Quality-Agnostic Image Recognition via Invertible Decoder

Insoo Kim[1] Seungju Han[1] Ji-won Baek[1] Seong-Jin Park[1] Jae-Joon Han[1*] Jinwoo Shin[2*]
[1]Samsung Advanced Institute of Technology (SAIT), South Korea
[2]Korea Advanced Institute of Science and Technology (KAIST), South Korea

Abstract

Despite the remarkable performance of deep models on image recognition tasks, they are known to be susceptible to common corruptions such as blur, noise, and low-resolution. Data augmentation is a conventional way to build a robust model by considering these common corruptions during the training. However, a naive data augmentation scheme may result in a non-specialized model for particular corruptions, as the model tends to learn the averaged distribution among corruptions. To mitigate the issue, we propose a new paradigm of training deep image recognition networks that produce clean-like features from any quality image via an invertible neural architecture. The proposed method consists of two stages. In the first stage, we train an invertible network with only clean images under the recognition objective. In the second stage, its inversion, i.e., the invertible decoder, is attached to a new recognition network and we train this encoder-decoder network using both clean and corrupted images by considering recognition and reconstruction objectives. Our two-stage scheme allows the network to produce clean-like and robust features from any quality images, by reconstructing their clean images via the invertible decoder. We demonstrate the effectiveness of our method on image classification and face recognition tasks.

1. Introduction

Deep learning models have shown remarkable performance for image recognition (or classification) tasks, even surpassing human-level performance [30, 41, 42, 17, 23, 4, 48, 22]. They typically assume high-quality (HQ) or clean images for their training/testing, while such an assumption may not hold in practice, e.g., images of various qualities can be encountered in their applications [9]. Moreover, deep models are known to be vulnerable to image distortions such as noise, blur, JPEG, contrast, weather, and low-resolution. On the other hand, the human visual system robustly extracts semantic information from such images due to its generalization ability [14].

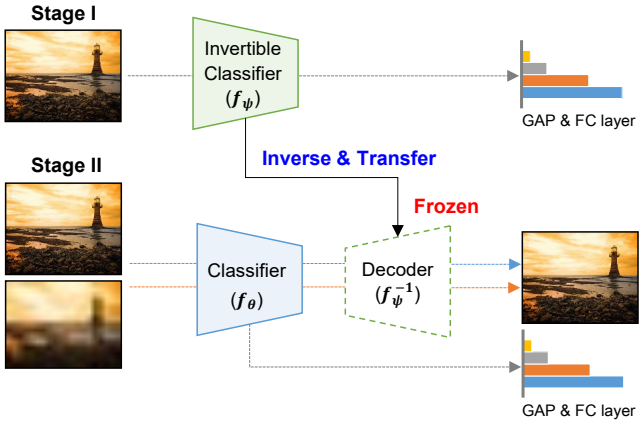


Figure 1. Quality-agnostic learning scheme based on classifier-decoder structure. We train an invertible classifier using a classification objective at the first stage. We use it as an invertible decoder of the second stage by its inversion and freezing the parameters. A new classifier is trained with this frozen decoder using HQ and LQ images at the second stage. As a result, the decoder evokes HQ features from both HQ and LQ input images.

In practice, data augmentation is a conventional and principled way to build models robust against various domains or corruptions of data.[1] For example, one can synthesize LQ images by using degradation procedures, such as blur, noise, and low-resolution, and then train a model using both HQ and LQ images. One can also train a model using HQ images, and then fine-tune it using both HQ and LQ images. However, training (or fine-tuning) with such diverse characteristics of data may generate the averaged distribution among corruptions [47], such that it results in an underfitted (or poor) model for a particular type of data, e.g., HQ images [14] or LQ images [44, 51, 10]. Indeed, learning a better single neural network handling such various types is a challenge to overcome.

To process various types of corruptions better, several quality-aware deep models have been studied [11, 2, 50]. They introduce an additional network module per quality type in order to handle multiple corruption types, and show

*Corresponding authors

[1]In this paper, we primarily assume that types of corruption or LQ images in test data are known at training time. Hence, our work is orthogonal to prior works building robust models without the assumption [20, 38, 19].

promising results with respect to both HQ and LQ images. However, as the number of corruption types increases, these approaches may require a large number of resources during the training and evaluation phases. Instead, we are interested in a more fundamental question: how to learn a single and unified model with high HQ and LQ accuracy, without using such additional quality-related modules?

