":1000,             # 크롤링 이미지 수
                 "print_urls":True,        # 이미지 url 출력
                 "no_directory":True,      #
                 'output_directory':dir,
                'chromedriver':'./driver/chromedriver.exe'}  # 크롤링 이미지를 저장할 폴더

    paths = response.download(arguments)
    print(paths)
```

In [2]:

```
imageCrawling('airplane','./download/airplane') # imageCrawling('검색어', '저장경로')
```

Methodology II

Image Selection: 크롤링 된 이미지에서 학습시킬 이미지를 선택



Methodology III

Coding: CNN 적용 모델 & VGG 모델, 총 2가지 모델을 선택 후 코드 작성

```
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(32, 3, 3, border_mode='Same',
    input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.25)) # dropout
    model.add(Convolution2D(64,
    model.add(Activation('relu'
    model.add(Convolution2D(64,
    model.add(MaxPooling2D(pool
    # dropout
    model.add(Flatten())
    model.add(Dense(512))
    model.add(Activation('relu'
    # dropout
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    model.compile(loss='binary_crossentropy',
    optimizer='rmsprop',
    metrics=['accuracy'])
    return model
```

Accuracy:

50% vs. 83%

```
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(64, 3, 3, border_mode='Same',
    input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(Convolution2D(64, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))

#     model.add(Dropout(0.25)) # dropout
    model.add(Convolution2D(128, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(128, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))

    D(256, 3, 3, border_mode='same'))
    relu'))
    D(256, 3, 3, border_mode='same'))
    relu'))
    D(256, 3, 3, border_mode='same'))
    relu'))
    pool_size=(2,2)))

    D(512, 3, 3, border_mode='same'))
    relu'))
    D(512, 3, 3, border_mode='same'))
    relu'))
    model.add(Convolution2D(512, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))

    model.add(Convolution2D(512, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(512, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(512, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    # dropout
    model.add(Flatten())
    model.add(Dense(4096))
    model.add(Activation('relu'))

    # dropout
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    adam = keras.optimizers.Adam(lr=0.00002) #학습률 수정 시킴
    model.compile(loss='categorical_crossentropy',
    optimizer=adam,
    metrics=['accuracy'])
    return model
```

Methodology IV

Alteration: Learning Rate -> Not Much Difference

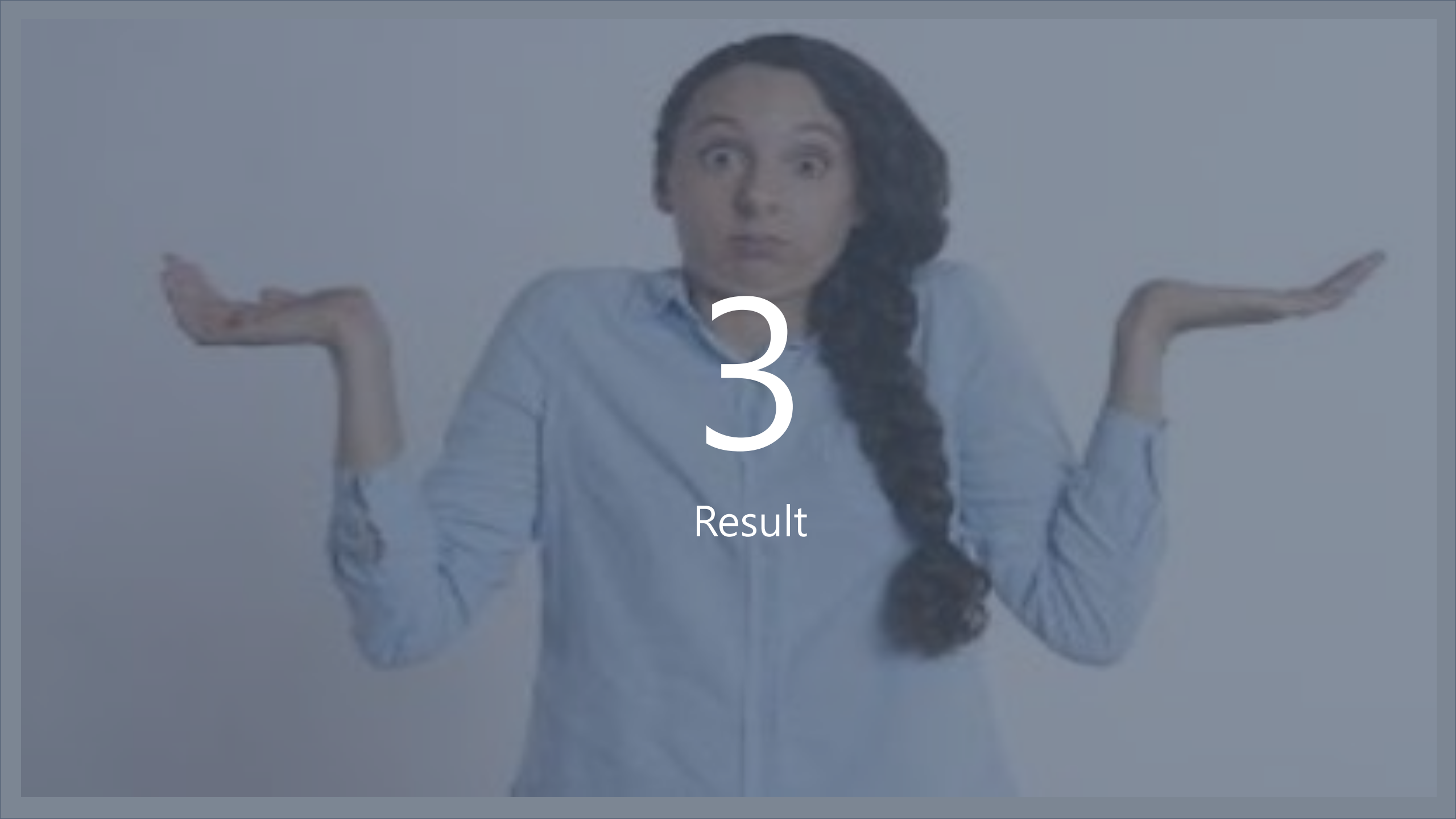
```python
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(64, 3, 3, border_mode='Same',
    input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(Convolution2D(64, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))

