Consolidating separate degradations model via weights fusion and distillation

Dinesh Daultani
Tokyo Institute of Technology
Tokyo, Japan
ddaultani@ok.sc.e.titech.ac.jp

Hugo Larochelle
Mila, Université de Montréal
Montreal, Canada
hugo.larochelle@umontreal.ca

Abstract

Real-world images prevalently contain different varieties of degradation, such as motion blur and luminance noise. Computer vision recognition models trained on clean images perform poorly on degraded images. Previously, several works have explored how to perform image classification of degraded images while training a single model for each degradation. Nevertheless, it becomes challenging to host several degradation models for each degradation on limited hardware applications and to estimate degradation parameters correctly at the run-time. This work proposes a method for effectively combining several models trained separately on different degradations into a single model to classify images with different types of degradations. Our proposed method is four-fold: (1) train a base model on clean images, (2) fine-tune the base model individually for all given image degradations, (3) perform a fusion of weights given the fine-tuned models for individual degradations, (4) perform fine-tuning on given task using distillation and cross-entropy loss. Our proposed method can outperform previous state-of-the-art methods of pretraining in out-of-distribution generalization based on degradations such as JPEG compression, salt-and-pepper noise, Gaussian blur, and additive white Gaussian noise by 2.5% on CIFAR-100 dataset and by 1.3% on CIFAR-10 dataset. Moreover, our proposed method can handle degradation used for training without any explicit information about degradation at the inference time. Code will be available at https://github.com/dineshdaultani/FusionDistill.

1. Introduction

Computer vision has been widely used in real-world applications nowadays. Considerable research has focused on the assumption that the images do not contain abnormalities and only ideal images. Real-world images frequently have different perturbations, like motion blur, noise (caused by low-light conditions), and compression, appearing in various digital versions of images/videos. Specifically, some computer vision domains face challenges with image degradation, leading to diminishing model performance or reliability. For example, self-driving systems in the form of adversarial attacks [1] can lead to accidents or safety-related issues, medical imaging due to additive white Gaussian noise incurred during acquisition [13] can lead to incorrect diagnosis of patients, remote sensing due to environment conditions such as clouds/low light [13] or atmospheric absorption and scattering [24] can lead to ineffective analysis. Hence, degradation in vision is often unavoidable, so it is crucial to handle degraded images properly. At the same time, often, the models trained are specific for a particular degradation. However, our study focuses on an essential aspect of this limitation: combining several models trained separately on individual degradations. To the best of our knowledge, this study is the first to investigate the method of combining separately trained degradation models into a single model for the classification of images with distinct types of degradation.

Figure 1 shows several approaches used in the later sections for comparison with our proposed method. The first method is when we have a separate model for each degradation; however, in that case, a single inferencing model cannot predict each degradation appropriately. Still, we include this approach as an Oracle to understand the performance expectation from separate models. Next, one of the most common ways in machine learning is to combine several weak learners that form a strong learner using ensemble [6, 10] methods. Hence, we also include the ensemble approach as a baseline where we combine individual models trained on each degradation using the ensemble approach. In the Vanilla fine-tuning (FT) method, we take a pre-trained clean image model for weight initialization and train a final model using all degradations. ModelSoups [26] and Fusing [4] are recent state-of-the-art approaches in pretraining and for out-of-distribution domains. On the other hand, the last approach is a block diagram to show our proposed method based on the fusion of individual model weights and distillation. We further explain our proposed

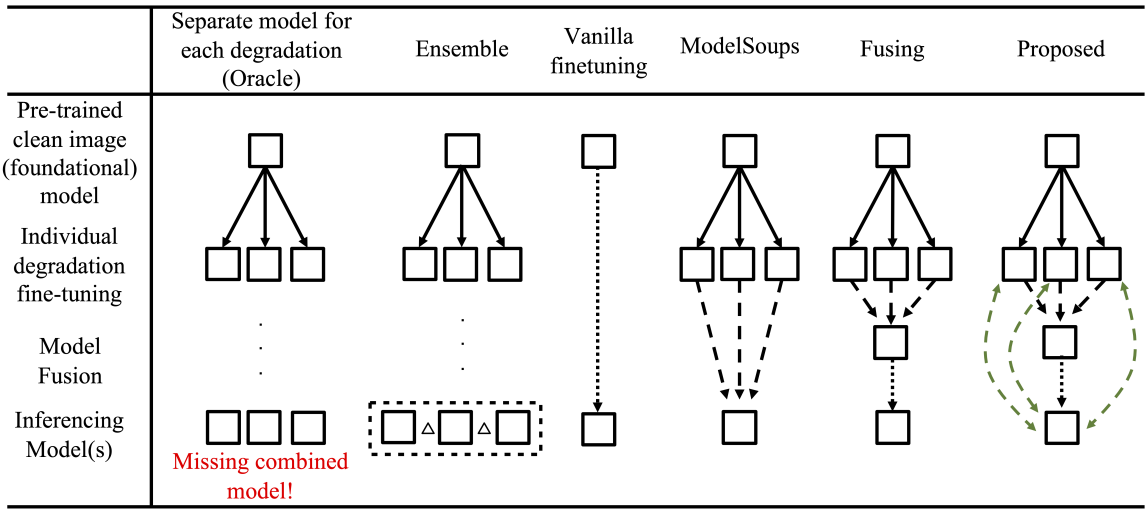


Figure 1. Variations of pretraining methods for combining several individual degradation models, where Model Ratatouille [21] study inspired this figure. The sequence of approaches is as follows: (1) Shows the traditional way to train several individual models for each degradation, i.e., Oracle, where a single model cannot classify several types of degradations correctly; (2) Ensemble method where we take ensemble (△) of all individual degradation models, (3) Vanilla fine-tuning (FT) where we start with pre-trained clean image model weights and fine-tune for all degradations at the same time, (4) ModelSoups [26] that average weights of individual degradation model, (5) Fusing [4] method goes one stage further from ModelSoups and involve fine-tuning on the target tasks, (6) Our proposed method. The arrow lines illustrated in the figure show different training/tuning processes. Solid arrow lines (⟶) represent individual degradation fine-tuning. Dashed arrow lines (- - - ▸) represent the averaging of model weights. Dotted arrow lines (····▸) represent combined degradations fine-tuning. Green dashed arrow lines (◂····▸) represent the distillation process.

method, FusionDistill, in the Section 3.

