Plant Pathology 2021 – FGVC8

DATA MINING PROJECT

Team SOGANG_MLLAB

120210211 이지환

120210181 HAMZA BASHIR

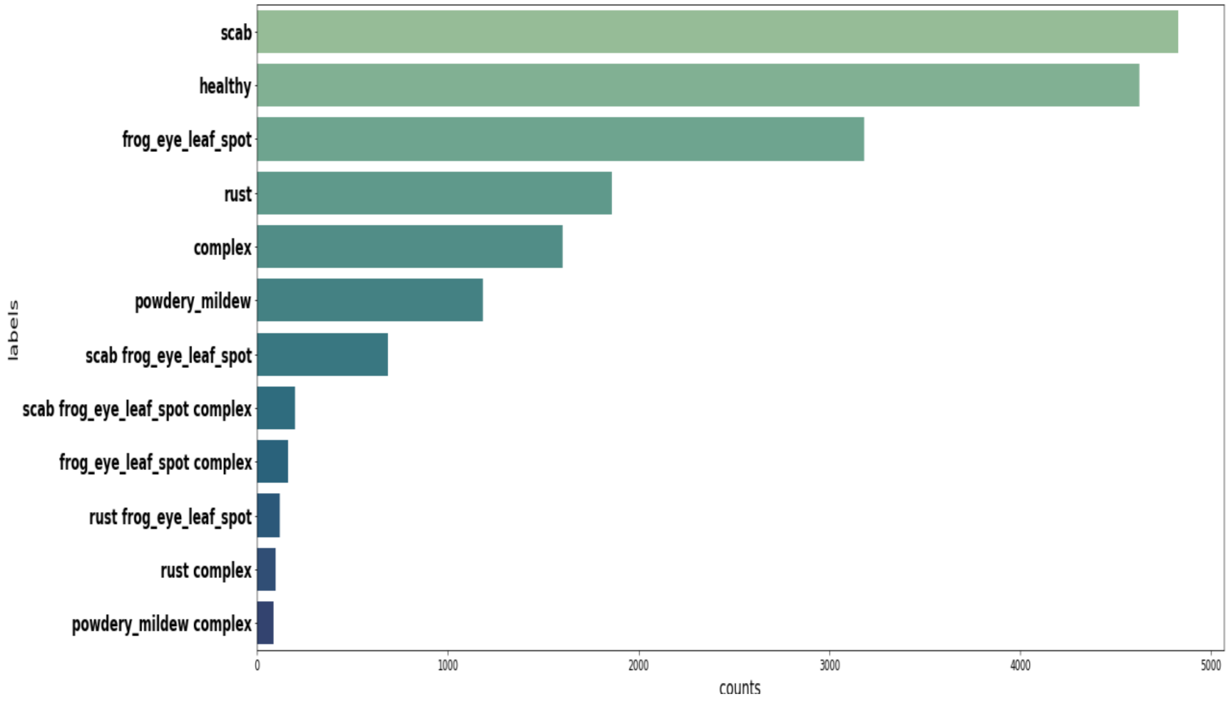
120210182 HAMSA PRIYA P

목차

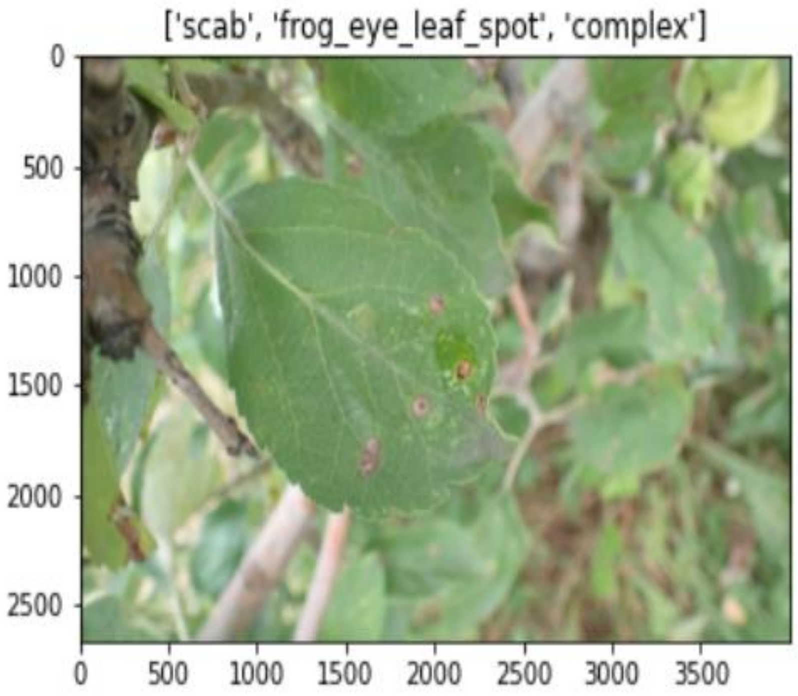
1. Plant Pathology 2021 – FGVC8

- Plant Pathology 2021 – FGVC8 (Fine-Grained Visual Categorization) – Identify the category of foliar diseases in apple trees.
- This competition is part of the Fine-Grained Visual Categorization FGVC8 workshop at the Computer Vision and Pattern Recognition Conference CVPR 2021.
- Can you help detect farmers detect apple diseases?
- This competition builds on last year's by challenging you to handle additional diseases and to provide more detailed information about leaves that have multiple infections.

1. Dataset



[그림 1] Label Distribution



[그림 2] Multi-Label Classification

1. Dataset

images

Binary Labels indicating plant's health condition
(Label Formatting using *MultiLabelBinarizer* instance)

	complex	frog_eye_leaf_spot	powdery_mildew	rust	scab	healthy
image						
800113bb65efe69e.jpg	0	0	0	0	0	1
8002cb321f8bfcdf.jpg	1	1	0	0	1	0
80070f7fb5e2ccaa.jpg	0	0	0	0	1	0
80077517781fb94f.jpg	0	0	0	0	1	0
800cbf0ff87721f8.jpg	1	0	0	0	0	0

