

통계적기계학습 과제3 Team project

# ResNet, VGG, EfficientNet 기반 분류 성능 비교: 데이터 크기와 증강의 중요성

팀명 : Statronics

전자공학과 202393158 박수연  
통계학과 202482120 박소정  
통계학과 202482123 이종철

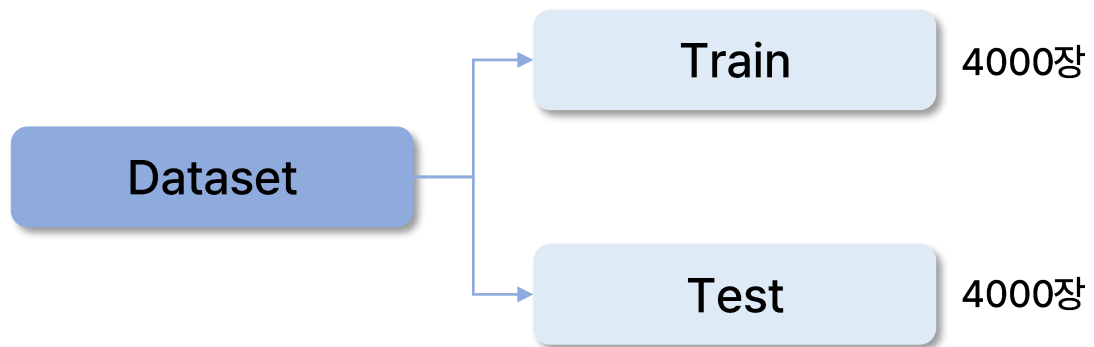


부산대학교  
PUSAN NATIONAL UNIVERSITY

# 분석 주제: 이미지 Classification

- 목적 : 10가지 종류를 가진 이미지 데이터 셋 분류하기
- Class : airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck

[ Dataset 구성 ]



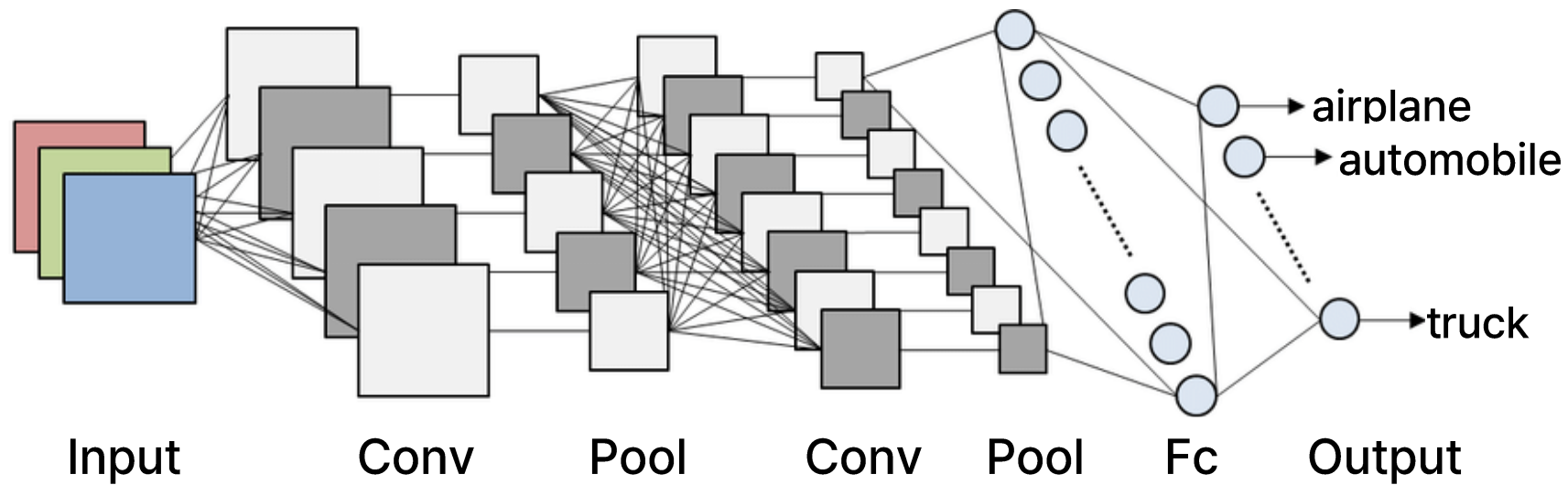
[ Class 종류 ]



# Deep Learning 분류 모델

## CNN

- CNN(Convolutional Neural Networks) : convolutional layer과 pooling layer을 통해 이미지의 특징을 추출한 후 fully connected layer을 통해 분류함



## 목표1.

# CNN 기반 모델 간의 성능 차이와 특징 분석

- 대표적인 CNN 모델을 비교하여 구조적 특징이 성능에 미치는 영향을 분석하고자 함

단순하면서  
깊은 네트워크

VGG  
(2014)

잔차 연결로  
깊이 한계 극복

ResNet  
(2015)

효율성과  
성능의 균형

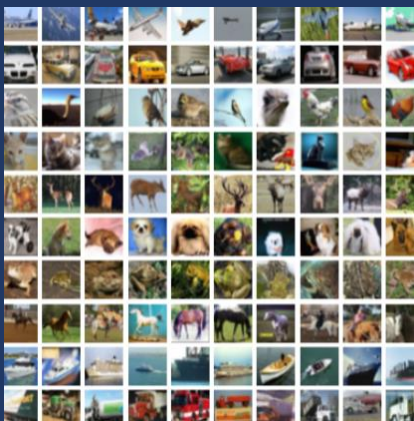
EfficientNet  
(2019)

## 목표2.

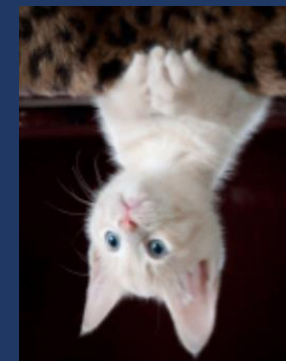
# 데이터 크기와 증강이 모델 성능에 미치는 영향 분석

- 데이터가 많아질 시 모델이 다양한 패턴과 특성을 학습할 기회가 증가해 일반화 성능이 향상됨
- 데이터 증강은 모델이 노이즈, 회전, 크기 변화, 밝기 변화 등 다양한 환경에서 일관된 성능을 발휘하도록 도와줌