The primary contributions of our work are as follows:

1. We propose an effective method for combining separately trained image classification models of individual degradations based on the fusion of individual degradations model weights and distillation from individual degradations models.

2. We exhibit that our proposed method can consistently outperform previous state-of-the-art pretraining methods in out-of-distribution generalization tasks on consolidating several degradations, such as JPEG compression, Gaussian blur, additive white Gaussian noise, and salt-and-pepper noise.

3. Our proposed method can handle degradation used during training without any explicit information about the degradation.

4. Moreover, we demonstrate that the initialization of our model using pretraining methods, specifically the fusion of model weights, leads to better robustness.

In the following section, we share prior work related to our study in Section 2. Next, we share the details of our proposed method, including the proposed method architecture diagram and corresponding loss equation in Section 3. Later, we explain the experimental details such as image processing in Section 4.1, evaluation metrics in Section 4.2, and experimental setup for each approach compared in our work in Section 4.3. Results and corresponding analysis are discussed in Sections 5.1 to 5.3. Next, we perform an ablation study to analyze the impact of different pretraining weights on our proposed method in Section 6. Then, we discuss the limitations and assumptions of our study in Section 7, and at last, we summarize our work in Section 8.

2. Related work

Various works have explored the performance issue of degraded images for diverse computer vision tasks such as super-resolution, image restoration, and image classification. Specifically, Zhang *et al*. [27] uses a separate training/testing network for joint super-resolution tasks and either degradation such as blur, hazy, or rainy images. Several approaches explore the restoration of multiple degradations on images [18, 29]. Meanwhile, Zamir *et al*. [28] proposed an image restoration network using a multi-stage approach for individual degradation such as rain, blur, and noise.

For the image classification task, Endo *et al*. [8, 9] and Daultani *et al*. [5] have explored the classification of degraded images. However, their approach leads to a separate network for each degradation. Similarly, [11, 25] have explored model training using JPEG/JPEG 2000 degradation,

and their approach seems to perform inferiorly on unseen degradations. Pei *et al*. [20] explore the train and test of single degradation at individual or mixed degradation levels. All the methods described above for image classification of degraded images do not explore training on multiple types of degradation. In real-world applications, this leads to the ineffective deployment of multiple individual degradation models for particular degradations on resource-constrained applications such as self-driving vehicles. Furthermore, since a single degradation trains each model, it requires information about degradation to send specific images to the relevant degradation model at inference time. Our proposed approach can handle any image with the trained degradations without any explicit information about the degradation and can apply it to different types of computer vision tasks.

Pretraining methods have been widely popular nowadays in computer vision [3, 7] and natural language processing [2, 3, 7, 19, 30] to retain/transfer information from commonly used datasets, for example vision datasets such as CIFAR-100 [17] and ImageNet [23]. Recently, several approaches have explored how to combine a set of deep learning models trained on different hyperparameters for the same task/or out-of-distribution tasks rather than picking up the best performance model on the validation set. Specifically, the ModelSoups approach [26] demonstrates that averaging weights of several models with different hyperparameters trained on a particular task leads to better performance and robustness. Additionally, their approach is practical on several image classification and natural language tasks. Later, Choshen *et al*. [4] demonstrate that fine-tuning on a target task after averaging the weights of source tasks leads to better performance on several English text classification tasks. Concurrently, knowledge distillation [15] has shown a promising impact in transferring information from a teacher network to a student network [12]. In our study, we combine the pretraining methods with distillation to achieve better generalization and performance in image classification of degraded images.

3. Proposed method

Our proposed method is split into four steps as follows:

1. Train a base model μ_{clean} on clean images.
2. Fine-tune the base model μ_{clean} individually for each degradation deg, i.e., μ_{deg}.
3. Perform fusion of weights given the fine-tuned models μ_{deg} for individual degradations as σ.
4. Perform fine-tuning on all degradation images using distillation and cross-entropy loss as σ_{tuned}.

Figure 2 elaborates step-4 of our proposed method. Clean images are the inputs of specific teacher networks' μ_{deg_1}, μ_{deg_2}, ..., μ_{deg_N}. Consequently, we get T^1_{pred}, T^2_{pred}, ..., T^N_{pred} from respective teacher networks trained on individual degradations separately. Conversely, we apply respective degradation on all clean images and then input them to the student network σ_{tuned}. In total, we have N degradation modules for each degradation where Operator $\bigoplus$ represents the batching of inputs from all specific degradation modules deg_1, deg_2, ... deg_N. In our study, $N = 4$, since we combine the individual degradation models for four degradations, i.e., JPEG compression, Gaussian blur, additive white Gaussian noise, and Salt-and-pepper noise. However, our proposed method can deal with several individual degradation models. Additionally, we initialize our target student network σ_{tuned} weights using σ, which is the fusion of all teacher network weights.

