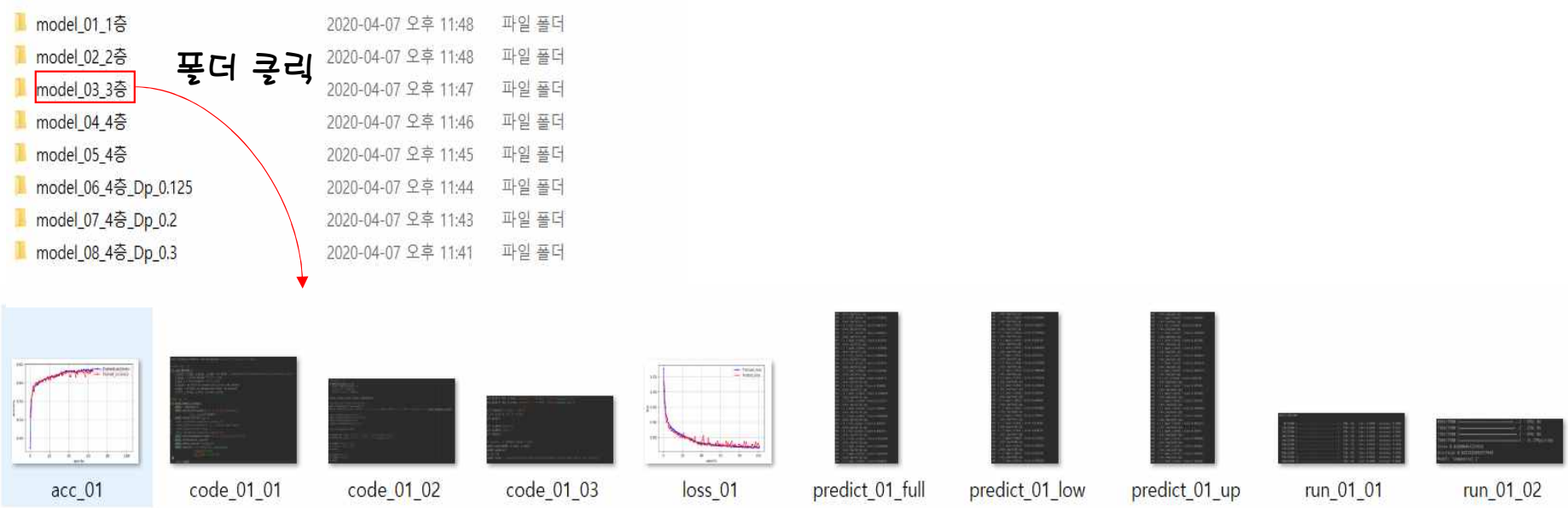
DeepFashion Data Set 상, 하의 분류 모델 분석

ViewCloset 팀
- 오 재 동 -

0. 개요

기존의 학습 모델을 수정해서 **총 8번의 수행**을 해 보았고,
한 번 수행할 때 마다 model_01, model_02, … 의 이름으로 폴더를 생성했다.




summary_01

폴더 내부에는 수행 후 얻은 정확도와 오차에 대해 시각화한 그래프 2개, 모델 생성 및 실행부 코드, 수행 결과를 보여주는 실행창, 생성한 학습망을 요약한 이미지, 그리고 한 번 옷, 상의, 하의에 대해 예측을 수행한 결과를 나타내는 사진들이 들어 있다.

모델을 수정할 때 이전에 받은 멘토님의 피드백을 적용해서 수정해 보았다.
피드백 받은 사항들을 정리해보면 아래와 같다.

1. 처음에는 층 수를 적게 가져간다.
- 학습망을 1층만 추가하고, 마지막에 Global Max Pooling을 적용한다. (Model_01)

2. 마찬가지로 처음에는 필터의 수도 적게 가져간다.
- 1층에는 필터의 개수를 16개 혹은 32개로 설정하고, 이후 신경망을 추가해 줄 때마다 32, 64, 128개 와 같이 점점 필터 수를 증가시킨다.

3. 모델의 마지막에는 Global Max Pooling 기법을 적용해준다.
- 1층: 신경망 1개 + Global Max Pooling, 2층: 신경망 2개 + Global Max Pooling …

4. 최적의 그래프 형태를 찾기 전에는 Dropout을 적용하지 않는다.
- 이상적인 그래프 형태가 나오면, 그 이후에 층마다 Dropout을 사용할 것

5. pool_size (필터 크기)는 통일 해 준다.
- 일관성 유지 목적..? ex) model.add(Convolution2D(16, 3, 3, border_mode='Same')…

6. 이후 Dropout을 적용할 때도 마찬가지로 Dropout 해주는 비율을 일관되게 적용한다.

모델을 수정할 때 이전에 받은 멘토님의 피드백을 적용해서 수정해 보았다.
피드백 받은 사항들을 정리해보면 아래와 같다.

7. 모델의 학습 정도를 살펴보기 위해 그래프를 사용하며, 그래프는 오차와 정확도에 대해 2개를 출력한다.

8. 그래프로 출력했을 때 어느 정도 최적화된 형태가 나오면, Dropout을 층마다 추가해준다.

9. Dropout도 과도하게 설정하지 말고, 0.125, 0.25 정도로 적용한다.

10. 생성한 모델을 컴파일 할 때 loss 함수는 'binary_crossentropy'가 아닌 'categorical_crossentropy'를 사용해야 한다.

11. 마찬가지로 모델 컴파일 할 때 optimizer는 'rmsprop' 보다는 'adam'을 사용하는 것이 좋다.

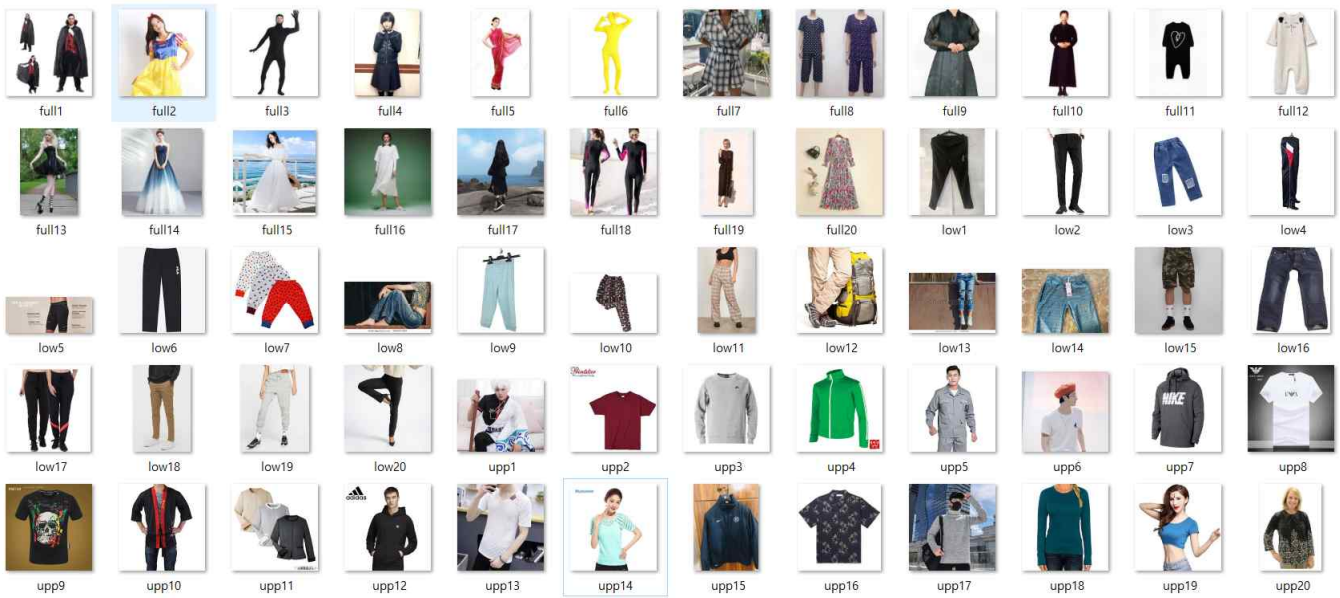
이렇게 피드백을 적용해서 생성한 모델 코드는 다음과 같은데, 잘 이해한 건지 모르겠다. (피드백 부탁드립니다!)

```
model.add(Convolution2D(16, 3, 3, border_mode='same', input_shape=in_shape))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.30))                                  # 층마다 MaxPooling과 Dropout을 적용..  맞나..?
model.add(Convolution2D(32, 3, 3, border_mode='Same'))
model.add(Activation('relu'))                             # 그리고 Flatten() 함수를 적용 안 해주었는데 상관없는지도.. 궁금하다.
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.30))
model.add(Convolution2D(64, 3, 3, border_mode='Same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.30))
model.add(Convolution2D(128, 3, 3, border_mode='Same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.30))
model.add(GlobalMaxPoolin2D(data_format='channels_last'))    # 마지막에 GlobalMaxPooling 적용
model.add(Dropout(0.30))       # GlobalMaxPooling 적용 후에도 Dropout을 추가해 주어야 하나..?
model.add(Dense(nb_classes)
model.add(Activation('softmax')
model.compile(loss='categorical crossentropy', optimizer='adam', metrics=['accuracy'])
```

이 다음부터는 모델을 수행할 때의 결과들을 모델 별로 정리해 둔 것으로,
가장 결과가 좋았던 모델은 4층 신경망에 Dropout 기법을 적용한 model_06,07,08 이다.

하지만 이 모델들을 먼저 보여주지 않고, model_01부터 순차적으로 보여줄 예정이다.

또, 생성한 모델의 결과를 예측할 때 사용한 이미지는 구글에서 검색하여 얻은 총 60장의 이미지로,
이 이미지들은 상의 20장, 하의 20장, 한 벌 옷 20장으로 구성되어 있다.



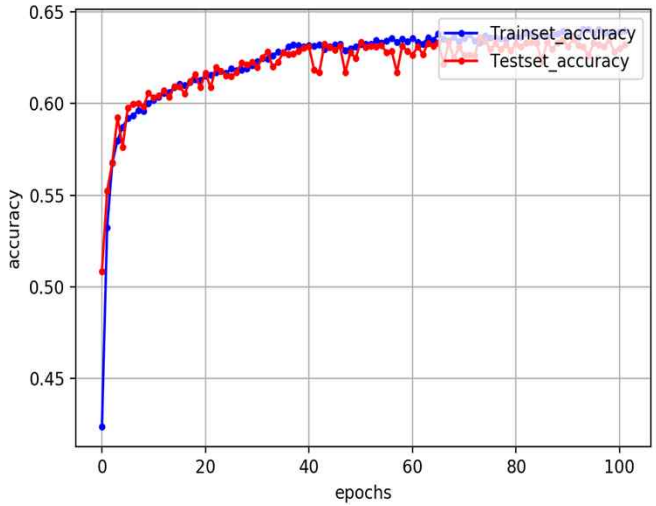
- 테스트 이미지 -

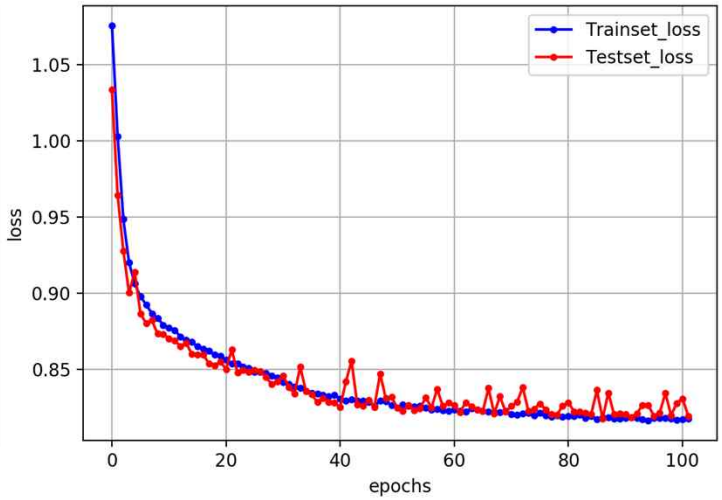
* 예측 결과는 편의상 결과가 좋았던 model_06,07,08에 대해서만 표시하였다.

