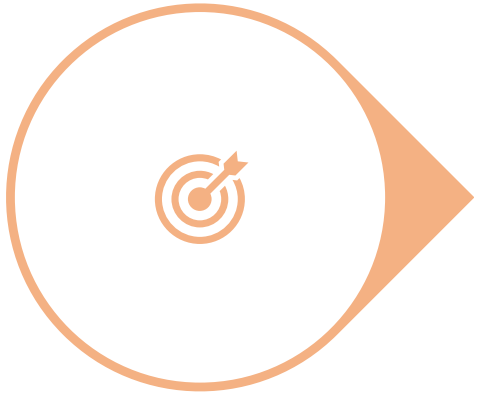
딥러닝 분반 (CV 프로젝트)

Human Protein Atlas Image Classification

13기 최해윤 · 14기 김종민 · 14기 이우준 · 14기 최대원

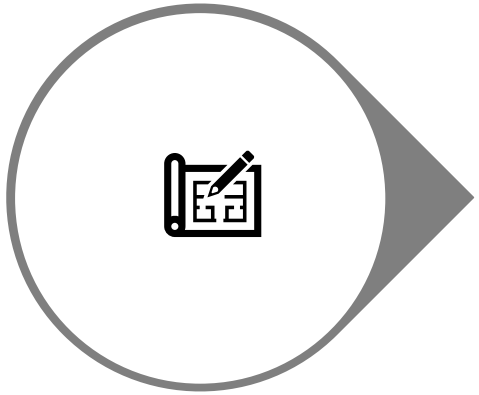
목차



1. 주제 및 배경지식

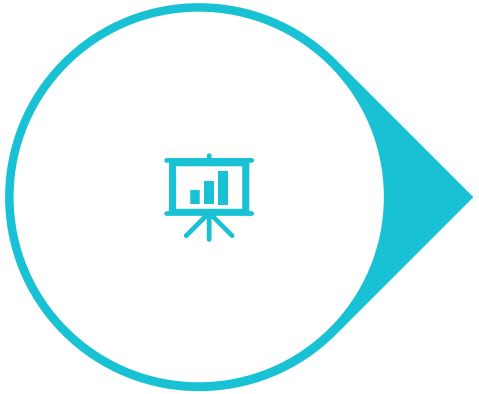
2. 데이터

- EDA
- 전처리 및 최종 데이터



3. 모델

- 모델 간 비교
- Filter 조절(InceptionResNetV2)



4. 결과 및 결론

- 최종 모델 submssion
-Discussion

목적: microscope image에서 mixed patterns of protein을 분류

0. Nucleoplasm	14. Microtubules
1. Nuclear membrane	15. Microtubule ends
2. Nucleoli	16. Cytokinetic bridge
3. Nucleoli fibrillar center	17. Mitotic spindle
4. Nuclear speckles	18. Microtubule organizing center
5. Nuclear bodies	19. Centrosome
6. Endoplasmic reticulum	20. Lipid droplets
7. Golgi apparatus	21. Plasma membrane
8. Peroxisomes	22. Cell junctions
9. Endosomes	23. Mitochondria
10. Lysosomes	24. Aggresome
11. Intermediate filaments	25. Cytosol
12. Actin filaments	26. Cytoplasmic bodies
13. Focal adhesion sites	27. Rods & rings

* 28가지의 단백질 존재

Macro-F1:
is the harmonic mean between precision and recall, where the average is calculated per label and then averaged across all labels. If p_j and r_j are the precision and recall for all $\lambda_j \in h(x_i)$ from $\lambda_j \in y_i$, the macro-F1 is

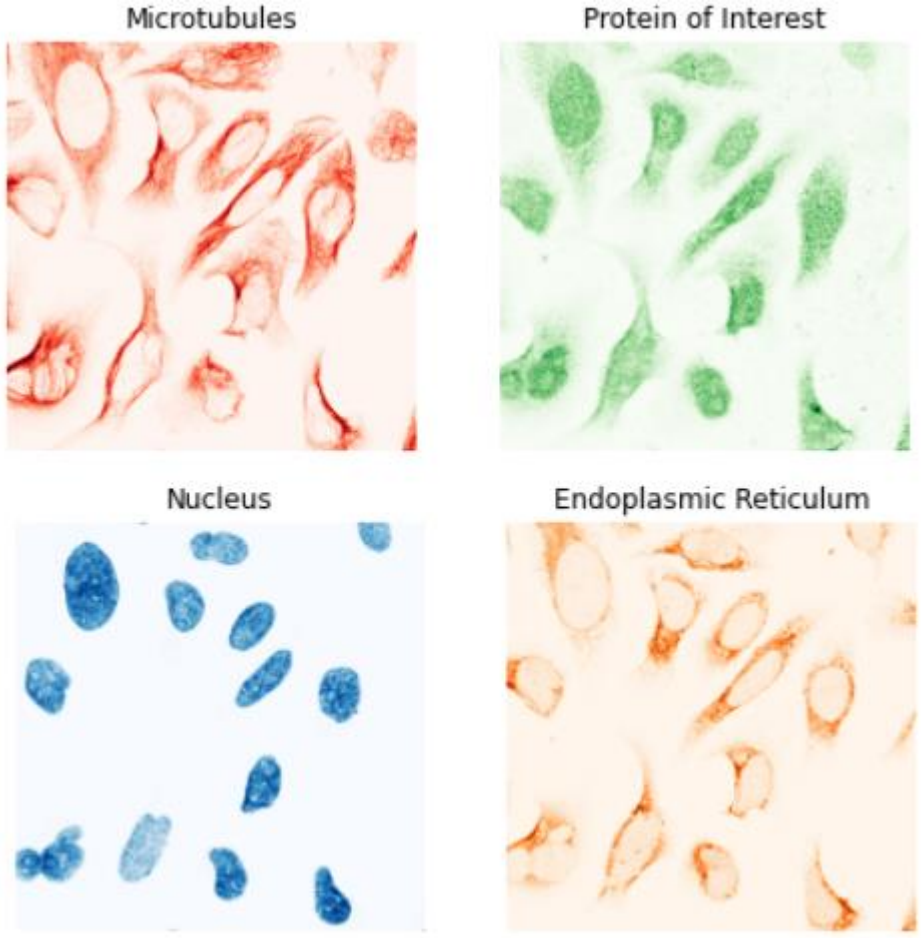
$$\text{Macro-F1} = \frac{1}{Q} \sum_{j=1}^{Q} \frac{2 \times pj \times rj}{pj + rj}$$