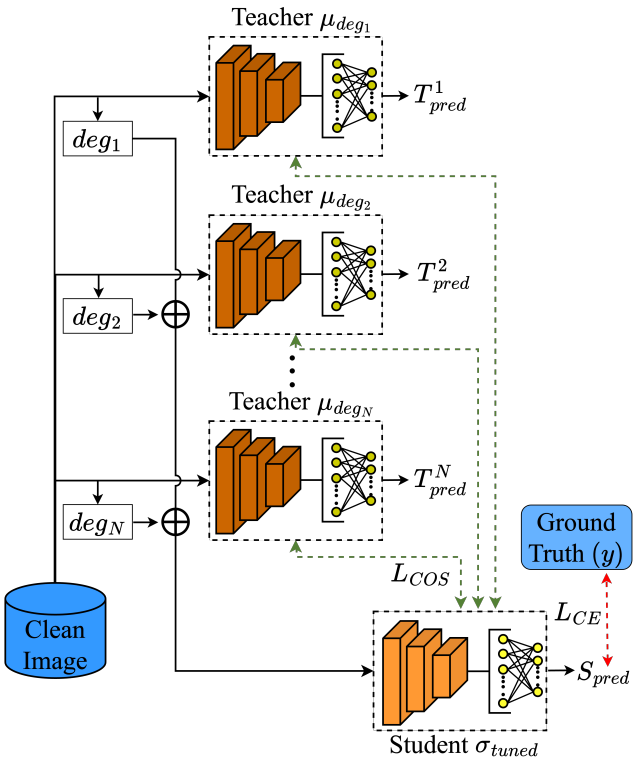


Figure 2. Fine-tuning of Student σ_{tuned} using fusion model initialized with weights σ and knowledge transfer from Teachers μ_{deg_1}, μ_{deg_2}, ..., μ_{deg_N} where N represents total individual degradations used for consolidation in the student network. Dark orange and dark yellow color blocks represent convolution and classifier blocks in the teacher networks where the weights are not tunable. On the other hand, light orange and yellow color blocks represent convolution and classifier blocks in the student network. The blue block represents input clean image data and ground truth. deg_1, deg_2, ..., deg_N represents degradation modules generating specific degradations *e.g*. JPEG compression where operator represents $\bigoplus$ represents the batching of degraded images.

Equation (1) shows the joint loss equation based on

cross-entropy loss and cosine similarity while training our proposed method as shown in Figure 2. α and γ represents weights of cross-entropy loss L_{CE} and cosine similarity loss L_{COS} respectively. L_{CE} is calculated based on the output of the student network, i.e., S_{pred} and the ground truth labels y. Accordingly, we perform a summation of all cosine similarity losses between the intermediate features from the target model Φ_{target} and Φ_{deg_i} for each degradation model. Based on the hyperparameter tuning of loss weights on the held-out validation set, we use $\alpha = 0.1$ and $\gamma = 1.0$, the same for all individual degradation models. A lower value of α makes the cross-entropy loss act like a regularization term between the labels y and S_{pred}; however, a higher value of γ helps to transfer knowledge from teacher network trained on individual degradations.

$$L = \alpha L_{CE} + \sum_{i=1}^{i=N} (\gamma_i L_{COS}(\Phi_{target}, \Phi_{deg_i})) \quad (1)$$

4. Experiments

4.1. Datasets

Similar to datasets used in [5, 9], we mainly use CIFAR-10 [17], CIFAR-100 [17], and Tiny Imagenet [23] datasets in our study. All the experiments have image augmentations similar to the ILIAC [5] method, i.e., random crop, random flip, and cutout for all approaches, since it is the current state-of-the-art method for image classification of degraded images. Moreover, we use a batch size of 128 for all experiments.

Meanwhile, it is an expensive and time-consuming process to capture, post-process, and annotate real-world images with degradations. We simulate different types of degradations on several commonly used datasets discussed above. Specifically, we simulate degradations on the clean images similar to [5, 9, 22] such as JPEG compression, Gaussian blur, additive white Gaussian noise, and salt-and-pepper noise, referred to as JPEG, Blur, AWGN, and SAPN respectively.

We derive degradation levels of JPEG compression from JPEG quality factors, specifically, $100 - Q$, where quality factors Q are uniformly sampled from 1 to 100 with a step size of 1. Next, Gaussian blur degradation levels represent the standard deviation of the Gaussian kernel ranging from 0 to 5 with a step size of 0.1. Additive white Gaussian noise degradation levels represent the standard deviation of the Gaussian distribution where degradation levels range from 0 to 50 with a step size of 1. Salt-and-pepper noise degradation levels represent the density of salt-and-pepper noise where degradation levels range from 0 to 0.25 with a step size of 0.1. Naturally, lower degradation levels refer to lower image degradation and vice versa.