To address the question, we propose a novel training scheme to build such a quality-agnostic model, i.e., having high performance on any input quality images. At a high level, the proposed training scheme encourages any quality images (even LQ) to have HQ-like features for the desired quality-agnostic performance. To this end, we suggest an additional training loss to reconstruct HQ images by using features of LQ images via an *auxiliary decoder*. Namely, we train the classification model using a multi-task learning strategy on the original (e.g., classification) task and the reconstruction task. However, in our early experiments, we found that training the classifier and the decoder in an end-to-end manner from scratch is not that effective for our purpose. This is because the additional reconstruction task may obstruct the original classification learning as their effective features may be different.

To effectively learn a classification model with the reconstruction decoder, we propose a *two-stage training scheme*. We first train a decoder with HQ images only, i.e., it takes some features as input and reconstructs the original HQ image as output. Then, we *freeze* the decoder and train a classifier-decoder architecture with both HQ and LQ images using two losses for classification and reconstruction. Under our scheme, LQ images are never fed to train the decoder. Hence, at the second stage, the classifier is encouraged to produce the features of LQ input images to be similar to those of HQ images by performing the reconstruction task well. To design a beneficial decoder for our two-stage strategy, we suggest training an *invertible* network under the classification objective at the first stage and use its *inversion* as a decoder of the second stage as shown in Fig. 1. As the decoder is trained for the classification objective at the first stage and is not updated at the second stage, the reconstruction task via the decoder can be regarded as a classification task in a backward perspective. Therefore, it can mitigate a potential conflict on our second multi-task learning stage. Furthermore, the decoder can reconstruct HQ images only from class-aware HQ features. Hence, under our training scheme, the classifier can be penalized by the reconstruction loss if the classifier does not output class-aware HQ features even in the case of LQ input images (see Fig. 2).

We demonstrate the superiority of our method under various tasks such as image classification and face recognition. Our extensive experiments show the effectiveness of our method on various benchmarks such as ImageNet [5], ImageNet-C [19], CFP-FP [39] and AgeDB-30 [34].

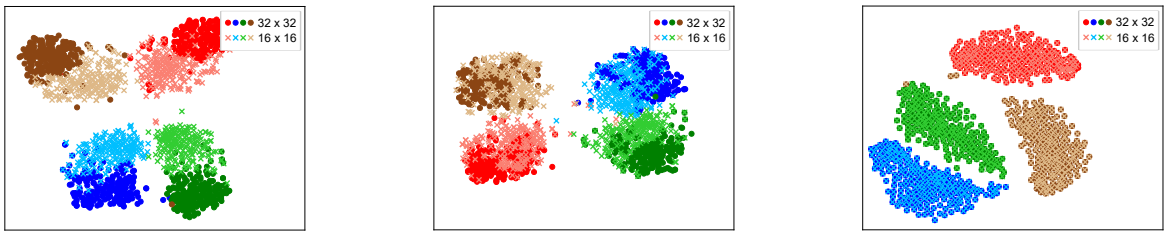
2. Related Works

Data augmentations and robustness. Data augmentation is a scheme to improve model performance on various quality images. A variety of data augmentation strategies has been studied to improve model robustness [20, 18, 13]. Another research line related to data augmentation attempts to inject noise to input images (or patches) for improving model robustness to noise corruptions as well as other types of corruptions [33, 38]. These data augmentation approaches assume that common corruptions or low-quality images in test scenarios are unknown. As a result, this may restrict performance improvement on the common corruptions. On the other hand, we exploit these common corruption types during the training because many types of corruptions are well-known and encountered in real-world scenarios. Furthermore, our method is specialized to learn a single model from those corruption types. Hence, it provides better robustness to the common corruptions.

Quality-aware methods. There have been efforts to learn a quality-resilient model. MixQualNet [11] presents a method that uses the predictions of multiple quality-expert networks. DeepCorrect [2] introduces additional units to particularly correct activation outputs of corrupted images so that their activation outputs are expected to be similar to clean activation outputs. Auxiliary Training [50] introduces auxiliary classifiers and selective batch normalization to effectively learn different types of corruptions. Each corruption type of input passes through its corresponding auxiliary classifier and batch normalization during the training. In the next step, it uses the knowledge transfer between primary and auxiliary classifiers, so that it enables only the use of primary classifier during the evaluation. Unlike relying on the individual components (e.g., expert network, correct unit, and auxiliary classifier) per corruption type, our method exploits the invertible decoder which is shared across all types of corruptions. In particular, our work covers 15 different corruption types [19] as a quality-agnostic model whereas they consider two or three corruption types, e.g., gaussian noise and blur during the training.