- 31,072개의 train data

	Id	Target
0	00070df0-bbc3-11e8-b2bc-ac1f6b6435d0	16 0
1	000a6c98-bb9b-11e8-b2b9-ac1f6b6435d0	7 1 2 0
2	000a9596-bbc4-11e8-b2bc-ac1f6b6435d0	5
3	000c99ba-bba4-11e8-b2b9-ac1f6b6435d0	1
4	001838f8-bbca-11e8-b2bc-ac1f6b6435d0	18
...	...	...
31067	ffed4430-bbac-11e8-b2ba-ac1f6b6435d0	21
31068	fff0a998-bbae-11e8-b2ba-ac1f6b6435d0	5
31069	fff189d8-bbab-11e8-b2ba-ac1f6b6435d0	7
31070	fffdf7e0-bbc4-11e8-b2bc-ac1f6b6435d0	25 2 21
31071	fffe0ffe-bbc0-11e8-b2bb-ac1f6b6435d0	2 0

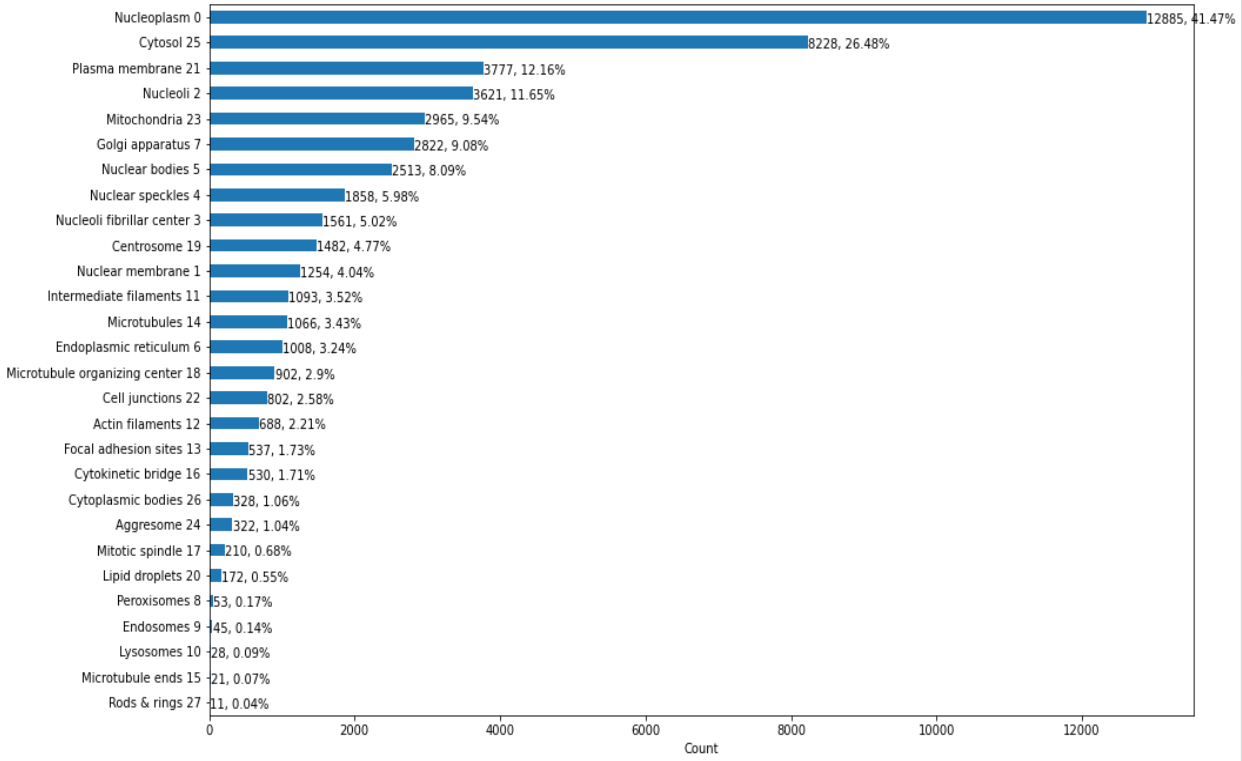
31072 rows × 2 columns

각 데이터는 R,G,B,Y 4개의 이미지로 구성
(green: protein of interest, blue: 핵, red: 미세관, yellow: 소포내막)