1. Model_01 - 1층 신경망 + Global Max Pooling

early_stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

모델 정확도: 0.63… 오차: 0.81… 최적 학습 횟수: 102





〈 모델 요약 〉

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 64, 64, 16)	448
activation_1 (Activation)	(None, 64, 64, 16)	0
global_max_pooling2d_1 (Glob	(None, 16)	0
dense_1 (Dense)	(None, 3)	51
activation_2 (Activation)	(None, 3)	0

Total params: 499
Trainable params: 499
Non-trainable params: 0

1. Model_01 - 1층 신경망 + Global Max Pooling

모델 정확도: 0.63… 오차: 0.81… 최적 학습 횟수: 102

```
early_stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

# 데이터 로딩 (1)
def load_dataset():
    x_train, x_test, y_train, y_test = np.load("../DeepFashion/attribute_predict/up_down/fashion.npy")
    x_train = x_train.astype("float") / 256
    x_test = x_test.astype("float") / 256
    y_train = np_utils.to_categorical(y_train, nb_classes)
    y_test = np_utils.to_categorical(y_test, nb_classes)
    return x_train, x_test, y_train, y_test

# 모델 구성 (2)
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(16, 3, 3, border_mode='Same',
                            input_shape=in_shape))
    model.add(Activation('relu'))
#     model.add(MaxPooling2D(pool_size=(2,2)))
#     model.add(Convolution2D(32, 3, 3, border_mode='Same'))
#     model.add(Activation('relu'))
#     model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(GlobalMaxPooling2D(data_format="channels_last"))
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model
```

1. Model_01 - 1층 신경망 + Global Max Pooling

실제 데이터 예측 결과 (한 벌 옷, 상의, 하의 순..)

```
입력: ./test_img/full1.jpg
예측: [ 0 ] full_clothes / Score 0.53228354
입력: ./test_img/full2.jpg
예측: [ 0 ] full_clothes / Score 0.46076772
입력: ./test_img/full3.jpg
예측: [ 0 ] full_clothes / Score 0.44969815
입력: ./test_img/full4.jpg
예측: [ 2 ] upper_clothes / Score 0.8477627
입력: ./test_img/full5.jpg
예측: [ 2 ] upper_clothes / Score 0.4190688
입력: ./test_img/full6.jpg
예측: [ 0 ] full_clothes / Score 0.66084546
입력: ./test_img/full7.jpg
예측: [ 0 ] full_clothes / Score 0.73233336
입력: ./test_img/full8.jpg
예측: [ 1 ] lower_clothes / Score 0.6261609
입력: ./test_img/full9.jpg
예측: [ 2 ] upper_clothes / Score 0.8644175
입력: ./test_img/full10.jpg
예측: [ 0 ] full_clothes / Score 0.4700689
입력: ./test_img/full11.jpg
예측: [ 2 ] upper_clothes / Score 0.51904285
입력: ./test_img/full12.jpg
예측: [ 1 ] lower_clothes / Score 0.5313327
입력: ./test_img/full13.jpg
예측: [ 2 ] upper_clothes / Score 0.360939
입력: ./test_img/full14.jpg
예측: [ 1 ] lower_clothes / Score 0.40560973
입력: ./test_img/full15.jpg
예측: [ 0 ] full_clothes / Score 0.6015779
입력: ./test_img/full16.jpg
예측: [ 0 ] full_clothes / Score 0.4963974
입력: ./test_img/full17.jpg
예측: [ 1 ] lower_clothes / Score 0.51432556
입력: ./test_img/full18.jpg
예측: [ 2 ] upper_clothes / Score 0.87910897
입력: ./test_img/full19.jpg
예측: [ 2 ] upper_clothes / Score 0.46235487
입력: ./test_img/full20.jpg
예측: [ 2 ] upper_clothes / Score 0.4599301
```

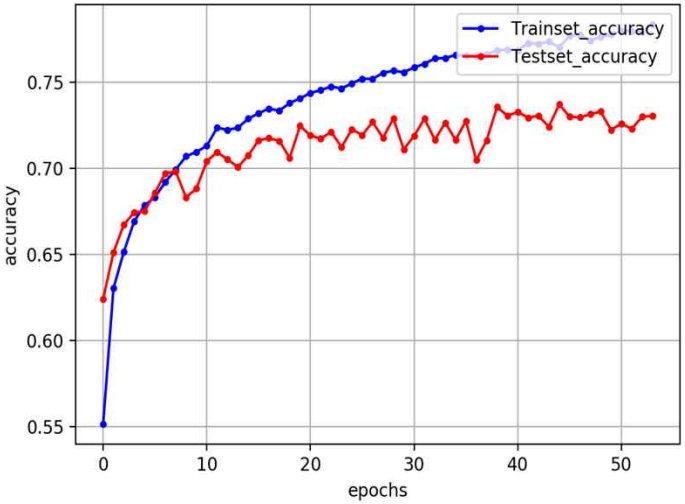
```
입력: ./test_img/low1.jpg
예측: [ 1 ] lower_clothes / Score 0.43125027
입력: ./test_img/low2.jpg
예측: [ 1 ] lower_clothes / Score 0.46070272
입력: ./test_img/low3.jpg
예측: [ 1 ] lower_clothes / Score 0.70918924
입력: ./test_img/low4.jpg
예측: [ 2 ] upper_clothes / Score 0.5067138
입력: ./test_img/low5.jpg
예측: [ 1 ] lower_clothes / Score 0.85042876
입력: ./test_img/low6.jpg
예측: [ 1 ] lower_clothes / Score 0.6973331
입력: ./test_img/low7.jpg
예측: [ 1 ] lower_clothes / Score 0.64129806
입력: ./test_img/low8.jpg
예측: [ 1 ] lower_clothes / Score 0.38062486
입력: ./test_img/low9.jpg
예측: [ 2 ] upper_clothes / Score 0.52512914
입력: ./test_img/low10.jpg
예측: [ 1 ] lower_clothes / Score 0.43648654
입력: ./test_img/low11.jpg
예측: [ 1 ] lower_clothes / Score 0.5930335
입력: ./test_img/low12.jpg
예측: [ 2 ] upper_clothes / Score 0.6501039
입력: ./test_img/low13.jpg
예측: [ 1 ] lower_clothes / Score 0.60760045
입력: ./test_img/low14.jpg
예측: [ 2 ] upper_clothes / Score 0.7249432
입력: ./test_img/low15.jpg
예측: [ 1 ] lower_clothes / Score 0.6147491
입력: ./test_img/low16.jpg
예측: [ 1 ] lower_clothes / Score 0.8297374
입력: ./test_img/low17.jpg
예측: [ 2 ] upper_clothes / Score 0.57418114
입력: ./test_img/low18.jpg
예측: [ 1 ] lower_clothes / Score 0.78582627
입력: ./test_img/low19.jpg
예측: [ 1 ] lower_clothes / Score 0.6617378
입력: ./test_img/low20.jpg
예측: [ 1 ] lower_clothes / Score 0.49955425
```

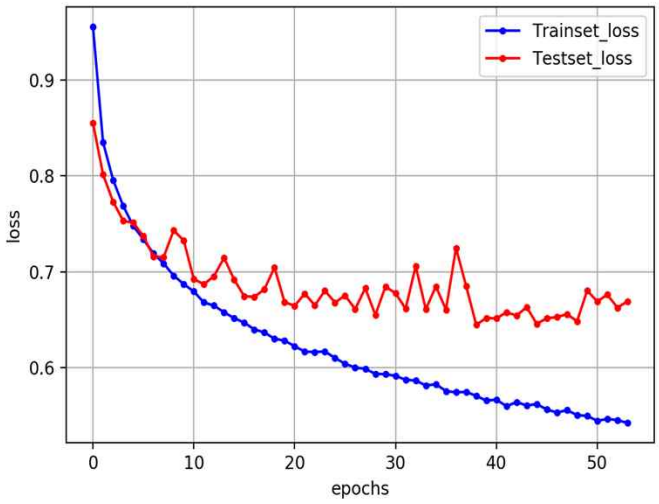
```
입력: ./test_img/upp1.jpg
예측: [ 2 ] upper_clothes / Score 0.6660368
입력: ./test_img/upp2.jpg
예측: [ 0 ] full_clothes / Score 0.4730402
입력: ./test_img/upp3.jpg
예측: [ 1 ] lower_clothes / Score 0.49388155
입력: ./test_img/upp4.jpg
예측: [ 1 ] lower_clothes / Score 0.4871926
입력: ./test_img/upp5.jpg
예측: [ 2 ] upper_clothes / Score 0.857736
입력: ./test_img/upp6.jpg
예측: [ 1 ] lower_clothes / Score 0.45265412
입력: ./test_img/upp7.jpg
예측: [ 1 ] lower_clothes / Score 0.49194235
입력: ./test_img/upp8.jpg
예측: [ 2 ] upper_clothes / Score 0.46502277
입력: ./test_img/upp9.jpg
예측: [ 0 ] full_clothes / Score 0.8687489
입력: ./test_img/upp10.jpg
예측: [ 2 ] upper_clothes / Score 0.9723681
입력: ./test_img/upp11.jpg
예측: [ 2 ] upper_clothes / Score 0.4214058
입력: ./test_img/upp12.jpg
예측: [ 2 ] upper_clothes / Score 0.9123665
입력: ./test_img/upp13.jpg
예측: [ 2 ] upper_clothes / Score 0.42655507
입력: ./test_img/upp14.jpg
예측: [ 2 ] upper_clothes / Score 0.8764256
입력: ./test_img/upp15.jpg
예측: [ 2 ] upper_clothes / Score 0.8051753
입력: ./test_img/upp16.jpg
예측: [ 2 ] upper_clothes / Score 0.565667
입력: ./test_img/upp17.jpg
예측: [ 2 ] upper_clothes / Score 0.5990648
입력: ./test_img/upp18.jpg
예측: [ 2 ] upper_clothes / Score 0.8050843
입력: ./test_img/upp19.jpg
예측: [ 2 ] upper_clothes / Score 0.9430784
입력: ./test_img/upp20.jpg
예측: [ 1 ] lower_clothes / Score 0.4816266
```

2. Model_02 - 2층 신경망 + Max Pooling + Global Max Pooling

early_stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

모델 정확도: 0.73… 오차: 0.66… 최적 학습 횟수: 54





〈 모델 요약 〉

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 64, 16)        448
_________________________________________________________________
activation_1 (Activation)    (None, 64, 64, 16)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 32)        4640
_________________________________________________________________
activation_2 (Activation)    (None, 32, 32, 32)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
global_max_pooling2d_1 (Glob (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 99
_________________________________________________________________
activation_3 (Activation)    (None, 3)                 0
=================================================================
Total params: 5,187
Trainable params: 5,187
Non-trainable params: 0
_________________________________________________________________
```

2. Model_02 - 2층 신경망 + Max Pooling + Global Max Pooling

모델 정확도: 0.73… 오차: 0.66… 최적 학습 횟수: 54

```
early_stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

# 데이터 로딩 (1)
def load_dataset():
    x_train, x_test, y_train, y_test = np.load("../DeepFashion/attribute_predict/up_down/fashion.npy")
    x_train = x_train.astype("float") / 256
    x_test = x_test.astype("float") / 256
    y_train = np_utils.to_categorical(y_train, nb_classes)
    y_test = np_utils.to_categorical(y_test, nb_classes)
    return x_train, x_test, y_train, y_test

# 모델 구성 (2)
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(16, 3, 3, border_mode='Same',
                            input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Convolution2D(32, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(GlobalMaxPooling2D(data_format="channels_last"))
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model
```

2. Model_02 - 2층 신경망 + Max Pooling + Global Max Pooling

실제 데이터 예측 결과 (한 벌 옷, 상의, 하의 순..)