#     model.add(Dropout(0.25)) # dropout
    model.add(Convolution2D(128, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(128, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))

    model.add(Convolution2D(256, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(256, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(256, 3, 3
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=

    model.add(Convolution2D(512, 3, 3
    model.add(Activation('relu'))
    model.add(Convolution2D(512, 3, 3
    model.add(Activation('relu'))
    model.add(Convolution2D(512, 3, 3
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=

    model.add(Convolution2D(512, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(512, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(512, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    # dropout
    model.add(Flatten())
    model.add(Dense(4096))
    model.add(Activation('relu'))

    # dropout
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    adam = keras.optimizers.Adam(lr=0.00001)
    model.compile(loss='categorical_crossentropy',
    optimizer=adam,
    metrics=['accuracy'])
    return model
```

```python
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(64, 3, 3, border_mode='Same',
    input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(Convolution2D(64, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))

#     model.add(Dropout(0.25)) # dropout
    model.add(Convolution2D(128, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(128, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))

    model.add(Convolution2D(256, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(256, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(256, 3, 3, border_mode='same'))

    model.add(Convolution2D(512, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(512, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(512, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    # dropout
    model.add(Flatten())
    model.add(Dense(4096))
    model.add(Activation('relu'))

    # dropout
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    adam = keras.optimizers.Adam(lr=0.00002) #학습률 수정 시킴
    model.compile(loss='categorical_crossentropy',
    optimizer=adam,
    metrics=['accuracy'])
    return model
```

```python
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(64, 3, 3, border_mode='Same',
    input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(Convolution2D(64, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))

#     model.add(Dropout(0.25)) # dropout
    model.add(Convolution2D(128, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(128, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))

    model.add(Convolution2D(256, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(256, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(256, 3, 3, border_mode='same'))

    model.add(Convolution2D(512, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(512, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(512, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    # dropout
    model.add(Flatten())
    model.add(Dense(4096))
    model.add(Activation('relu'))