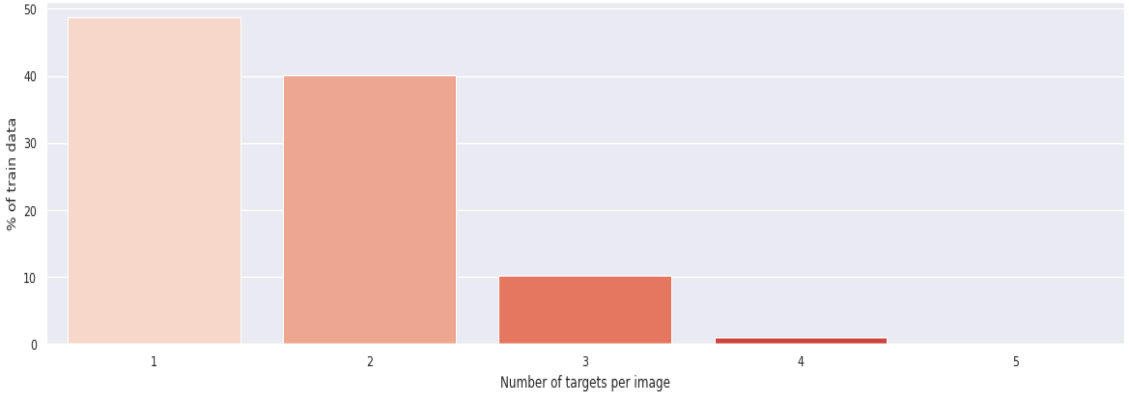


1) EDA

- 단백질 빈도



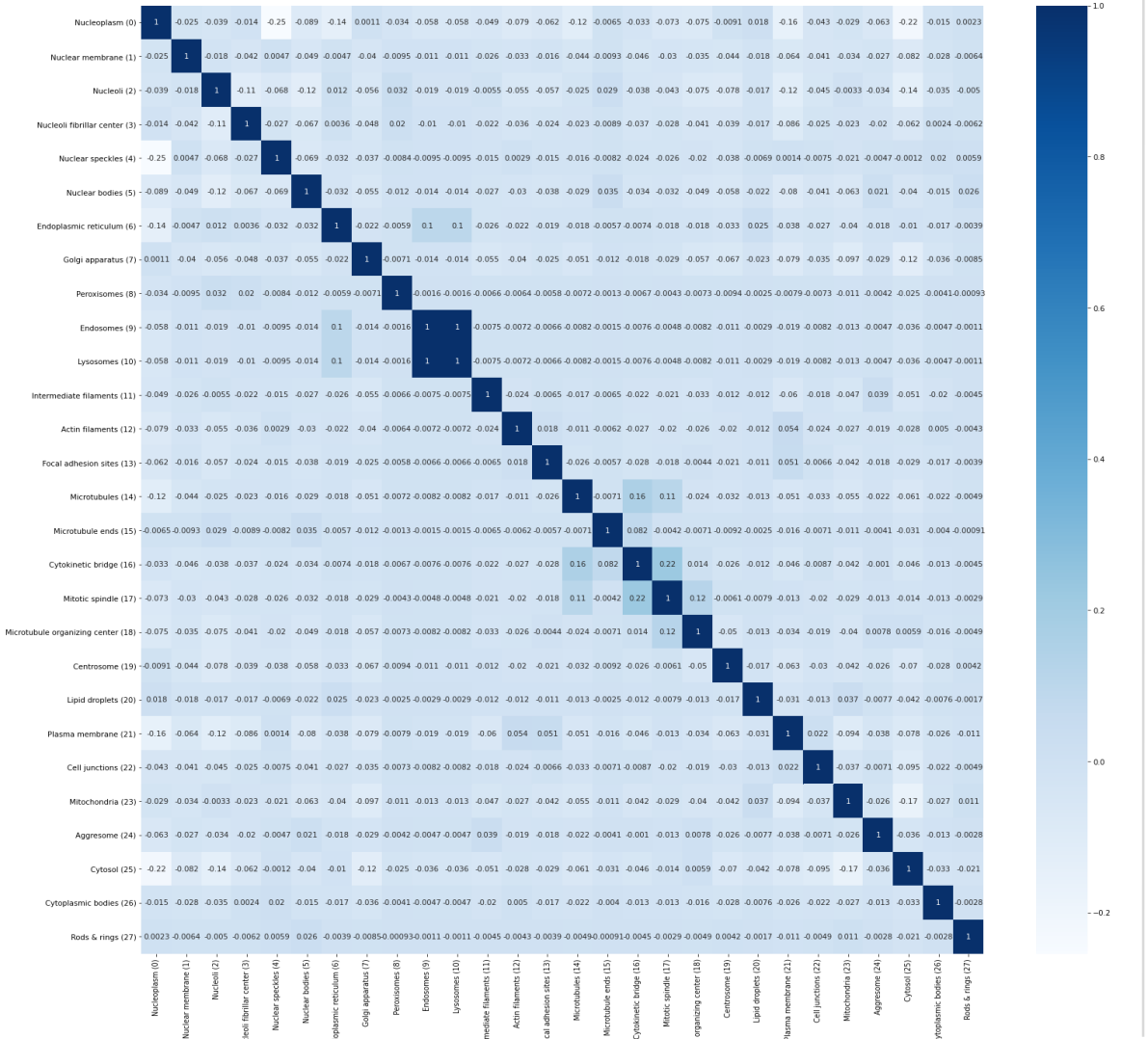
- 데이터 imbalance 문제

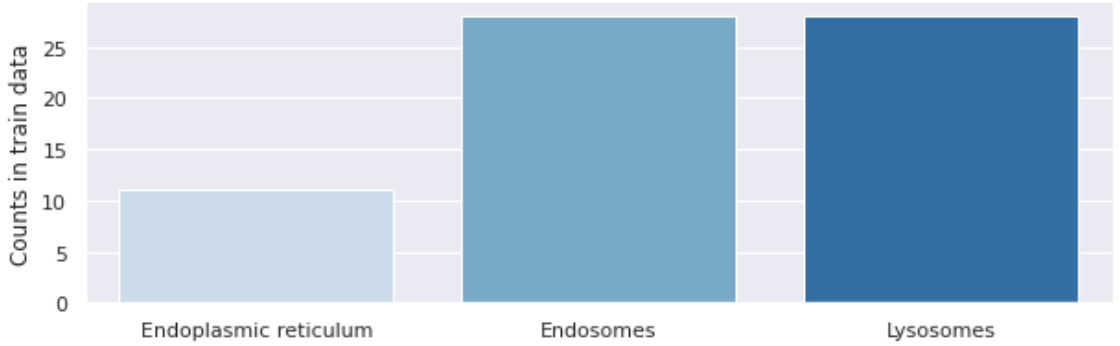


	image_count	pct_of_dataset
1	15126	48.680484
2	12485	40.180870
3	3160	10.169928
4	299	0.962281
5	2	0.006437

1) EDA

- 단백질 간 Correlation





Lysomes & Endosomes

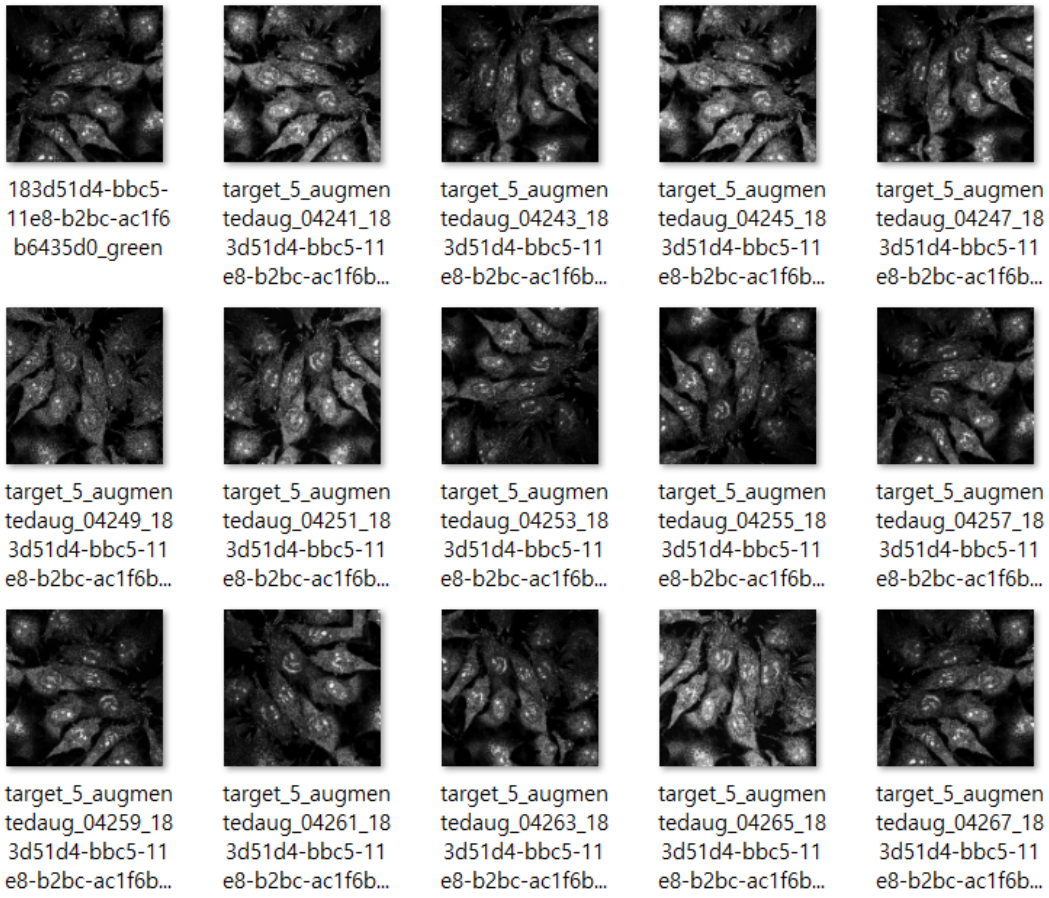
상관관계가 높음

Lysomes & Endosomes은 항상 동시 관찰

2) 전처리 및 최종 데이터

Albumentations Library 이용하여 **데이터 증강**

```
def strong_aug(p=1):
    return Compose([
        RandomRotate90(),
        Flip(),
        Transpose(),
        OneOf([
            IAAAdditiveGaussianNoise(),
            GaussNoise(),
        ], p=0.2),
        OneOf([
            MotionBlur(p=.2),
            MedianBlur(blur_limit=3, p=.1),
            Blur(blur_limit=3, p=.1),
        ], p=0.2),
        ShiftScaleRotate(shift_limit=0.0625, scale_limit=0.2,
                         rotate_limit=45, p=.2),
        OneOf([
            OpticalDistortion(p=0.3),
            GridDistortion(p=.1),
            IAAPiecewiseAffine(p=0.3),
        ], p=0.2),
        OneOf([
            CLAHE(clip_limit=2),
            IAASharpen(),
            IAAEmboss(),
            RandomContrast(),
            RandomBrightness(),
        ], p=0.3),
    ], p=p)
```



