IDL FINAL PROJECT

IMPROVING WASTE IMAGE CLASSIFICATION PERFORMANCE OF EFFICIENT DL MODELS

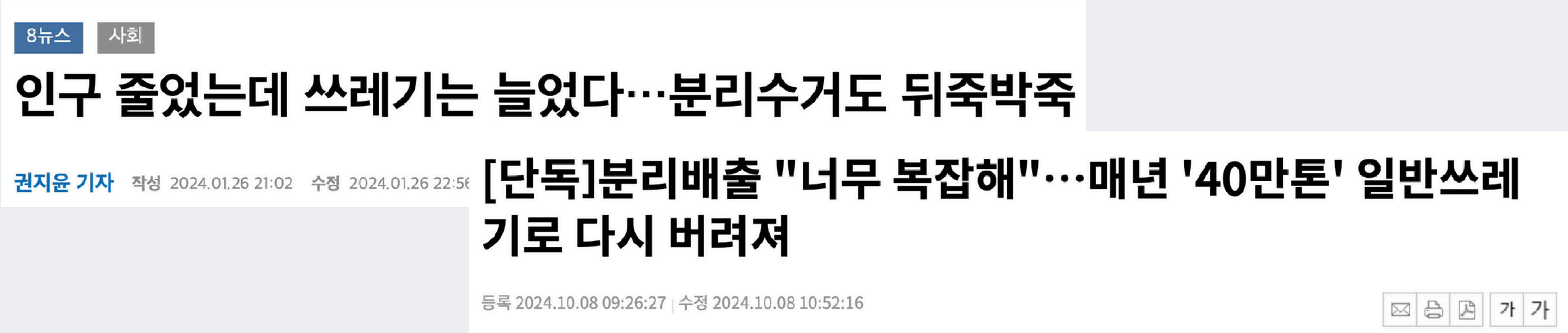
Group 9

PRESENTED BY

문강륜 백정은 이상혁 이동률 최해민

TABLE OF CONTENTS

PROBLEM DEFINITION: Objectives



Problem Definition

Problems

- 1990 ~ 현재까지 폐기물 발생량의 지속적 증가
- 분리수거가 올바르게 진행되지 않아 폐기물을 분리하는 과정이 필요
- 수거 과정에서 모두 섞여 다시 분리하는 과정 발생

AI 적용을 통해 빠르고, 정확하게 폐기물을 분류

Image Classification !

Major Objectives

1. Accuracy

- 폐기물의 **LABEL을 정확하게 예측**하는 모델의 설계
- 정확한 폐기물 작업에 기여할 수 있는 알고리즘 설계
- 실제로 **적용할 수 있는 수준의 정확도**(>90%)를 기록할 수 있도록 설계

2. Faster Inference

- 수기 기반 폐기물 분류를 대체 -> **빠르게 분류**하는 것이 중요함
- 현업 사용가능한 추론 속도를 위해 기존의 모델 아키텍처를 간소화/수정

RELATED RESEARCH ：Efficient DL Models

EfficientNetV2

- **Adaptive Regularization**

  Used task & data specific regularization, which improves generalization performance

- **Use of Fused-MBConv Block**

  Removed Depthwise Convolution of MBConv

- **Non-uniform Scaling**

  More flexibility by assigning different scaling rates to depth, width, and resolution.

MobileViT

- **Hybrid Architecture**

  Combined convolutional layers with transformer-based layers

- **Efficient Tokenization**

  Process the image at multiple resolutions, applying convolutions in the lower layers

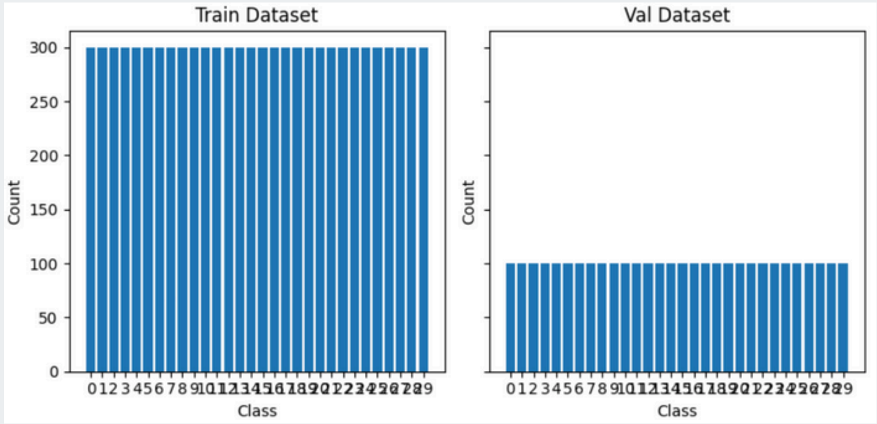
- **Lightweight Transformer Blocks**

  Integrate depth-wise separable convolutions in its transformer blocks

INITIAL IMPLEMENTATIONS

Dataset

- **15,000** images (256 x 256)
- **30** Classes of Recycle & Household Waste
- Includes **in-the-wild** and **normal** images
- **Dataset Split |** Tr : Val : Te = **3 : 1 : 1**



