

# **2021 SMARCLE 겨울방학 스터디 총결산**

## **HOW TO Deal With Pretrained Model**

# Contents



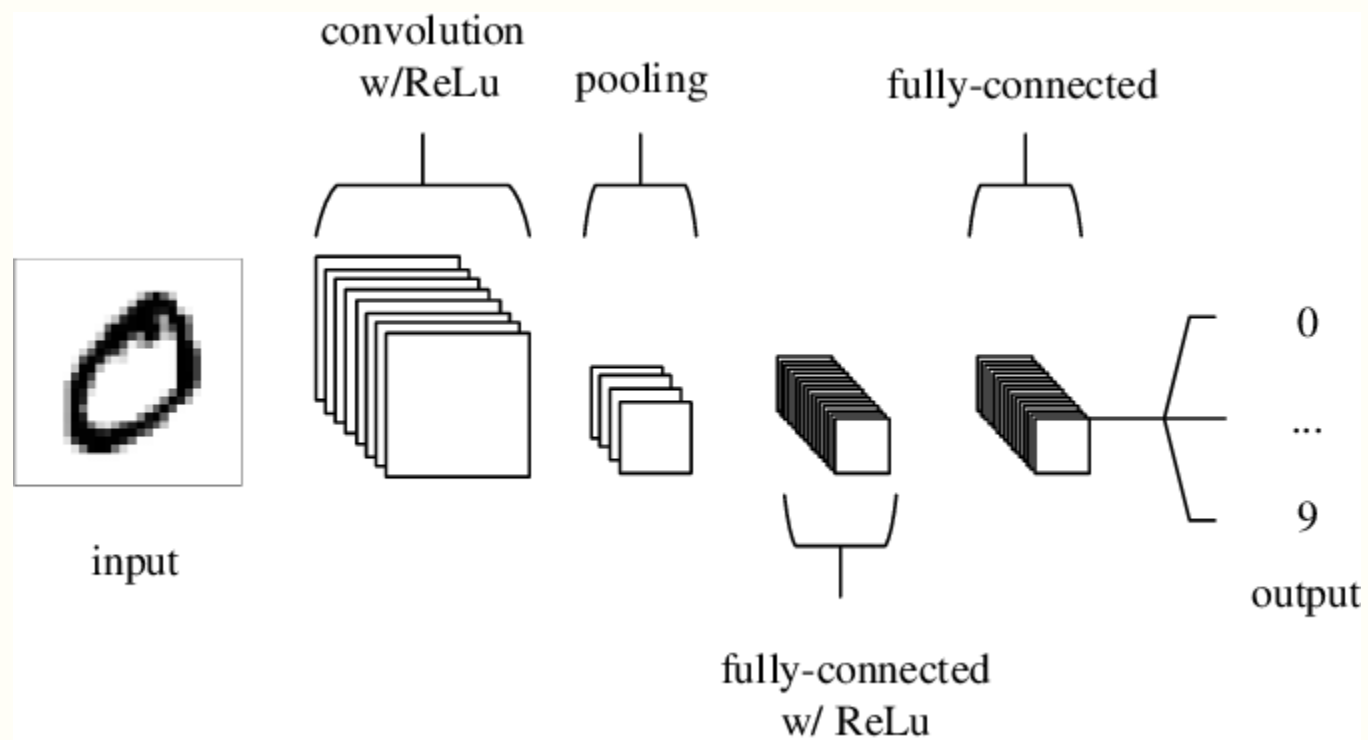
그저 Deeeeeeeep 한게 좋은 것인가?

다양한 접근

Transfer Learning With Pretrained Model

닭 잡는데 소 잡는 칼 쓰지 말자

# 그저 Deeeeeeeep 한게 좋은 것인가?



# 그저 Deeeeeeeep 한게 좋은 것인가?

@ 9회차 모뎀 9회차 실습예시(이은지).ipynb

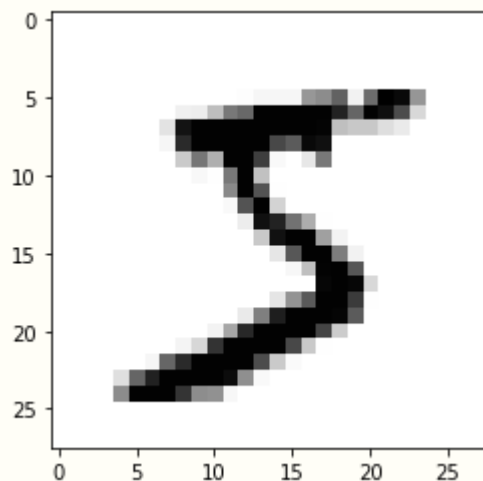
```
Epoch 15/5000  
118/118 [=====] - 3s 22ms/step - loss: 0.0062 - accuracy: 0.9982 - val_loss: 0.0307 - val_accuracy: 0.9901  
Epoch 16/5000  
118/118 [=====] - 3s 22ms/step - loss: 0.0082 - accuracy: 0.9974 - val_loss: 0.0314 - val_accuracy: 0.9912  
Epoch 00016: early stopping
```