```
입력: ./test_img/full1.jpg
예측: [ 2 ] upper_clothes / Score 0.4946722
입력: ./test_img/full2.jpg
예측: [ 2 ] upper_clothes / Score 0.5524478
입력: ./test_img/full3.jpg
예측: [ 1 ] lower_clothes / Score 0.8205817
입력: ./test_img/full4.jpg
예측: [ 2 ] upper_clothes / Score 0.98131967
입력: ./test_img/full5.jpg
예측: [ 2 ] upper_clothes / Score 0.4882938
입력: ./test_img/full6.jpg
예측: [ 0 ] full_clothes / Score 0.3790063
입력: ./test_img/full7.jpg
예측: [ 0 ] full_clothes / Score 0.5389483
입력: ./test_img/full8.jpg
예측: [ 0 ] full_clothes / Score 0.60262746
입력: ./test_img/full9.jpg
예측: [ 2 ] upper_clothes / Score 0.95714957
입력: ./test_img/full10.jpg
예측: [ 0 ] full_clothes / Score 0.9933483
입력: ./test_img/full11.jpg
예측: [ 0 ] full_clothes / Score 0.3977783
입력: ./test_img/full12.jpg
예측: [ 0 ] full_clothes / Score 0.5548896
입력: ./test_img/full13.jpg
예측: [ 1 ] lower_clothes / Score 0.5441869
입력: ./test_img/full14.jpg
예측: [ 0 ] full_clothes / Score 0.78901905
입력: ./test_img/full15.jpg
예측: [ 1 ] lower_clothes / Score 0.97305727
입력: ./test_img/full16.jpg
예측: [ 0 ] full_clothes / Score 0.95030284
입력: ./test_img/full17.jpg
예측: [ 0 ] full_clothes / Score 0.49465823
입력: ./test_img/full18.jpg
예측: [ 2 ] upper_clothes / Score 0.5211045
입력: ./test_img/full19.jpg
예측: [ 0 ] full_clothes / Score 0.58330524
입력: ./test_img/full20.jpg
예측: [ 0 ] full_clothes / Score 0.7738935
```

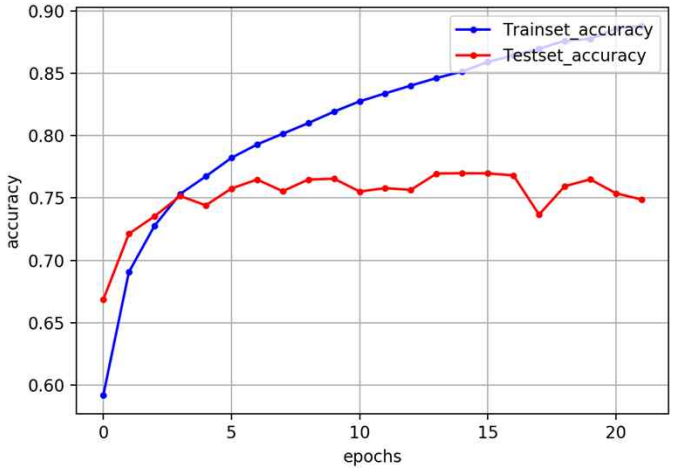
```
입력: ./test_img/low1.jpg
예측: [ 2 ] upper_clothes / Score 0.60363686
입력: ./test_img/low2.jpg
예측: [ 1 ] lower_clothes / Score 0.9598263
입력: ./test_img/low3.jpg
예측: [ 1 ] lower_clothes / Score 0.667447
입력: ./test_img/low4.jpg
예측: [ 2 ] upper_clothes / Score 0.85787594
입력: ./test_img/low5.jpg
예측: [ 1 ] lower_clothes / Score 0.61909425
입력: ./test_img/low6.jpg
예측: [ 1 ] lower_clothes / Score 0.9681445
입력: ./test_img/low7.jpg
예측: [ 1 ] lower_clothes / Score 0.9641728
입력: ./test_img/low8.jpg
예측: [ 1 ] lower_clothes / Score 0.7146975
입력: ./test_img/low9.jpg
예측: [ 2 ] upper_clothes / Score 0.8245531
입력: ./test_img/low10.jpg
예측: [ 1 ] lower_clothes / Score 0.57491237
입력: ./test_img/low11.jpg
예측: [ 0 ] full_clothes / Score 0.5843786
입력: ./test_img/low12.jpg
예측: [ 0 ] full_clothes / Score 0.66595787
입력: ./test_img/low13.jpg
예측: [ 2 ] upper_clothes / Score 0.57134974
입력: ./test_img/low14.jpg
예측: [ 2 ] upper_clothes / Score 0.8322225
입력: ./test_img/low15.jpg
예측: [ 1 ] lower_clothes / Score 0.9714234
입력: ./test_img/low16.jpg
예측: [ 1 ] lower_clothes / Score 0.94803673
입력: ./test_img/low17.jpg
예측: [ 1 ] lower_clothes / Score 0.93392277
입력: ./test_img/low18.jpg
예측: [ 1 ] lower_clothes / Score 0.9715703
입력: ./test_img/low19.jpg
예측: [ 0 ] full_clothes / Score 0.46695712
입력: ./test_img/low20.jpg
예측: [ 1 ] lower_clothes / Score 0.7983542
```

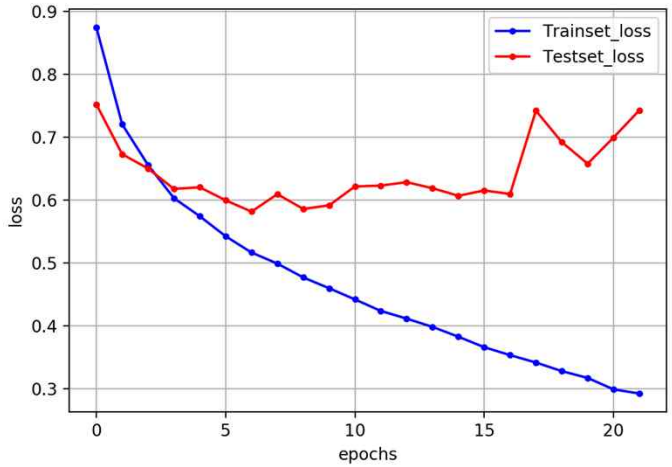
```
입력: ./test_img/upp1.jpg
예측: [ 2 ] upper_clothes / Score 0.7344687
입력: ./test_img/upp2.jpg
예측: [ 2 ] upper_clothes / Score 0.69566256
입력: ./test_img/upp3.jpg
예측: [ 2 ] upper_clothes / Score 0.56172276
입력: ./test_img/upp4.jpg
예측: [ 2 ] upper_clothes / Score 0.9908125
입력: ./test_img/upp5.jpg
예측: [ 2 ] upper_clothes / Score 0.95953244
입력: ./test_img/upp6.jpg
예측: [ 2 ] upper_clothes / Score 0.42299548
입력: ./test_img/upp7.jpg
예측: [ 2 ] upper_clothes / Score 0.9043825
입력: ./test_img/upp8.jpg
예측: [ 0 ] full_clothes / Score 0.81747836
입력: ./test_img/upp9.jpg
예측: [ 1 ] lower_clothes / Score 0.52488196
입력: ./test_img/upp10.jpg
예측: [ 2 ] upper_clothes / Score 0.9973347
입력: ./test_img/upp11.jpg
예측: [ 2 ] upper_clothes / Score 0.8656794
입력: ./test_img/upp12.jpg
예측: [ 2 ] upper_clothes / Score 0.9969118
입력: ./test_img/upp13.jpg
예측: [ 2 ] upper_clothes / Score 0.7229989
입력: ./test_img/upp14.jpg
예측: [ 2 ] upper_clothes / Score 0.8285236
입력: ./test_img/upp15.jpg
예측: [ 2 ] upper_clothes / Score 0.6676505
입력: ./test_img/upp16.jpg
예측: [ 2 ] upper_clothes / Score 0.6219814
입력: ./test_img/upp17.jpg
예측: [ 2 ] upper_clothes / Score 0.7705793
입력: ./test_img/upp18.jpg
예측: [ 2 ] upper_clothes / Score 0.969403
입력: ./test_img/upp19.jpg
예측: [ 2 ] upper_clothes / Score 0.9853388
입력: ./test_img/upp20.jpg
예측: [ 1 ] lower_clothes / Score 0.6344588
```

3. Model_03 - 3층 신경망 + Max Pooling + Global Max Pooling

early_stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

모델 정확도: 0.74… 오차: 0.74… 최적 학습 횟수: 22





〈 모델 요약 〉

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 64, 16)        448
_________________________________________________________________
activation_1 (Activation)    (None, 64, 64, 16)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 32)        4640
_________________________________________________________________
activation_2 (Activation)    (None, 32, 32, 32)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
activation_3 (Activation)    (None, 16, 16, 64)        0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
global_max_pooling2d_1 (Glob (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 195
_________________________________________________________________
activation_4 (Activation)    (None, 3)                 0
=================================================================
Total params: 23,779
Trainable params: 23,779
Non-trainable params: 0
_________________________________________________________________
```

3. Model_03 - 3층 신경망 + Max Pooling + Global Max Pooling

모델 정확도: 0.74… 오차: 0.74… 최적 학습 횟수: 22

```
early_stopping_callback = EarlyStopping(monitor='val_loss',patience=15)

# 데이터 로딩 (1)
def load_dataset():
    x_train, x_test, y_train, y_test = np.load("../DeepFashion/attribute_predict/up_down/fashion.npy")
    x_train = x_train.astype("float") / 256
    x_test = x_test.astype("float") / 256
    y_train = np_utils.to_categorical(y_train, nb_classes)
    y_test = np_utils.to_categorical(y_test, nb_classes)
    return x_train, x_test, y_train, y_test

# 모델 구성 (2)
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(16, 3, 3, border_mode='Same',
                            input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Convolution2D(32, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Convolution2D(64, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(GlobalMaxPooling2D(data_format="channels_last"))
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model
```