CIFAR-10 데이터 사용

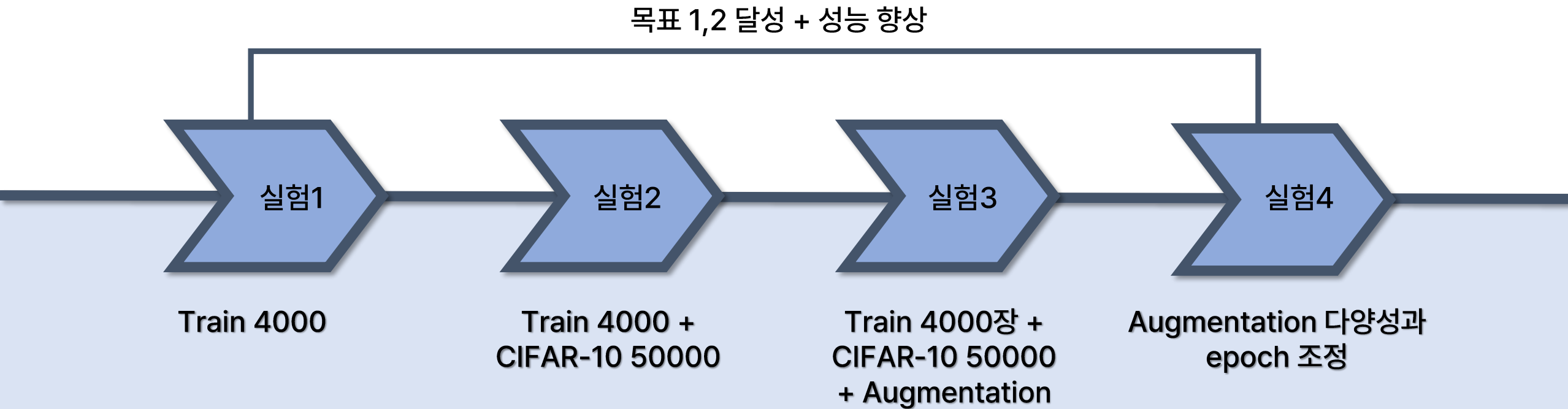


여러가지 데이터 증강법 사용



# 본론

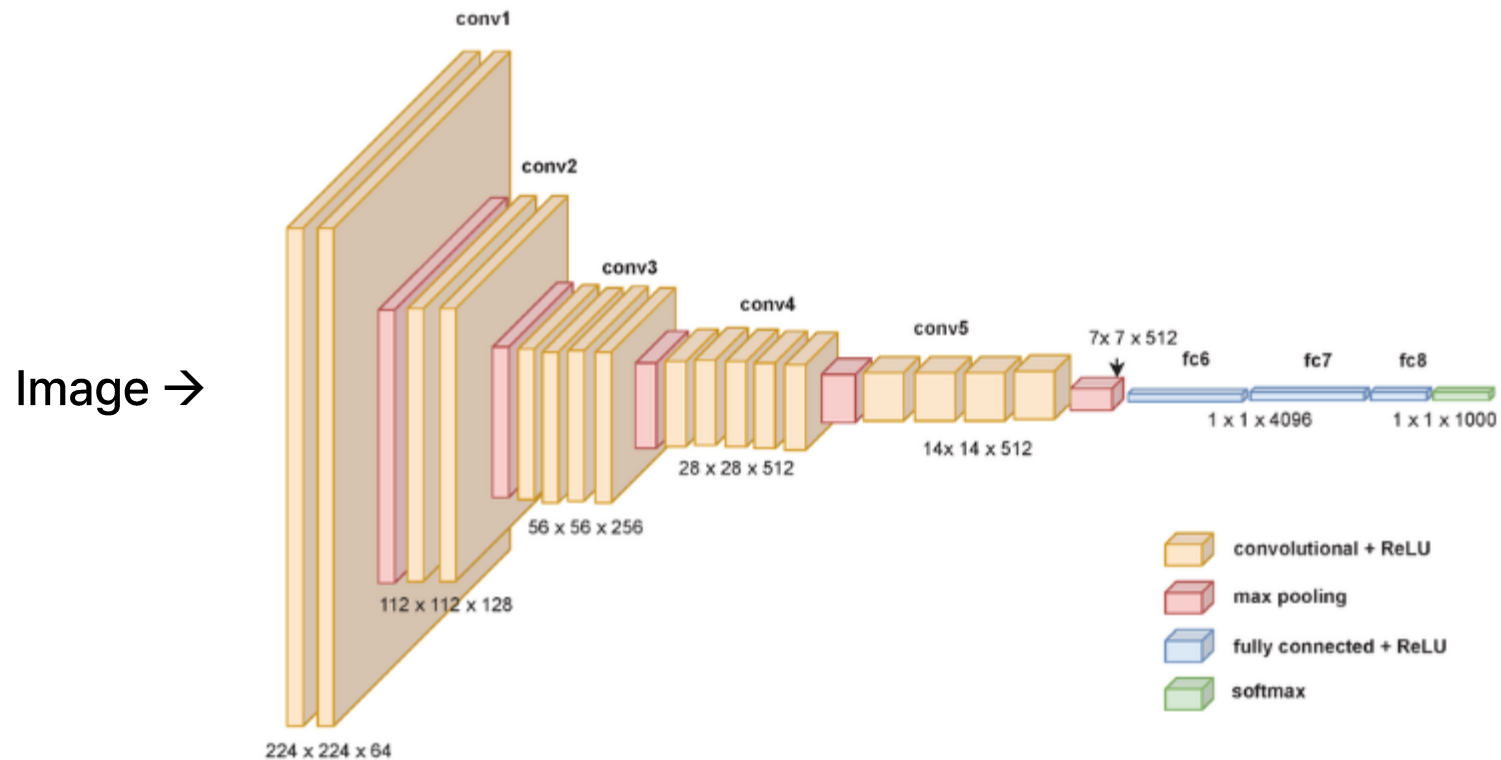
# 목표 학습과 성능 향상을 위한 Process



# 단순하면서 깊은 네트워크

## VGG19

- 일관된 구조로 구성된 매우 깊은 네트워크



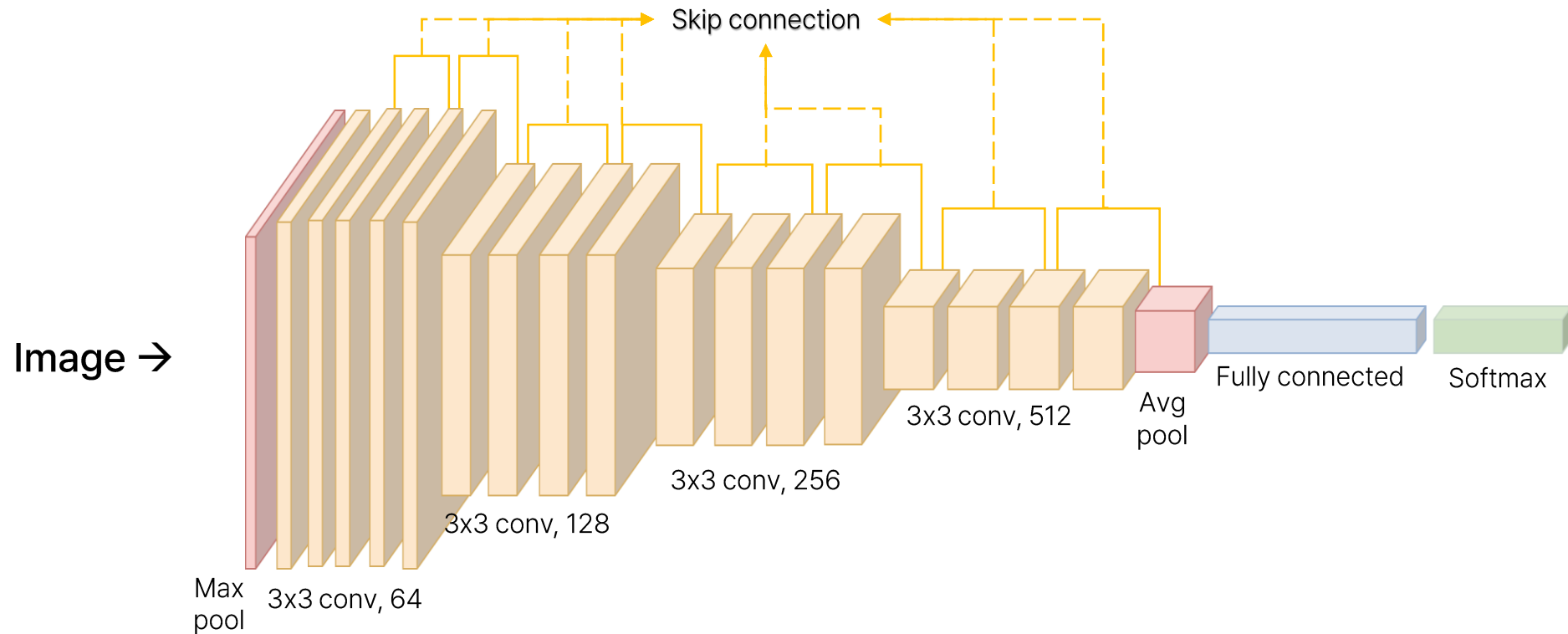
[VGG19 구조]