4.2. Evaluation metric

In our work, we use a variation of the interval mean accuracy metric defined previously [8, 9] to measure the accuracy of degraded image classification. We define $\overline{Acc}$ metric as accuracy on a given particular degradation interval where degradation levels vary between the lower bound and upper bound, i.e., $q \in [Q_l, Q_u]$. Equation (2) represents the interval mean accuracy equation, where the given parameters include model M, input clean images X, and ground truth labels y. deg represents a specific degradation module. Furthermore, we fundamentally use $\overline{Acc}$(All) for measuring the performance on several degradations where "All" represents all degradation intervals specified for each degradation in the Section 4.1. For example, in the case of JPEG compression, $\overline{Acc}$(All) represents the interval mean accuracy between the lower bound Q_l of 0 and upper bound Q_u of 100.

$$\overline{Acc}(M, Q_l, Q_u) \stackrel{def}{=} \frac{\sum_{q=Q_l}^{Q_u} Acc(M(deg(X, q)), y)}{Q_u - Q_l + 1} * 100 \quad (2)$$

4.3. Experiment settings

To illustrate the difference between all comparison approaches, we show the main distinct parameters in the Table 1. The first column lists all comparison approaches, including our proposed approach, FusionDistill. The second column represents the pre-trained model, either used to initialize the model while training or to perform inferencing out-of-the-box. Oracle and Vanilla fine-tuning (FT) methods use clean image model C weights as initialization. The Ensemble and Scratch methods use individual degradation fine-tuned model I and random weights initialization R, respectively. ModelSoups and the Fusing method use fused individual degradation fine-tuned model weights F. FusionDistill uses I and F weights for teacher and student networks. The third column represents whether the fusion

Approach	Pre-trained model	Fusion	Combined deg FT	Loss	Final model size
Oracle	C	No	No	-	-
Ensemble	I	No	No	-	$N\times$
Scratch	R	No	Yes	CE	$1\times$
Vanilla FT	C	No	Yes	CE	$1\times$
ModelSoups	F	Yes	No	-	$1\times$
Fusing	F	Yes	Yes	CE	$1\times$
Ours	I and F	Yes	Yes	CE, COS	$1\times$

Table 1. Experiment details for all methods discussed in this study. Pretrained model variation includes random weights, clean image model, fused individual degradation model, and individual degradation model abbreviated as R, C, F, and I in the table.

of individual degradation fine-tuned model weights is utilized in the approach or not. Oracle, Ensemble, Scratch, and Vanilla FT methods do not utilize fusion. On the other hand, ModelSoups, Fusing, and FusionDistill do utilize fusion.

The next column indicates whether we use combined degradation fine-tuning or not. Oracle, Ensemble, and ModelSoups methods do not utilize combined degradation fine-tuning. Conversely, Scratch, Vanilla FT, Fusing, and FusionDistill utilize combined degradation fine-tuning. The next column represents loss functions applied during the fine-tuning phase of combined degradation images, where Scratch, Vanilla FT, and Fusing methods utilize cross-entropy (CE) loss. However, FusionDistill utilizes both cross-entropy and cosine similarity (COS) losses as shared in Section 3. Lastly, since the Ensemble method contains all individual degradation models, i.e., N, the inferencing model size is N times. In all other cases, the inferencing model size is only 1 times. Moreover, further experimental details are provided specific to each approach in the following sub-sections.

4.3.1 Common setup

For all the fine-tuning methods defined in Figure 1, we tune hyperparameters such as optimizer, learning rate, weight decay, and our proposed method loss weights on a 90%/10% train/validation split on the CIFAR-10 dataset. Specifically, we use the RAdam optimizer with a learning rate of 1×10^{-3} and weight decay of 1×10^{-4}. Furthermore, we use a multi-step learning rate scheduler with gamma of 0.2 and milestones of 30, 70, and 90. Consequently, we apply the same hyperparameters to all fine-tuning approaches, such as Vanilla FT, Fusing, and our proposed method. For combined degradation fine-tuning steps, the number of images used for training/testing becomes four times the usual size since we simulate the degradations for four types of degradations discussed in Section 4.1. Additionally, all the experiment results in this work are calculated based on three runs with different random seeds. Subsequently, Tables 2 to 4 shows the results with mean and standard deviation based on the three runs. We perform all the experiments on Pytorch library version $1.12.0$ and the Torchvision library $0.13.0$.

4.3.2 Base separate (Oracle)

The base separate method acts like an Oracle since this method contains four separate models trained on specific degradation. Out of the box, we use trained individual degradation models proposed in the ILIAC [5] approach as the Oracle method. Subsequently, we apply the same residual convolutional neural network, i.e., ResNet56 [14] network, for both teacher and student networks since it is comparatively lightweight and frequently used.

4.3.3 Ensemble

We apply the Ensemble method based on plurality voting [10] since it is a common strategy to combine outputs of the base learners, which typically results in enhanced performance. Besides, whenever there is a tie, the final output label is chosen based on the maximum sum of probabilities from softmax introduced by Kokkinos and Margaritis [16].

4.3.4 Vanilla fine-tuning and Scratch

To compensate for the total epochs used in the Fusing and FusionDistill, i.e., individual degradation fine-tuning and combined degradations after fusion, i.e., $100 + 100 = 200$ epochs. We train Vanilla FT and the Scratch methods for 200 epochs. Furthermore, we use a multi-step learning rate scheduler with gamma of 0.2 and milestones of 60, 140, and 180. Since the Scratch method is not a fine-tuning method, we apply SGD optimizer with a learning rate of 0.1, momentum of 0.9, and weight decay of 1×10^{-4}.

4.3.5 ModelSoups

ModelSoups [26] mainly proposed uniform soup and greedy soup recipes where uniform soup averages weights of several models, also used as term fusion in our study. On the other hand, the greedy soup recipe iteratively adds those individual models, which leads to better accuracy on a held-out validation set. Since uniform soup and greedy soup have comparable performance and uniform soup is a comparatively simple approach to incorporate in our problem setting, we primarily use the uniform soup method to perform a fusion of individual degradation models.