(기존 31072개 데이터 + 11539개 데이터)

단백질을 딥러닝 모델을 통해 예측

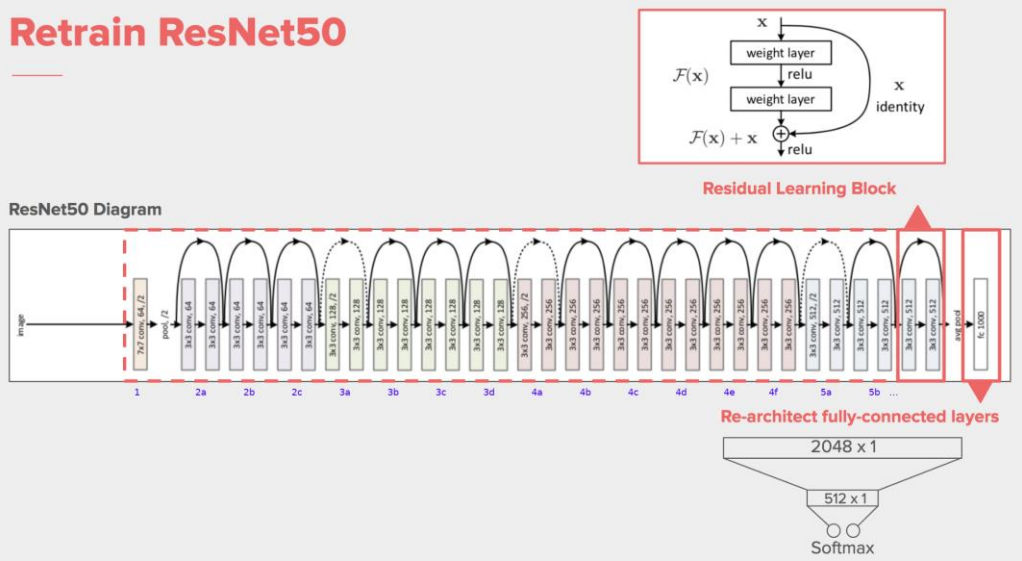
HOW?

①Fine-tuning 기법을 통한 모델링 -> 여러가지 모델을 훈련, 평가 후 비교

② 데이터가 R,G,B,Y 로 주어져 있음-> 채널 (channel)을 여러가지 방식으로 구성 후 비교

1) 모델 간 비교

①ResNet50 fine-tune
(R/2+Y/2, G/2+Y/2, B)

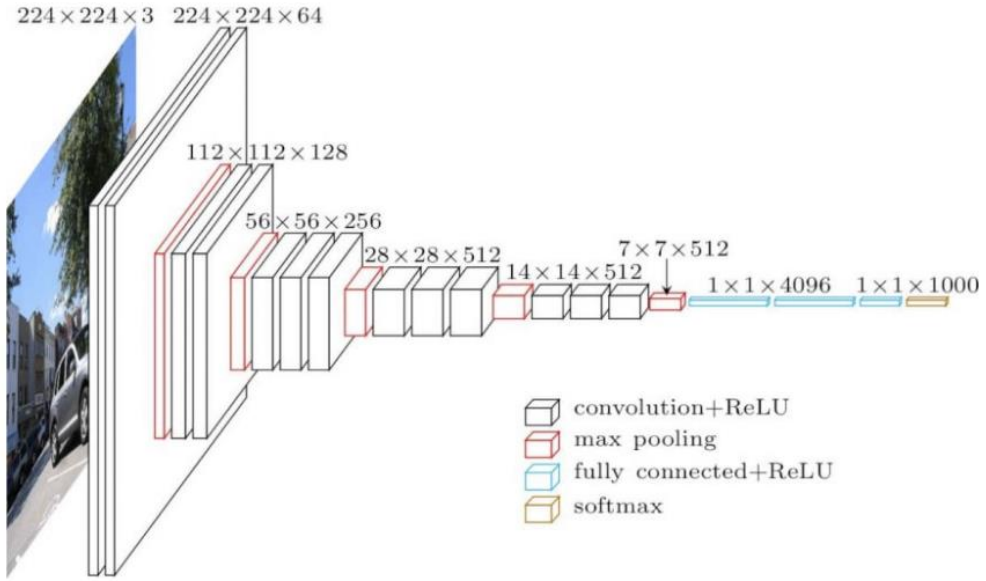



```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 299, 299, 3)]     0
_________________________________________________________________
batch_normalization (BatchNo (None, 299, 299, 3)       12
_________________________________________________________________
resnet50 (Functional)        (None, 10, 10, 2048)      23587712
_________________________________________________________________
conv2d (Conv2D)              (None, 10, 10, 32)        65568
_________________________________________________________________
flatten (Flatten)            (None, 3200)              0
_________________________________________________________________
dropout (Dropout)            (None, 3200)              0
_________________________________________________________________
dense (Dense)                (None, 1024)              3277824
_________________________________________________________________
dropout_1 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 28)                28700
=================================================================
Total params: 26,959,816
Trainable params: 26,906,690
Non-trainable params: 53,126
```

```
Epoch 7/10
4528/4528 [==============================] - 1419s 313ms/step - loss: 0.1729 - binary_accuracy: 0.9393 - f1: 0.1290 - val_loss: 0.1660 - val_binary_accuracy: 0.9416 - val_f1: 0.1753

Epoch 00007: val_loss improved from 0.16707 to 0.16598, saving model to best_model3.h5
Epoch 8/10
4528/4528 [==============================] - 1403s 310ms/step - loss: 0.1721 - binary_accuracy: 0.9394 - f1: 0.1298 - val_loss: 0.1721 - val_binary_accuracy: 0.9393 - val_f1: 0.1687

Epoch 00008: val_loss did not improve from 0.16598
Epoch 9/10
4528/4528 [==============================] - 1387s 306ms/step - loss: 0.1728 - binary_accuracy: 0.9392 - f1: 0.1290 - val_loss: 0.1685 - val_binary_accuracy: 0.9410 - val_f1: 0.1742

Epoch 00009: val_loss did not improve from 0.16598
Epoch 10/10
4528/4528 [==============================] - 1397s 309ms/step - loss: 0.1754 - binary_accuracy: 0.9383 - f1: 0.1248 - val_loss: 0.1700 - val_binary_accuracy: 0.9408 - val_f1: 0.1725

Epoch 00010: val_loss did not improve from 0.16598
```