    # dropout
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    adam = keras.optimizers.Adam(lr=0.00003) #학습률 수정 시킴
    model.compile(loss='categorical_crossentropy',
    optimizer=adam,
    metrics=['accuracy'])
    return model
```

Accuracy:

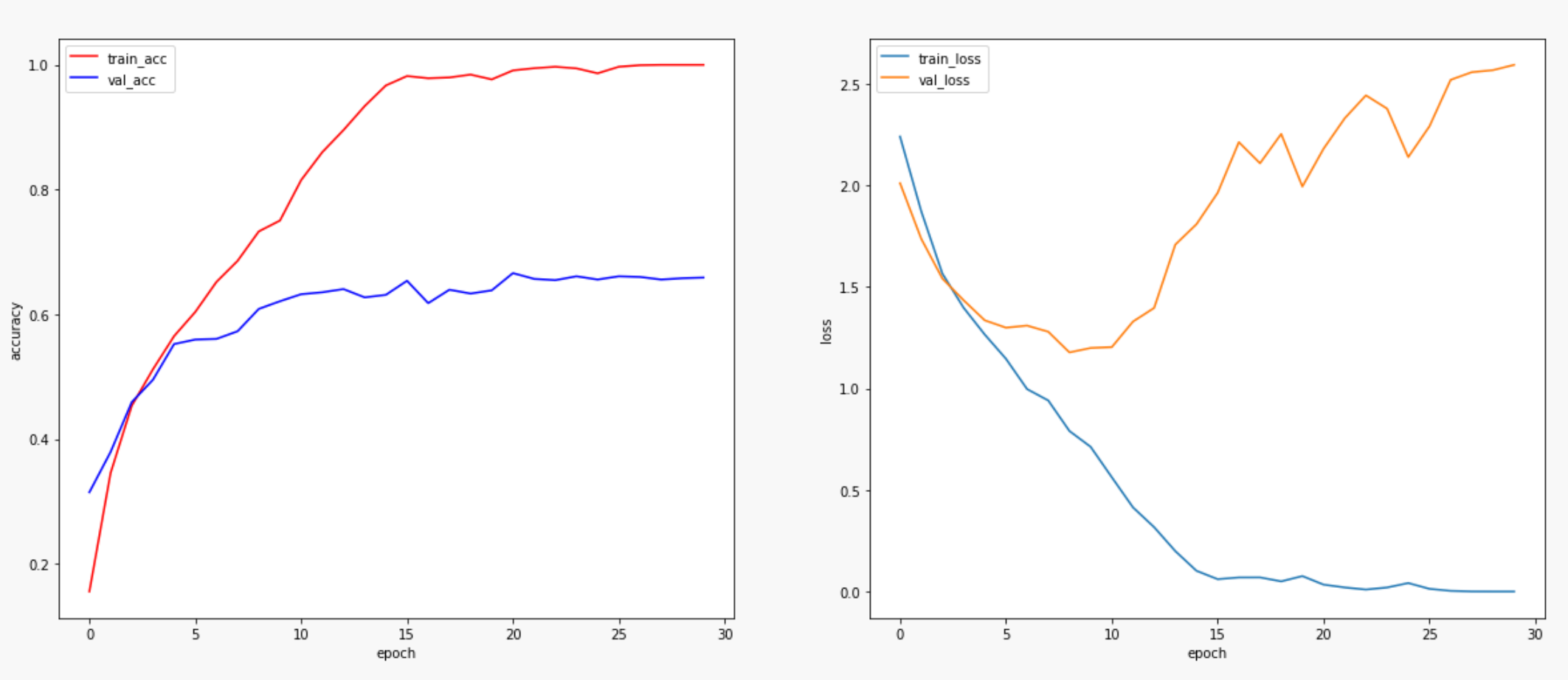
0.8543% vs. 0.8439% vs. 0.8184%



3

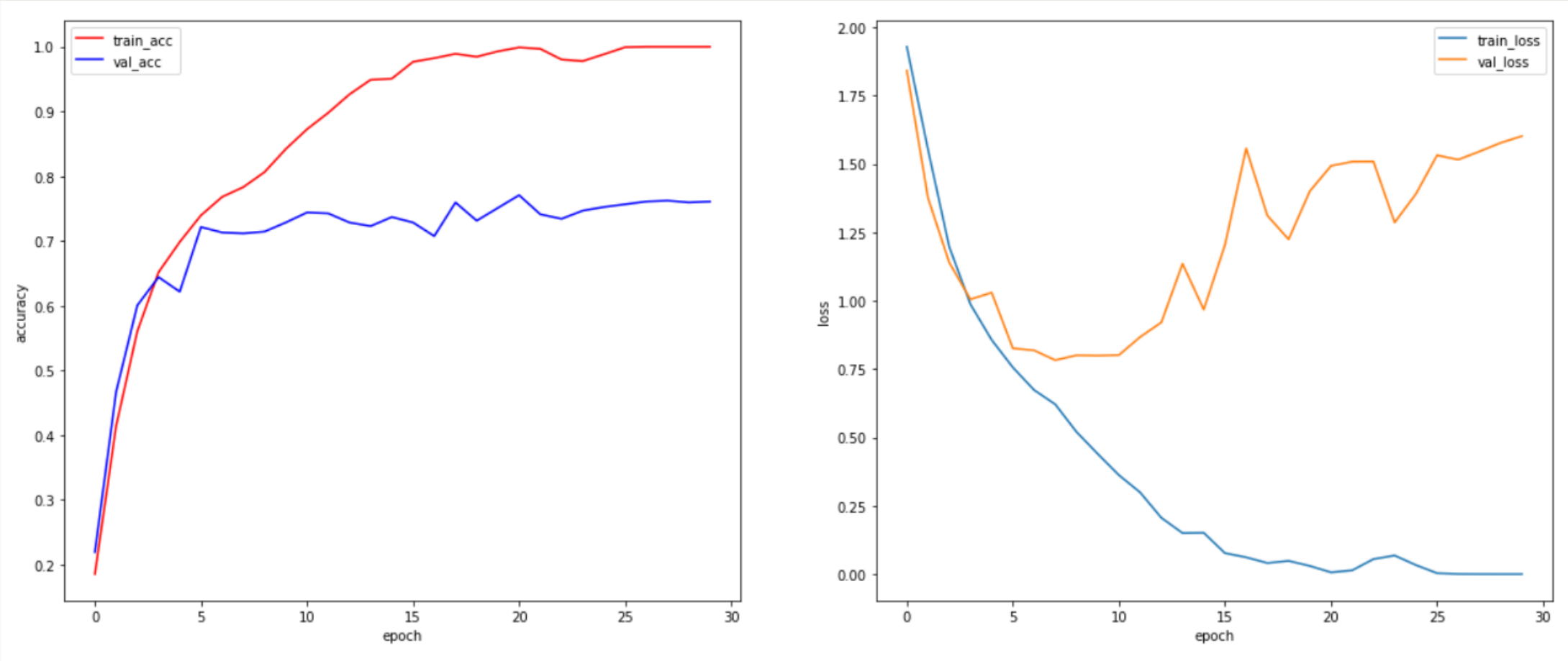