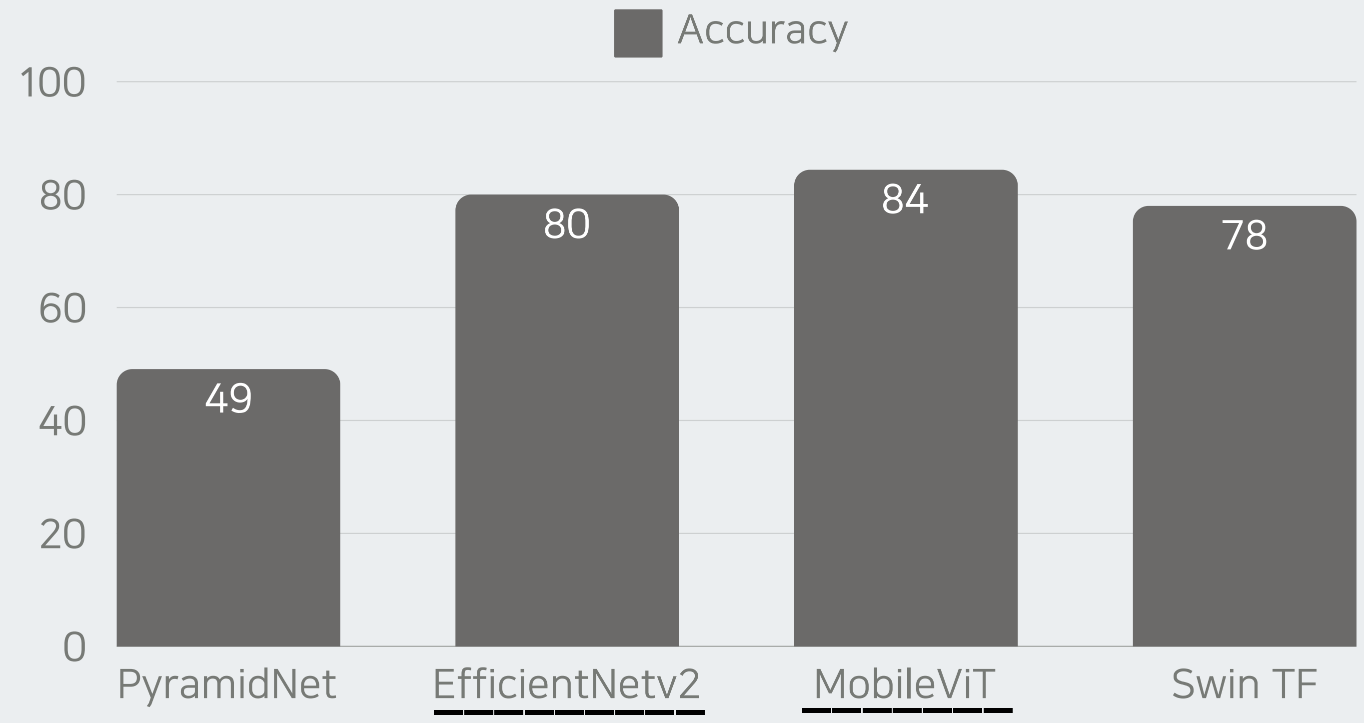
Dataset Distribution

Dataset Samples

Train Configs

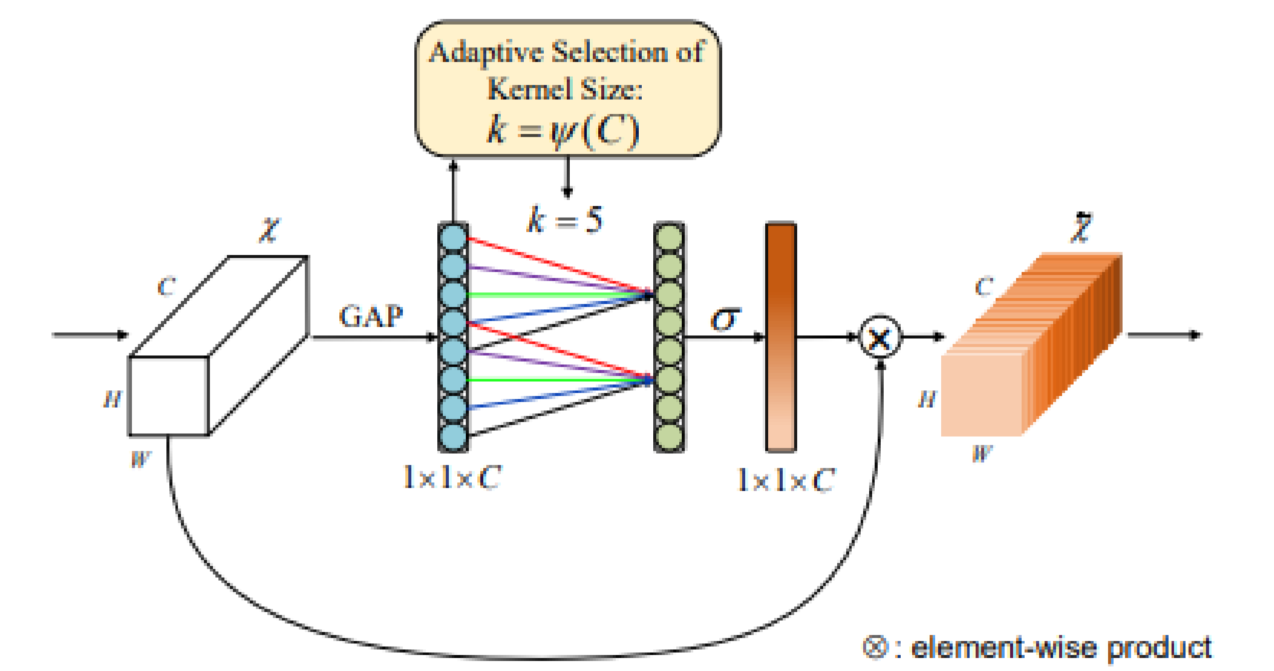
- Training Environment : 1 NVIDIA T4 / A100 GPU
- Random Seed : 17
- Optimizer : Adam