Out[ ]: <tensorflow.python.keras.callbacks.History at 0x7fbc344d94e0>

```
In [ ]: model.evaluate(x_test,y_test)[1]
```

```
313/313 [=====] - 1s 2ms/step - loss: 0.0314 - accuracy: 0.9912
```

Out[ ]: 0.9911999702453613



# 그저 Deeeeeeeep 한게 좋은 것인가?

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 26, 26, 32)	320
-----		
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
-----		
flatten (Flatten)	(None, 5408)	0
-----		
dense (Dense)	(None, 10)	54090
=====		
Total params: 54,410		
Trainable params: 54,410		
Non-trainable params: 0		

```
Epoch 1/5
1875/1875 [=====] - 3s 2ms/step - loss: 0.6184 - accuracy: 0.84
Epoch 2/5
1875/1875 [=====] - 3s 2ms/step - loss: 0.2654 - accuracy: 0.92
Epoch 3/5
1875/1875 [=====] - 3s 2ms/step - loss: 0.2152 - accuracy: 0.93
Epoch 4/5
1875/1875 [=====] - 3s 2ms/step - loss: 0.1801 - accuracy: 0.94
Epoch 5/5
1875/1875 [=====] - 3s 2ms/step - loss: 0.1536 - accuracy: 0.95
313/313 [=====] - 0s 2ms/step - loss: 0.1376 - accuracy: 0.96
[0.1376066356897354, 0.9620000123977661]
```

# 그저 Deeeeeeeep 한게 좋은 것인가?

Model: "sequential\_11"

Layer (type)	Output Shape	Param #
conv2d_32 (Conv2D)	(None, 26, 26, 32)	320
conv2d_33 (Conv2D)	(None, 24, 24, 32)	9248
max_pooling2d_29 (MaxPooling)	(None, 12, 12, 32)	0
dropout_8 (Dropout)	(None, 12, 12, 32)	0
conv2d_34 (Conv2D)	(None, 10, 10, 64)	18496
max_pooling2d_30 (MaxPooling)	(None, 5, 5, 64)	0
conv2d_35 (Conv2D)	(None, 3, 3, 128)	73856
max_pooling2d_31 (MaxPooling)	(None, 1, 1, 128)	0
flatten_11 (Flatten)	(None, 128)	0
dense_11 (Dense)	(None, 10)	1290
Total params: 103,210		
Trainable params: 103,210		
Non-trainable params: 0		

```
Epoch 1/5
1875/1875 [=====] - 6s 3ms/step - loss: 0.4507 - accuracy: 0.86
Epoch 2/5
1875/1875 [=====] - 6s 3ms/step - loss: 0.1364 - accuracy: 0.95
Epoch 3/5
1875/1875 [=====] - 7s 4ms/step - loss: 0.0991 - accuracy: 0.97
Epoch 4/5
1875/1875 [=====] - 7s 4ms/step - loss: 0.0802 - accuracy: 0.97
Epoch 5/5
1875/1875 [=====] - 7s 4ms/step - loss: 0.0677 - accuracy: 0.97
313/313 [=====] - 1s 2ms/step - loss: 0.0554 - accuracy: 0.98
[0.05540342256426811, 0.9814000129699707]
```

