Ablation-Aware Fine-Tuning of Lightweight Vision CNNs under Edge Constraints

Jimin Lee
University of Liverpool
sgjlee13@liverpool.ac.uk

Abstract

As Computer Vision becomes overwhelmingly popular for real-time use cases, edge devices such as micro-controllers and low-powered embedded systems are increasingly being used for inference on deep learning vision models. To meet strict memory and energy requirements, lightweight architectures such as MobileNetV2 are commonly used to fine-tune on specific subdomains of larger datasets. This paper investigates the benefits and drawbacks of using advanced data augmentation strategies (Mixup, CutMix, RandAugment) when applied to an ImageNet-100 subset. Experimental results show that ablation-based training achieves slightly lower accuracy (83.50% at epoch 30) compared to classical fine-tuning (84.96% at epoch 28) whilst requiring more epochs to converge. With inference efficiency staying constant between both models, results show that the increased cost at training introduced by ablations does not provide improved performance at inference. These findings suggest that, for smaller datasets and resource-constrained models, classical fine-tuning remains more efficient compared to ablation-aware techniques.

1 Introduction

Whilst deep vision neural neural networks achieve high accuracy with the flexibility to infer across a large class, due to the heavy memory requirements, this is infeasible on edge devices. MobileNetV2 is a widely used vision CNN optimised for mobile/embedded inference which is optimal for fine-tuning with smaller subsets on the larger pretrained dataset. However, in scenarios where only limited subsets of data are available, fine-tuned models risk overfitting and may not generalise effectively.

To address the shortcomings of smaller datasets, ablation-aware augmentation strategies have been proposed, including Mixup, Cutmix and RandAugment. These methods aim to improve the robustness of a model by synthesising a derivative of the training data which reduces overfitting by making memorisation harder thereby forcing the model to learn the general features of the data instead. Due to the nature of such techniques, models trained with ablations tend to converge slower and reduce accuracy during earlier epochs of training.

Ablation-aware techniques have been shown to provide benefits in accuracy in larger models with longer training operations however their effects on smaller architectures and datasets remain less clear. Furthermore, their significant computational expense during training raises further questions about their practical and real-world value with inference on edge hardware in mind especially when fine-tuned models rarely fit on the smaller memory sizes and thus are quantized before deployment which is shown to cause further accuracy losses.

In this work, we explore the effectiveness of ablation-aware fine-tuning compared to classical methods on a custom ImageNet-100 subset. Using MobileNetV2 as our architecture, we will evaluate the validation accuracy of both approaches. By viewing our results in the context of edge deployment, we aim to provide an insight as to whether the added computational expenses of augmentation are justified in light of its use in constrained environments.

2 Method

We investigate the effect of fine-tuning MobileNetV2 on a custom ImageNet-100 subset using different augmentation methods. Our goal is to evaluate how classical fine-tuning compares to ablation-aware fine-tuning in terms of accuracy for edge deployment.

We define a single training sample as (x, y), where $x \in \mathbb{R}^{W \times H \times C}$ is the input image and y is its corresponding one-hot encoded label. The model is represented by a function f_θ, and the loss function is $\mathcal{L}$ (in this case, cross-entropy).

Classical Training

In classical supervised training, the model learns by minimising the loss on individual, unaltered training samples. For each sample (x, y), the objective is to minimise the loss, calculated as:

$$\mathcal{L}_{\text{classical}} = \mathcal{L}(f_\theta(x), y) \tag{1}$$

The model's parameters, θ, are updated by minimising this loss over the entire training batch. The key point of importance here is that the input x and label y are used directly without modification.

Mixup

Mixup(1) creates new virtual training data by linearly interpolating two different samples from the training batch. This encourages the model to learn smoother decision boundaries. Given two random samples (x_i, y_i) and (x_j, y_j), a new sample $(\tilde{x}, \tilde{y})$ is generated as:

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j \tag{2}$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j \tag{3}$$

Here, the mixing coefficient λ is sampled from a Beta distribution, $\lambda \sim \text{Beta}(\alpha, \alpha)$, where α is a hyperparameter. The loss is then computed on this new virtual sample:

$$\mathcal{L}_{\text{mixup}} = \mathcal{L}(f_\theta(\tilde{x}), \tilde{y}) \tag{4}$$

CutMix

CutMix(2) also creates new samples from two training examples. However, it cuts a random rectangular patch from one image (x_j) and pastes it onto another (x_i). The labels are then mixed proportionally to the area of the patch. The new sample $(\tilde{x}, \tilde{y})$ is generated as:

$$\tilde{x} = \mathbf{M} \odot x_i + (\mathbf{1} - \mathbf{M}) \odot x_j \tag{5}$$

Here, $\mathbf{M} \in \{0, 1\}^{W \times H}$ is a **binary mask** indicating where to drop out pixels from x_i and fill in with pixels from x_j. $\mathbf{1}$ is a mask of all ones, and $\odot$ is element-wise multiplication. The new target label is a weighted combination based on the patch size:

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j \tag{6}$$

The mixing ratio λ is determined by the area of the cutout region. The loss is calculated on this composite sample:

$$\mathcal{L}_{\text{cutmix}} = \mathcal{L}(f_\theta(\tilde{x}), \tilde{y}) \tag{7}$$

RandAugment

RandAugment(3) does not combine multiple samples. Instead, it applies a sequence of N randomly selected data augmentation transformations with a magnitude M to a single image. For a given sample (x, y), the augmented input $\tilde{x}$ is created by applying the transformation sequence T:

$$\tilde{x} = T_{N,M}(x) = T_N(\dots T_2(T_1(x))\dots) \tag{8}$$

Where T_k is a transformation selected uniformly at random from a predefined set. The crucial point is that the **label remains unchanged**:

$$\tilde{y} = y \tag{9}$$

The model is then trained on this heavily distorted sample that preserves its label:

$$\mathcal{L}_{\text{randaugment}} = \mathcal{L}(f_\theta(\tilde{x}), y) \tag{10}$$

Training

The Adam (Adaptive Moment Estimation) optimizer was used with lr $= 1 \times 10^{-3}$. The scheduler was set to cosine annealing, batch size was 128 with mixed precision enabled via torch.cuda.amp. The accuracy and loss were computed on the validation split after each epoch.