3. Model_03 - 2층 신경망 + Max Pooling + Global Max Pooling

실제 데이터 예측 결과 (한 벌 옷, 상의, 하의 순..)

```
입력: ./test_img/full1.jpg
예측: [ 0 ] full_clothes / Score 0.647424
입력: ./test_img/full2.jpg
예측: [ 0 ] full_clothes / Score 0.9062372
입력: ./test_img/full3.jpg
예측: [ 1 ] lower_clothes / Score 0.99076486
입력: ./test_img/full4.jpg
예측: [ 0 ] full_clothes / Score 0.6197109
입력: ./test_img/full5.jpg
예측: [ 1 ] lower_clothes / Score 0.47405833
입력: ./test_img/full6.jpg
예측: [ 2 ] upper_clothes / Score 0.8136901
입력: ./test_img/full7.jpg
예측: [ 0 ] full_clothes / Score 0.86153615
입력: ./test_img/full8.jpg
예측: [ 0 ] full_clothes / Score 0.95914805
입력: ./test_img/full9.jpg
예측: [ 2 ] upper_clothes / Score 0.8052461
입력: ./test_img/full10.jpg
예측: [ 0 ] full_clothes / Score 0.6740558
입력: ./test_img/full11.jpg
예측: [ 1 ] lower_clothes / Score 0.44437334
입력: ./test_img/full12.jpg
예측: [ 2 ] upper_clothes / Score 0.5497249
입력: ./test_img/full13.jpg
예측: [ 0 ] full_clothes / Score 0.6314046
입력: ./test_img/full14.jpg
예측: [ 0 ] full_clothes / Score 0.8765461
입력: ./test_img/full15.jpg
예측: [ 1 ] lower_clothes / Score 0.89521486
입력: ./test_img/full16.jpg
예측: [ 0 ] full_clothes / Score 0.87149924
입력: ./test_img/full17.jpg
예측: [ 1 ] lower_clothes / Score 0.82572085
입력: ./test_img/full18.jpg
예측: [ 0 ] full_clothes / Score 0.75271547
입력: ./test_img/full19.jpg
예측: [ 0 ] full_clothes / Score 0.9214165
입력: ./test_img/full20.jpg
예측: [ 2 ] upper_clothes / Score 0.66165584
```

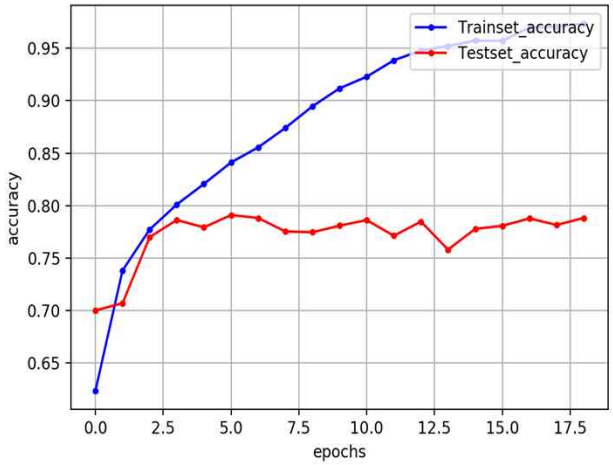
```
입력: ./test_img/low1.jpg
예측: [ 2 ] upper_clothes / Score 0.6630996
입력: ./test_img/low2.jpg
예측: [ 1 ] lower_clothes / Score 0.9957718
입력: ./test_img/low3.jpg
예측: [ 1 ] lower_clothes / Score 0.9792739
입력: ./test_img/low4.jpg
예측: [ 1 ] lower_clothes / Score 0.5626568
입력: ./test_img/low5.jpg
예측: [ 0 ] full_clothes / Score 0.83025604
입력: ./test_img/low6.jpg
예측: [ 1 ] lower_clothes / Score 0.9979735
입력: ./test_img/low7.jpg
예측: [ 1 ] lower_clothes / Score 0.9913829
입력: ./test_img/low8.jpg
예측: [ 0 ] full_clothes / Score 0.6899881
입력: ./test_img/low9.jpg
예측: [ 1 ] lower_clothes / Score 0.7646284
입력: ./test_img/low10.jpg
예측: [ 0 ] full_clothes / Score 0.804222
입력: ./test_img/low11.jpg
예측: [ 1 ] lower_clothes / Score 0.998872
입력: ./test_img/low12.jpg
예측: [ 0 ] full_clothes / Score 0.9471226
입력: ./test_img/low13.jpg
예측: [ 1 ] lower_clothes / Score 0.7793892
입력: ./test_img/low14.jpg
예측: [ 2 ] upper_clothes / Score 0.8555949
입력: ./test_img/low15.jpg
예측: [ 1 ] lower_clothes / Score 0.99870455
입력: ./test_img/low16.jpg
예측: [ 1 ] lower_clothes / Score 0.9774811
입력: ./test_img/low17.jpg
예측: [ 1 ] lower_clothes / Score 0.9988771
입력: ./test_img/low18.jpg
예측: [ 1 ] lower_clothes / Score 0.9995328
입력: ./test_img/low19.jpg
예측: [ 1 ] lower_clothes / Score 0.9970175
입력: ./test_img/low20.jpg
예측: [ 1 ] lower_clothes / Score 0.93193096
```

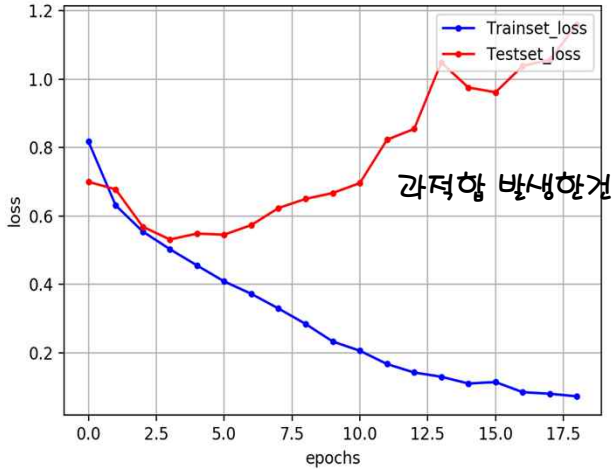
```
입력: ./test_img/upp1.jpg
예측: [ 1 ] lower_clothes / Score 0.471712
입력: ./test_img/upp2.jpg
예측: [ 2 ] upper_clothes / Score 0.9860802
입력: ./test_img/upp3.jpg
예측: [ 2 ] upper_clothes / Score 0.867757
입력: ./test_img/upp4.jpg
예측: [ 2 ] upper_clothes / Score 0.9827116
입력: ./test_img/upp5.jpg
예측: [ 2 ] upper_clothes / Score 0.7759768
입력: ./test_img/upp6.jpg
예측: [ 0 ] full_clothes / Score 0.51530665
입력: ./test_img/upp7.jpg
예측: [ 2 ] upper_clothes / Score 0.77606845
입력: ./test_img/upp8.jpg
예측: [ 1 ] lower_clothes / Score 0.43039277
입력: ./test_img/upp9.jpg
예측: [ 1 ] lower_clothes / Score 0.9005415
입력: ./test_img/upp10.jpg
예측: [ 2 ] upper_clothes / Score 0.9838356
입력: ./test_img/upp11.jpg
예측: [ 2 ] upper_clothes / Score 0.9991781
입력: ./test_img/upp12.jpg
예측: [ 2 ] upper_clothes / Score 0.9962081
입력: ./test_img/upp13.jpg
예측: [ 1 ] lower_clothes / Score 0.561097
입력: ./test_img/upp14.jpg
예측: [ 2 ] upper_clothes / Score 0.9880199
입력: ./test_img/upp15.jpg
예측: [ 2 ] upper_clothes / Score 0.70302856
입력: ./test_img/upp16.jpg
예측: [ 2 ] upper_clothes / Score 0.9871722
입력: ./test_img/upp17.jpg
예측: [ 2 ] upper_clothes / Score 0.62738407
입력: ./test_img/upp18.jpg
예측: [ 2 ] upper_clothes / Score 0.9930062
입력: ./test_img/upp19.jpg
예측: [ 1 ] lower_clothes / Score 0.75510514
입력: ./test_img/upp20.jpg
예측: [ 2 ] upper_clothes / Score 0.93511814
```

4. Model_04 - 4층 신경망 + Max Pooling + Global Max Pooling

early_Stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

모델 정확도: 0.78… 오차: 1.15… 최적 학습 횟수: 19





과적합 발생한건가..?

〈 모델 요약 〉

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 64, 16)        448
_________________________________________________________________
activation_1 (Activation)    (None, 64, 64, 16)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 32)        4640
_________________________________________________________________
activation_2 (Activation)    (None, 32, 32, 32)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
activation_3 (Activation)    (None, 16, 16, 64)        0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 8, 128)         73856
_________________________________________________________________
activation_4 (Activation)    (None, 8, 8, 128)         0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 4, 4, 128)         0
_________________________________________________________________
global_max_pooling2d_1 (Glob (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 387
_________________________________________________________________
activation_5 (Activation)    (None, 3)                 0
=================================================================
Total params: 97,827
Trainable params: 97,827
Non-trainable params: 0
_________________________________________________________________
```

4. Model_04 - 4층 신경망 + Max Pooling + Global Max Pooling

모델 정확도: 0.78… 오차: 1.15… 최적 학습 횟수: 19

```
early_stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

# 데이터 로딩 (1)
def load_dataset():
    x_train, x_test, y_train, y_test = np.load("../DeepFashion/attribute_predict/up_down/fashion.npy")
    x_train = x_train.astype("float") / 256
    x_test = x_test.astype("float") / 256
    y_train = np_utils.to_categorical(y_train, nb_classes)
    y_test = np_utils.to_categorical(y_test, nb_classes)
    return x_train, x_test, y_train, y_test

# 모델 구성 (2)
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(16, 3, 3, border_mode='Same',
                            input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Convolution2D(32, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Convolution2D(64, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Convolution2D(128, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(GlobalMaxPooling2D(data_format="channels_last"))
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model
```

4. Model_04 - 4층 신경망 + Max Pooling + Global Max Pooling

실제 데이터 예측 결과 (한 벌 옷, 상의, 하의 순..)