1) 모델 간 비교

②VGG16 fine-tune
(R/2+Y/2, G/2+Y/2, B)



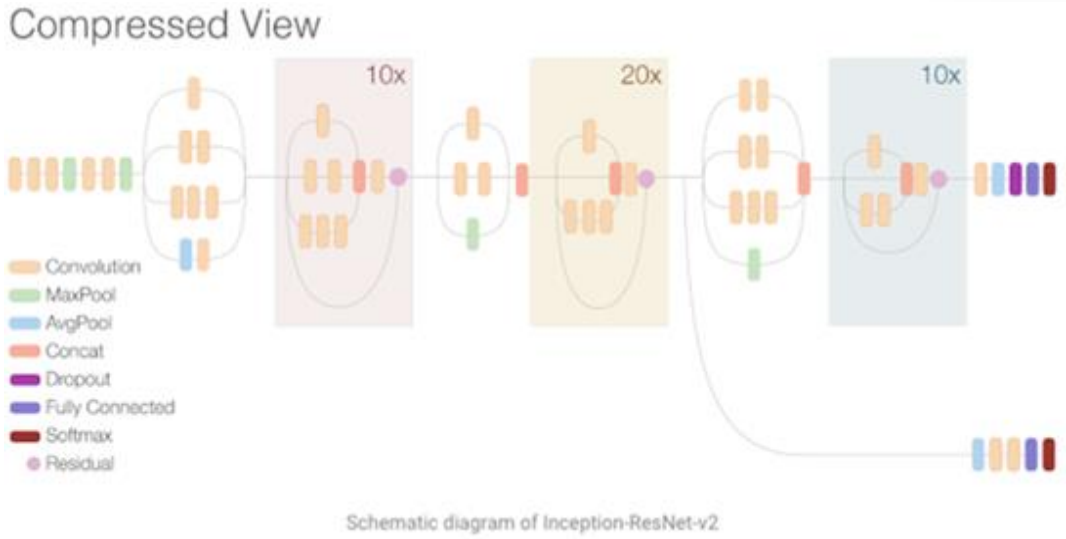
VGG16 Architecture

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_4 (InputLayer)         [(None, 299, 299, 3)]     0
_________________________________________________________________
batch_normalization_204 (Bat (None, 299, 299, 3)       12
_________________________________________________________________
vgg16 (Functional)           (None, 9, 9, 512)         14714688
_________________________________________________________________
conv2d_204 (Conv2D)          (None, 9, 9, 32)          16416
_________________________________________________________________
flatten_1 (Flatten)          (None, 2592)              0
_________________________________________________________________
dropout_2 (Dropout)          (None, 2592)              0
_________________________________________________________________
dense_2 (Dense)              (None, 1024)              2655232
_________________________________________________________________
dropout_3 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_3 (Dense)              (None, 28)                28700
=================================================================
Total params: 17,415,048
Trainable params: 17,415,042
Non-trainable params: 6
_________________________________________________________________
```

```
Epoch 5/10
4528/4528 [==============================] - 1451s 320ms/step - loss: 0.2207 - binary_accuracy: 0.9155 - f1: 0.0659 - val_loss: 0.2176 - val_binary_accuracy: 0.9165 - val_f1: 0.0776

Epoch 00005: val_loss improved from 0.21793 to 0.21763, saving model to best_model2.h5
Epoch 6/10
4528/4528 [==============================] - 1647s 364ms/step - loss: 0.2192 - binary_accuracy: 0.9162 - f1: 0.0655 - val_loss: 0.2176 - val_binary_accuracy: 0.9166 - val_f1: 0.0782

Epoch 00006: val_loss did not improve from 0.21763
Epoch 7/10
4528/4528 [==============================] - 1769s 391ms/step - loss: 0.2199 - binary_accuracy: 0.9158 - f1: 0.0660 - val_loss: 0.2176 - val_binary_accuracy: 0.9166 - val_f1: 0.0783