# 잔차 연결로 깊이 한계 극복

## ResNet18

- 잔차 연결로 깊은 신경망의 성능 열화 문제 완화

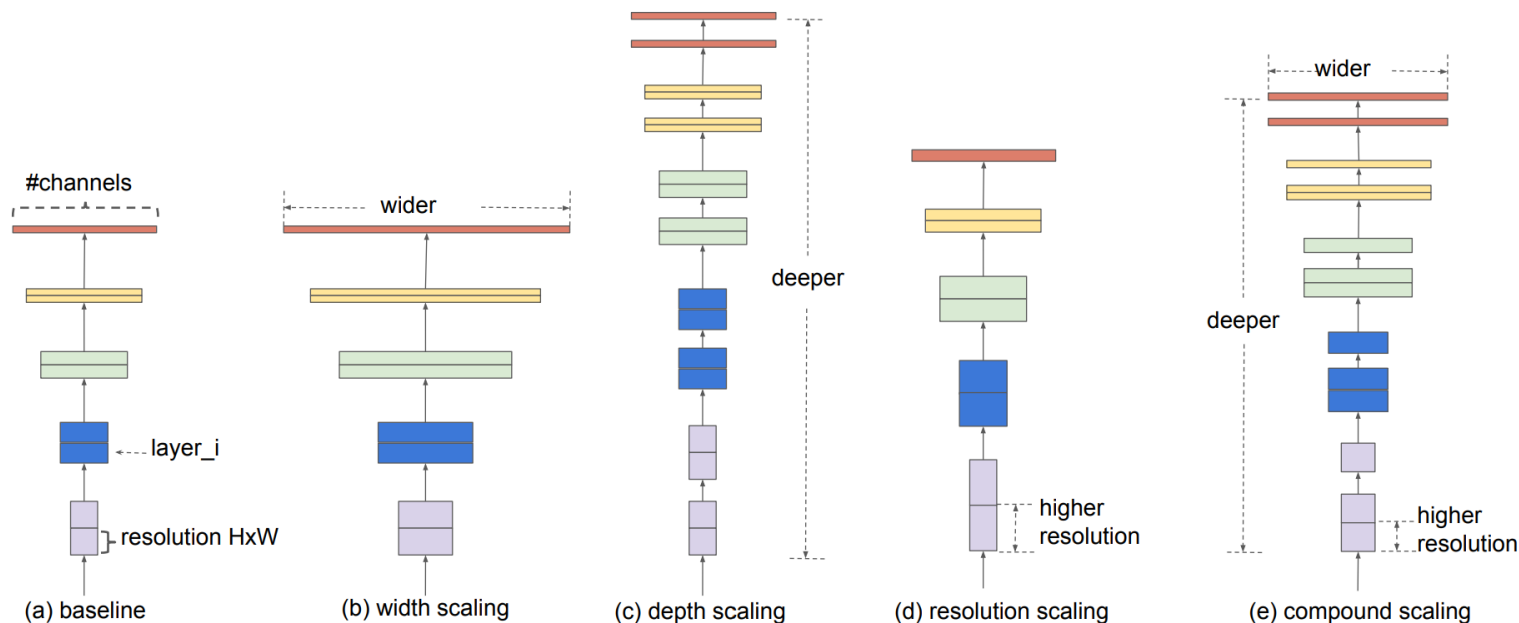


[ResNet18 구조]

# 효율성과 성능의 균형

## EfficientNet

- 모델의 깊이, 너비, 해상도를 균형 있게 확장하는 컴파운드 스케일링을 활용한 모델

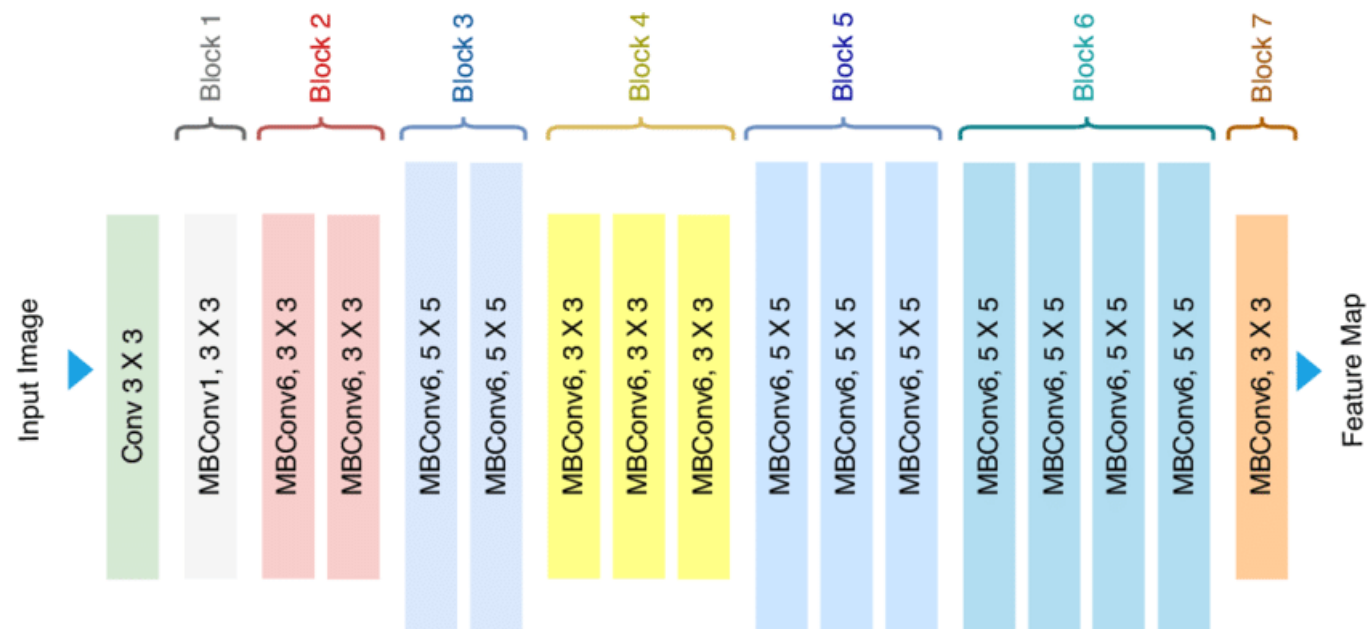


**Figure 2. Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

# 효율성과 성능의 균형

## EfficientNet

- 모델의 깊이, 너비, 해상도를 균형 있게 확장하는 컴파운드 스케일링을 활용한 모델

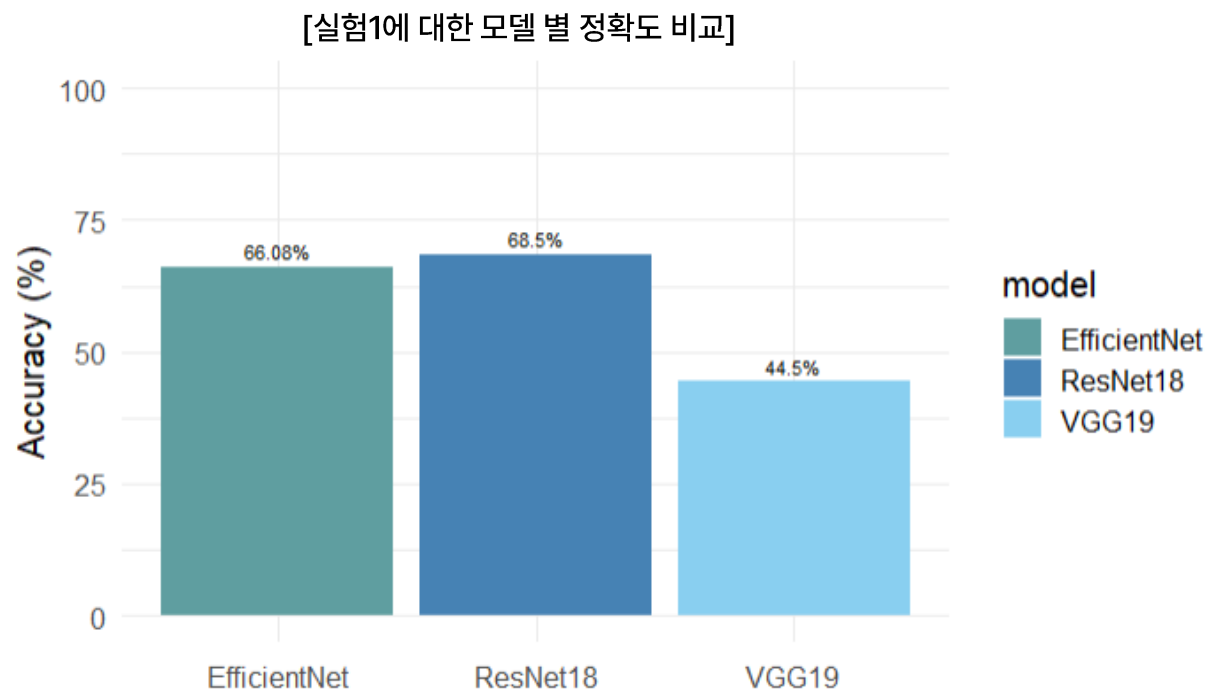


[EfficientNet 구조]

# 실험1.

## Train(4000)

- 실험 목적 : CNN 기반 모델 간의 비교
- 기본 설정 : batch size = 32, epoch = 30, optimizer = Adam, validation set = 20% 사용