- Loss Function : Cross Entropy Loss
- Metric : Accuracy (%)

Model Performance

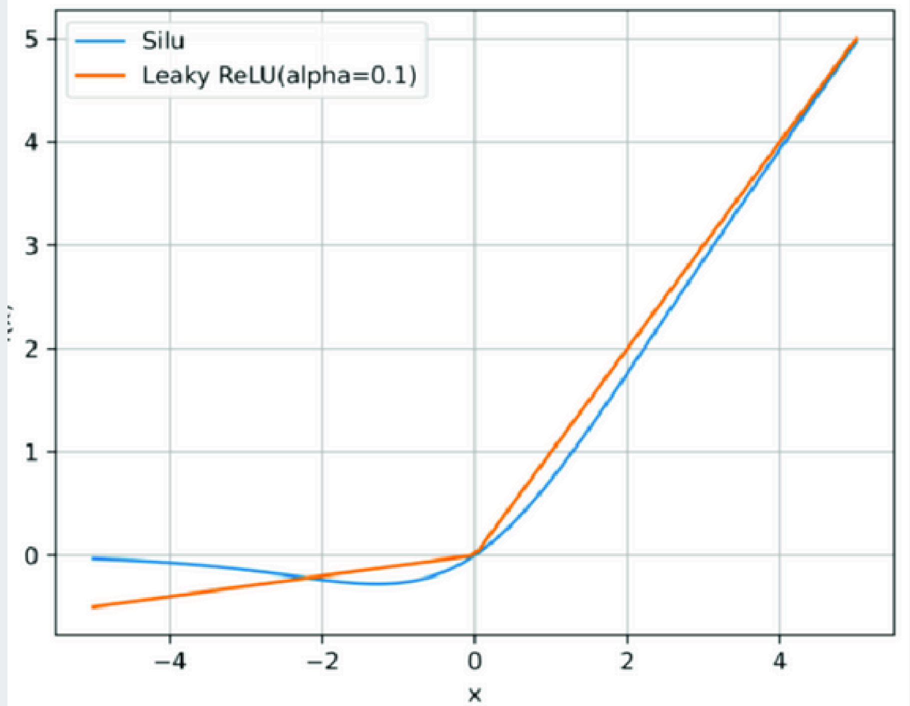


PROPOSED METHOD 01

>>> Replacing Squeeze and Excitation(SE) with Efficient Channel Attention(ECA)