One-Hot Encoding

1. Data Preprocessing

Removed duplicates

Images resized 600x600

One-hot Encoding

Making k-fold tf records

1. Data Augmentation

Original

VerticalFlip

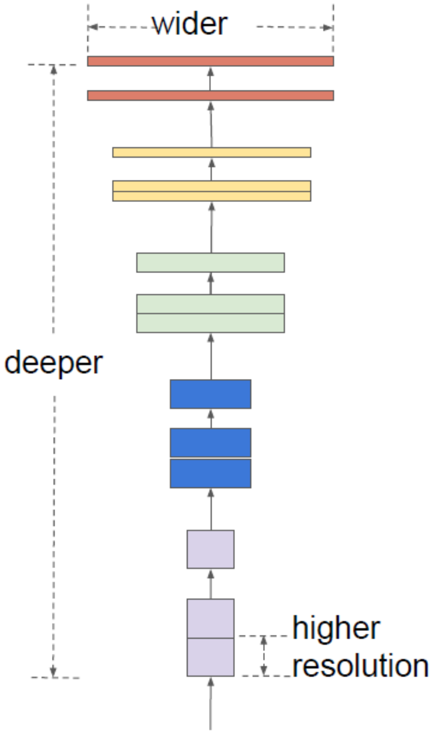
ShiftScaleRotate

Cutout

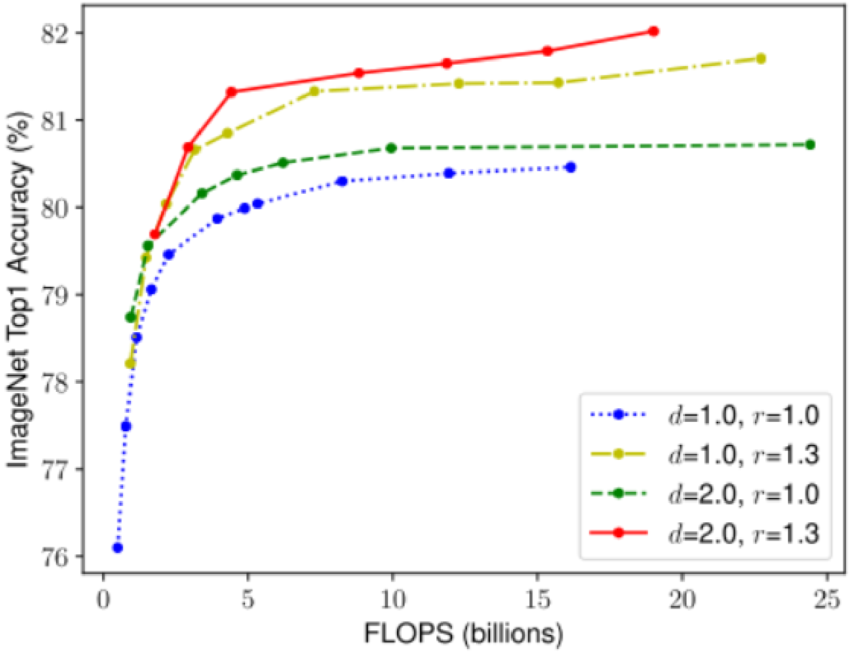
Gaussian blur…

다양한 Augmentation 기법들 사용

2. EfficientNet (ICML 2019)



[그림 3] Compound Scaling



[그림 4] Scaling Network Width for Different Baseline Network

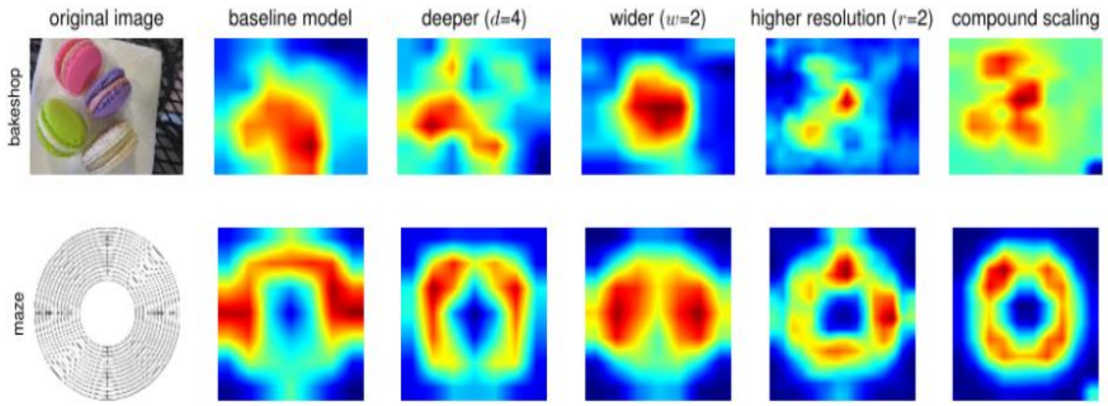
width, depth, resolution 3가지 방식을 모두 고려한 모델

Mingxing Tan, Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." *arXiv preprint arXiv:1905.11946* (2019).

2. EfficientNet (ICML 2019)

$$
\begin{aligned}
\text{depth: } d &= \alpha^{\phi} \\
\text{width: } w &= \beta^{\phi} \\
\text{resolution: } r &= \gamma^{\phi} \\
\text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 &\approx 2 \\
\alpha \geq 1, \beta \geq 1, \gamma &\geq 1
\end{aligned}
$$

[그림 5] Compound Scaling Notation



[그림 6] Class Activation Map Experiments

- grid search를 통해 α, β, γ가 균형을 이루는 적절한 값 찾기
- Scaling factor 고정 후 φ 값 키우기 (B0 to B7)

- 3가지 Scaling Factor를 동시에 고려 → 더 정교한 CAM

Mingxing Tan, Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." *arXiv preprint arXiv:1905.11946* (2019).