# 그저 Deeeeeeeep 한게 좋은 것인가?

cifar 10 dataset

airplane



automobile



bird



cat



deer



dog



frog



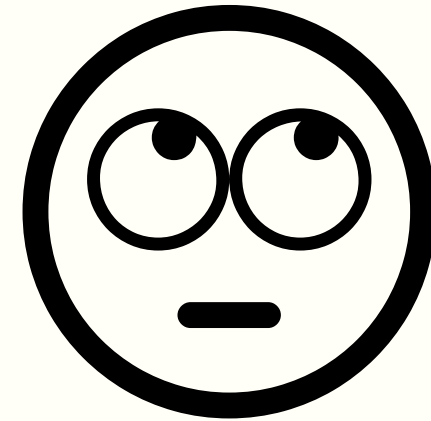
horse



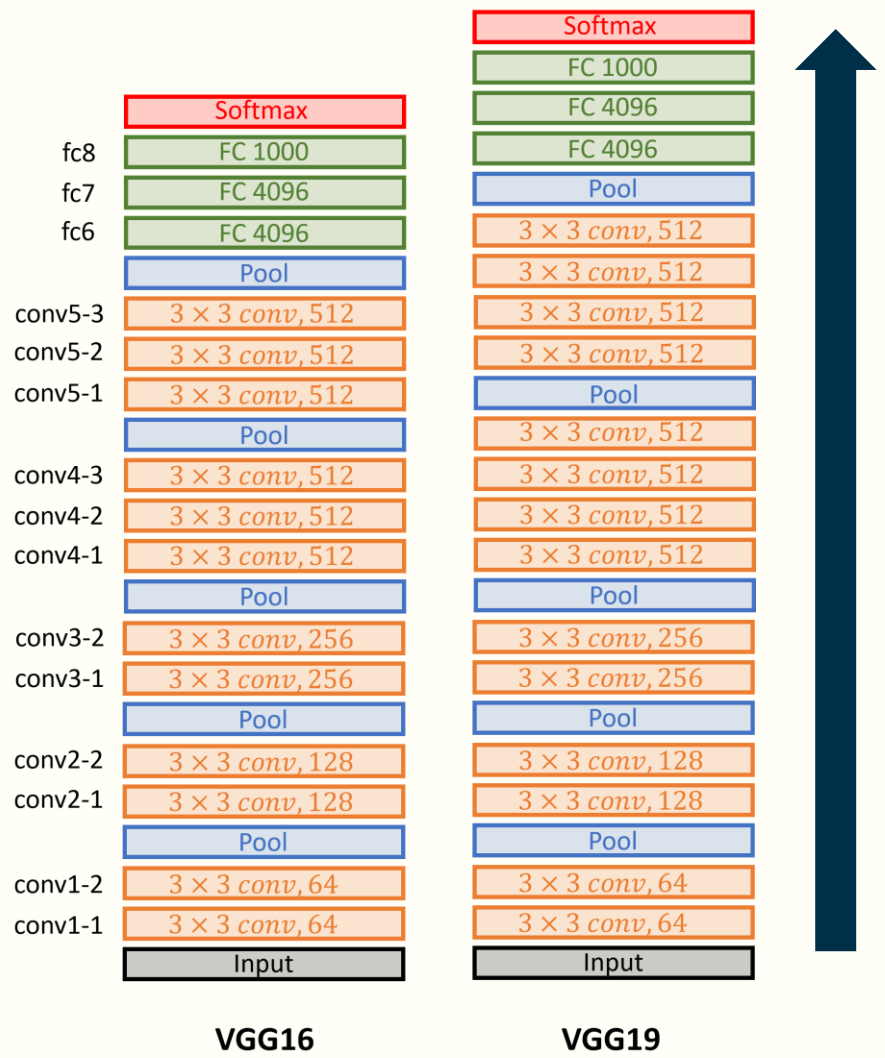
ship



truck

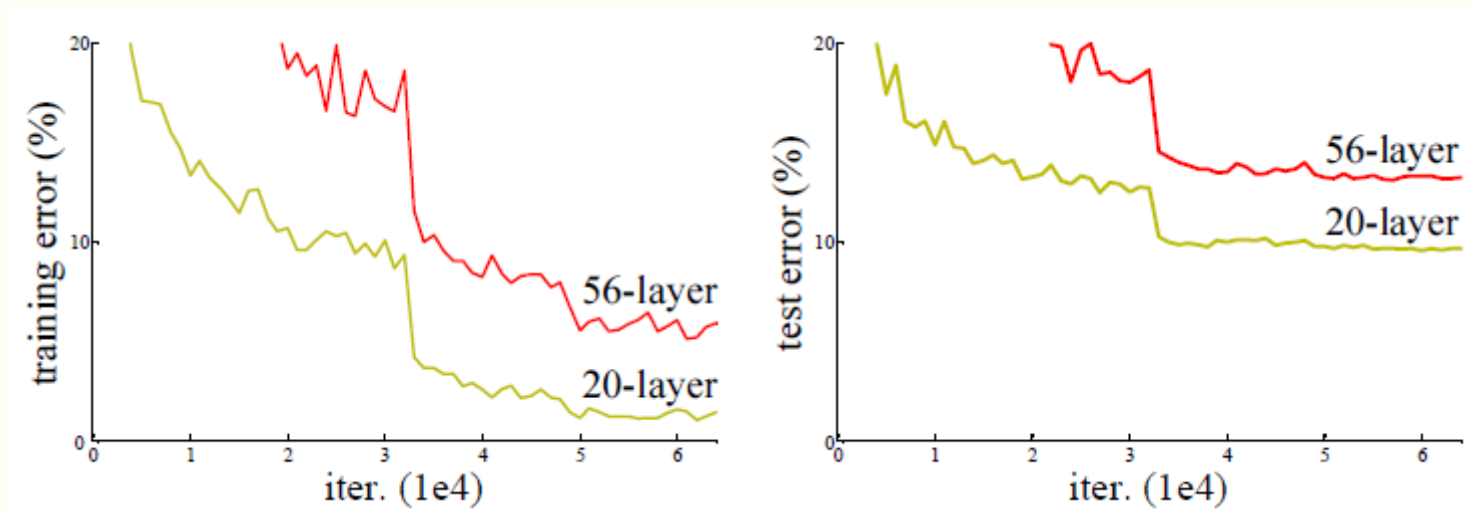