3 Results and Discussion

Epoch	Classical	Ablated
1	71.10	67.02
5	76.10	72.40
10	79.06	76.46
15	81.76	78.46
20	84.04	80.90
25	84.68	82.86
28	84.96	83.38
30	84.84	83.50

Table 1: Validation accuracy (%) per epoch for classical fine-tuning and various ablation methods.

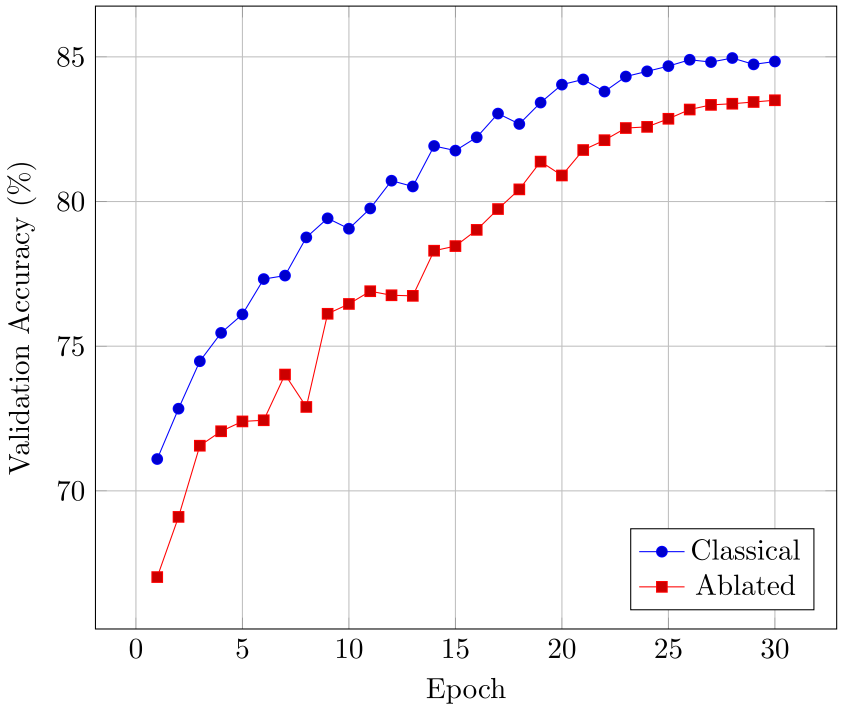

Figure 1: Validation accuracy per epoch for classical and ablation methods.

In the data we can see that the model trained with ablation-aware techniques was slower to converge and ultimately resulted in a lower valuation accuracy after 30 epochs of fine-tuning. However, throughout the training, the ablation-aware approach yielded better generalisation due to the smaller validation gap between training and valuation data. We can also confirm this as whilst the model with augmentation kept improving albeit at a lower rate, classical training plateaued early which indicated stronger generalisation with augmentation.

Accuracy-wise, ablation-aware techniques underperformed compared to classical training as using augmentations on the smaller dataset size likely served to make the supervision signal noisier. Whilst this works to increase generalisation in bigger datasets, in our case there was too little training data to get to the point of overfitting and thus failed to learn strong class-specific features quickly which is critical to the performance of small datasets.

Furthermore, MobileNetV2 is a relatively small model and thus has a lower capacity however the aggressive augmentations implemented in our training cycle demand that the model has enough parameters to decode noisy signals and extract the significant class boundaries which it failed to do.

In smaller models and datasets like ours, Mixup is likely to struggle as it assumes class linearity in creating mixed labels however with smaller datasets, the scale required to make meaningful augmentations of different samples is unlikely to be present. CutMix also assumes locality of discriminative features - such that cutting/pasting preserves features used to identify class boundaries - however in smaller datasets, the effects of these assumptions failing is drastically accentuated.

4 Conclusion

In this work, we evaluated the effect of augmentation-based ablation strategies and their effects on inference accuracy and evaluated this with regards to the increased computational overhead that is introduced with such techniques and their deployment on resource-constrained edge devices. Our results showed that classical fine-tuning marginally outperformed the ablation-aware approach on the ImageNet-100 subset, reaching a higher validation accuracy after 30 epochs. We attributed this to the small dataset size and the limited capacity of MobileNetV2 which is likely to have accentuated the noise introduced by the aggressive augmentation techniques.

Nevertheless, ablation-aware techniques were able to produce models that had stronger generalisation and so were likely to be more robust in real-world use which is often much more important especially in use cases where edge devices are used. The small trade-off in accuracy is justified by the much greater gains in regularisation. Whilst these results show that ablation-aware techniques do not always provide immediate benefits in accuracy, it is still important to acknowledge their effectiveness in larger datasets with longer training operations.

References

[1] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. *mixup: Beyond Empirical Risk Minimization.* arXiv preprint arXiv:1710.09412, 2017.

[2] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. *CutMix: Regularization Strategy*

to Train Strong Classifiers with Localizable Features. arXiv preprint arXiv:1905.04899, 2019. URL: `https://arxiv.org/abs/1905.04899`

[3] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. *RandAugment: Practical automated data augmentation with a reduced search space*. arXiv preprint arXiv:1909.13719, 2019. URL: `https://arxiv.org/abs/1909.13719`