```
입력: ./test_img/full1.jpg
예측: [ 2 ] upper_clothes / Score 0.9690766
입력: ./test_img/full2.jpg
예측: [ 2 ] upper_clothes / Score 0.5167343
입력: ./test_img/full3.jpg
예측: [ 1 ] lower_clothes / Score 0.9999999
입력: ./test_img/full4.jpg
예측: [ 0 ] full_clothes / Score 1.0
입력: ./test_img/full5.jpg
예측: [ 1 ] lower_clothes / Score 0.79746723
입력: ./test_img/full6.jpg
예측: [ 0 ] full_clothes / Score 0.9927448
입력: ./test_img/full7.jpg
예측: [ 1 ] lower_clothes / Score 0.64977396
입력: ./test_img/full8.jpg
예측: [ 0 ] full_clothes / Score 0.99999547
입력: ./test_img/full9.jpg
예측: [ 2 ] upper_clothes / Score 0.99998176
입력: ./test_img/full10.jpg
예측: [ 0 ] full_clothes / Score 0.99984145
입력: ./test_img/full11.jpg
예측: [ 2 ] upper_clothes / Score 0.62761337
입력: ./test_img/full12.jpg
예측: [ 2 ] upper_clothes / Score 0.9773557
입력: ./test_img/full13.jpg
예측: [ 0 ] full_clothes / Score 0.93634164
입력: ./test_img/full14.jpg
예측: [ 0 ] full_clothes / Score 0.9999678
입력: ./test_img/full15.jpg
예측: [ 1 ] lower_clothes / Score 0.93082565
입력: ./test_img/full16.jpg
예측: [ 0 ] full_clothes / Score 0.97702867
입력: ./test_img/full17.jpg
예측: [ 1 ] lower_clothes / Score 0.59254545
입력: ./test_img/full18.jpg
예측: [ 0 ] full_clothes / Score 0.9998041
입력: ./test_img/full19.jpg
예측: [ 0 ] full_clothes / Score 0.9991605
입력: ./test_img/full20.jpg
예측: [ 0 ] full_clothes / Score 0.8920268
```

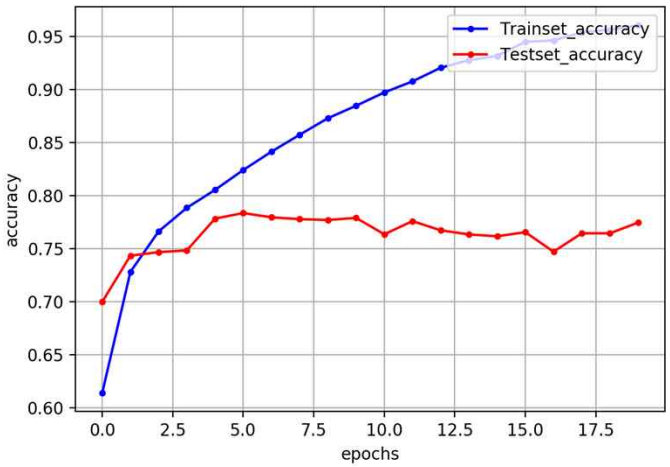
```
입력: ./test_img/low1.jpg
예측: [ 1 ] lower_clothes / Score 0.9666569
입력: ./test_img/low2.jpg
예측: [ 1 ] lower_clothes / Score 1.0
입력: ./test_img/low3.jpg
예측: [ 1 ] lower_clothes / Score 1.0
입력: ./test_img/low4.jpg
예측: [ 2 ] upper_clothes / Score 0.9983871
입력: ./test_img/low5.jpg
예측: [ 2 ] upper_clothes / Score 0.7974402
입력: ./test_img/low6.jpg
예측: [ 1 ] lower_clothes / Score 1.0
입력: ./test_img/low7.jpg
예측: [ 1 ] lower_clothes / Score 0.9973073
입력: ./test_img/low8.jpg
예측: [ 2 ] upper_clothes / Score 0.9997954
입력: ./test_img/low9.jpg
예측: [ 2 ] upper_clothes / Score 0.75572294
입력: ./test_img/low10.jpg
예측: [ 0 ] full_clothes / Score 0.8455669
입력: ./test_img/low11.jpg
예측: [ 1 ] lower_clothes / Score 0.9996605
입력: ./test_img/low12.jpg
예측: [ 2 ] upper_clothes / Score 0.91227764
입력: ./test_img/low13.jpg
예측: [ 2 ] upper_clothes / Score 0.7372328
입력: ./test_img/low14.jpg
예측: [ 0 ] full_clothes / Score 0.9679088
입력: ./test_img/low15.jpg
예측: [ 1 ] lower_clothes / Score 1.0
입력: ./test_img/low16.jpg
예측: [ 1 ] lower_clothes / Score 1.0
입력: ./test_img/low17.jpg
예측: [ 1 ] lower_clothes / Score 0.9999752
입력: ./test_img/low18.jpg
예측: [ 1 ] lower_clothes / Score 0.999998
입력: ./test_img/low19.jpg
예측: [ 1 ] lower_clothes / Score 0.9970059
입력: ./test_img/low20.jpg
예측: [ 1 ] lower_clothes / Score 0.9589567
```

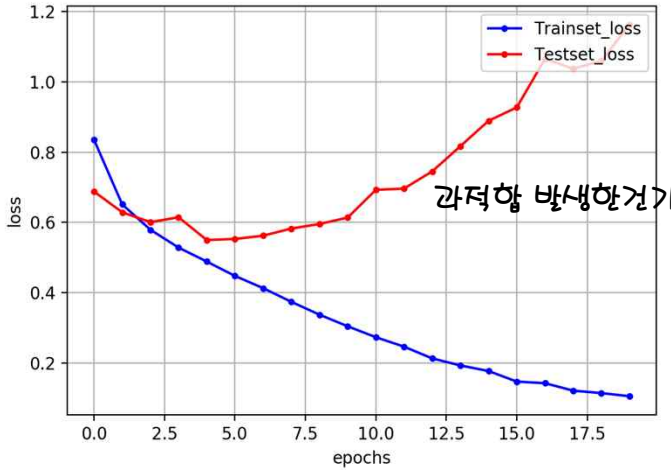
```
입력: ./test_img/upp1.jpg
예측: [ 1 ] lower_clothes / Score 0.564375
입력: ./test_img/upp2.jpg
예측: [ 2 ] upper_clothes / Score 0.5727372
입력: ./test_img/upp3.jpg
예측: [ 2 ] upper_clothes / Score 0.99999774
입력: ./test_img/upp4.jpg
예측: [ 2 ] upper_clothes / Score 0.9999999
입력: ./test_img/upp5.jpg
예측: [ 2 ] upper_clothes / Score 0.9999356
입력: ./test_img/upp6.jpg
예측: [ 2 ] upper_clothes / Score 0.9998493
입력: ./test_img/upp7.jpg
예측: [ 2 ] upper_clothes / Score 0.9871317
입력: ./test_img/upp8.jpg
예측: [ 0 ] full_clothes / Score 0.99996614
입력: ./test_img/upp9.jpg
예측: [ 1 ] lower_clothes / Score 0.9999083
입력: ./test_img/upp10.jpg
예측: [ 2 ] upper_clothes / Score 0.9999988
입력: ./test_img/upp11.jpg
예측: [ 2 ] upper_clothes / Score 0.99998486
입력: ./test_img/upp12.jpg
예측: [ 2 ] upper_clothes / Score 0.99999666
입력: ./test_img/upp13.jpg
예측: [ 2 ] upper_clothes / Score 0.9437882
입력: ./test_img/upp14.jpg
예측: [ 2 ] upper_clothes / Score 1.0
입력: ./test_img/upp15.jpg
예측: [ 2 ] upper_clothes / Score 0.58446383
입력: ./test_img/upp16.jpg
예측: [ 2 ] upper_clothes / Score 0.9887194
입력: ./test_img/upp17.jpg
예측: [ 2 ] upper_clothes / Score 0.9999695
입력: ./test_img/upp18.jpg
예측: [ 2 ] upper_clothes / Score 0.9999989
입력: ./test_img/upp19.jpg
예측: [ 1 ] lower_clothes / Score 0.98938227
입력: ./test_img/upp20.jpg
예측: [ 2 ] upper_clothes / Score 0.99999
```

5. Model_05 - 4층 신경망 + Max Pooling + Global Max Pooling

early_Stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

모델 정확도: 0.77… 오차: 1.16… 최적 학습 횟수: 20





과적합 발생한건가..?

〈 모델 요약 〉

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 64, 16)        448
_________________________________________________________________
activation_1 (Activation)    (None, 64, 64, 16)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 32)        4640
_________________________________________________________________
activation_2 (Activation)    (None, 32, 32, 32)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
activation_3 (Activation)    (None, 16, 16, 64)        0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 8, 90)          51930
_________________________________________________________________
activation_4 (Activation)    (None, 8, 8, 90)          0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 4, 4, 90)          0
_________________________________________________________________
global_max_pooling2d_1 (Glob (None, 90)                0
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 273
_________________________________________________________________
activation_5 (Activation)    (None, 3)                 0
=================================================================
Total params: 75,787
Trainable params: 75,787
Non-trainable params: 0
_________________________________________________________________
```

5. Model_05 - 4층 신경망 + Max Pooling + Global Max Pooling

모델 정확도: 0.77… 오차: 1.16… 최적 학습 횟수: 20

```
early_stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

# 데이터 로딩 (1)
def load_dataset():
    x_train, x_test, y_train, y_test = np.load("../DeepFashion/attribute_predict/up_down/fashion.npy")
    x_train = x_train.astype("float") / 256
    x_test = x_test.astype("float") / 256
    y_train = np_utils.to_categorical(y_train, nb_classes)
    y_test = np_utils.to_categorical(y_test, nb_classes)
    return x_train, x_test, y_train, y_test

# 모델 구성 (2)
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(16, 3, 3, border_mode='Same',
                            input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Convolution2D(32, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Convolution2D(64, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Convolution2D(90, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(GlobalMaxPooling2D(data_format="channels_last"))
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model
```

model_04 에서는 Convolution2D(128, 3, 3)으로 적용 - 여기선 90으로 좀 줄여보았다.

5. Model_05 - 4층 신경망 + Max Pooling + Global Max Pooling

실제 데이터 예측 결과 (한 벌 옷, 상의, 하의 순..)

```
입력: ./test_img/full1.jpg
예측: [ 2 ] upper_clothes / Score 0.99785346
입력: ./test_img/full2.jpg
예측: [ 2 ] upper_clothes / Score 0.5594618
입력: ./test_img/full3.jpg
예측: [ 1 ] lower_clothes / Score 0.9969897
입력: ./test_img/full4.jpg
예측: [ 2 ] upper_clothes / Score 0.99511504
입력: ./test_img/full5.jpg
예측: [ 1 ] lower_clothes / Score 0.97341657
입력: ./test_img/full6.jpg
예측: [ 0 ] full_clothes / Score 0.9999945
입력: ./test_img/full7.jpg
예측: [ 1 ] lower_clothes / Score 0.9784937
입력: ./test_img/full8.jpg
예측: [ 0 ] full_clothes / Score 0.999892
입력: ./test_img/full9.jpg
예측: [ 1 ] lower_clothes / Score 0.99118656
입력: ./test_img/full10.jpg
예측: [ 1 ] lower_clothes / Score 0.73652357
입력: ./test_img/full11.jpg
예측: [ 2 ] upper_clothes / Score 0.9999995
입력: ./test_img/full12.jpg
예측: [ 0 ] full_clothes / Score 0.9537245
입력: ./test_img/full13.jpg
예측: [ 2 ] upper_clothes / Score 0.8484007
입력: ./test_img/full14.jpg
예측: [ 0 ] full_clothes / Score 0.99158597
입력: ./test_img/full15.jpg
예측: [ 1 ] lower_clothes / Score 0.5929195
입력: ./test_img/full16.jpg
예측: [ 0 ] full_clothes / Score 0.8847423
입력: ./test_img/full17.jpg
예측: [ 1 ] lower_clothes / Score 0.77492505
입력: ./test_img/full18.jpg
예측: [ 1 ] lower_clothes / Score 0.99261236
입력: ./test_img/full19.jpg
예측: [ 0 ] full_clothes / Score 0.9999449
입력: ./test_img/full20.jpg
예측: [ 0 ] full_clothes / Score 0.65691084
```