2. EfficientNet (ICML 2019)

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPS	Ratio-to-EfficientNet
EfficientNet-B0	**76.3%**	**93.2%**	**5.3M**	**1x**	**0.39B**	**1x**
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	**78.8%**	**94.4%**	**7.8M**	**1x**	**0.70B**	**1x**
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	**79.8%**	**94.9%**	**9.2M**	**1x**	**1.0B**	**1x**
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	**81.1%**	**95.5%**	**12M**	**1x**	**1.8B**	**1x**
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	**82.6%**	**96.3%**	**19M**	**1x**	**4.2B**	**1x**
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	**83.3%**	**96.7%**	**30M**	**1x**	**9.9B**	**1x**
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	**84.0%**	**96.9%**	**43M**	**1x**	**19B**	**1x**
EfficientNet-B7	**84.4%**	**97.1%**	**66M**	**1x**	**37B**	**1x**
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

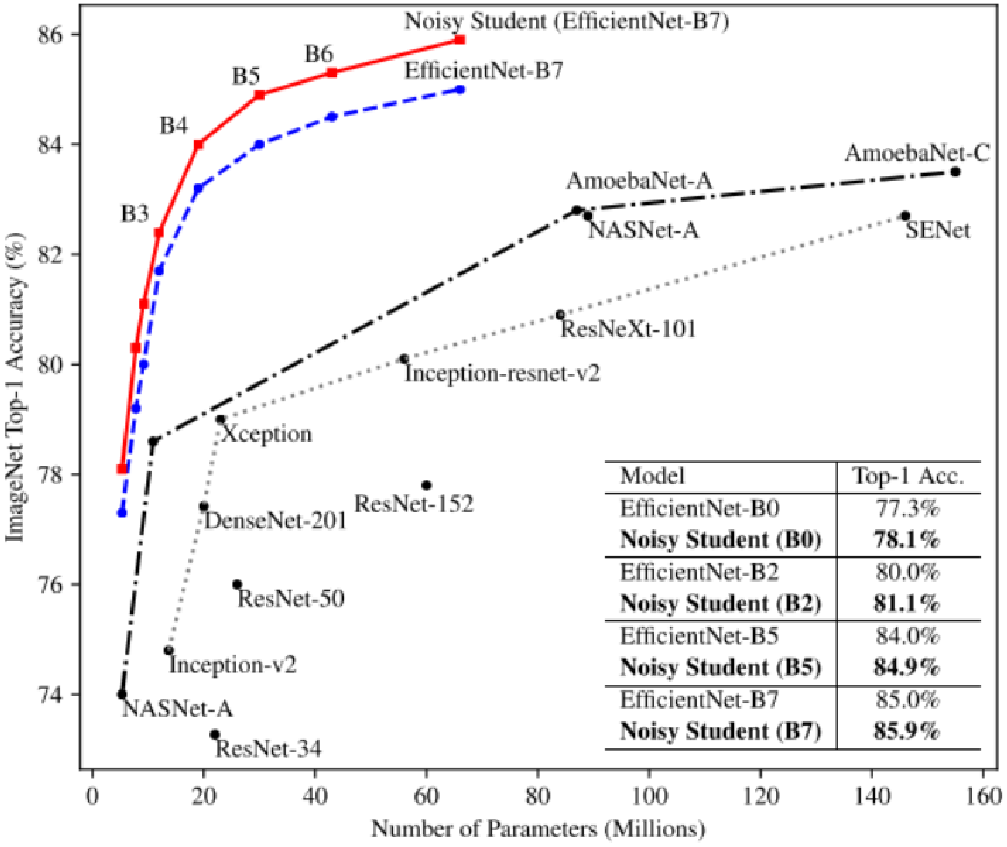
We omit ensemble and multi-crop models (Hu et al., 2018), or models pretrained on 3.5B Instagram images (Mahajan et al., 2018).

[그림 7] Experimental results based on ImageNet

다른 모델에 비해 훨씬 적은 Parameter, FLOPs수를 가짐

Mingxing Tan, Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." *arXiv preprint arXiv:1905.11946* (2019).