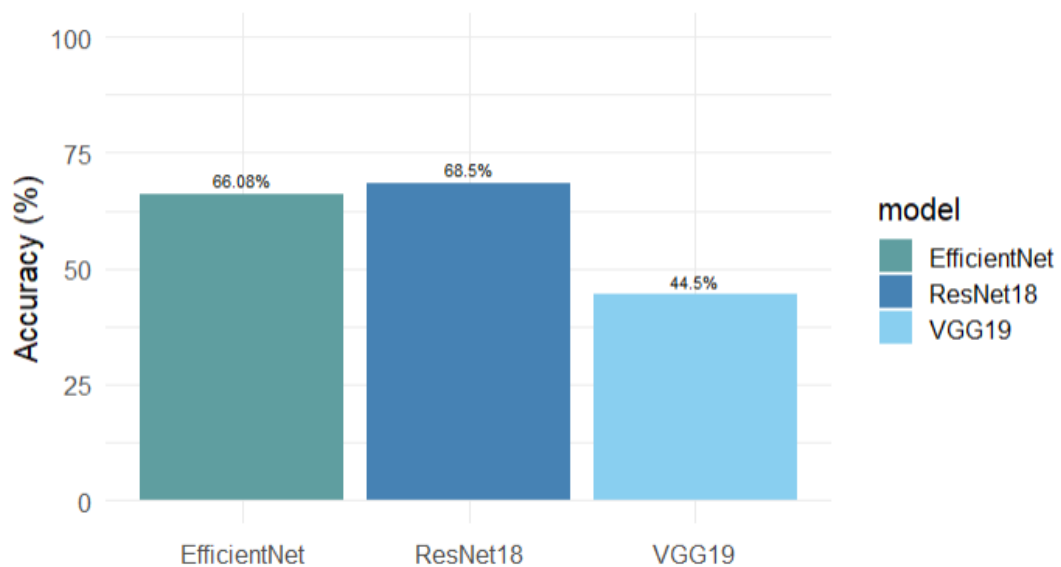


## 실험2.

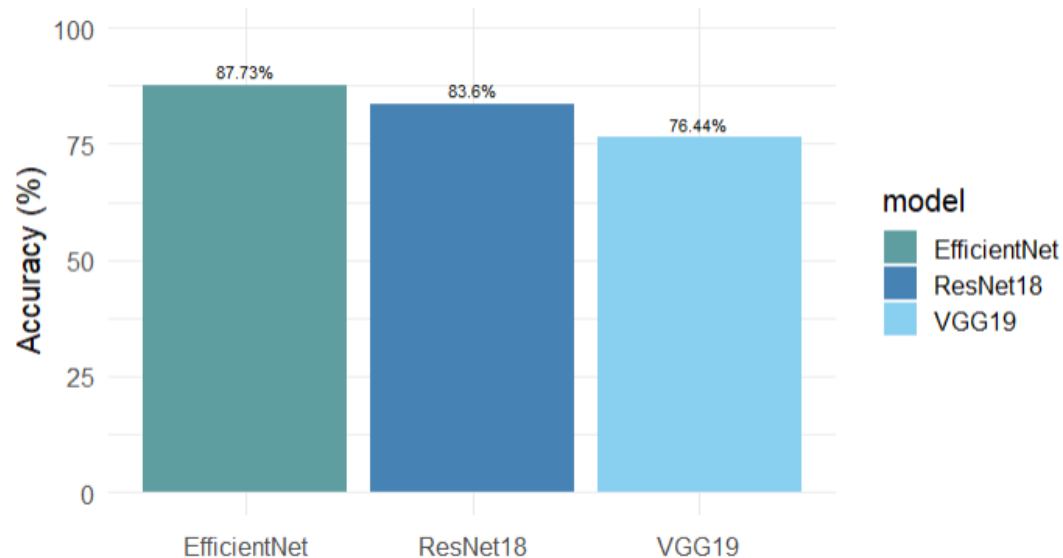
# Train(4000) + CIFAR-10(50000)

- 실험 목적 : 학습 데이터 증가에 따른 성능 변화 비교
- 기본 설정 : batch size = 32, epoch = 30, optimizer = Adam, validation set = 20% 사용

[실험1에 대한 모델 별 정확도 비교]



[실험2에 대한 모델 별 정확도 비교]

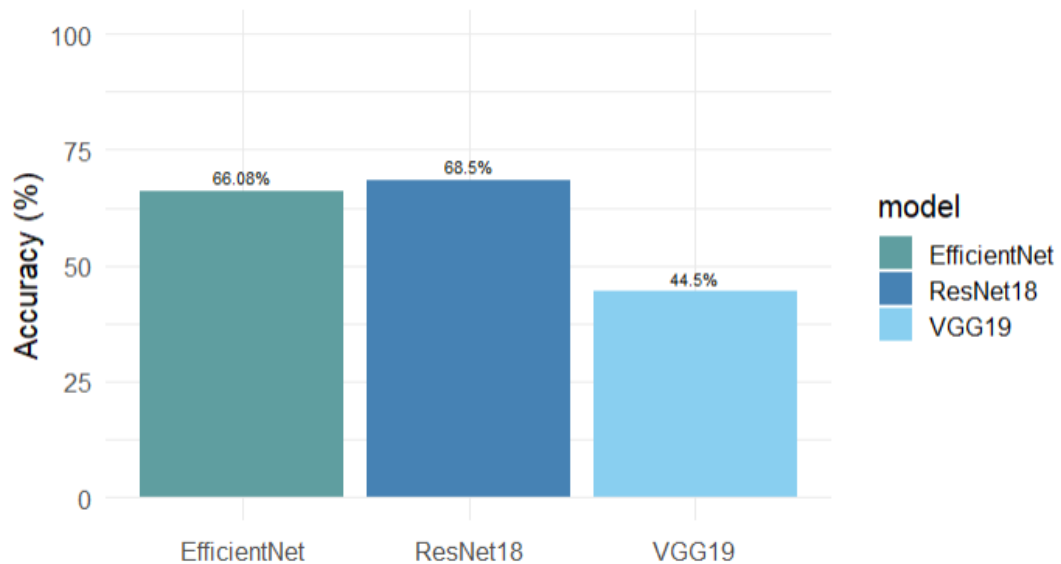


## 실험2.

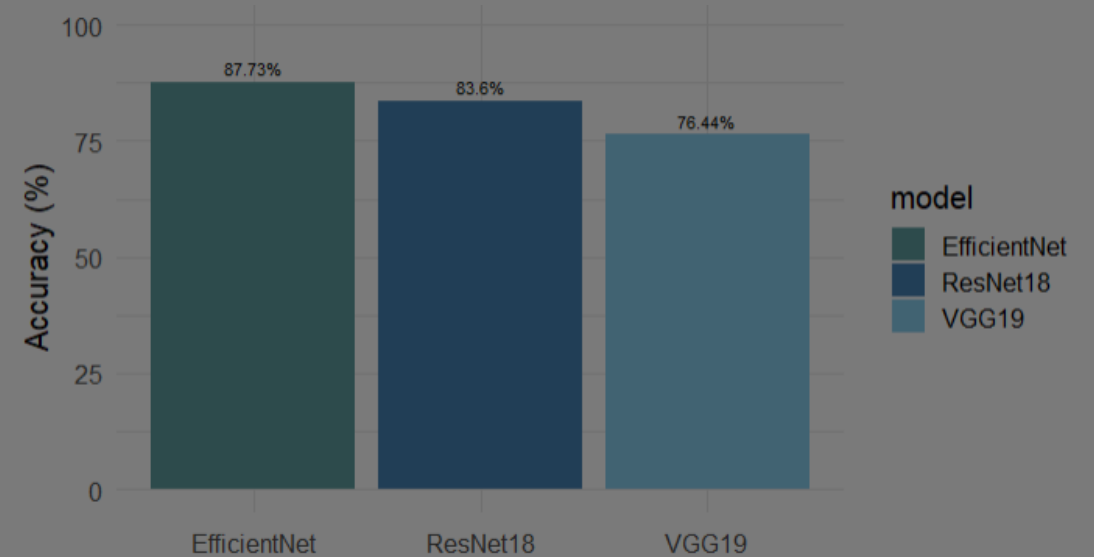
# Train(4000) + CIFAR-10(50000)

- 실험 목적 : 학습 데이터 증가에 따른 성능 변화 비교
- 기본 설정 : batch size = 32, epoch = 30, optimizer = Adam, validation set = 20% 사용

[실험1에 대한 모델 별 정확도 비교]



[실험2에 대한 모델 별 정확도 비교]