Efficient Channel Attention(ECA) [1]



SiLU & Leaky ReLU Activation Function

Problems aim to resolve

1. Computationally **expensive operations of SE**
   (e.g., additional fully connected layers and global average pooling)
2. Susceptibility to overfitting due to limited dataset size

Changes

- Replace SE blocks with ECA blocks
- Add ECA Blocks to Fused-MBConv
- Switch the SiLU activation function to Leaky ReLU

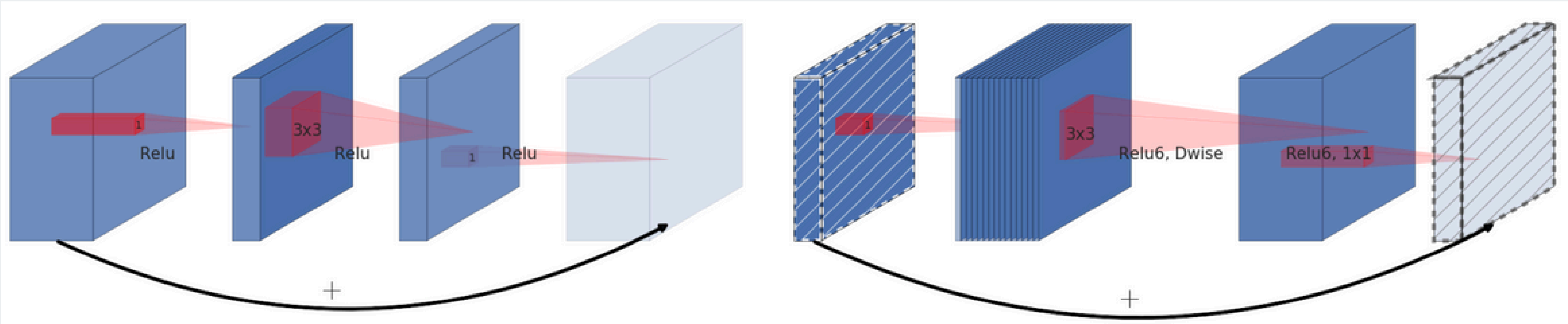
Anticipated Result

- Faster Inference while maintaining accuracy :
  - Eliminate fully connected layers in SE blocks
  - Improve channel-wise attention
  - Prevent additional computational overhead in shallow models

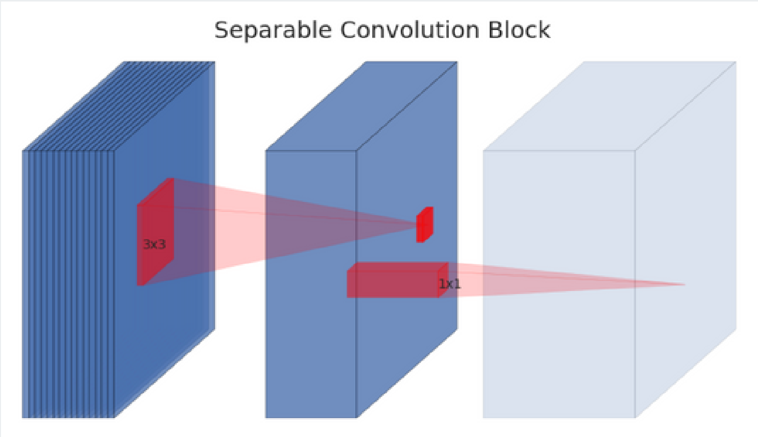
[1] Wang, Qilong, et al. "ECA-Net: Efficient channel attention for deep convolutional neural networks."

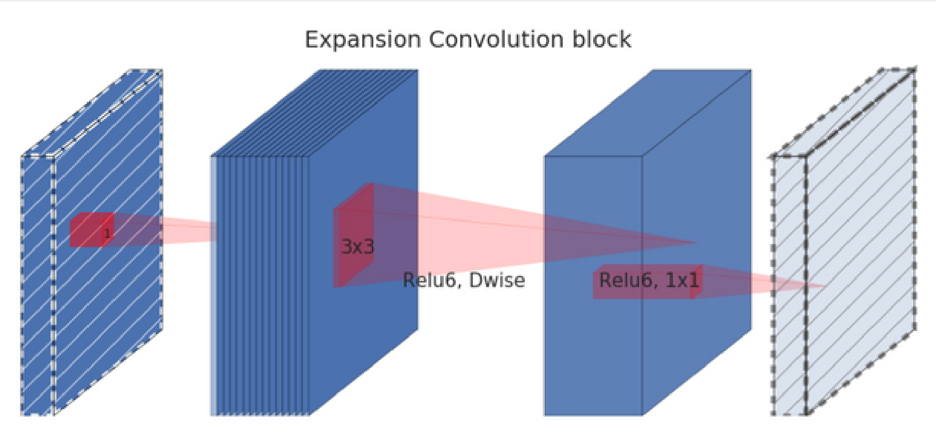
PROPOSED METHOD 02

>>> Replacing Self-Attention(SA) with Inverted Residuals(IR)