2. Noisy-Student Weights (CVPR 2020)



Model	Top-1 Acc.
EfficientNet-B0	77.3%
Noisy Student (B0)	**78.1%**
EfficientNet-B2	80.0%
Noisy Student (B2)	**81.1%**
EfficientNet-B5	84.0%
Noisy Student (B5)	**84.9%**
EfficientNet-B7	85.0%
Noisy Student (B7)	**85.9%**

EfficientNet을 기반으로 ImageNet 데이터셋에 대해 SOTA를 갱신

Qizhe Xie et al. "Self-training with Noisy Student improves ImageNet classification." *arXiv preprint arXiv:1911.04252* (2020).

2. Noisy-Student Weights (CVPR 2020)

Require: Labeled images $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ and unlabeled images $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_m\}$.

1: Learn teacher model θ^t_* which minimizes the cross entropy loss on labeled images

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta^t))$$

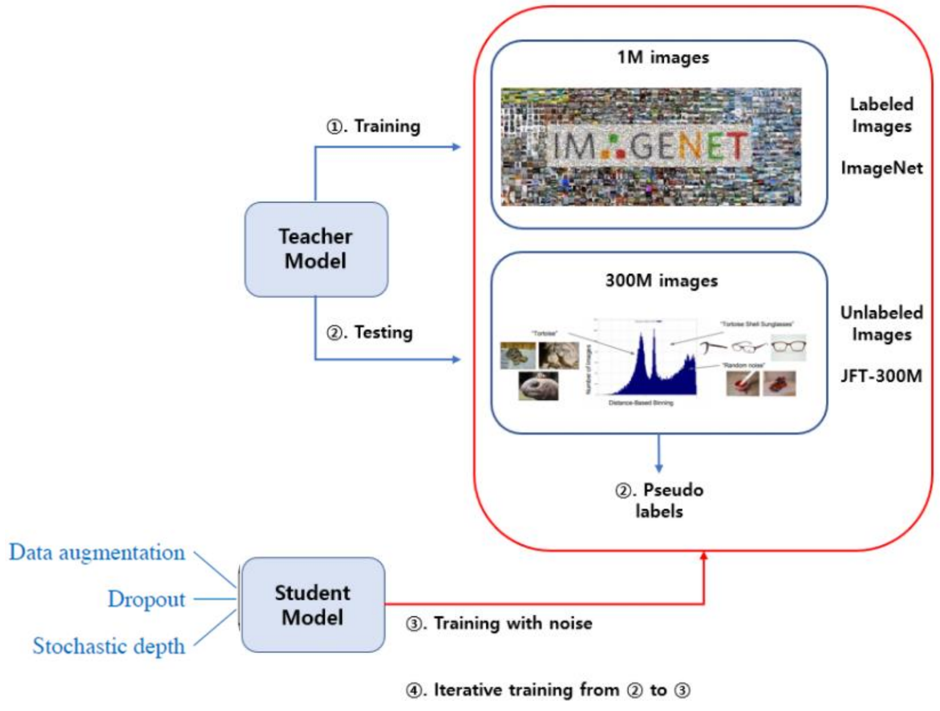
2: Use a normal (i.e., not noised) teacher model to generate soft or hard pseudo labels for clean (i.e., not distorted) unlabeled images

$$\tilde{y}_i = f(\tilde{x}_i, \theta^t_*), \forall i = 1, \cdots, m$$

3: Learn an **equal-or-larger** student model θ^s_* which minimizes the cross entropy loss on labeled images and unlabeled images with **noise** added to the student model

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta^s)) + \frac{1}{m}\sum_{i=1}^{m} \ell(\tilde{y}_i, f^{noised}(\tilde{x}_i, \theta^s))$$

4: Iterative training: Use the student as a teacher and go back to step 2.



[그림 8] Self-Training Algorithm

Labeled 데이터셋 뿐만 아니라 Unlabeled 데이터셋도 활용하여 성능을 높인 방법

Qizhe Xie et al. "Self-training with Noisy Student improves ImageNet classification." *arXiv preprint arXiv:1911.04252* (2020).