Epoch 00007: val_loss did not improve from 0.21763
Epoch 8/10
4528/4528 [==============================] - 2197s 485ms/step - loss: 0.2195 - binary_accuracy: 0.9162 - f1: 0.0655 - val_loss: 0.2177 - val_binary_accuracy: 0.9166 - val_f1: 0.0776

Epoch 00008: val_loss did not improve from 0.21763

Epoch 00008: ReduceLROnPlateau reducing learning rate to 1.0000000474974514e-05.
```

1) 모델 간 비교

③ InceptionResNetV2
fine-tune
(R/2+Y/2, G/2+Y/2, B)



Schematic diagram of Inception-ResNet-v2

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 299, 299, 3)]     0
_________________________________________________________________
batch_normalization_203 (Bat (None, 299, 299, 3)       12
_________________________________________________________________
inception_resnet_v2 (Functio (None, 8, 8, 1536)        54336736
_________________________________________________________________
conv2d_203 (Conv2D)          (None, 8, 8, 32)          49184
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0
_________________________________________________________________
dropout (Dropout)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 1024)              2098176
_________________________________________________________________
dropout_1 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 28)                28700
=================================================================
Total params: 56,512,808
Trainable params: 2,176,066
Non-trainable params: 54,336,742
_________________________________________________________________
```

```
Epoch 00005: val_loss improved from 0.14463 to 0.14262, saving model to best_model.h5
Epoch 6/10
4528/4528 [==============================] - 1425s 315ms/step - loss: 0.1422 - binary_accuracy: 0.9502 - f1: 0.1671 - val_loss: 0.1321 - val_binary_accuracy: 0.9534 - val_f1: 0.2381

Epoch 00006: val_loss improved from 0.14262 to 0.13214, saving model to best_model.h5
Epoch 7/10
4528/4528 [==============================] - 1400s 309ms/step - loss: 0.1408 - binary_accuracy: 0.9509 - f1: 0.1678 - val_loss: 0.1452 - val_binary_accuracy: 0.9492 - val_f1: 0.2157

Epoch 00007: val_loss did not improve from 0.13214
Epoch 8/10
4528/4528 [==============================] - 1407s 311ms/step - loss: 0.1467 - binary_accuracy: 0.9489 - f1: 0.1621 - val_loss: 0.2111 - val_binary_accuracy: 0.9269 - val_f1: 0.1030

Epoch 00008: val_loss did not improve from 0.13214
Epoch 9/10
4528/4528 [==============================] - 1440s 318ms/step - loss: 0.1504 - binary_accuracy: 0.9476 - f1: 0.1549 - val_loss: 0.1417 - val_binary_accuracy: 0.9506 - val_f1: 0.2224

Epoch 00009: val_loss did not improve from 0.13214

Epoch 00009: ReduceLROnPlateau reducing learning rate to 0.00010000000474974513.
```

2) 채널 조절(InceptionResNetV2 fine-tune)

① (R/2+Y/2, G/2+Y/2, B)

```
Epoch 00005: val_loss improved from 0.14463 to 0.14262, saving model to best_model.h5
Epoch 6/10
4528/4528 [==============================] - 1425s 315ms/step - loss: 0.1422 - binary_accuracy: 0.9502 - f1: 0.1671 - val_loss: 0.1321 - val_binary_accuracy: 0.9534 - val_f1: 0.2381

Epoch 00006: val_loss improved from 0.14262 to 0.13214, saving model to best_model.h5
Epoch 7/10
4528/4528 [==============================] - 1400s 309ms/step - loss: 0.1408 - binary_accuracy: 0.9509 - f1: 0.1678 - val_loss: 0.1452 - val_binary_accuracy: 0.9492 - val_f1: 0.2157

Epoch 00007: val_loss did not improve from 0.13214
Epoch 8/10
4528/4528 [==============================] - 1407s 311ms/step - loss: 0.1467 - binary_accuracy: 0.9489 - f1: 0.1621 - val_loss: 0.2111 - val_binary_accuracy: 0.9269 - val_f1: 0.1030

Epoch 00008: val_loss did not improve from 0.13214
Epoch 9/10
4528/4528 [==============================] - 1440s 318ms/step - loss: 0.1504 - binary_accuracy: 0.9476 - f1: 0.1549 - val_loss: 0.1417 - val_binary_accuracy: 0.9506 - val_f1: 0.2224

Epoch 00009: val_loss did not improve from 0.13214

Epoch 00009: ReduceLROnPlateau reducing learning rate to 0.000100000000474974513.
```

② (R=0, G, B=0)

```
Epoch 17/20
3019/3019 [==============================] - 810s 268ms/step - loss: 0.1095 - binary_accuracy: 0.9606 - f1: 0.2055 - val_loss: 0.1081 - val_binary_accuracy: 0.9617 - val_f1: 0.3130

Epoch 00017: val_loss did not improve from 0.10740
Epoch 18/20
3019/3019 [==============================] - 753s 250ms/step - loss: 0.1084 - binary_accuracy: 0.9610 - f1: 0.2098 - val_loss: 0.1079 - val_binary_accuracy: 0.9620 - val_f1: 0.3136

Epoch 00018: val_loss did not improve from 0.10740
Epoch 19/20
3019/3019 [==============================] - 745s 247ms/step - loss: 0.1074 - binary_accuracy: 0.9614 - f1: 0.2104 - val_loss: 0.1050 - val_binary_accuracy: 0.9627 - val_f1: 0.3216

Epoch 00019: val_loss improved from 0.10740 to 0.10505, saving model to best_model_sig_g-only.h5
Epoch 20/20
3019/3019 [==============================] - 781s 259ms/step - loss: 0.1065 - binary_accuracy: 0.9618 - f1: 0.2113 - val_loss: 0.1051 - val_binary_accuracy: 0.9627 - val_f1: 0.3190

Epoch 00020: val_loss did not improve from 0.10505
```