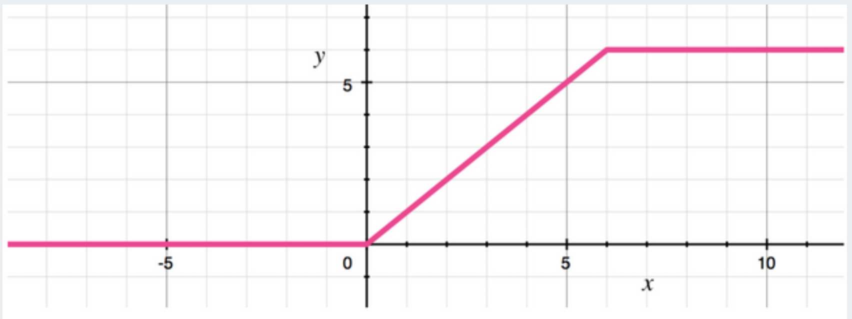


Residual and Inverted Residual Blocks [2]





Separable and Expension Convolution Blocks



ReLU6 Activation Function

Problems aim to resolve

1. Computationally **expensive operations of SA** (e.g., query-key matrix multiplication)
2. **Dependency** on **small** train dataset (overfitting)

Changes

- Replace SA layer with IR block (stacked MV2 blocks)
- Linear Bottleneck layers (w/ ReLU6, & exclude at final layer)
- Strong data augmentations

Anticipated Result

- Reduced Time Complexity : Linear to multiplication of H, W

$$O\left(\frac{H^2 \cdot W^2}{P^4} \cdot d_{\text{attn}}\right) \text{ to } O(H \cdot W \cdot C_{exp} \cdot k^2)$$

- Stable Training : Prevent large activations with better optimizations

[2] *Sandler, Mark, et al. "MobileNetV2: Inverted residuals and linear bottlenecks."*

APPLIED TRAINING TECHNIQUES

1. Weight Decay

Encourage smaller weights
- L2 Regularization

$$L_{new}(w) = L_{original}(w) + \lambda w^T w$$

Weight Decay of L2 Regularization

2. Data Augmentation

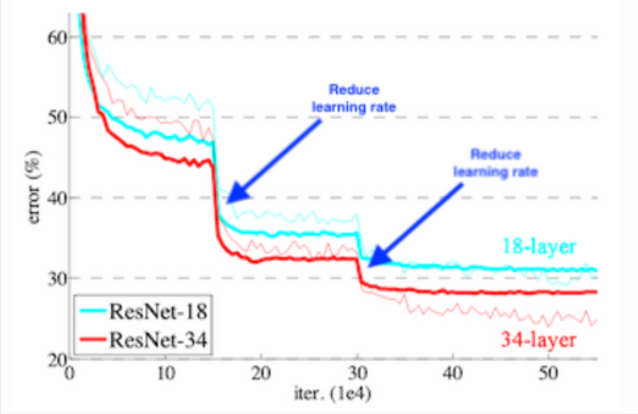
Consider Real World Scenarios
- Random Resized Crop
- Random Horizontal Flip

Example of Random Crop [3]

---

3. Learning Rate Scheduler

Find a better local optima
- Reduce LR on Plateau
- Warmup LR



Reduce LR on Plateau [4]

4. Weight Initialization

Prevent Vanishing / Exploding Gradients
- CNN layers : He Init.
- Linear layers : Normal Init.