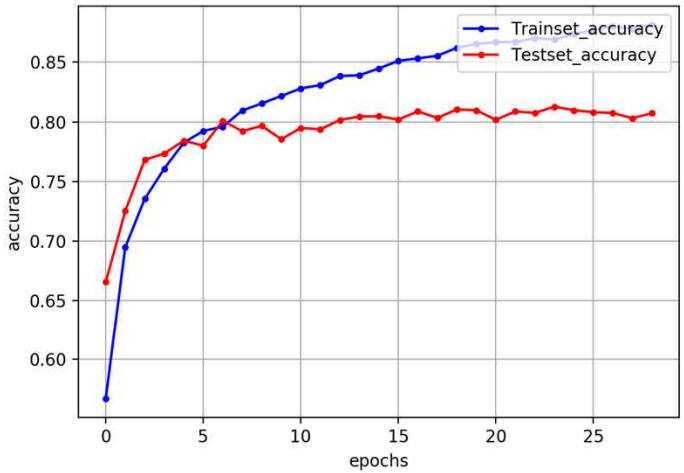
```
입력: ./test_img/low1.jpg
예측: [ 2 ] upper_clothes / Score 0.9354637
입력: ./test_img/low2.jpg
예측: [ 1 ] lower_clothes / Score 1.0
입력: ./test_img/low3.jpg
예측: [ 1 ] lower_clothes / Score 0.806592
입력: ./test_img/low4.jpg
예측: [ 2 ] upper_clothes / Score 0.9982723
입력: ./test_img/low5.jpg
예측: [ 2 ] upper_clothes / Score 0.8354702
입력: ./test_img/low6.jpg
예측: [ 1 ] lower_clothes / Score 1.0
입력: ./test_img/low7.jpg
예측: [ 1 ] lower_clothes / Score 0.99666023
입력: ./test_img/low8.jpg
예측: [ 1 ] lower_clothes / Score 0.91336447
입력: ./test_img/low9.jpg
예측: [ 1 ] lower_clothes / Score 0.58226943
입력: ./test_img/low10.jpg
예측: [ 1 ] lower_clothes / Score 0.999729
입력: ./test_img/low11.jpg
예측: [ 1 ] lower_clothes / Score 0.953235
입력: ./test_img/low12.jpg
예측: [ 1 ] lower_clothes / Score 0.9556869
입력: ./test_img/low13.jpg
예측: [ 1 ] lower_clothes / Score 0.79239684
입력: ./test_img/low14.jpg
예측: [ 0 ] full_clothes / Score 0.7929369
입력: ./test_img/low15.jpg
예측: [ 1 ] lower_clothes / Score 1.0
입력: ./test_img/low16.jpg
예측: [ 1 ] lower_clothes / Score 0.9999981
입력: ./test_img/low17.jpg
예측: [ 1 ] lower_clothes / Score 0.94242936
입력: ./test_img/low18.jpg
예측: [ 1 ] lower_clothes / Score 0.9999995
입력: ./test_img/low19.jpg
예측: [ 1 ] lower_clothes / Score 0.97552574
입력: ./test_img/low20.jpg
예측: [ 2 ] upper_clothes / Score 0.9999938
```

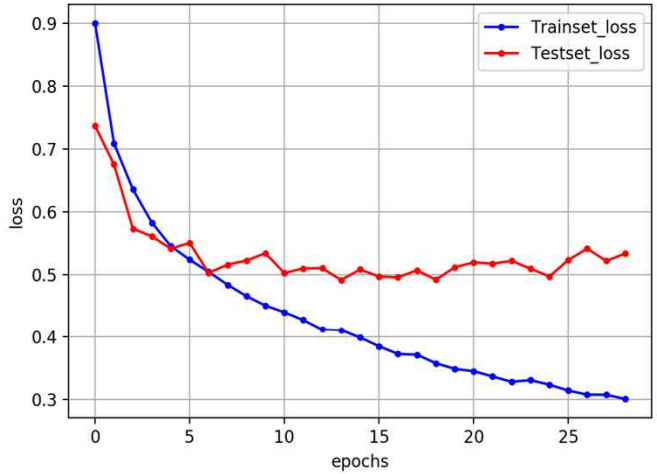
```
입력: ./test_img/upp1.jpg
예측: [ 2 ] upper_clothes / Score 0.92910874
입력: ./test_img/upp2.jpg
예측: [ 2 ] upper_clothes / Score 0.99387556
입력: ./test_img/upp3.jpg
예측: [ 2 ] upper_clothes / Score 0.9998604
입력: ./test_img/upp4.jpg
예측: [ 2 ] upper_clothes / Score 1.0
입력: ./test_img/upp5.jpg
예측: [ 2 ] upper_clothes / Score 0.99999344
입력: ./test_img/upp6.jpg
예측: [ 2 ] upper_clothes / Score 0.99864393
입력: ./test_img/upp7.jpg
예측: [ 2 ] upper_clothes / Score 0.9580036
입력: ./test_img/upp8.jpg
예측: [ 0 ] full_clothes / Score 0.99994576
입력: ./test_img/upp9.jpg
예측: [ 1 ] lower_clothes / Score 0.99865973
입력: ./test_img/upp10.jpg
예측: [ 2 ] upper_clothes / Score 1.0
입력: ./test_img/upp11.jpg
예측: [ 2 ] upper_clothes / Score 0.9999999
입력: ./test_img/upp12.jpg
예측: [ 2 ] upper_clothes / Score 0.99988174
입력: ./test_img/upp13.jpg
예측: [ 1 ] lower_clothes / Score 0.9738728
입력: ./test_img/upp14.jpg
예측: [ 2 ] upper_clothes / Score 1.0
입력: ./test_img/upp15.jpg
예측: [ 2 ] upper_clothes / Score 0.9995009
입력: ./test_img/upp16.jpg
예측: [ 2 ] upper_clothes / Score 0.99995804
입력: ./test_img/upp17.jpg
예측: [ 0 ] full_clothes / Score 0.9893785
입력: ./test_img/upp18.jpg
예측: [ 2 ] upper_clothes / Score 1.0
입력: ./test_img/upp19.jpg
예측: [ 2 ] upper_clothes / Score 0.97917366
입력: ./test_img/upp20.jpg
예측: [ 2 ] upper_clothes / Score 0.99988365
```

6. Model_06 - 4층 신경망 + Max Pooling + Global Max Pooling + DropOut(0.125)

early_Stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

모델 정확도: 0.80… 오차: 0.53… 최적 학습 횟수: 29





〈 모델 요약 〉

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 64, 16)        448
_________________________________________________________________
activation_1 (Activation)    (None, 64, 64, 16)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 16)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 32, 32, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 32)        4640
_________________________________________________________________
activation_2 (Activation)    (None, 32, 32, 32)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
activation_3 (Activation)    (None, 16, 16, 64)        0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
dropout_3 (Dropout)          (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 8, 128)         73856
_________________________________________________________________
activation_4 (Activation)    (None, 8, 8, 128)         0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 4, 4, 128)         0
_________________________________________________________________
global_max_pooling2d_1 (Glob (None, 128)               0
_________________________________________________________________
dropout_4 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 387
_________________________________________________________________
activation_5 (Activation)    (None, 3)                 0
=================================================================
Total params: 97,827
Trainable params: 97,827
Non-trainable params: 0
```

6. Model_06 - 4층 신경망 + Max Pooling + Global Max Pooling + DropOut(0.125)

모델 정확도: 0.80… 오차: 0.53… 최적 학습 횟수: 29

```
early_stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

# 데이터 로딩 (1)
def load_dataset():
    x_train, x_test, y_train, y_test = np.load("../DeepFashion/attribute_predict/up_down/fashion.npy")
    x_train = x_train.astype("float") / 256
    x_test = x_test.astype("float") / 256
    y_train = np_utils.to_categorical(y_train, nb_classes)
    y_test = np_utils.to_categorical(y_test, nb_classes)
    return x_train, x_test, y_train, y_test

# 모델 구성 (2)
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(16, 3, 3, border_mode='Same',
                            input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.125))
    model.add(Convolution2D(32, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.125))
    model.add(Convolution2D(64, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.125))
    model.add(Convolution2D(128, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(GlobalMaxPooling2D(data_format="channels_last"))
    model.add(Dropout(0.125))
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model
```

6. Model_06 - 4층 신경망 + Max Pooling + Global Max Pooling + DropOut(0.125)

실제 데이터 예측 결과

```
입력: ./test_img/full1.jpg
예측: [ 2 ] upper_clothes / Score 0.9756496
입력: ./test_img/full2.jpg
예측: [ 1 ] lower_clothes / Score 0.4721738
입력: ./test_img/full3.jpg
예측: [ 1 ] lower_clothes / Score 0.72904134
입력: ./test_img/full4.jpg
예측: [ 0 ] full_clothes / Score 0.9643534
입력: ./test_img/full5.jpg
예측: [ 1 ] lower_clothes / Score 0.81603014
입력: ./test_img/full6.jpg
예측: [ 0 ] full_clothes / Score 0.76431984
입력: ./test_img/full7.jpg
예측: [ 0 ] full_clothes / Score 0.954797
입력: ./test_img/full8.jpg
예측: [ 2 ] upper_clothes / Score 0.56706077
입력: ./test_img/full9.jpg
예측: [ 2 ] upper_clothes / Score 0.6527321
입력: ./test_img/full10.jpg
예측: [ 0 ] full_clothes / Score 0.90252334
입력: ./test_img/full11.jpg
예측: [ 1 ] lower_clothes / Score 0.7012841
입력: ./test_img/full12.jpg
예측: [ 0 ] full_clothes / Score 0.7069121
입력: ./test_img/full13.jpg
예측: [ 2 ] upper_clothes / Score 0.49874723
입력: ./test_img/full14.jpg
예측: [ 0 ] full_clothes / Score 0.99782276
입력: ./test_img/full15.jpg
예측: [ 0 ] full_clothes / Score 0.60792345
입력: ./test_img/full16.jpg
예측: [ 0 ] full_clothes / Score 0.87502307
입력: ./test_img/full17.jpg
예측: [ 0 ] full_clothes / Score 0.6827216
입력: ./test_img/full18.jpg
예측: [ 0 ] full_clothes / Score 0.98096406
입력: ./test_img/full19.jpg
예측: [ 0 ] full_clothes / Score 0.64843047
입력: ./test_img/full20.jpg
```

한벌 옷 - 12개 맞춤