3. Experiments

[표 1] 실험 환경

Hyper-parameter	Setting
Batch Size	64
Optimizer	adam
Learning Rate	1e-3
Epochs	20

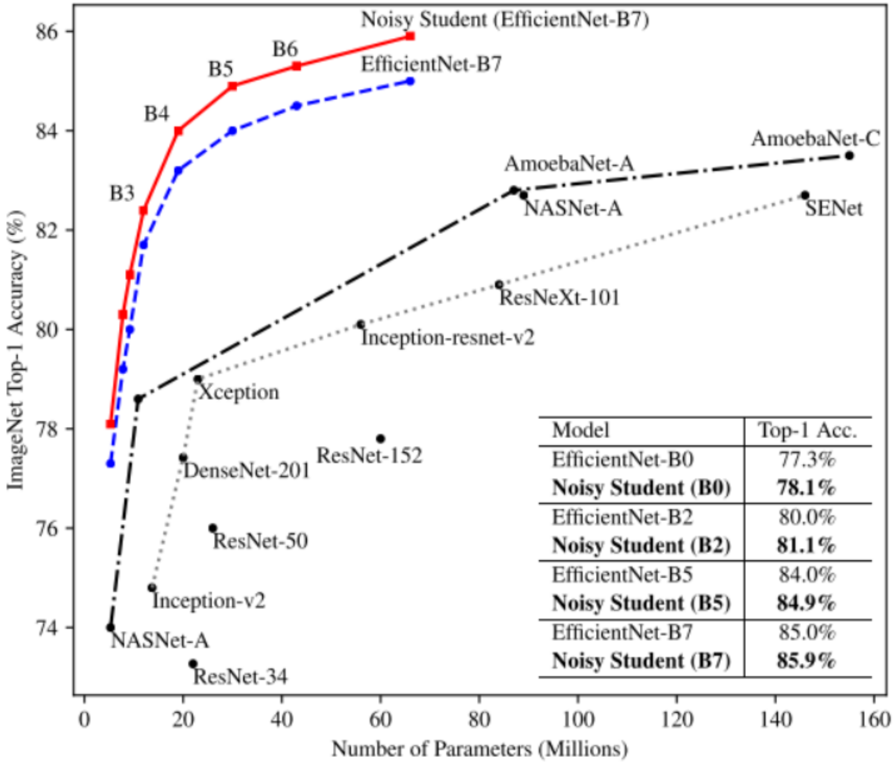
[그림 9] 성능 평가 지표

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

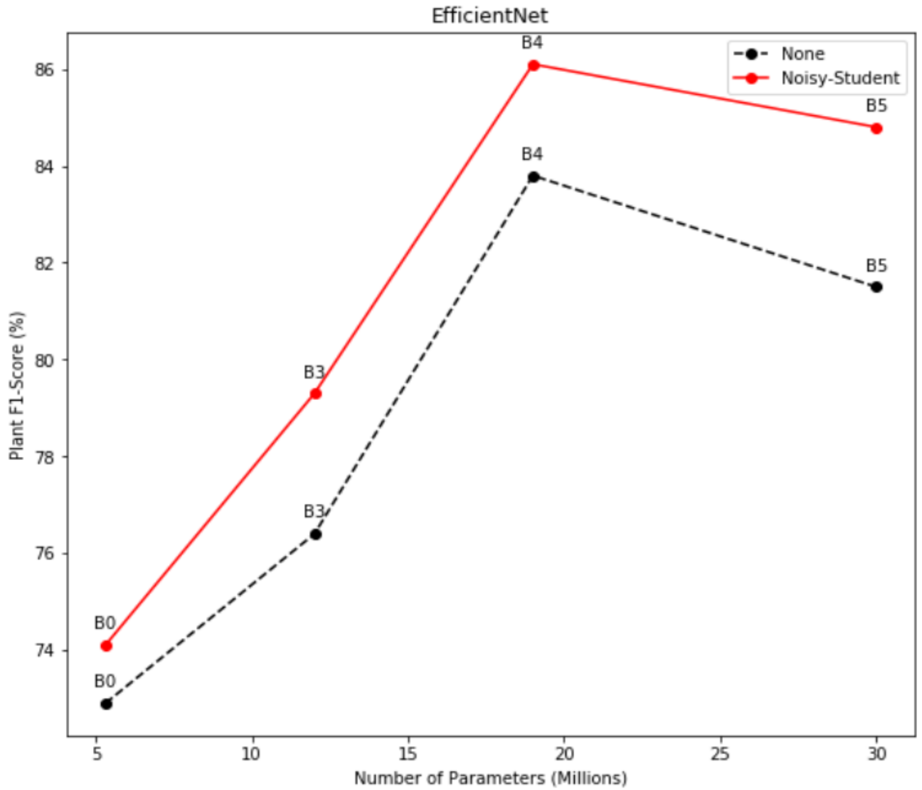
- 모델마다 하이퍼파라미터 동일하게 설정
- F1-score로 성능 평가
- 실험 1, 실험 2 진행