5. Results

5.1. Performance analysis for results on CIFAR-10 and CIFAR-100 datasets

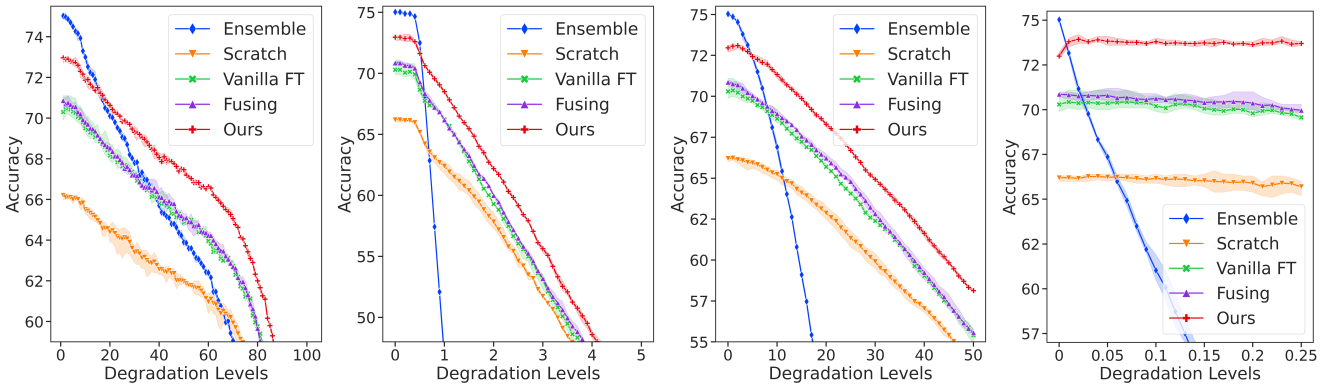
Table 2 shows the performance for comparisons approaches discussed in Section 1, i.e., Base separate (Oracle), Ensemble, Scratch, Vanilla FT, ModelSoups, Fusing, and our proposed method FusionDistill on the CIFAR-10 and CIFAR-100 datasets. For more details on the experiment settings for each approach, refer to the Section 4.3. The first row represents the Base separate method, i.e., the Oracle. The second row, i.e., the Ensemble method, performs well on JPEG compression with $\overline{Acc}$(All) of 84.9 and 59.3 on CIFAR-10 and CIFAR-100 datasets; however, it lags on other degradations in comparison with our proposed method. Hence, overall, the Ensemble method's performance is low compared to the other methods except the

Approach	CIFAR-10 Dataset: $\overline{Acc}$(All)					CIFAR-100 Dataset: $\overline{Acc}$(All)				
	JPEG	Blur	AWGN	SAPN	Avg	JPEG	Blur	AWGN	SAPN	Avg
Base separate (Oracle)	88.2 ± 0.1	84.5 ± 0.0	90.1 ± 0.1	$\underline{94.6} \pm 0.1$	89.4	63.4 ± 0.1	58.2 ± 0.1	65.4 ± 0.1	$\underline{74.1} \pm 0.2$	65.3
Ensemble	84.9 ± 0.1	47.9 ± 0.5	63.0 ± 1.3	87.1 ± 0.2	70.7	59.3 ± 0.1	25.3 ± 0.3	43.6 ± 0.2	58.7 ± 0.5	46.7
Scratch	86.3 ± 0.1	82.2 ± 0.1	87.8 ± 0.1	91.3 ± 0.1	86.9	59.1 ± 0.3	54.3 ± 0.2	61.0 ± 0.3	66.0 ± 0.3	60.1
Vanilla fine-tuning	87.6 ± 0.1	83.2 ± 0.1	89.3 ± 0.1	92.7 ± 0.1	88.2	62.2 ± 0.2	56.0 ± 0.1	63.8 ± 0.1	70.2 ± 0.4	63.1
ModelSoups	10.8 ± 0.9	10.8 ± 0.8	10.6 ± 0.7	10.9 ± 1.3	10.8	1.1 ± 0.1	1.0 ± 0.1	1.2 ± 0.2	1.0 ± 0.1	1.1
Fusing	87.7 ± 0.1	83.1 ± 0.1	89.4 ± 0.1	92.9 ± 0.1	88.3	62.3 ± 0.1	56.4 ± 0.1	64.1 ± 0.2	70.5 ± 0.2	63.3
FusionDistill (Ours)	$\underline{\mathbf{88.8}} \pm 0.1$	$\underline{\mathbf{84.8}} \pm 0.1$	$\underline{\mathbf{90.6}} \pm 0.1$	$\mathbf{94.1} \pm 0.1$	$\underline{\mathbf{89.6}}$	$\underline{\mathbf{64.5}} \pm 0.1$	$\underline{\mathbf{58.6}} \pm 0.1$	$\underline{\mathbf{66.4}} \pm 0.1$	$\mathbf{73.7} \pm 0.1$	$\underline{\mathbf{65.8}}$

Table 2. Performance evaluation for comparison approaches discussed in Section 4.3, applied to both CIFAR-10 and CIFAR-100 datasets on ResNet56 backbones. These datasets undergo assessment under four distinct degradations, i.e., JPEG compression, Gaussian blur, additive white Gaussian noise, and salt-and-pepper noise, denoted as JPEG, Blur, AWGN, and SAPN, respectively. The "Avg" column contains the average for the above four degradations. Moreover, results in **bold** and underline represent the best performance for combined degradation and separate/combined models, respectively.