[3] *PyTorch Official Docs*
[4] *What learning rate should I use? (B. D. Hammel)*

RESULT ANALYSIS 01 : EFFICIENTNETV2

Ablation 1 | Effect of Data Augmentation

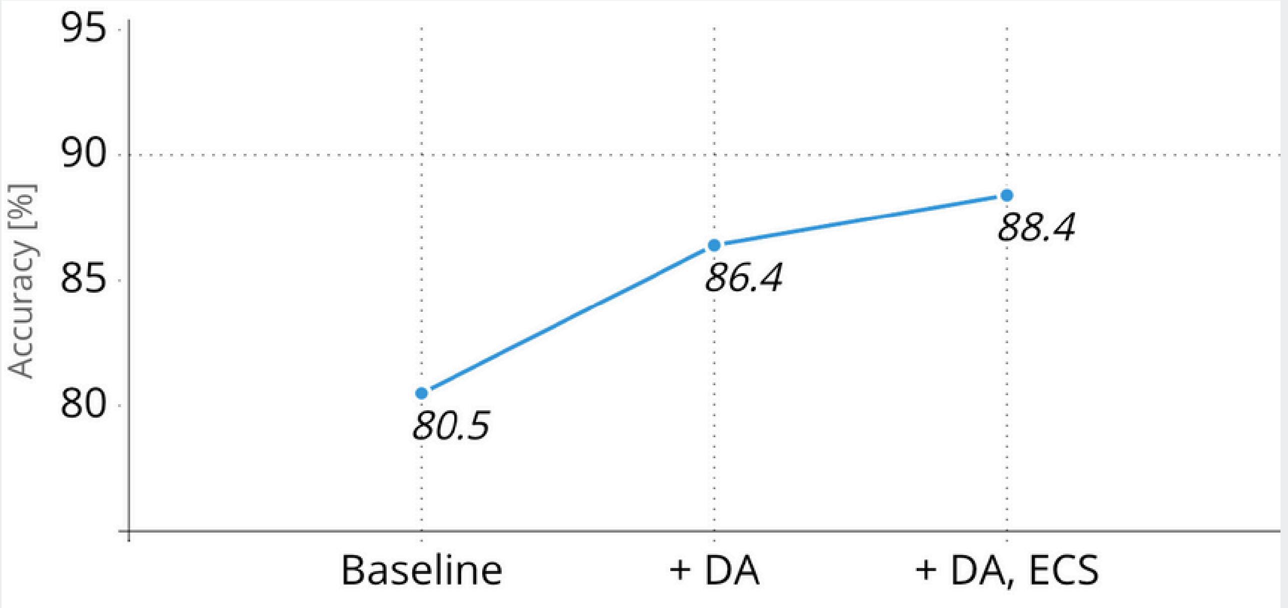
- Required more iterations to converge 50 → 100
- Overfitting alleviation: Validation loss 1.15 → **0.59**
- Acc. Increment : **5.9%** increase on Test Set (80.5 → 86.4%)

Ablation 2 | Effect of ECA Block

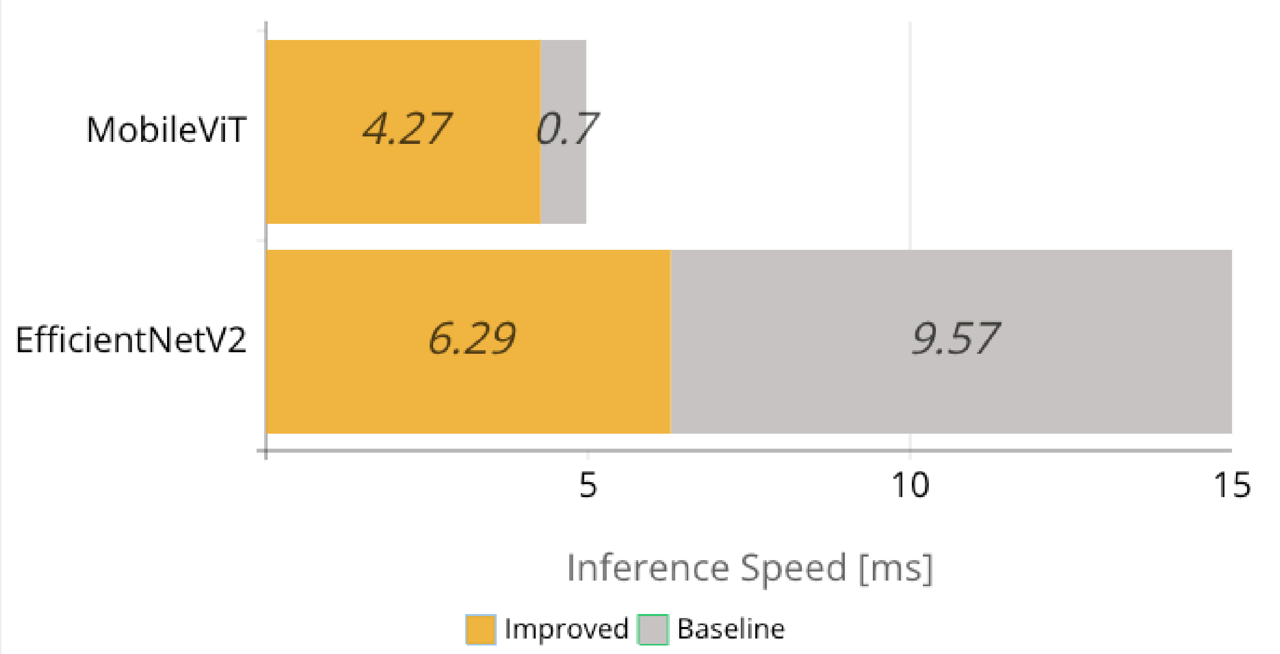
- Faster Infer. Speed : **44.6%,** (15.86ms → 8.79ms) on NVIDIA T4 GPU
- Memory Consumption