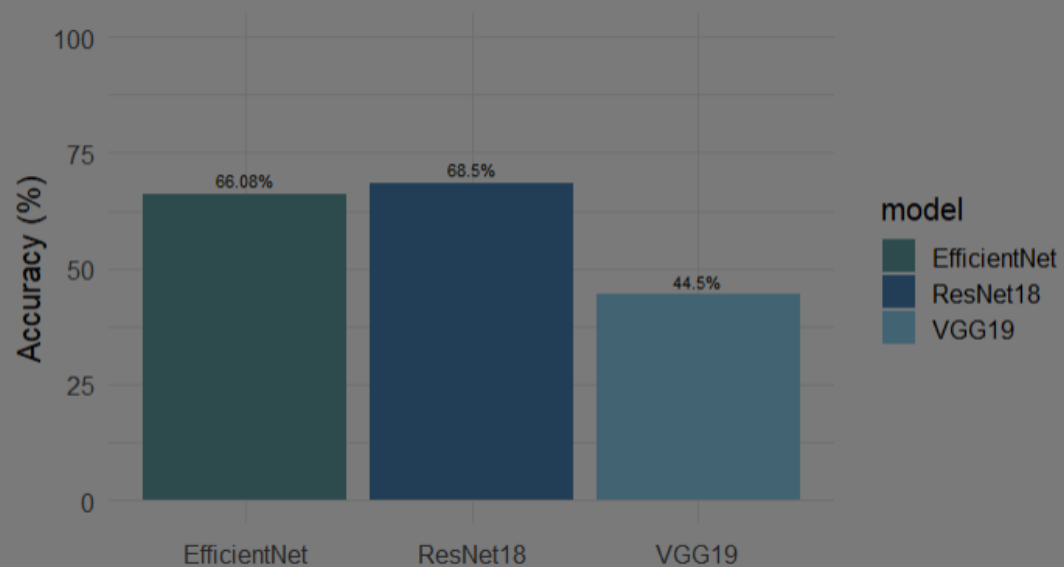


## 실험2.

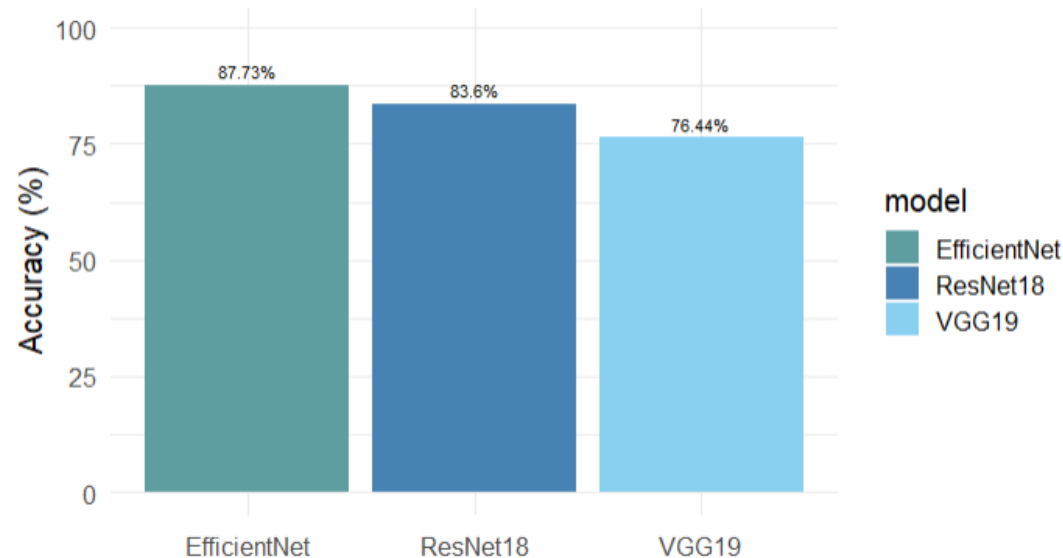
# Train(4000) + CIFAR-10(50000)

- 실험 목적 : 학습 데이터 증가에 따른 성능 변화 비교
- 기본 설정 : batch size = 32, epoch = 30, optimizer = Adam, validation set = 20% 사용

[실험1에 대한 모델 별 정확도 비교]



[실험2에 대한 모델 별 정확도 비교]