ModelSoups method. Next, the Scratch method achieves fourth best results if we consider the Avg $\overline{Acc}$(All) for all degradations, i.e., the performance of 86.9 and 60.1 on CIFAR-10 and CIFAR-100 dataset. Vanilla FT performs decently on all degradations and achieves overall third-best performance. Since the ModelSoups method omits combined fine-tuning on all degradation, it leads to relatively low performance on both datasets. We cover more details on the analysis related to ModelSoups and Ensemble methods in section Section 5.3. The Fusing approach achieves almost similar performance as the Vanilla FT method. Meanwhile, our proposed method, FusionDistill, performs best on all degradation intervals simulated on CIFAR-10 and CIFAR-100 datasets. However, when considered along with the Oracle, just for Salt-and-pepper noise degradation, our proposed method is slightly lower, i.e., 0.5% on CIFAR-10 and 0.4% on CIFAR-100 dataset. It shows that a consolidated single model based on our proposed method can consistently achieve the best performance compared to all other methods.

Figure 3 represents the performance at specific relevant degradation levels for Ensemble, Scratch, Vanilla FT (FT), Fusing, and our proposed method (FusionDistill) on the CIFAR-100 dataset with JPEG compression, Gaussian blur, Additive white Gaussian noise, and Salt-and-pepper noise in Figures 3a to 3d respectively. We remove the ModelSoups [26] method in the comparisons of Figure 3 since the performance is relatively low, and removal of it leads to a better representation of the graph for proper analysis. First, Figure 3a shows our proposed method can outperform at mid to higher degradation levels, i.e., > 20 of JPEG compression. At lower degradation levels, i.e., < 20, the Ensemble method outperforms all other methods remarkably; however, with higher degradation, the performance significantly decays. Specifically, accuracy is even less than the Fusing and Vanilla FT method on higher degradation levels, i.e., > 40. Fusing method performance at all degradation levels is comparable to the Vanilla FT approach. The Scratch method performance is the lowest at the degradation levels < 65.



(a) JPEG compression (b) Gaussian blur (c) Additive white Gaussian noise (d) Salt-and-pepper noise

Figure 3. The accuracy of different approaches on ResNet56 backbone for the degradation of JPEG, Gaussian blur, AWGN, and salt-and-pepper noise in (a), (b), (c), and (d) respectively on CIFAR-100 dataset. The shaded area for each line shows the 95% confidence interval for a particular method over three runs with different random seeds.

Next, Figure 3b shows that our proposed method (FusionDistill) leads to the best performance on degradation levels > 0.6 on Gaussian blur degradation. The Ensemble method performs again remarkably on degradation levels < 0.6; however, degradation levels > 0.6 lead to severe performance deterioration as it achieves only $\overline{Acc}$(All) of 25.3 on Gaussian blur as shown in Table 2. Like JPEG compression results, Fusing and Vanilla FT lead to comparable performance on all the degradation levels. The Scratch method leads to lower performance on lower degradation levels < 2.5; however, at higher degradation levels, the performance is slightly lower than the Fusing and Vanilla FT methods.

Figure 3c demonstrates that our proposed method is the best at degradation levels > 6 on additive white Gaussian noise degradation. Similar to Figure 3b, the Ensemble method performs well on lower degradation levels, i.e., < 6; however, performance decays substantially at higher degradation levels > 6. Next, the Fusing method leads to slightly better performance at degradation levels < 30; however, the performance is almost like the Vanilla FT method at higher degradation levels. Scratch method consistently leads to relatively lower performance than Fusing and Vanilla FT methods. Though concurrently, the Scratch method outperforms the Ensemble method at degradation levels > 10.

Lastly, Figure 3d shows that our proposed method outperforms Fusing and Vanilla FT methods by around 1.5% accuracy on all degradation levels for the degradation of Salt-and-pepper noise. The Ensemble method performs reasonably only on clean images, i.e., degradation level 0; however, performance declines substantially at all other degradation levels. The Fusing method slightly outperforms the Vanilla FT method on all degradation levels. The Scratch method performs inferiorly, about 2% less in accuracy on all degradation levels than the Vanilla FT method.

5.2. Performance analysis for results on Tiny ImageNet dataset

To further evaluate our proposed method compared to other methods, we apply all approaches on the Tiny ImageNet dataset, and Table 3 shows the corresponding experimental results. The first row shows the results for the Base separate method, i.e., Oracle, based on four different models on each corresponding degradation. Next, the Ensemble method performs considerably well on JPEG compression; however, it performs erroneously on other degradations. Scratch method performs interestingly well on all degradations with Avg $\overline{Acc}$(All) of 53.9, falling behind only Vanilla FT with 54.9 and our proposed method 56.3. The Vanilla FT method provides the second-best performance with Avg $\overline{Acc}$(All) of 54.9. Next, the ModelSoups method leads to unsurprisingly relatively lower performance due to a lack of fine-tuning after the fusion process of weight averaging. The Fusing method yields a slightly lower Avg $\overline{Acc}$(All) of 53.8 compared to the Vanilla FT method, which has an Avg $\overline{Acc}$(All) of 54.9. Nevertheless, the Fusing and Vanilla FT methods' performance was similar on CIFAR-10 and CIFAR-100 datasets. At last, our proposed method performance is the best with Avg $\overline{Acc}$(All) of 56.3 compared to other approaches. The closest one is Vanilla FT with Avg $\overline{Acc}$(All) of 54.9, i.e., 1.4% lower. Overall, patterns of all methods on the Tiny ImageNet dataset are pretty analogous with the results on the CIFAR-10 and CIFAR-100 datasets as shown in Table 2 except the Scratch method's decent performance.