# 그저 Deeeeeeeep 한게 좋은 것인가?



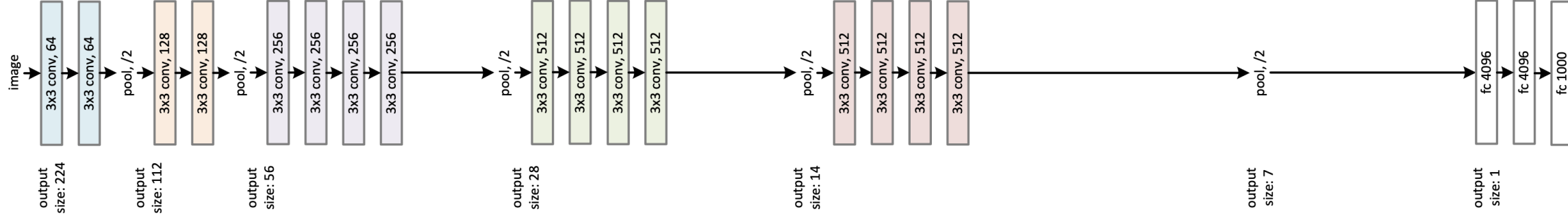


# 그저 Deeeeeeep 한게 좋은 것인가?

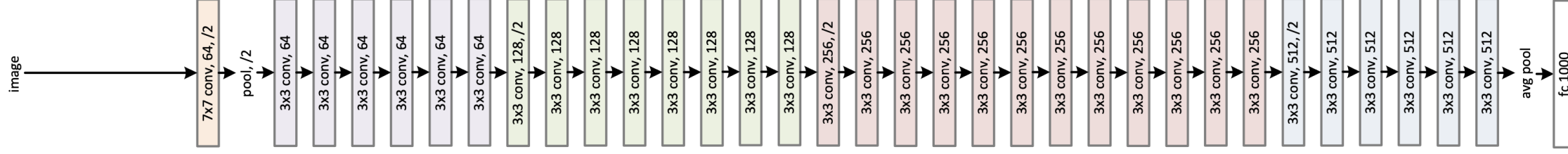


# 다양한 접근

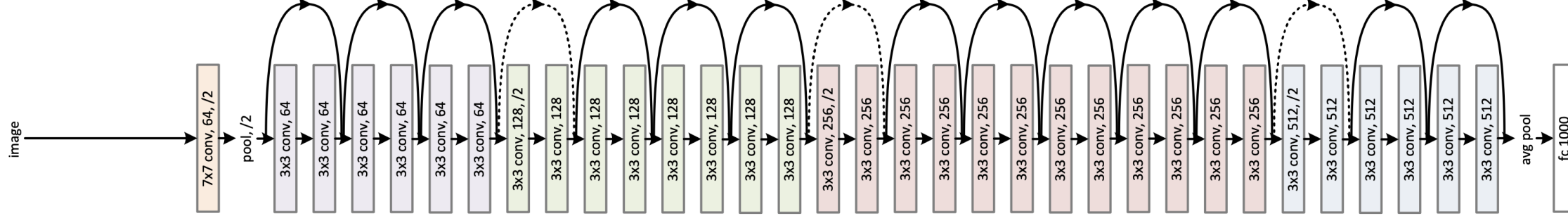
VGG-19



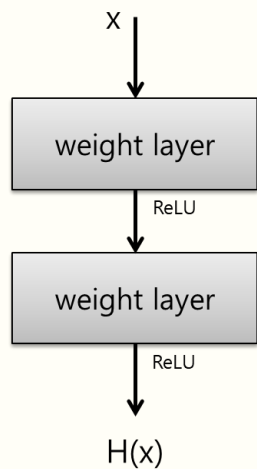
34-layer plain



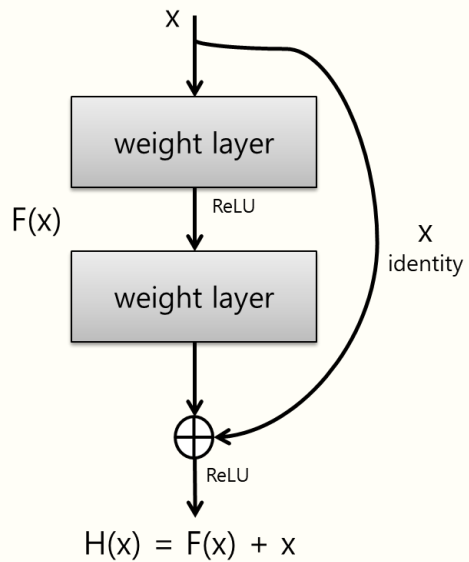