3. Experiments

실험 1- Noisy Student



[그림 10] ImageNet Dataset

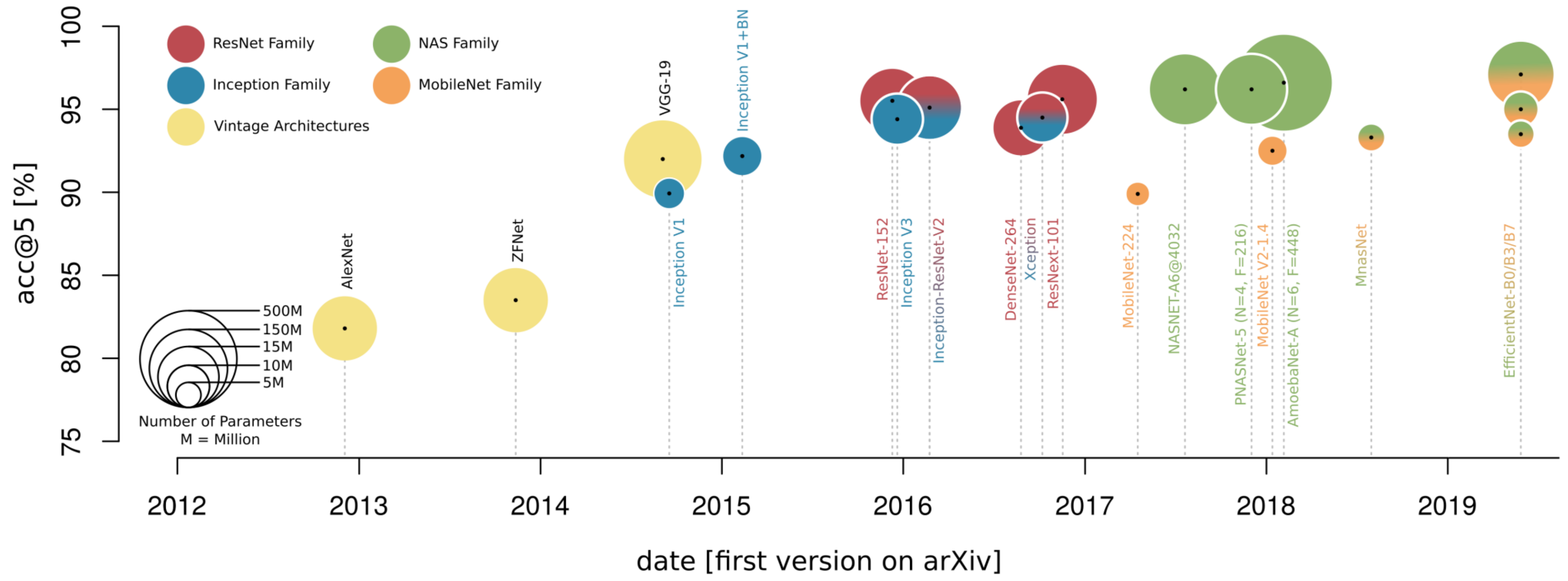


[그림 11] Plant Dataset

Plant Dataset에서 Noisy-Student를 적용한 pre-trained 모델을 활용하는 것이 더 높은 성능을 보임

3. Experiments

실험 2- CNN Backbone



CNN Architecture별 실험 진행

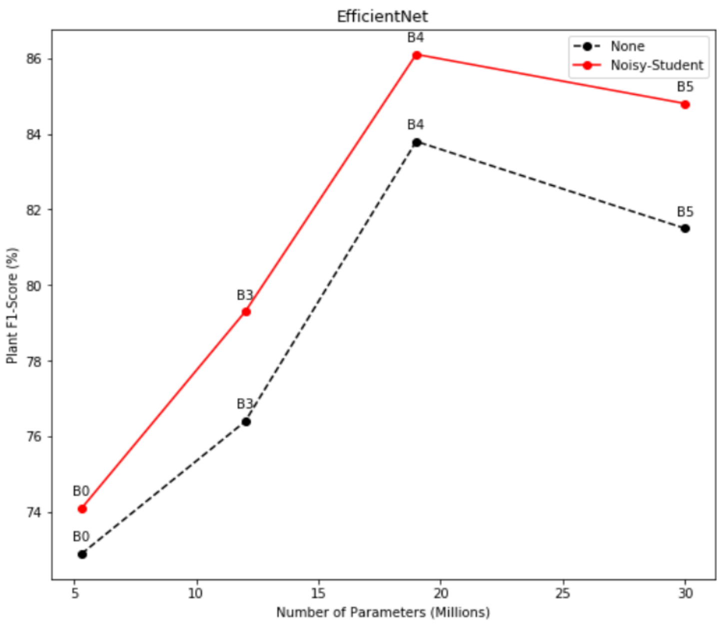
3. Experiments

실험 2- CNN Backbone

Model	F1-Score	#Params	#FLOPs
ResNet-50 (He et al., 2016)	74.6%	26M	4.1B
ResNet-101 (He et al., 2016)	72.4%	43M	7.9B
ResNet-152 (He et al., 2016)	71.2%	60.3M	11B
Inception-v3 (Szegedy et al., 2016)	56.7%	24M	5.7B
NASNet (Zoph et al., 2018)	46.2%	89M	24B
DenseNet-121 (Huang et al., 2017)	75.6%	7.2M	2.83B
DenseNet-169 (Huang et al., 2017)	80.9%	14M	3.5B
MobileNetV2 (Howard et al., 2017)	52.4%	3.4M	1.3B
EfficientNet-B0 (Mingxing et al., 2019)	74.1%	5.3M	0.39B
EfficientNet-B3 (Mingxing et al., 2019)	79.3%	12M	1.8B
EfficientNet-B4 (Mingxing et al., 2019)	86.1%	19M	4.2B
EfficientNet-B5 (Mingxing et al., 2019)	84.8%	30M	9.9B