Approach	Tiny Imagenet Dataset: $\overline{Acc}$(All)				
	JPEG	Blur	AWGN	SAPN	Avg
Base separate (Oracle)	56.3	50.3	57.7	60.6	56.2
Ensemble	53.1	17.4	45.4	49.9	35.5
Scratch	54.0	48.2	55.3	58.2	53.9
Vanilla fine-tuning	55.2	48.1	56.6	59.7	54.9
ModelSoups	0.5	0.5	0.5	0.5	0.5
Fusing	54.2	47.1	55.2	58.8	53.8
FusionDistill (Ours)	**56.7**	**48.5**	**58.2**	**61.9**	**56.3**

Table 3. Performance evaluation for comparison approaches discussed in Section 4.3, applied to Tiny ImageNet dataset. These datasets undergo assessment under four distinct degradations, i.e., JPEG, Blur, AWGN, and SAPN. The "Avg" column contains the average for the above four degradations. Moreover, results in **bold** and underline represent the best performance for combined degradation and separate/combined models, respectively.

5.3. Performance of Base separate (Oracle) on all degradations

We examine the performance of the Base separate (Oracle) models on seen and unseen degradations using a heatmap as shown in Figure 4. Expectedly, the diagonal of the heatmap achieves the best $\overline{Acc}$(All) performance for all degradations since the diagonal represents the training and testing degradations to be the same. Next, we can observe that irrespective of the training degradation, if we perform the test on the degradation of JPEG compression, i.e., the first column in the heatmap, performance is satisfactory. Consequently, this provides rationale since the Ensemble method contains all four Base separate models and achieves reasonable performance on JPEG compression as shown in Tables 2 and 3; yet, it lags on other degradations. Additionally, if we train on AWGN, performance on SAPN is decent enough. However, in all other cases, the performance is inadequate. Therefore, we can conclude the discussion based on the above points: Base separate models struggle to handle degradations that are not used for training except for JPEG compression. Inherently, it leads to low performance in Ensemble and ModelSoups methods. Moreover, to achieve reasonable performance on all degradations, we

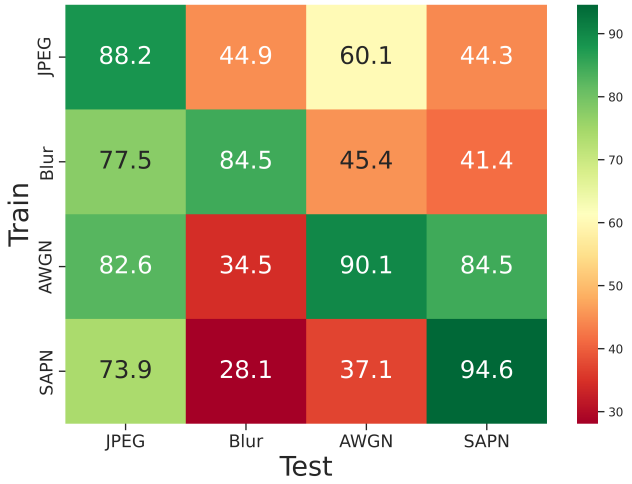


Figure 4. Heatmap that shows the $\overline{Acc}$(All) performance of Base separate (Oracle) models on CIFAR-10 dataset, with training and testing performed on particular individual degradations, representing the Y-axis and X-axis, respectively. For example, the cell value of 44.9 on the first row and second column represents the $\overline{Acc}$(All) performance of a model trained on JPEG compressed images and tested on Gaussian blurred images.

need to perform fine-tuning on all those degradations.

6. Ablation study: Effect of different pretraining weights

To test the effectiveness of initializing network in the step-4 of our proposed method with pretraining weights for the image classification of degraded images, we explore three types of pretraining weights, i.e., random initialization, clean image model initialization, and ModelSoups initialization represented as Random, Clean, and Fusion in the Table 4.

Weights initialization does play a role in performance improvement in our proposed method if we compare Random initialization with Clean initialization or Fusion initialization as shown in Table 4. Specifically, Clean initialization (row 2) leads to Avg $\overline{Acc}$(All) of 89.5 and Fusion initialization (row 3) of 89.6, which is 1.2% and 1.3% better as compared to Random initialization (row 1) respectively. It shows that pretraining weight initialization does lead to better robustness. However, Fusion and Clean initialization lead to comparable performance, i.e., Avg $\overline{Acc}$(All) of 89.5 and 89.6, respectively. Hence, as long as we initialize the weights with either of these methods, the main improvement in our proposed method is due to the distillation. At the same time, Fusion initialization does lead to slightly better performance than Clean image initialization. Therefore, our proposed method uses the Fusion method for the weights initialization.

Approach	CIFAR-10 Dataset: $\overline{Acc}$(All)				
	JPEG	Blur	AWGN	SAPN	Avg
Random	87.7	83.3	89.5	92.8	88.3
Clean	**88.8**	84.5	90.5	94.0	89.5
Fusion	**88.8**	**84.8**	**90.6**	**94.1**	**89.6**

Table 4. Performance evaluation for three variations in initialization of our proposed method, applied to CIFAR-10 dataset.

7. Limitation and assumptions

There are several limitations of our study, as follows:

1. Although our model can handle multiple degradations used during training at the inference time, we do not apply multiple degradations together on the same clean image. That means each image contains a specific type of degradation from the degradations such as JPEG, Gaussian blur, AWGN, and SAPN. We need to explore further when we apply multiple degradations on the same image and how it performs.

2. Since we simulate the degraded images using clean and degraded image pairs while training all the approaches explored in this study, the problem setting would be different if we did not have the clean and degraded image pairs available. Hence, we cannot directly apply our proposed method in those scenarios.