34-layer residual



# 다양한 접근

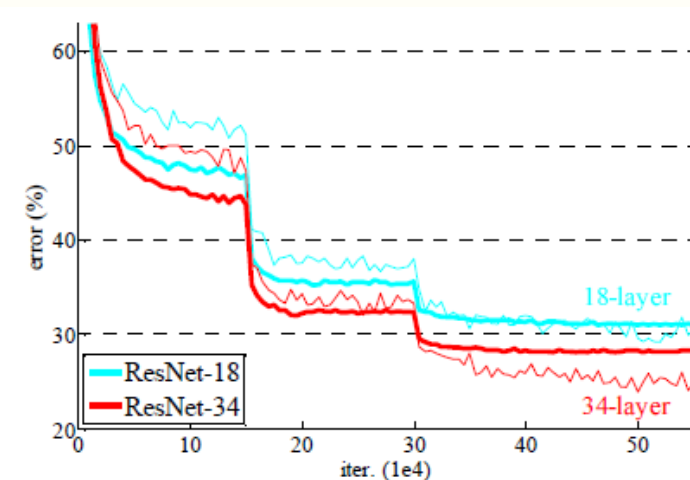
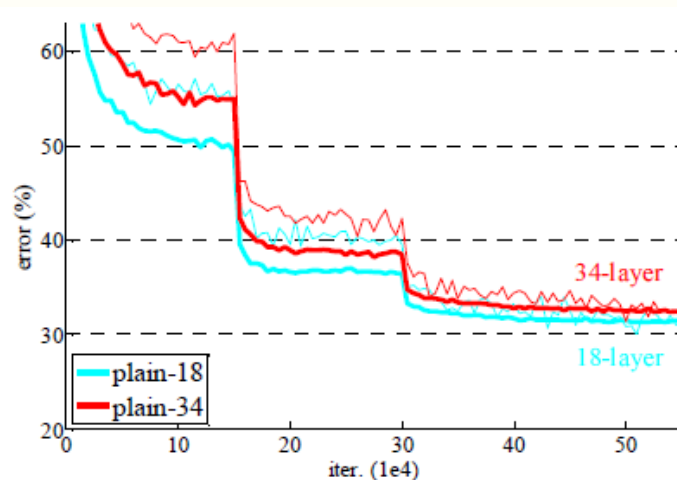


기존 방식



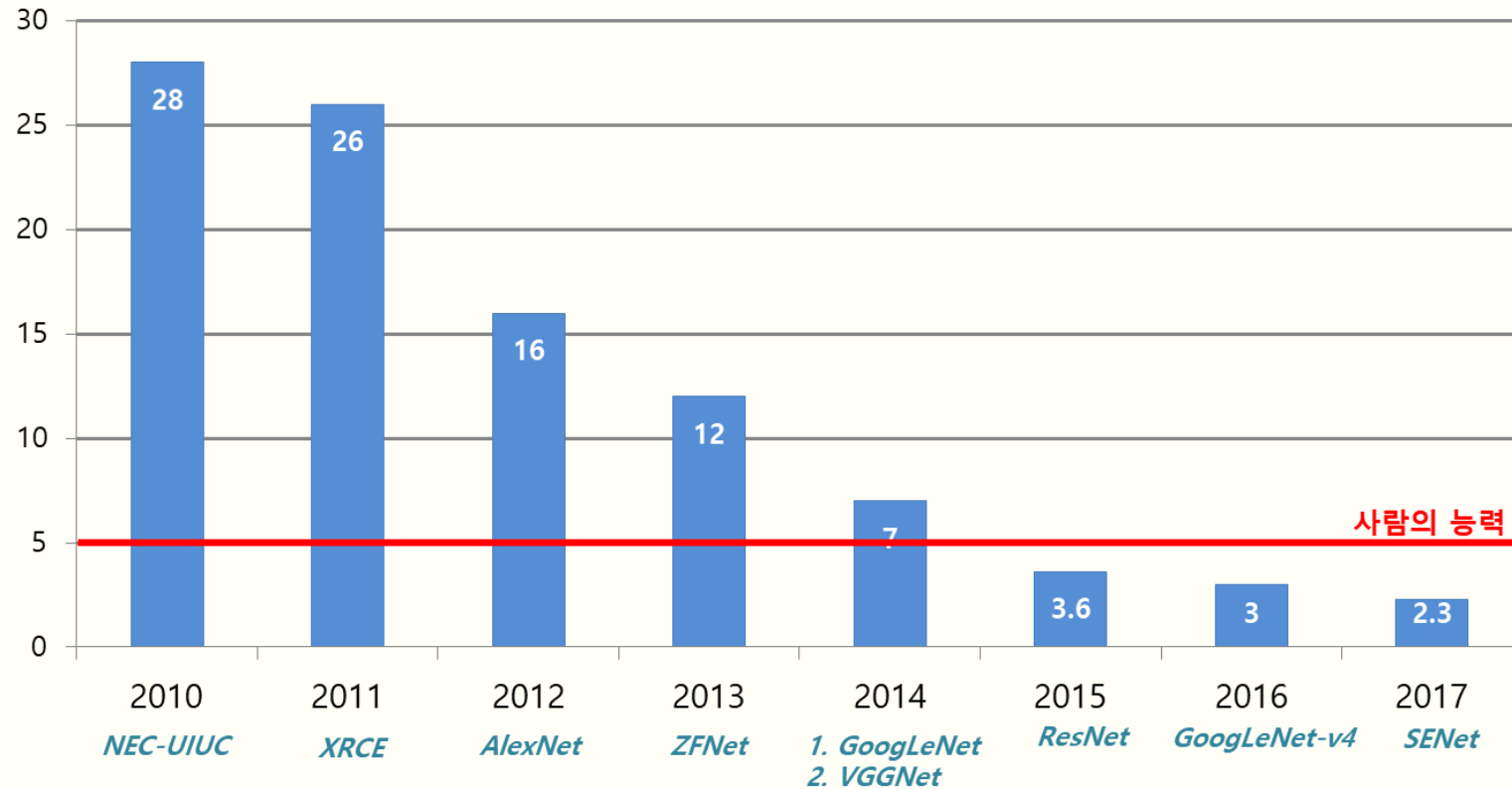
Residual block

2015 ILSVRC: ImageNet Large Scale Visual Recognition Competition



# 다양한 접근

우승 알고리즘의 분류 에러율(%)