2) 채널 조절(InceptionResNetV2 fine-tune)

③ (R,G,B)

```
Epoch 1/10
4528/4528 [==============================] - 14804s 3s/step - loss: 0.1820 - binary_accuracy: 0.9381 - f1: 0.1267 - val_loss: 0.1557 - val_binary_accuracy: 0.9468 - val_f1: 0.1968

Epoch 00001: val_loss improved from inf to 0.15575, saving model to best_model.h5
Epoch 2/10
4528/4528 [==============================] - 14754s 3s/step - loss: 0.1484 - binary_accuracy: 0.9490 - f1: 0.1571 - val_loss: 0.1450 - val_binary_accuracy: 0.9489 - val_f1: 0.2223

Epoch 00002: val_loss improved from 0.15575 to 0.14497, saving model to best_model.h5
Epoch 3/10
4528/4528 [==============================] - 22215s 5s/step - loss: 0.1392 - binary_accuracy: 0.9515 - f1: 0.1692 - val_loss: 0.1296 - val_binary_accuracy: 0.9534 - val_f1: 0.2456

Epoch 00003: val_loss improved from 0.14497 to 0.12958, saving model to best_model.h5
Epoch 4/10
4528/4528 [==============================] - 14368s 3s/step - loss: 0.1349 - binary_accuracy: 0.9527 - f1: 0.1748 - val_loss: 0.1402 - val_binary_accuracy: 0.9490 - val_f1: 0.2389

Epoch 00004: val_loss did not improve from 0.12958
```

④ (B/2+Y/2, R/2+Y/2, G)

```
Epoch 17/20
3019/3019 [==============================] - 969s 321ms/step - loss: 0.1003 - binary_accuracy: 0.9633 - f1: 0.2211 - val_loss: 0.0998 - val_binary_accuracy: 0.9640 - val_f1: 0.3374

Epoch 00017: val_loss did not improve from 0.09979
Epoch 18/20
3019/3019 [==============================] - 1070s 354ms/step - loss: 0.1004 - binary_accuracy: 0.9633 - f1: 0.2196 - val_loss: 0.0996 - val_binary_accuracy: 0.9641 - val_f1: 0.3363

Epoch 00018: val_loss improved from 0.09979 to 0.09956, saving model to best_model_sig_rgby.h5
Epoch 19/20
3019/3019 [==============================] - 855s 283ms/step - loss: 0.0989 - binary_accuracy: 0.9640 - f1: 0.2223 - val_loss: 0.0979 - val_binary_accuracy: 0.9652 - val_f1: 0.3459

Epoch 00019: val_loss improved from 0.09956 to 0.09793, saving model to best_model_sig_rgby.h5
Epoch 20/20
3019/3019 [==============================] - 982s 325ms/step - loss: 0.0981 - binary_accuracy: 0.9640 - f1: 0.2225 - val_loss: 0.0975 - val_binary_accuracy: 0.9648 - val_f1: 0.3409

Epoch 00020: val_loss improved from 0.09793 to 0.09753, saving model to best_model_sig_rgby.h5
```

1) 최종 모델 submission

최종모델:
데이터 증강 O, InceptionResNetV2 fine-tune,
(R,G,B,Y)

Submission 결과

Name	Submitted	Wait time	Execution time	Score
submit_best_model_rgby.csv	just now	1 seconds	0 seconds	0.33828

Complete

Jump to your position on the leaderboard

1465	▲ 50	C_heng		0.33888	3	3y
1466	▼ 34	LarsHulstaert		0.33878	9	3y
1467	▼ 20	masih4		0.33829	6	3y
1468	▲ 29	miyashita		0.33807	1	3y
1469	▲ 35	Ant0n		0.33797	8	3y
1470	▲ 18	newber than IG		0.33765	6	3y
1471	▼ 76	Krishnadasar Sudheer Kumar		0.33719	183	3y

leaderboard:
1468/2160

1) 최종 모델 submission

최종모델:
데이터 증강 O, InceptionResNetV2 fine-tune,
(R,G,B,Y)

비록 리더보드의 등수는 높진 않지만

① 1등의 f1 score가 0.59369라는 점에서 **많은 차이를 보이지 않음.**

② 최종모델의 경우 시간관계 상 20epoch에서 멈추었지만, **val loss가 계속 줄고 있어서** 더 훈련할 경우, 더 좋은 성능을 보였을 것.

2) Discussion

① Activation function에 sigmoid 사용한 이유

Multi Class: 하나의 이미지에 하나의 객체만 존재, 각 객체는 2개 이상의 카테고리로 분류

Softmax 사용

Mutli Label: 하나의 이미지에 여러 객체가 존재. 각 객체는 2개 이상의 카테고리로 분류

Sigmoid 사용

Q&A