```
입력: ./test_img/low1.jpg
예측: [ 1 ] lower_clothes / Score 0.8083703
입력: ./test_img/low2.jpg
예측: [ 1 ] lower_clothes / Score 0.9790583
입력: ./test_img/low3.jpg
예측: [ 1 ] lower_clothes / Score 0.9810394
입력: ./test_img/low4.jpg
예측: [ 2 ] upper_clothes / Score 0.7604095
입력: ./test_img/low5.jpg
예측: [ 0 ] full_clothes / Score 0.85061914
입력: ./test_img/low6.jpg
예측: [ 1 ] lower_clothes / Score 0.999995
입력: ./test_img/low7.jpg
예측: [ 2 ] upper_clothes / Score 0.9403193
입력: ./test_img/low8.jpg
예측: [ 1 ] lower_clothes / Score 0.8396491
입력: ./test_img/low9.jpg
예측: [ 1 ] lower_clothes / Score 0.5479467
입력: ./test_img/low10.jpg
예측: [ 1 ] lower_clothes / Score 0.79935163
입력: ./test_img/low11.jpg
예측: [ 1 ] lower_clothes / Score 0.9667391
입력: ./test_img/low12.jpg
예측: [ 2 ] upper_clothes / Score 0.90548843
입력: ./test_img/low13.jpg
예측: [ 1 ] lower_clothes / Score 0.51806796
입력: ./test_img/low14.jpg
예측: [ 2 ] upper_clothes / Score 0.6341971
입력: ./test_img/low15.jpg
예측: [ 1 ] lower_clothes / Score 0.99985754
입력: ./test_img/low16.jpg
예측: [ 1 ] lower_clothes / Score 0.9999534
입력: ./test_img/low17.jpg
예측: [ 1 ] lower_clothes / Score 0.99725085
입력: ./test_img/low18.jpg
예측: [ 1 ] lower_clothes / Score 0.9984262
입력: ./test_img/low19.jpg
예측: [ 1 ] lower_clothes / Score 0.9973387
입력: ./test_img/low20.jpg
예측: [ 2 ] upper_clothes / Score 0.70259523
```

하의 - 14개 맞춤

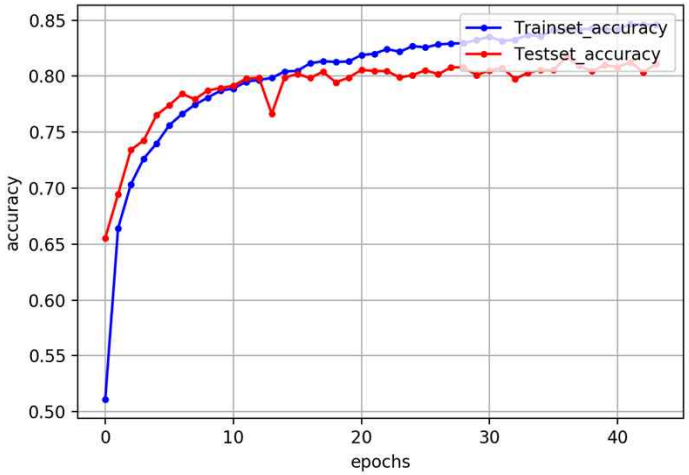
```
입력: ./test_img/upp1.jpg
예측: [ 2 ] upper_clothes / Score 0.6027518
입력: ./test_img/upp2.jpg
예측: [ 2 ] upper_clothes / Score 0.95239234
입력: ./test_img/upp3.jpg
예측: [ 2 ] upper_clothes / Score 0.8539214
입력: ./test_img/upp4.jpg
예측: [ 2 ] upper_clothes / Score 0.9999676
입력: ./test_img/upp5.jpg
예측: [ 2 ] upper_clothes / Score 0.98077
입력: ./test_img/upp6.jpg
예측: [ 2 ] upper_clothes / Score 0.6276259
입력: ./test_img/upp7.jpg
예측: [ 2 ] upper_clothes / Score 0.86898196
입력: ./test_img/upp8.jpg
예측: [ 0 ] full_clothes / Score 0.55596876
입력: ./test_img/upp9.jpg
예측: [ 2 ] upper_clothes / Score 0.76674885
입력: ./test_img/upp10.jpg
예측: [ 2 ] upper_clothes / Score 0.96211076
입력: ./test_img/upp11.jpg
예측: [ 2 ] upper_clothes / Score 0.9889326
입력: ./test_img/upp12.jpg
예측: [ 2 ] upper_clothes / Score 0.99575716
입력: ./test_img/upp13.jpg
예측: [ 2 ] upper_clothes / Score 0.9623886
입력: ./test_img/upp14.jpg
예측: [ 2 ] upper_clothes / Score 0.9999516
입력: ./test_img/upp15.jpg
예측: [ 2 ] upper_clothes / Score 0.9301461
입력: ./test_img/upp16.jpg
예측: [ 2 ] upper_clothes / Score 0.99958545
입력: ./test_img/upp17.jpg
예측: [ 2 ] upper_clothes / Score 0.67072225
입력: ./test_img/upp18.jpg
예측: [ 2 ] upper_clothes / Score 0.9929819
입력: ./test_img/upp19.jpg
예측: [ 2 ] upper_clothes / Score 0.49025315
입력: ./test_img/upp20.jpg
예측: [ 2 ] upper_clothes / Score 0.69724363
```

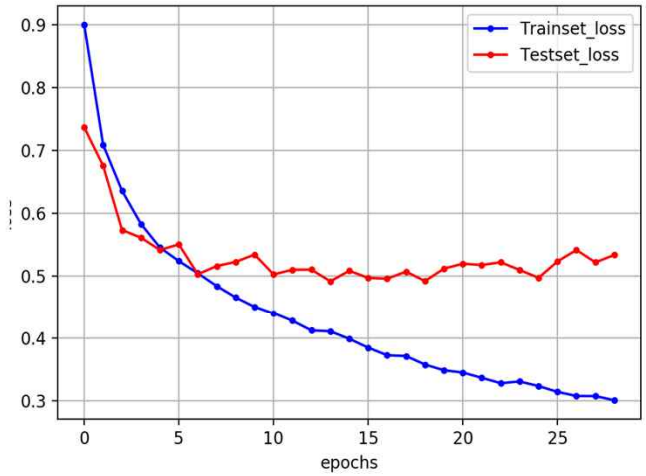
상의 - 19개 맞춤

7. Model_07 - 4층 신경망 + Max Pooling + Global Max Pooling + DropOut(0.20)

early_Stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

모델 정확도: 0.80… 오차: 0.53… 최적 학습 횟수: 29





〈 모델 요약 〉

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 64, 16)        448
_________________________________________________________________
activation_1 (Activation)    (None, 64, 64, 16)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 16)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 32, 32, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 32)        4640
_________________________________________________________________
activation_2 (Activation)    (None, 32, 32, 32)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
activation_3 (Activation)    (None, 16, 16, 64)        0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
dropout_3 (Dropout)          (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 8, 128)         73856
_________________________________________________________________
activation_4 (Activation)    (None, 8, 8, 128)         0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 4, 4, 128)         0
_________________________________________________________________
global_max_pooling2d_1 (Glob (None, 128)               0
_________________________________________________________________
dropout_4 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 387
_________________________________________________________________
activation_5 (Activation)    (None, 3)                 0
=================================================================
Total params: 97,827
Trainable params: 97,827
Non-trainable params: 0
```

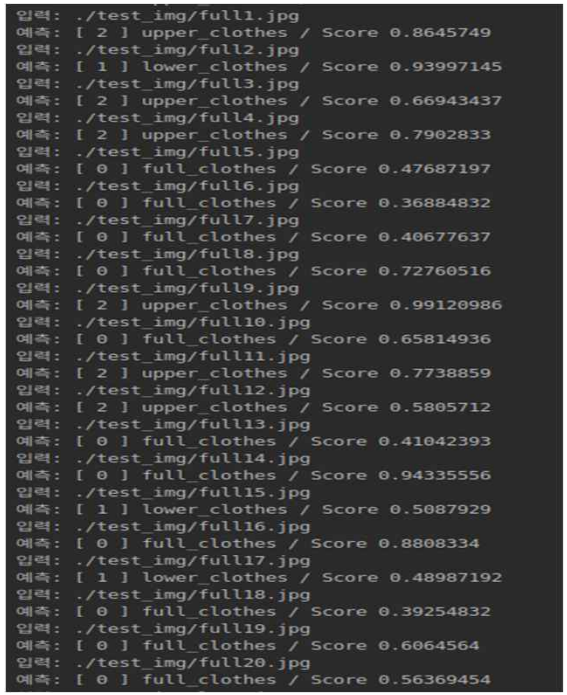
7. Model_07 - 4층 신경망 + Max Pooling + Global Max Pooling + DropOut(0.20)

모델 정확도: 0.81… 오차: 0.48… 최적 학습 횟수: 44

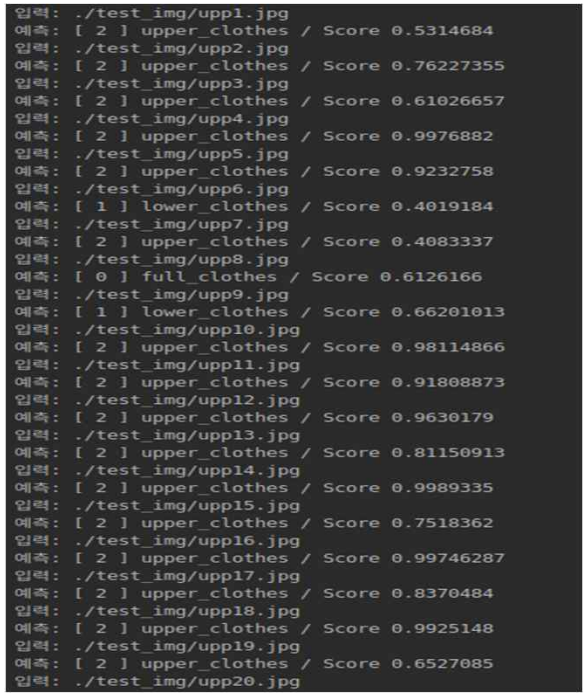
```
early_stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

# 데이터 로딩 (1)
def load_dataset():
    x_train, x_test, y_train, y_test = np.load("../DeepFashion/attribute_predict/up_down/fashion.npy")
    x_train = x_train.astype("float") / 256
    x_test = x_test.astype("float") / 256
    y_train = np_utils.to_categorical(y_train, nb_classes)
    y_test = np_utils.to_categorical(y_test, nb_classes)
    return x_train, x_test, y_train, y_test

# 모델 구성 (2)
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(16, 3, 3, border_mode='Same',
                            input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.125))
    model.add(Convolution2D(32, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.125))
    model.add(Convolution2D(64, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.125))
    model.add(Convolution2D(128, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(GlobalMaxPooling2D(data_format="channels_last"))
    model.add(Dropout(0.125))
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model
```