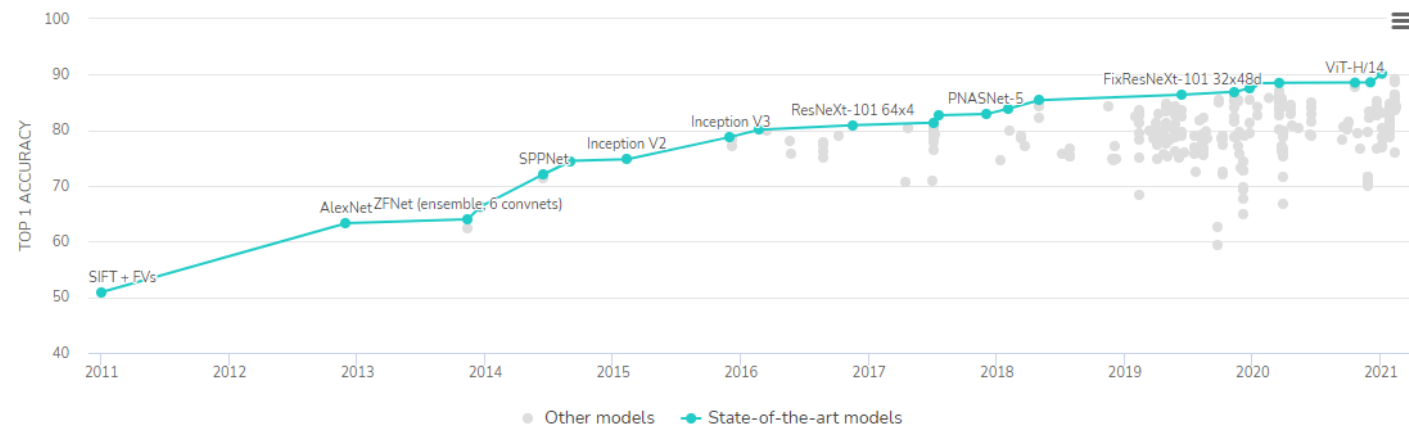


# 다양한 접근

## Image Classification on ImageNet

Leaderboard

Dataset



View Top 1 Accuracy

All models

Edit

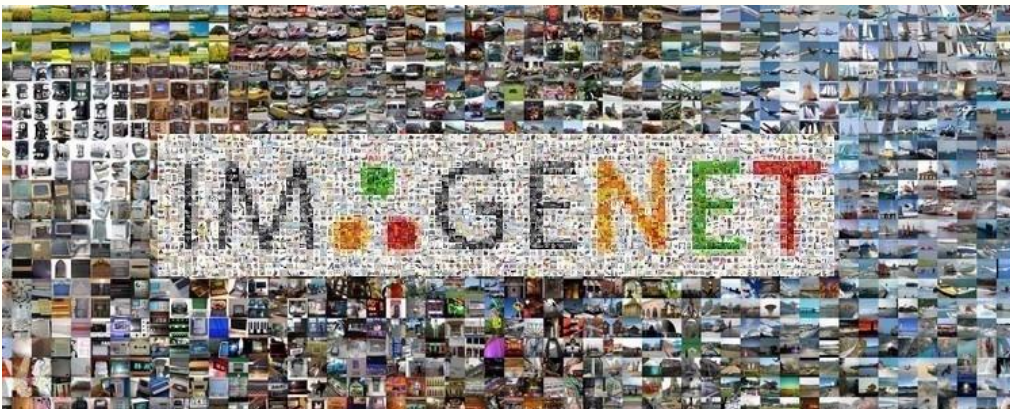
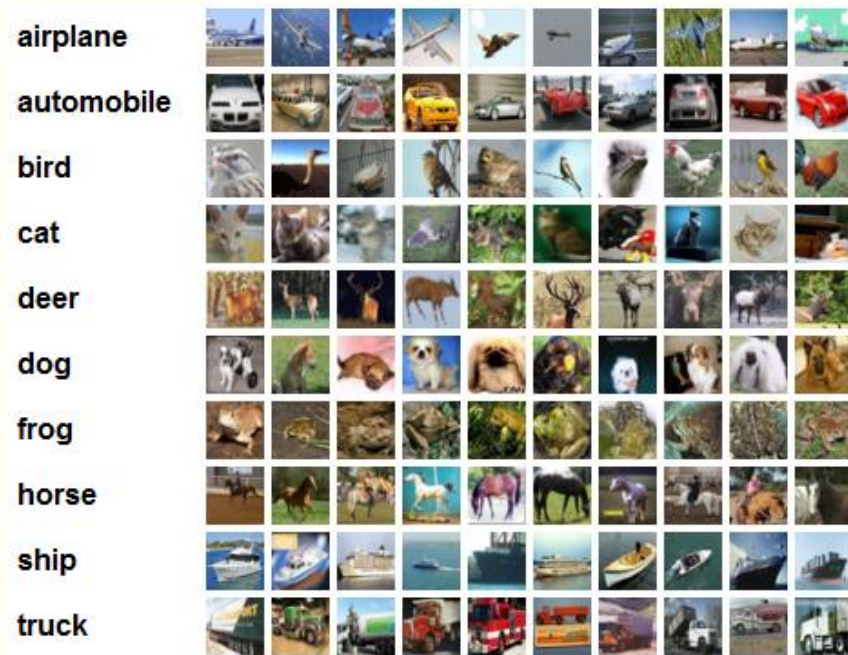
RANK	MODEL	TOP 1 ACCURACY	TOP 5 ACCURACY	NUMBER OF PARAMS	EXTRA TRAINING DATA	PAPER	CODE	RESULT	YEAR
1	Meta Pseudo Labels (EfficientNet-L2)	90.2%	98.8%	480M	✓	Meta Pseudo Labels	<a href="#">GitHub</a>	<a href="#">Result</a>	2021
2	Meta Pseudo Labels (EfficientNet-B6-Wide)	90%	98.7%	390M	✓	Meta Pseudo Labels	<a href="#">GitHub</a>	<a href="#">Result</a>	2021
3	NFNet-F4+	89.2%		527M	✓	High-Performance Large-Scale Image Recognition Without Normalization	<a href="#">GitHub</a>	<a href="#">Result</a>	2021