## 실험3. Image Augmentation

- 더 많은 데이터를 학습시켜 일반화 성능 향상 가능
- 각도, 크기, 색상 등 다양한 요소를 변형

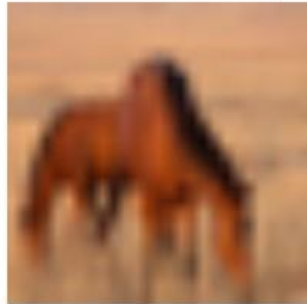
Original image



Random Horizontal Flip



Random Resized Crop



Random Rotation



ColorJitter



Cut out



Hue Saturation Value

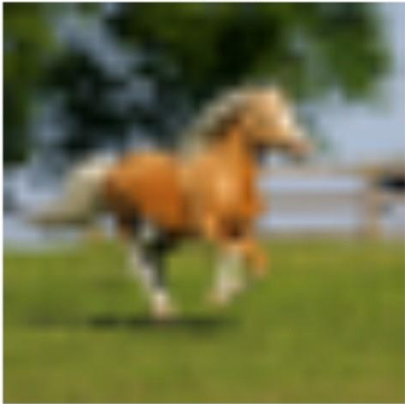




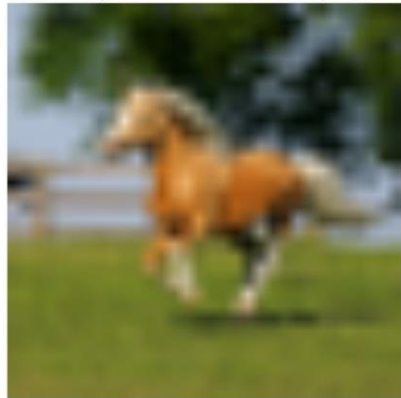
## 실험3. Implentation

- 저화질 이미지라는 점을 고려해 본래 특성을 가능한 해치지 않는 단순한 방법 적용

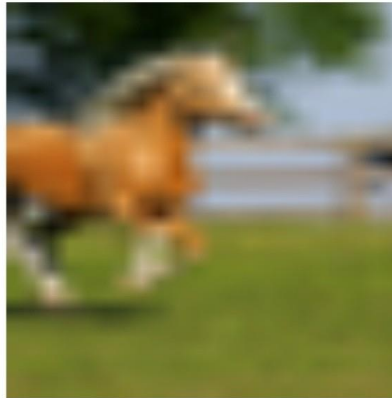
Original image



Horizontal Flipping P : 0.5



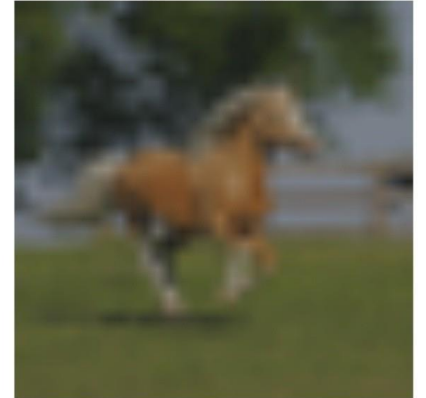
Cropping size : 0.8~1.0



Rotation angle :  $-45^{\circ} \sim 45^{\circ}$

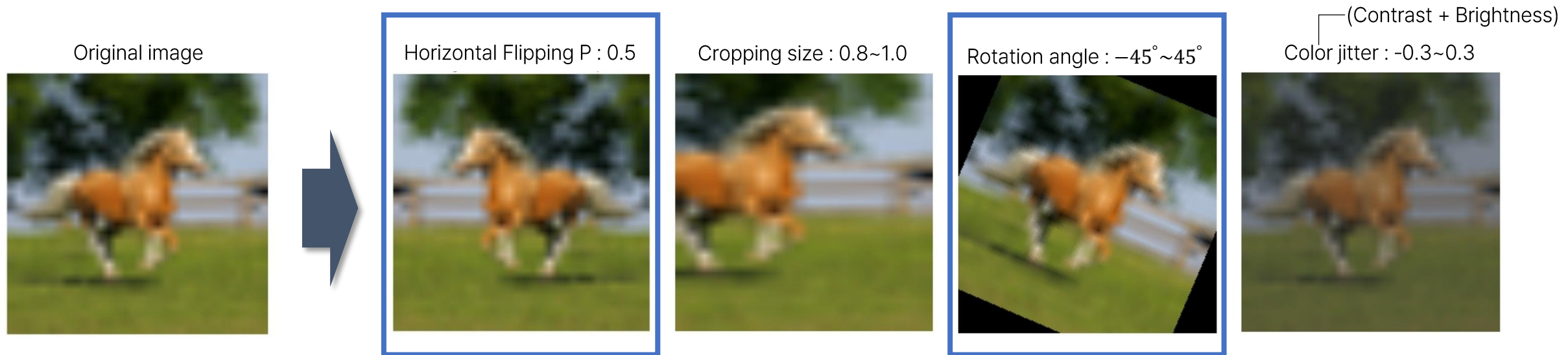


(Contrast + Brightness)  
Color jitter : -0.3~0.3



## 실험3. Implentation

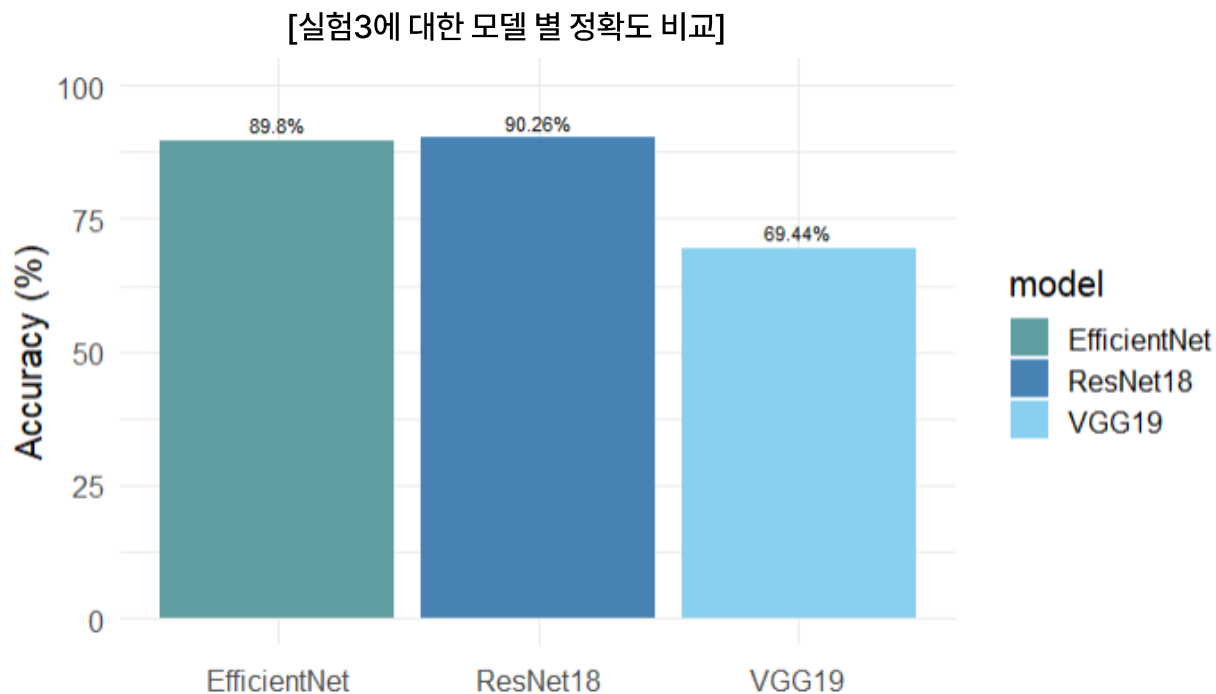
- 저화질 이미지라는 점을 고려해 본래 특성을 가능한 해치지 않는 단순한 방법 적용



## 실험3.

# Train(4000) + CIFAR-10(50000) + Augmentation

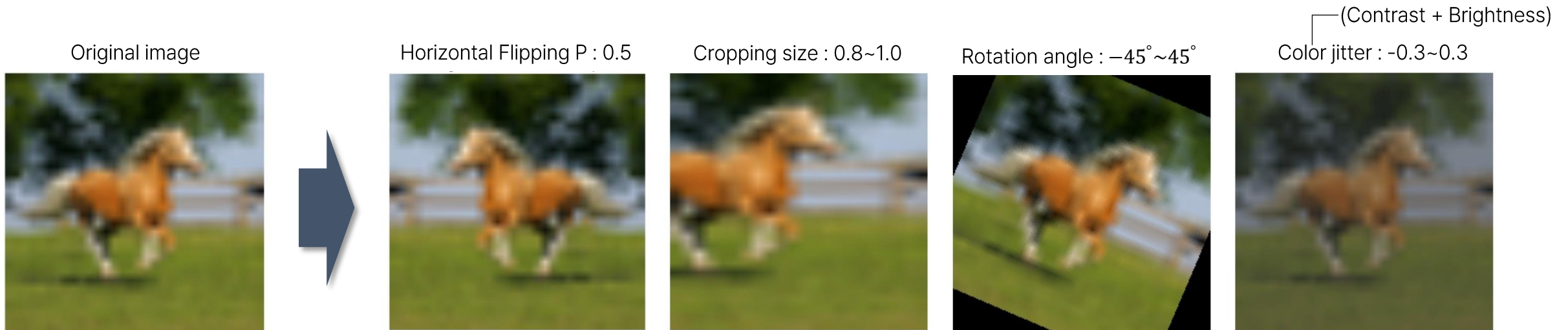
- 실험 목적 : 증강 데이터 추가에 따른 성능 비교
- 증강 데이터 : Random rotation(45) + Random Horizontal Flip(0.5) = 15000
- 기본 설정 : batch size = 32, epoch = 30, optimizer = Adam, validation set = 20% 사용



## 실험4.

# Augmentation 다양성과 epoch 조정

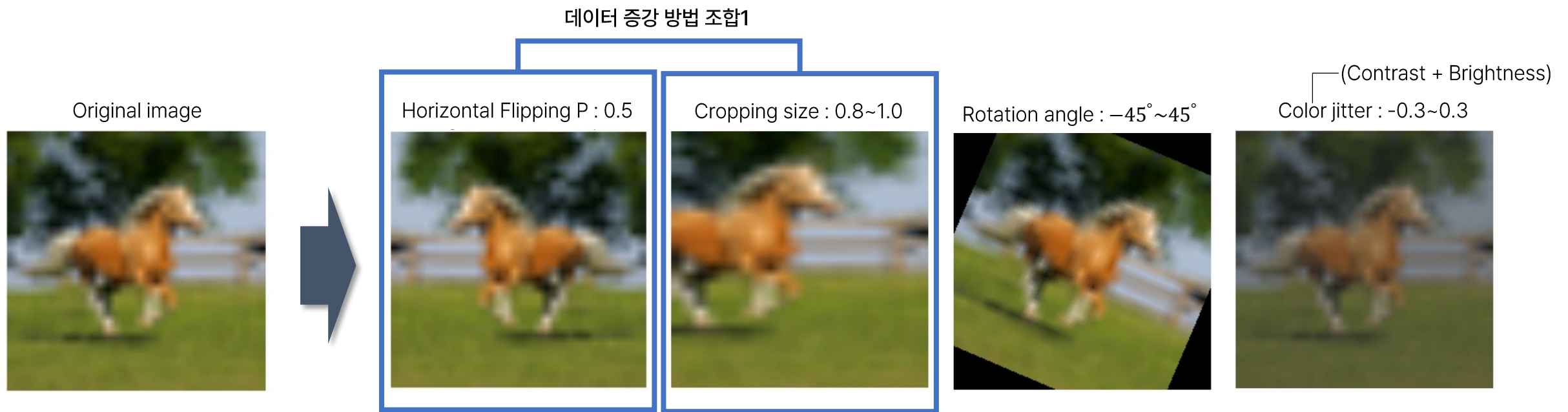
- Augmentation 방법을 다양하게 사용하여 증강 데이터를 추가함