Ablation 3 | Effect of Leaky ReLU

- Faster Infer. Speed : **28%,** (8.79 ms → 6.29ms) on NVIDIA T4 GPU
- Memory Consumption



Ablation Study



Comparison of Inference Speed

RESULT ANALYSIS 02 : MOBILEVIT

Ablation 1 | Effect of Train Techniques

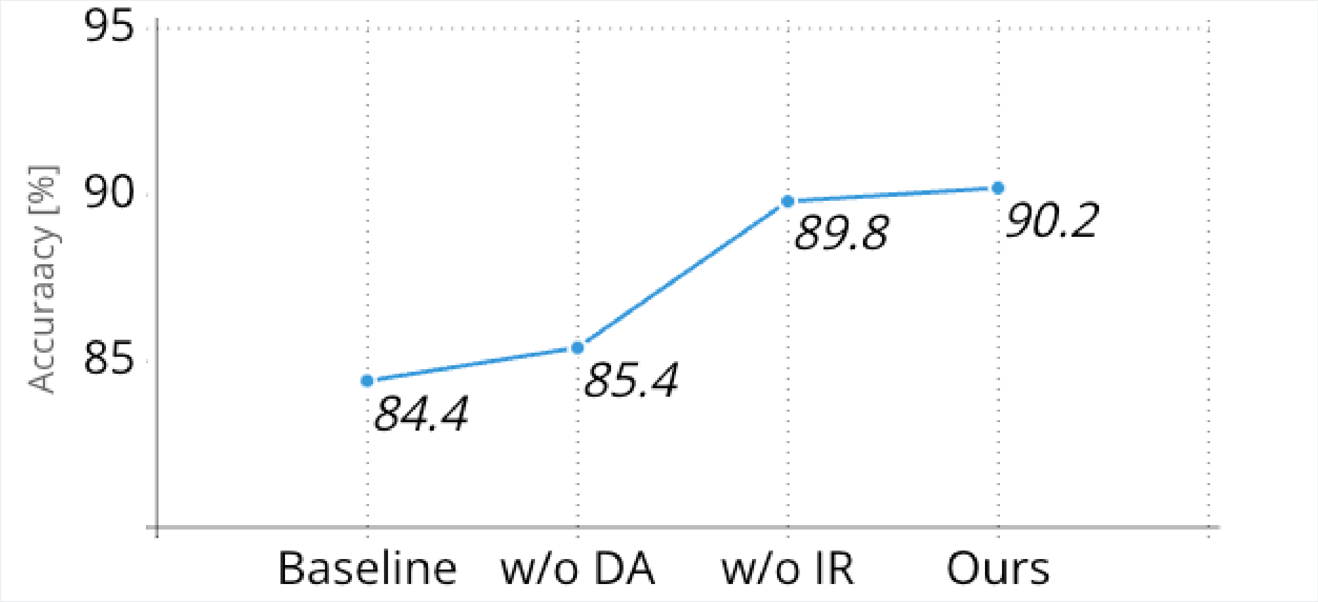
- Helps the model finetune to local minima before final convergence
- Acc. increment : about **1.0%**