# Transfer Learning With Pretrained Model





# Transfer Learning With Pretrained Model



# Transfer Learning With Pretrained Model



KNITR©T





# Transfer Learning With Pretrained Model

- tf.image
- tf.initializers
- tf.io
- ▼ tf.keras
  - Overview
  - Input
  - Model
  - Sequential
  - activations
  - ▼ applications
    - Overview
    - DenseNet121
    - DenseNet169
    - DenseNet201
    - InceptionResNetV2
    - InceptionV3
    - MobileNet
    - MobileNetV2
    - NASNetLarge
    - NASNetMobile
    - ResNet101
    - ResNet101V2
    - ResNet152
    - ResNet152V2
    - ResNet50
    - ResNet50V2
    - VGG16
    - VGG19
    - Xception

## Modules

`densenet` module: DenseNet models for Keras.

`efficientnet` module: EfficientNet models for Keras.

`imagenet_utils` module: Utilities for ImageNet data preprocessing & prediction decoding.

`inception_resnet_v2` module: Inception-ResNet V2 model for Keras.

`inception_v3` module: Inception V3 model for Keras.

`mobilenet` module: MobileNet v1 models for Keras.

`mobilenet_v2` module: MobileNet v2 models for Keras.

`mobilenet_v3` module: MobileNet v3 models for Keras.

`nasnet` module: NASNet-A models for Keras.

`resnet` module: ResNet models for Keras.

`resnet50` module: Public API for tf.keras.applications.resnet50 namespace.

`resnet_v2` module: ResNet v2 models for Keras.

`vgg16` module: VGG16 model for Keras.

`vgg19` module: VGG19 model for Keras.

`xception` module: Xception V1 model for Keras.

# Transfer Learning With Pretrained Model

tf.keras.applications.VGG16



TensorFlow 1 version



View source on GitHub

Instantiates the VGG16 model.



View aliases

```
tf.keras.applications.VGG16(  
    include_top=True, weights='imagenet', input_tensor=None,  
    input_shape=None, pooling=None, classes=1000,  
    classifier_activation='softmax'  
)
```



# Transfer Learning With Pretrained Model

tf.keras.applications.VGG16



TensorFlow 1 version



View source on GitHub

Instantiates the VGG16 model.



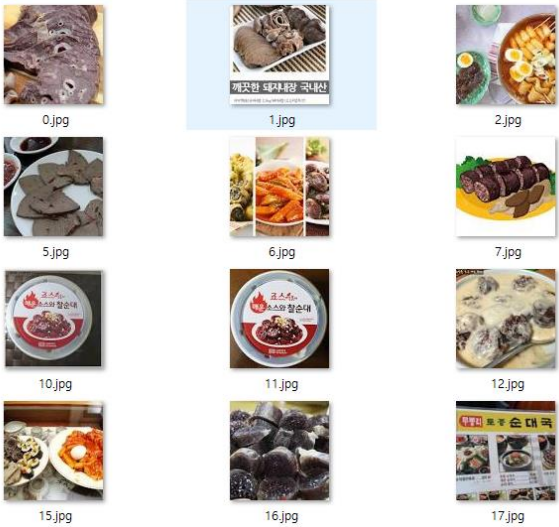
View aliases

```
tf.keras.applications.VGG16(  
    include_top=True, weights='imagenet', input_tensor=None,  
    input_shape=None, pooling=None, classes=1000,  
    classifier_activation='softmax'  
)
```