## 실험4.

# Augmentation 다양성과 epoch 조정

- Augmentation 방법을 다양하게 사용하여 증강 데이터를 추가함



## 실험4.

# Augmentation 다양성과 epoch 조정

- Augmentation 방법을 다양하게 사용하여 증강 데이터를 추가함



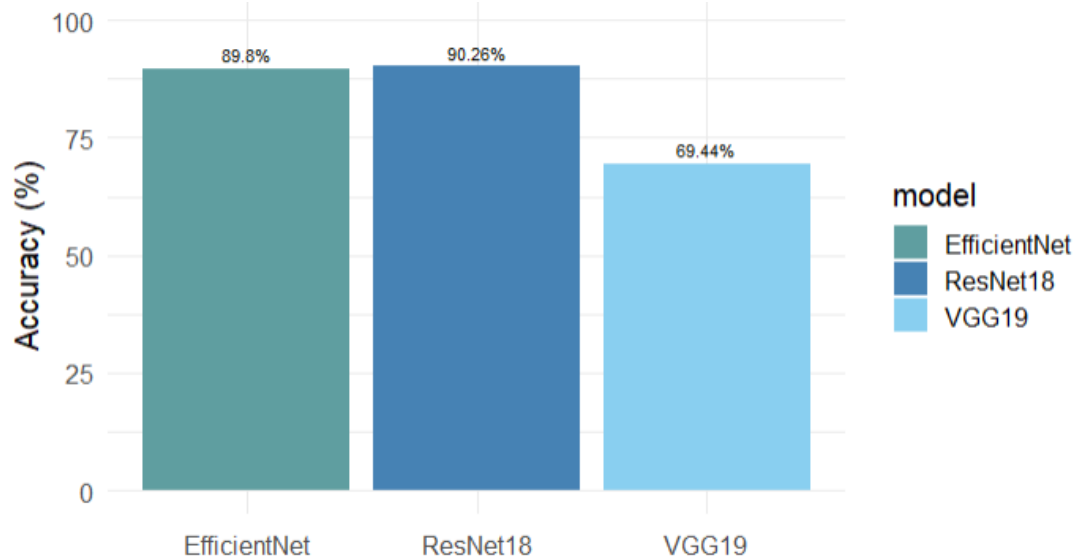


## 실험4.

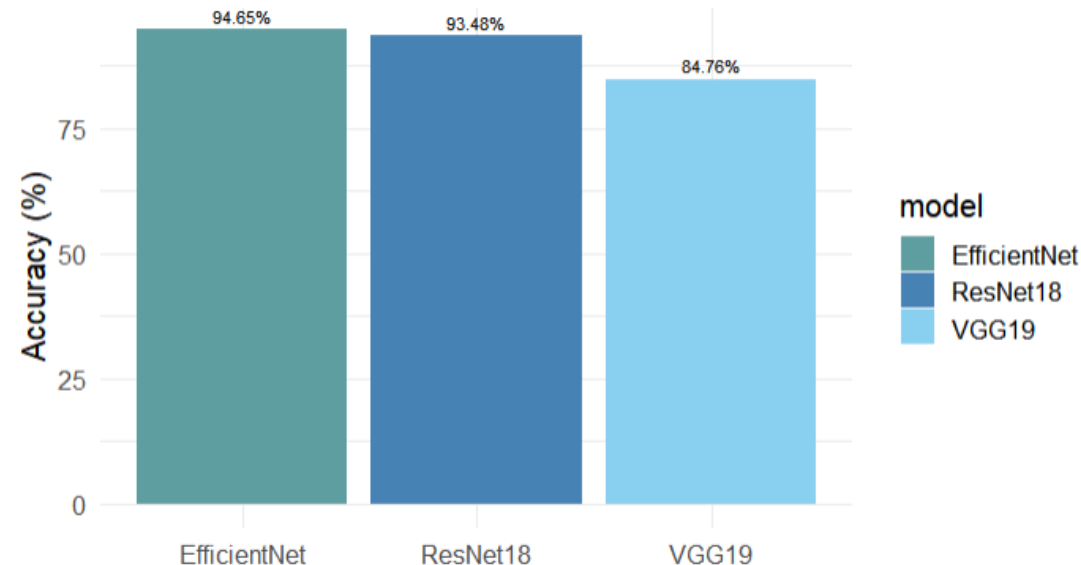
# Augmentation 다양성과 epoch 조정

- 실험 목적 : 두가지 증강 데이터를 혼합함으로써 원본 데이터를 보존하면서 더 다양한 증강 적용
- 사용 데이터 : train 4000 + CIFAR-10 50000 + 증강 조합1 15000 + 증강 조합2 15000 = 84000
- 기본 설정 : batch size = 32, epoch = 60, optimizer = Adam, validation set = 20% 사용

[실험3에 대한 모델 별 정확도 비교]



[실험4에 대한 모델 별 정확도 비교]

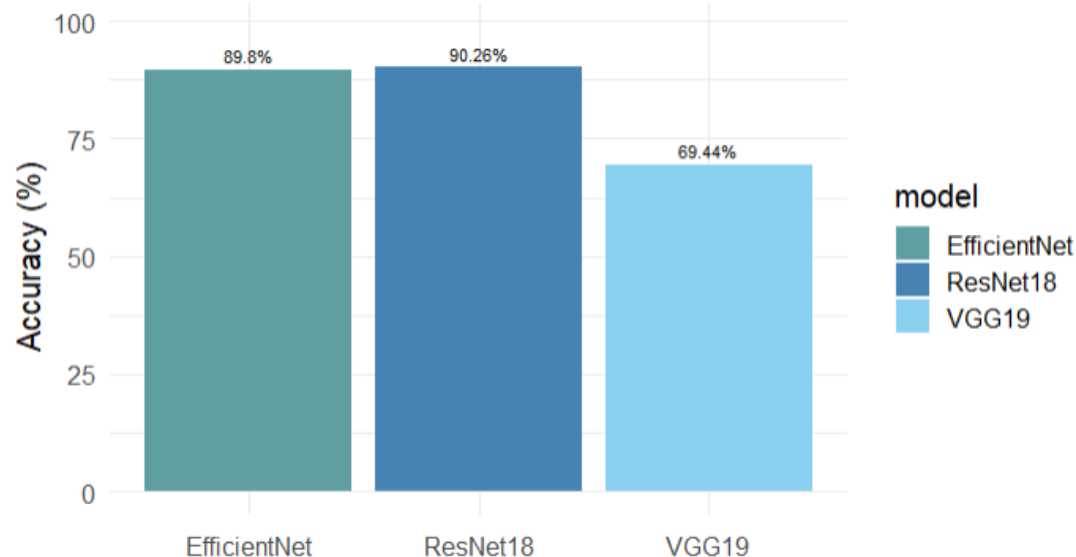


## 실험4.

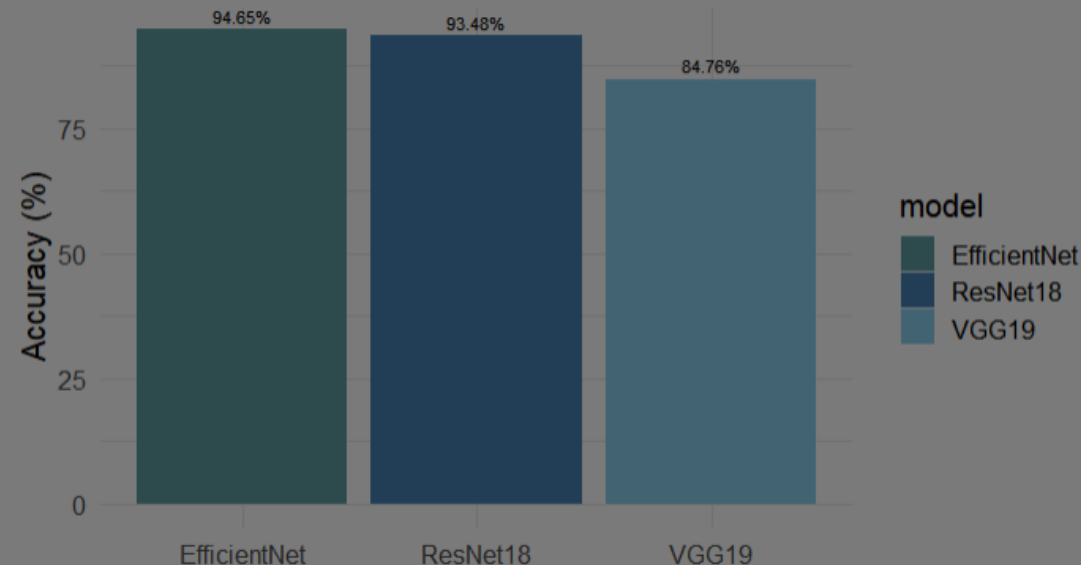
# Augmentation 다양성과 epoch 조정

- 실험 목적 : 두가지 증강 데이터를 혼합함으로써 원본 데이터를 보존하면서 더 다양한 증강 적용
- 사용 데이터 : train 4000 + CIFAR-10 50000 + 증강 조합1 15000 + 증강 조합2 15000 = 84000
- 기본 설정 : batch size = 32, epoch = 60, optimizer = Adam, validation set = 20% 사용

[실험3에 대한 모델 별 정확도 비교]



[실험4에 대한 모델 별 정확도 비교]