Ablation 2 | Effect of Augmentation

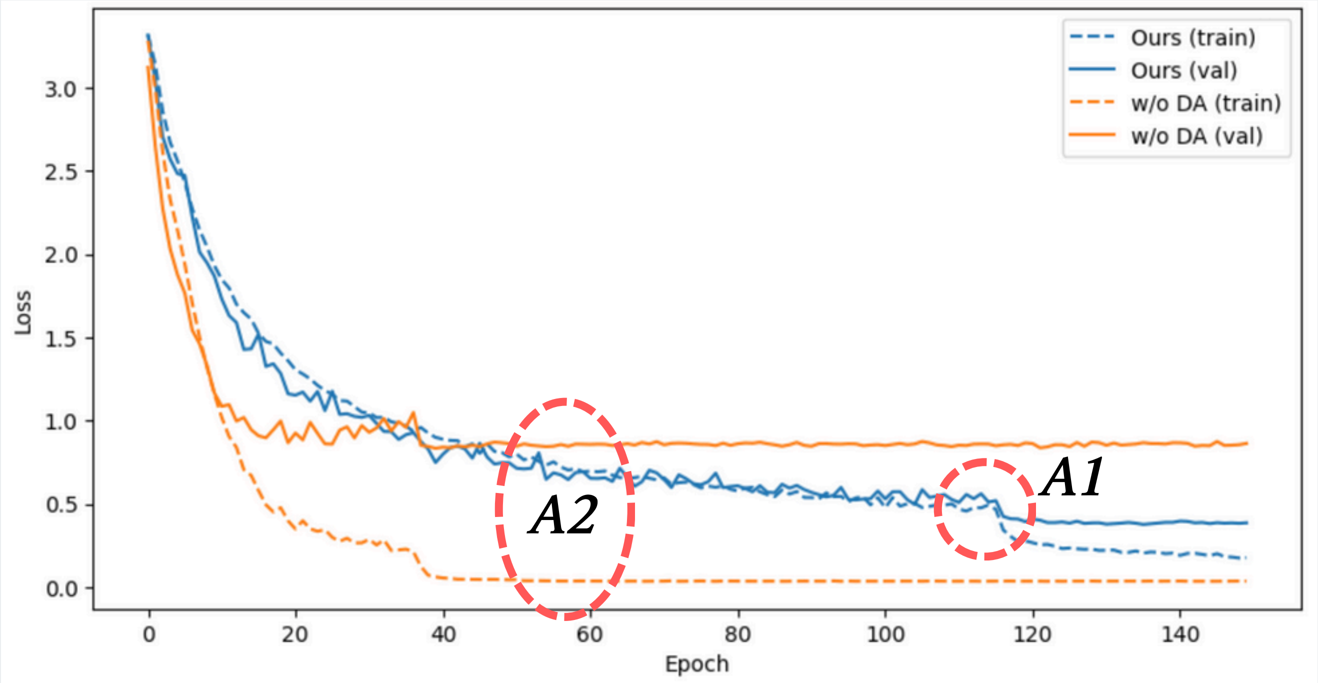
- Required more iterations to converge
- Overfitting alleviation : Shortened gap between train/val loss
- Acc. increment : about **4%**

Ablation 3 | Effect of Substitution to IR Blocks

- Faster Infer. Speed : **14.1%**, 4.97 -> 4.27(ms) / NVIDIA T4 GPU
- Final Accuracy : **90.2%**



Ablation Study



Visualization of Loss

CONCLUSION

Strong points

- Real time inference with appropriate accuracy (>*30 fps*)
- Support low computation devices (*even mobiles!*)

Use cases

- Industrial fields for recycling
- Private apartments

Limitation

- Should consider cases of multiple objects in a frame
- Consider rotation/occlusion of objects

[5] EBS 극한직업, *https://www.youtube.com/watch?v=T3l2aF0z6Bo&t=246s*텍스트

Group 9

Q / A

THANK YOU

APPENDIX - WORK DIVISION

1. Baseline Model Implementation

EfficientNetV2 : 백정은, 최해민 / PyramidNet + ASAM : 이상혁
MobileViT : 이동률 / Swin Transformer : 문강륜

2. Preparation of Presentation Materials

문강륜 백정은 이상혁 최해민

* Equal contribution, in Korean alphabetical order

3. Improvement of Key Models

EfficientNetV2

백정은 : Replaced SE with ECA and added ECA in Fused-MBconv block. Used Leaky ReLU.

- Accuracy Increment, Overfitting Alleviation, Faster Infer. Speed.

최해민 : Replaced SE block with GLU (Gated Linear Unit).

- Overfitting Alleviation, 2 Times Faster Infer. Speed at CPU and GPU.

이상혁 : Used ImageNet-1K pre-trained weights and added Attention module after feature extractor.

- 5 Times Faster Infer. Speed at GPU.

MobileViT

문강륜 : Replaced SA with IR, Apply stronger DA and additional train techniques.

- Accuracy Increment, Overfitting Alleviation, Faster Infer. Speed.

이동률 : Added 5x5 Conv layer in Local Representation of MobileViT Block.

- 2.4 Times Faster Infer. Speed at CPU.