EfficientNetV2 (Mingxing et al., 2021)	83.9%	24M	8.8B

[그림 12] Model별 F1-Score 값 (Plant Dataset)

4. Conclusion



Model	F1-Score	#Params	#FLOPs
ResNet-50 (He et al., 2016)	74.6%	26M	4.1B
ResNet-101 (He et al., 2016)	72.4%	43M	7.9B
ResNet-152 (He et al., 2016)	71.2%	60.3M	11B
Inception-v3 (Szegedy et al., 2016)	56.7%	24M	5.7B
NASNet (Zoph et al., 2018)	46.2%	89M	24B
DenseNet-121 (Huang et al., 2017)	75.6%	7.2M	2.83B
DenseNet-169 (Huang et al., 2017)	80.9%	14M	3.5B
MobileNetV2 (Howard et al., 2017)	52.4%	3.4M	1.3B
EfficientNet-B0 (Mingxing et al., 2019)	74.1%	5.3M	0.39B
EfficientNet-B3 (Mingxing et al., 2019)	79.3%	12M	1.8B
EfficientNet-B4 (Mingxing et al., 2019)	86.1%	19M	4.2B
EfficientNet-B5 (Mingxing et al., 2019)	84.8%	30M	9.9B
EfficientNetV2 (Mingxing et al., 2021)	83.9%	24M	8.8B

i) Unlabeled Data를 활용한 pre-trained model을 사용하는 것이 성능 향상에 도움을 준다.

ii) 모델의 크기가 클수록 성능이 좋은 것은 아님. ex〉 ResNet-50 VS ResNet-152, B4 VS B5

iii) 최종 제출 모델: EfficientNet-B4