Result

Result



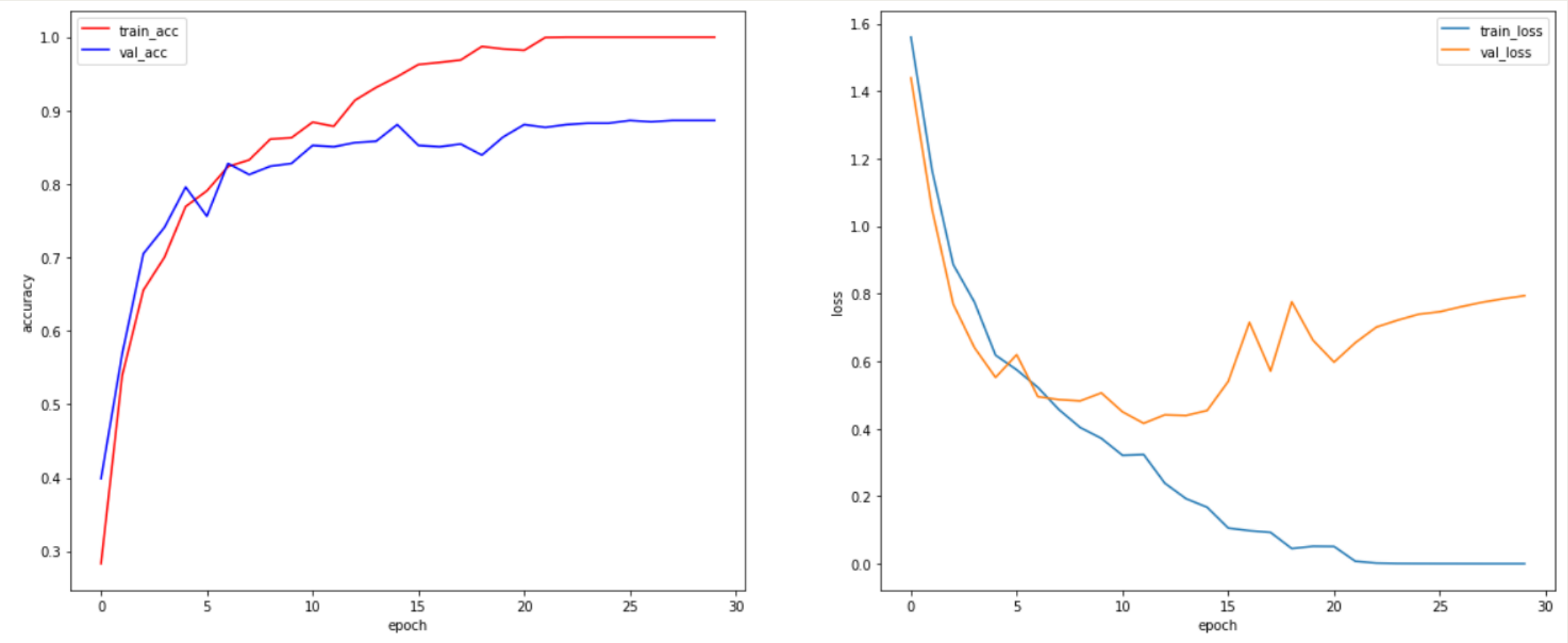
10개의 카테고리: val_accuracy가 67%정도에서 멈춘다.

Result



10개의 카테고리 중(-차/버스/트럭) : val_accuracy가 76% 가 나온다.

Result



10개의 카테고리 중(-차/버스/트럭/비행기/헬리콥터) : val_accuracy가 88% 가 나온다.

4

Summary

Challenges

Issues:

낮은 컴퓨터 사양

배경화면이 없는 순수한 타겟 이미지 획득의 어려움

문제의 원인을 너무 적은 데이터 셋 개수(500개 정도)라고 판단함.