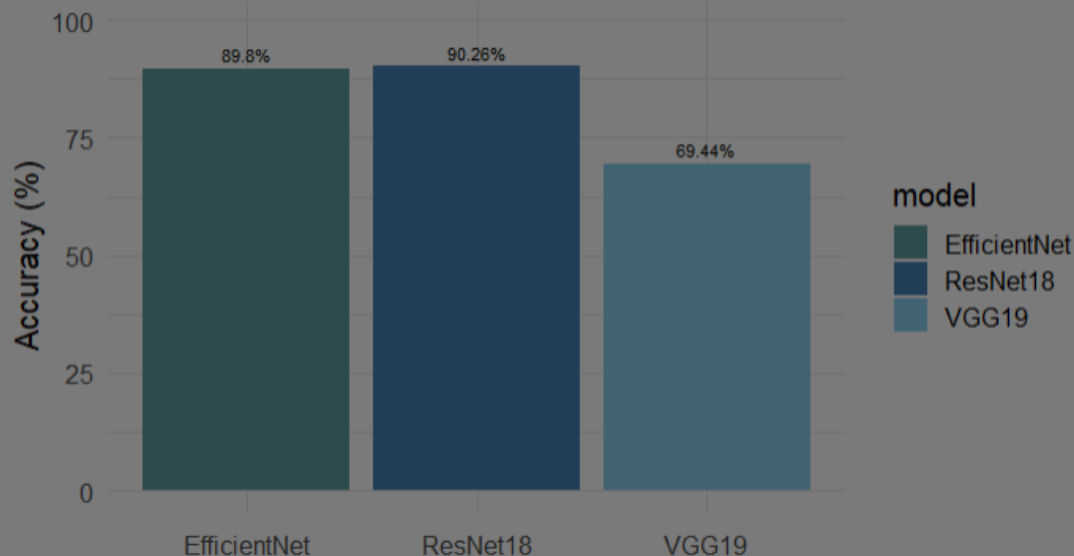


## 실험4.

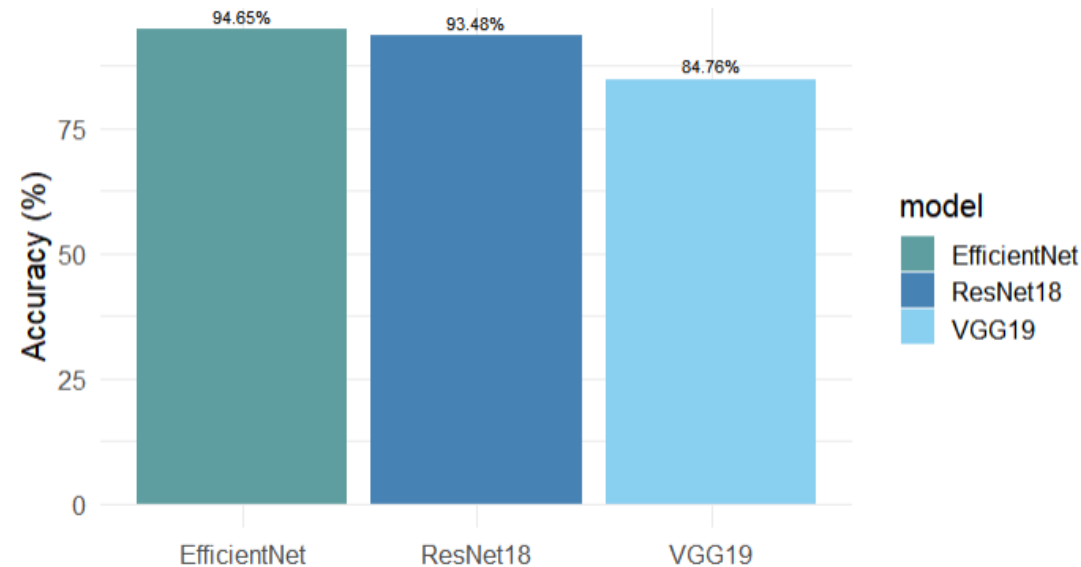
# Augmentation 다양성과 epoch 조정

- 실험 목적 : 두가지 증강 데이터를 혼합함으로써 원본 데이터를 보존하면서 더 다양한 증강 적용
- 사용 데이터 : train 4000 + CIFAR-10 50000 + 증강 조합1 15000 + 증강 조합2 15000 = 84000
- 기본 설정 : batch size = 32, epoch = 60, optimizer = Adam, validation set = 20% 사용

[실험3에 대한 모델 별 정확도 비교]



[실험4에 대한 모델 별 정확도 비교]

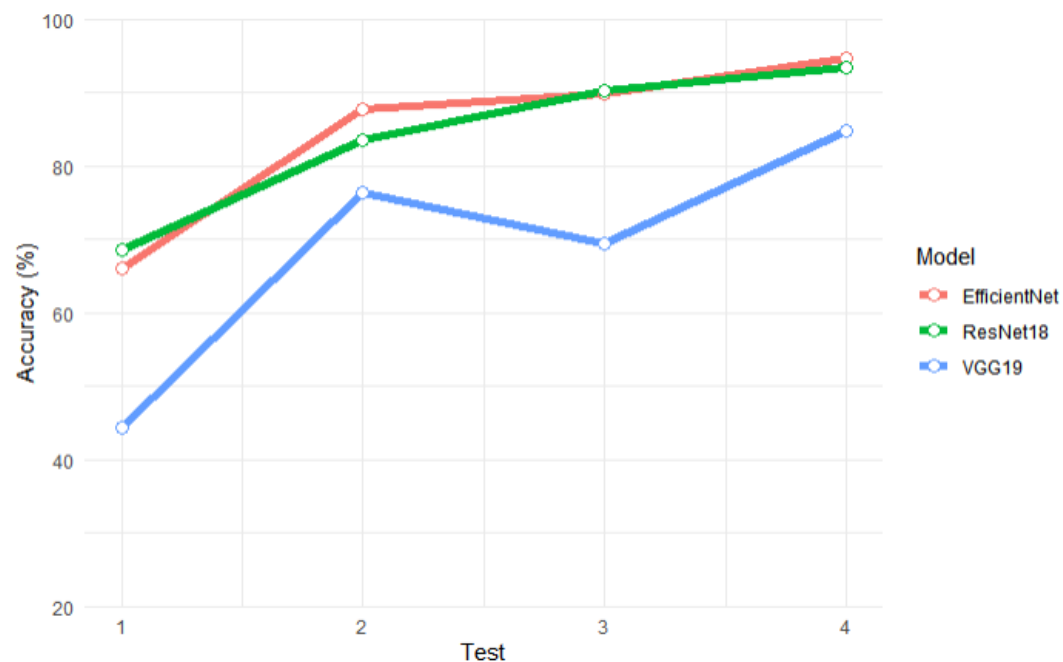


# 결론

# 결과 최종 성능 비교

- 최종 제출 모델: 실험 4에서 사용된 EfficientNet 모델

[실험별 모델 정확도 비교\_선그래프]



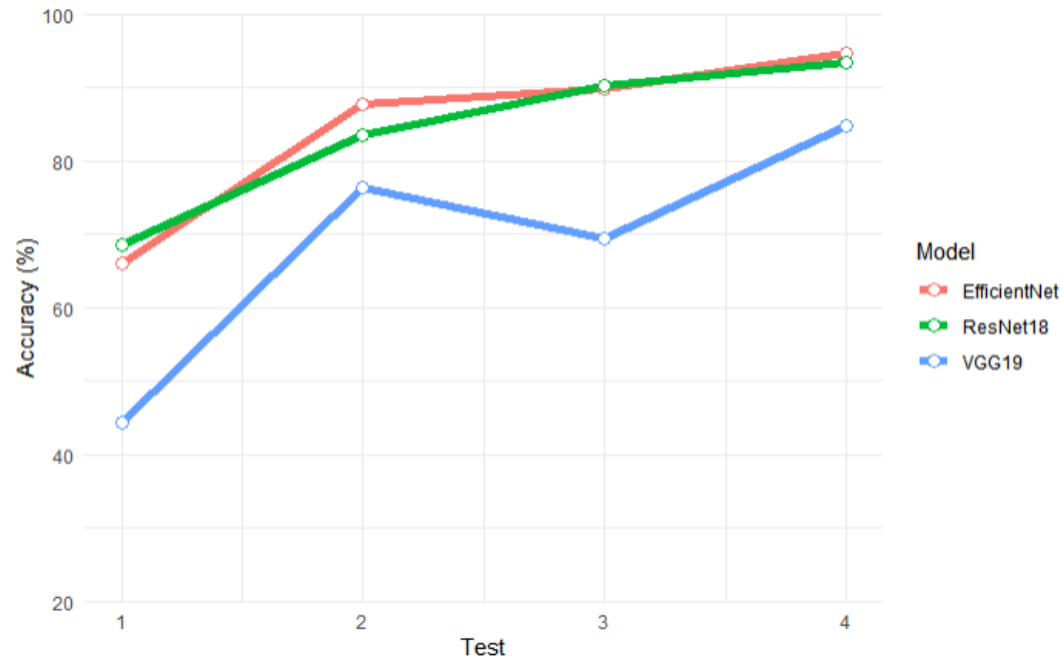
[실험별 모델 정확도 비교\_표]

Model	test1	test2	test3	test4
EfficientNet	66.08	87.73	89.80	94.65
ResNet18	68.50	83.60	90.26	93.48
VGG19	44.50	76.44	69.44	84.76

# 결과 최종 성능 비교

- 최종 제출 모델: 실험 4에서 사용된 EfficientNet 모델

[실험별 모델 정확도 비교\_선그래프]



[실험별 모델 정확도 비교\_표]

Model	test1	test2	test3	test4
EfficientNet	66.08	87.73	89.80	94.65
ResNet18	68.50	83.60	90.26	93.48
VGG19	44.50	76.44	69.44	84.76

## 소감 & 회고



### 통계학과 이종철

구성원들에게 많이  
배울 수 있어서 좋았습니다.  
또한 여러 방법을 적용해보고, 특정 요소  
의 변화가 모델 성능에 주는 영향을 관찰  
하는 과정이 정말 재밌었습니다.



### 전자공학과 박수연

이전까지는 딥러닝 모델을 짜고  
구상하는 부분을 위주로 연구를 진행하여  
데이터를 가공하여 학습에 활용해 볼 수 있  
는 기회가 적었는데, 팀원 분들의 도움으로  
다양하게 데이터를 가공하고 결과를 내볼 수  
있어서 좋은 경험이 되었습니다.  
짧은 시간임에도 불구하고 다양한 방법을 시  
도해 볼 수 있어서 좋은 시간이었습니다.



### 통계학과 박소정

유능한 팀원들 덕분에 순차적으로 진행했던  
것 같습니다! 이번 팀플의 계기로 딥러닝의  
경우 많은 시도들이 중요하다는 것을 알게  
되었고 모델 적용시에는 여러 많은 시도를  
스스로 진행해야겠다고 생각했습니다.

# Q&A