7. Model_07 - 4층 신경망 + Max Pooling + Global Max Pooling + DropOut(0.2)

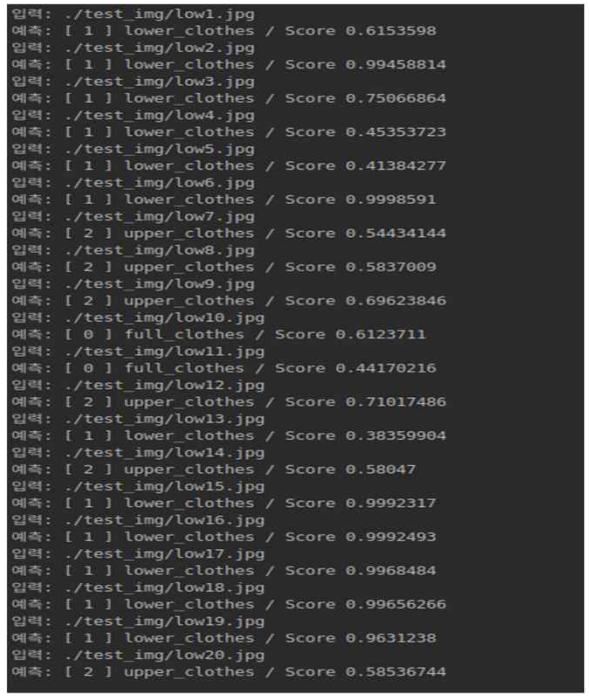
실제 데이터 예측 결과



한벌 옷 - 11개 맞춤



상의 - 17개 맞춤

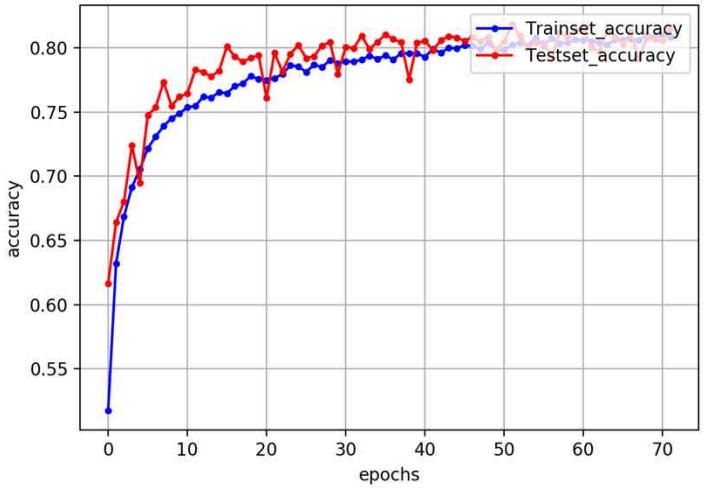


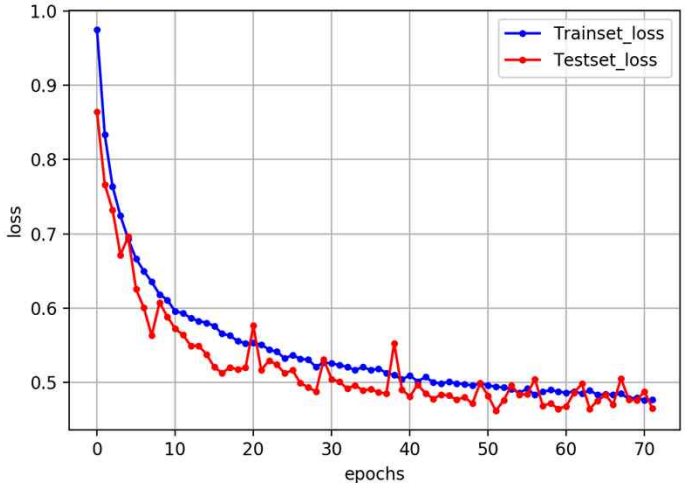
하의 - 12개 맞춤

8. Model_08 - 4층 신경망 + Max Pooling + Global Max Pooling + DropOut(0.3)

early_Stopping_callback = EarlyStopping(monitor='val_loss', patience=15)

모델 정확도: 0.81… 오차: 0.46… 최적 학습 횟수: 72





〈 모델 요약 〉

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 64, 16)        448
_________________________________________________________________
activation_1 (Activation)    (None, 64, 64, 16)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 16)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 32, 32, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 32)        4640
_________________________________________________________________
activation_2 (Activation)    (None, 32, 32, 32)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
activation_3 (Activation)    (None, 16, 16, 64)        0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
dropout_3 (Dropout)          (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 8, 128)         73856
_________________________________________________________________
activation_4 (Activation)    (None, 8, 8, 128)         0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 4, 4, 128)         0
_________________________________________________________________
dropout_4 (Dropout)          (None, 4, 4, 128)         0
_________________________________________________________________
global_max_pooling2d_1 (Glob (None, 128)               0
_________________________________________________________________
dropout_5 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 387
_________________________________________________________________
activation_5 (Activation)    (None, 3)                 0
=================================================================
Total params: 97,827
Trainable params: 97,827
Non-trainable params: 0
```

8. Model_08 - 4층 신경망 + Max Pooling + Global Max Pooling + DropOut(0.3)

모델 정확도: 0.81… 오차: 0.46… 최적 학습 횟수: 72

```
early_stopping_callback = EarlyStopping(monitor='val_loss', patience=20)

# 데이터 로딩 (1)
def load_dataset():
    x_train, x_test, y_train, y_test = np.load("../DeepFashion/attribute_predict/up_down/fashion.npy")
    x_train = x_train.astype("float") / 256
    x_test = x_test.astype("float") / 256
    y_train = np_utils.to_categorical(y_train, nb_classes)
    y_test = np_utils.to_categorical(y_test, nb_classes)
    return x_train, x_test, y_train, y_test

# 모델 구성 (2)
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(16, 3, 3, border_mode='Same',
                            input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.30))
    model.add(Convolution2D(32, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.30))
    model.add(Convolution2D(64, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.30))
    model.add(Convolution2D(128, 3, 3, border_mode='Same'))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.30))
    model.add(GlobalMaxPooling2D(data_format="channels_last"))
    model.add(Dropout(0.30))
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model
```

8. Model_08 - 4층 신경망 + Max Pooling + Global Max Pooling + DropOut(0.3)

실제 데이터 예측 결과

```
입력: ./test_img/full1.jpg
예측: [ 0 ] full_clothes / Score 0.47087273
입력: ./test_img/full2.jpg
예측: [ 0 ] full_clothes / Score 0.7023816
입력: ./test_img/full3.jpg
예측: [ 2 ] upper_clothes / Score 0.54072636
입력: ./test_img/full4.jpg
예측: [ 0 ] full_clothes / Score 0.681783
입력: ./test_img/full5.jpg
예측: [ 1 ] lower_clothes / Score 0.5021409
입력: ./test_img/full6.jpg
예측: [ 0 ] full_clothes / Score 0.6111427
입력: ./test_img/full7.jpg
예측: [ 0 ] full_clothes / Score 0.49102798
입력: ./test_img/full8.jpg
예측: [ 0 ] full_clothes / Score 0.59686935
입력: ./test_img/full9.jpg
예측: [ 2 ] upper_clothes / Score 0.92886853
입력: ./test_img/full10.jpg
예측: [ 0 ] full_clothes / Score 0.6481034
입력: ./test_img/full11.jpg
예측: [ 2 ] upper_clothes / Score 0.64378166
입력: ./test_img/full12.jpg
예측: [ 0 ] full_clothes / Score 0.6437298
입력: ./test_img/full13.jpg
예측: [ 1 ] lower_clothes / Score 0.49152508
입력: ./test_img/full14.jpg
예측: [ 0 ] full_clothes / Score 0.9305884
입력: ./test_img/full15.jpg
예측: [ 1 ] lower_clothes / Score 0.4907147
입력: ./test_img/full16.jpg
예측: [ 0 ] full_clothes / Score 0.7330772
입력: ./test_img/full17.jpg
예측: [ 0 ] full_clothes / Score 0.43024376
입력: ./test_img/full18.jpg
예측: [ 0 ] full_clothes / Score 0.6587686
입력: ./test_img/full19.jpg
예측: [ 0 ] full_clothes / Score 0.71338373
입력: ./test_img/full20.jpg
예측: [ 0 ] full_clothes / Score 0.8314269
```

한벌 옷 - 14개 맞춤

```
입력: ./test_img/low1.jpg
예측: [ 2 ] upper_clothes / Score 0.634558
입력: ./test_img/low2.jpg
예측: [ 1 ] lower_clothes / Score 0.98777384
입력: ./test_img/low3.jpg
예측: [ 2 ] upper_clothes / Score 0.62805
입력: ./test_img/low4.jpg
예측: [ 1 ] lower_clothes / Score 0.4347358
입력: ./test_img/low5.jpg
예측: [ 1 ] lower_clothes / Score 0.45046985
입력: ./test_img/low6.jpg
예측: [ 1 ] lower_clothes / Score 0.9993293
입력: ./test_img/low7.jpg
예측: [ 2 ] upper_clothes / Score 0.48110464
입력: ./test_img/low8.jpg
예측: [ 1 ] lower_clothes / Score 0.70844406
입력: ./test_img/low9.jpg
예측: [ 2 ] upper_clothes / Score 0.74533284
입력: ./test_img/low10.jpg
예측: [ 0 ] full_clothes / Score 0.4398575
입력: ./test_img/low11.jpg
예측: [ 1 ] lower_clothes / Score 0.63432896
입력: ./test_img/low12.jpg
예측: [ 1 ] lower_clothes / Score 0.5142531
입력: ./test_img/low13.jpg
예측: [ 1 ] lower_clothes / Score 0.69356084
입력: ./test_img/low14.jpg
예측: [ 2 ] upper_clothes / Score 0.55263364
입력: ./test_img/low15.jpg
예측: [ 1 ] lower_clothes / Score 0.9650162
입력: ./test_img/low16.jpg
예측: [ 1 ] lower_clothes / Score 0.9856608
입력: ./test_img/low17.jpg
예측: [ 1 ] lower_clothes / Score 0.9945686
입력: ./test_img/low18.jpg
예측: [ 1 ] lower_clothes / Score 0.99005824
입력: ./test_img/low19.jpg
예측: [ 1 ] lower_clothes / Score 0.9983853
입력: ./test_img/low20.jpg
예측: [ 1 ] lower_clothes / Score 0.4344037
```

하의 - 14개 맞춤

```
입력: ./test_img/upp1.jpg
예측: [ 2 ] upper_clothes / Score 0.8831399
입력: ./test_img/upp2.jpg
예측: [ 2 ] upper_clothes / Score 0.56683856
입력: ./test_img/upp3.jpg
예측: [ 2 ] upper_clothes / Score 0.7037547
입력: ./test_img/upp4.jpg
예측: [ 2 ] upper_clothes / Score 0.9788028
입력: ./test_img/upp5.jpg
예측: [ 2 ] upper_clothes / Score 0.98827374
입력: ./test_img/upp6.jpg
예측: [ 2 ] upper_clothes / Score 0.4708245
입력: ./test_img/upp7.jpg
예측: [ 2 ] upper_clothes / Score 0.6112917
입력: ./test_img/upp8.jpg
예측: [ 1 ] lower_clothes / Score 0.45924893
입력: ./test_img/upp9.jpg
예측: [ 1 ] lower_clothes / Score 0.6276492
입력: ./test_img/upp10.jpg
예측: [ 2 ] upper_clothes / Score 0.9652637
입력: ./test_img/upp11.jpg
예측: [ 2 ] upper_clothes / Score 0.85597396
입력: ./test_img/upp12.jpg
예측: [ 2 ] upper_clothes / Score 0.98270345
입력: ./test_img/upp13.jpg
예측: [ 2 ] upper_clothes / Score 0.64060956
입력: ./test_img/upp14.jpg
예측: [ 2 ] upper_clothes / Score 0.9987091
입력: ./test_img/upp15.jpg
예측: [ 1 ] lower_clothes / Score 0.4681727
입력: ./test_img/upp16.jpg
예측: [ 2 ] upper_clothes / Score 0.9913237
입력: ./test_img/upp17.jpg
예측: [ 2 ] upper_clothes / Score 0.5103139
입력: ./test_img/upp18.jpg
예측: [ 2 ] upper_clothes / Score 0.96140516
입력: ./test_img/upp19.jpg
예측: [ 2 ] upper_clothes / Score 0.6889144
입력: ./test_img/upp20.jpg
예측: [ 2 ] upper_clothes / Score 0.85291773
```

상의 - 17개 맞춤