8. Conclusion

In this work, we proposed the FusionDistill method to combine separate models trained on individual degradations into a single model for the degraded image classification. Our proposed method consistently outperforms previous state-of-the-art methods for pretraining in out-of-distribution generalization, Ensemble, and Vanilla fine-tuning methods on CIFAR-10, CIFAR-100, and Tiny Imagenet datasets. Besides, we show that pretraining with fusion weights leads to better generalization than random and clean image model initialization. We can apply our proposed method to other computer vision tasks such as object detection, super-resolution, semantic segmentation, and image restoration of degraded images, showcasing its vast potential. We also exhibit that a model trained on one degradation typically performs poorly on unseen degradations. A few possible future directions are as follows: first, how to deal with multiple degradations in a single image rather than a single degradation in the current experiment setting using pre-trained network weights and how to transfer the information from trained individual degradation networks. Second, solve the clean and degraded image pairing limitation with preprocessed degraded images on separately held-out datasets.

References

[1] Mohammad R. Alam and Chris M. Ward. Adversarial examples in self-driving: A review of available datasets and attacks. In *2022 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–6, 2022. 1

[2] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. In *Conference on Empirical Methods in Natural Language Processing*, 2019. 3

[3] Feilong Chen, Duzhen Zhang, Minglun Han, Xiuyi Chen, Jing Shi, Shuang Xu, and Bo Xu. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20:38–56, 2022. 3

[4] Leshem Choshen, Elad Venezian, Noam Slonim, and Yoav Katz. Fusing finetuned models for better pretraining. *arXiv preprint arXiv:2204.03044*, 2022. 1, 2, 3

[5] Dinesh Daultani, Masayuki Tanaka, Masatoshi Okutomi, and Kazuki Endo. Iliac: Efficient classification of degraded images using knowledge distillation with cutout data augmentation. *Electronic Imaging*, 35(9):296–1, 2023. 2, 4, 5

[6] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000. 1

[7] Yifan Du, Zikang Liu, Junyi Li, and Wayne Xin Zhao. A survey of vision-language pre-trained models. In *International Joint Conference on Artificial Intelligence*, 2022. 3

[8] Kazuki Endo, Masayuki Tanaka, and Masatoshi Okutomi. Cnn-based classification of degraded images. *Electronic Imaging*, 2020(10):28–1–28–7, 2020. 2, 4

[9] Kazuki Endo, Masayuki Tanaka, and Masatoshi Okutomi. Cnn-based classification of degraded images without sacrificing clean images. *IEEE Access*, 9:116094–116104, 2021. 2, 4

[10] M.A. Ganaie, Minghui Hu, A.K. Malik, M. Tanveer, and P.N. Suganthan. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, oct 2022. 1, 5

[11] Sanjukta Ghosh, Rohan Shet, Peter Amon, Andreas Hutter, and André Kaup. Robustness of deep convolutional neural networks for image degradations. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2916–2920, 2018. 2

[12] Jianping Gou, B. Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789 – 1819, 2020. 3

[13] Bhawna Goyal, Sunil Agrawal, and BS Sohi. Noise issues prevailing in various types of medical images. *Biomedical & Pharmacology Journal*, 11(3):1227, 2018. 1

[14] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. 5

[15] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. 3

[16] Yiannis Kokkinos and Konstantinos G. Margaritis. Breaking ties of plurality voting in ensembles of distributed neural network classifiers using soft max accumulations. In Lazaros Iliadis, Ilias Maglogiannis, and Harris Papadopoulos, editors, *Artificial Intelligence Applications and Innovations*, pages 20–28, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. 5

[17] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 3, 4

[18] Minhua Liu, Yuanman Li, Rongqin Liang, Jiaxiang You, and Xia Li. Multiple degraded image restoration via degradation history estimation. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 528–533, 2023. 2

[19] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. *ArXiv*, abs/1908.08345, 2019. 3

[20] Yanting Pei, Yaping Huang, and Xingyuan Zhang. Consistency guided network for degraded image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 31:2231–2246, 2021. 3

[21] Alexandre Ram'e, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In *International Conference on Machine Learning*, 2022. 2

[22] Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. Effects of degradations on deep neural network architectures. *ArXiv*, abs/1807.10108, 2018. 4

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 3, 4

[24] Shuyin Tao, Huajun Feng, Zhihai Xu, and Qi Li. Image degradation and recovery based on multiple scattering in remote sensing and bad weather condition. *Opt. Express*, 20(15):16584–16595, Jul 2012. 1

[25] Sheng Wan, Tung-Yu Wu, Heng-Wei Hsu, Wing Hung Wong, and Chen-Yi Lee. Feature consistency training with jpeg compressed images. *IEEE Transactions on Circuits and Systems for Video Technology*, 30:4769–4780, 2020. 2

[26] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 23965–23998. PMLR, 17–23 Jul 2022. 1, 2, 3, 5, 6

[27] Zhang Xinyi, Dong Hang, Hu Zhe, Lai Wei-Sheng, Wang Fei, and Yang Ming-Hsuan. Gated fusion network for degraded image super resolution. *International Journal of Computer Vision*, pages 1 – 23, 2020. 2

[28] Syed Waqas Zamir, Aditya Arora, Salman Hameed Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration.

2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 14816–14826, 2021. 2

[29] Lujun Zhai, Yonghui Wang, Suxia Cui, and Yu Zhou. A comprehensive review of deep learning-based real-world image restoration. *IEEE Access*, 11:21049–21067, 2023. 2

[30] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guan Wang, Kaichao Zhang, Cheng Ji, Qi Yan, Lifang He, Hao Peng, Jianxin Li, Jia Wu, Ziwei Liu, Pengtao Xie, Caiming Xiong, Jian Pei, Philip S. Yu, and Lichao Sun. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *ArXiv*, abs/2302.09419, 2023. 3