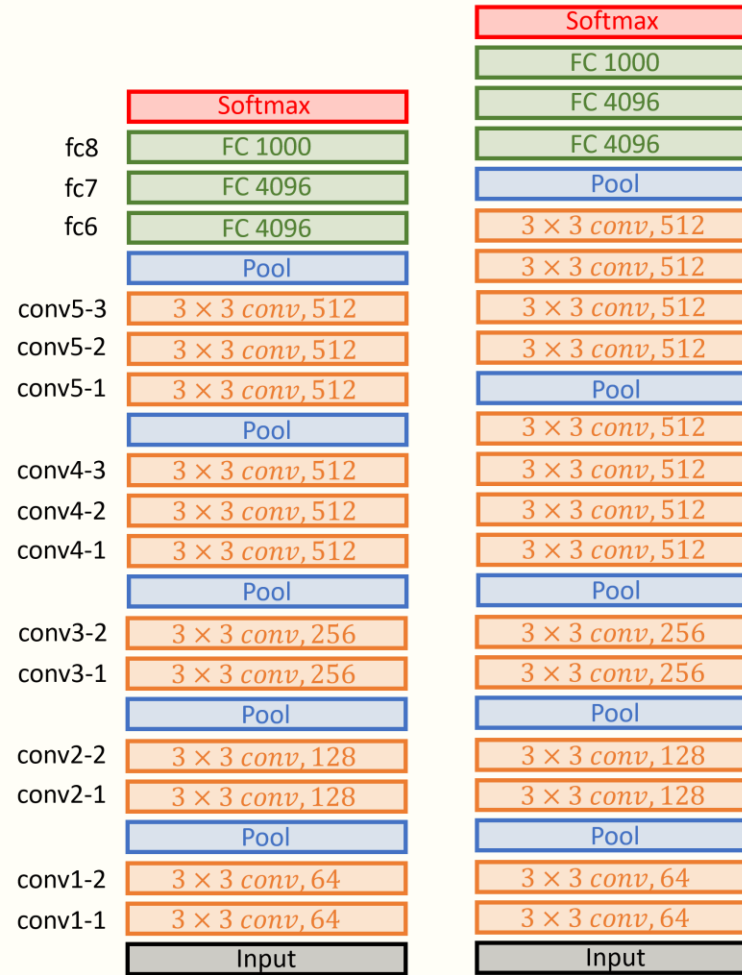
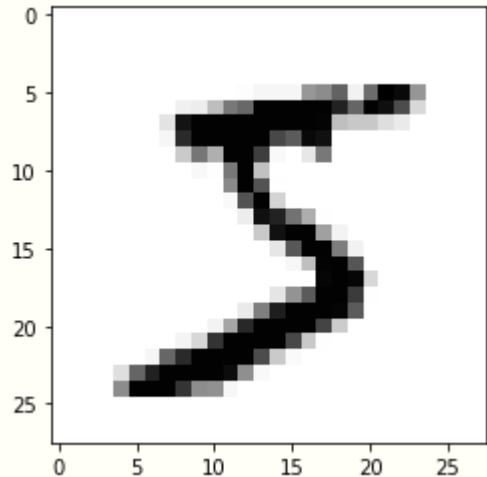
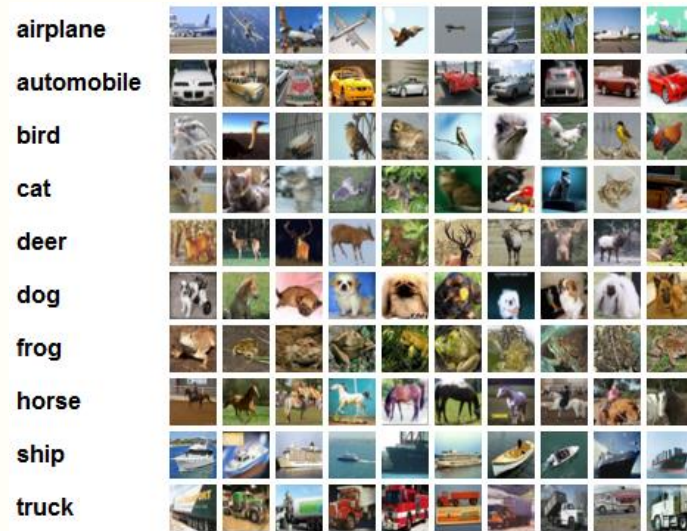
# Transfer Learning With Pretrained Model

@ crawling code by 김찬영



# Transfer Learning With Pretrained Model

# 닭 잡는데 소 잡는 칼 쓰지 말자



VGG16

VGG19

# QnA