3. Quality-Agnostic Image Recognition via Invertible Decoder

In this section, we describe a new paradigm of training deep classification networks, that produces HQ-like features from any quality input images via an invertible decoder. This decoder imitates the human ability to imagine enhanced images of the same semantic information given limited information from any quality images. First, we describe the conventional recognition task of our interest in Section 3.1. Then, in Section 3.2, we introduce the necessity of our key component, invertible decoder, designed to



(a) (b) (c)

Figure 2. t-SNE feature visualization on the CIFAR-10 [29] validation set. (a) Naïve data augmentation, (b) Feature-based knowledge distillation [37], and (c) Proposed method. Note that we train CIFAR-10 with all classes and exhibit four classes of the validation set. The LQ images (16x16 resolution) are generated by bicubic down-sampling of the HQ images (32x32 resolution). The colored symbols are feature vectors, and the color of symbols denotes the corresponding class. The dark colors or "o" indicate HQ features, and light colors or "x" indicate LQ features. The results of (c) show that an individual LQ feature converges to its corresponding HQ feature by our method.

facilitate the classifier to have the quality-agnostic performance. The proposed method is summarized in Section 3.3. Finally, we argue that our method substantially differs from knowledge distillation methods in Section 3.4.

3.1. Preliminaries on image recognition

Consider a dataset $\mathcal{D} = \{(x, y)\}$, which consists of high-quality (e.g., clean or high-resolution) image $x \in \mathcal{X}$ and its corresponding label (or class) $y \in \mathcal{Y} = \{1, 2, \dots, c\}$. We also let $\tilde{\mathcal{D}} = \{(\tilde{x}, y)\}$ denote an image dataset of various low-qualities, where low-quality images can be obtained by augmenting from the high-quality dataset $\mathcal{D}$. Formally, one can write the relation between high-quality (HQ) and low-quality (LQ) images by

$$\tilde{x} = \varphi(x). \tag{1}$$

where φ is a corruption or down-sampling operation in the set of functions $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_k\}$. In general, the dataset $\tilde{\mathcal{D}}$ can contain HQ images as φ_k can be the identity function ($x_i = \varphi_k(x_i)$). We are interested in learning robust and accurate model parameterized by $\{\theta, w\}$ that outputs a final feature map $f_\theta(\tilde{x}) \in \mathbb{R}^d$ and a classification score $w^T\mathcal{H}(f_\theta(\tilde{x})) \in \mathbb{R}^c$, where $\mathcal{H}$ is a feature vector extractor (e.g., global average pooling). The model parameters θ, w can be jointly found by minimizing the following softmax loss:

$$L_{\texttt{softmax}}(\theta, w; \tilde{\mathcal{D}}) = \frac{1}{|\tilde{\mathcal{D}}|} \sum_{(\tilde{x},y)\in\tilde{\mathcal{D}}} \log \frac{e^{w_y^T\mathcal{H}(f_\theta(\tilde{x}))}}{\sum_{y\in\mathcal{Y}} e^{w_y^T\mathcal{H}(f_\theta(\tilde{x}))}}. \tag{2}$$

3.2. Key ideas: training with pre-trained decoder

When one trains a model with both HQ and LQ images by using (2) from scratch, several studies show that accuracy on HQ images is often decreased [50, 47]. This is because it is difficult to train diverse types of quality images in a single model, e.g., see Fig. 2 (a). To tackle this issue, one naïve solution is to use several specialized models in order to maximize performance on various types of quality images, e.g., a deep neural network trained with a single distortion alone (a specialized model) consistently outperforms humans under i.i.d train/test conditions [14]. However, maintaining several specialized networks leads to a large number of resources, and requires accurate quality assessment for a given image: a certain type specialized network gives a poor performance on other types [14]. We are interested in incorporating these specialized models into a single and unified model. Namely, we aim to learn a single model that has high performance on various types of quality images, comparable to those of the specialized models.

To this end, we consider the following *two-stage training scheme*. We train a specialized model f_ψ (parameterized by ψ) with only HQ images at the first stage. Then, we train a target classifier f_θ (parameterized by θ) with both HQ and LQ images by using the knowledge of HQ specialized model f_ψ. Namely, for training f_θ, we consider an additional loss (similar to one of knowledge distillation methods, i.e., FitNet [37]) to minimize the distance between the features in the two models as follows:

$$L_{\texttt{quality}}(\theta; \tilde{\mathcal{D}}) = \frac{1}{|\tilde{\mathcal{D}}|} \sum_{(\tilde{x},y)\in\tilde{\mathcal{D}}} d(f_\psi(x), f_\theta(\tilde{x})), \tag{3}$$

where x is the HQ image corresponding to an any quality image $\tilde{x}$ (e.g., HQ or LQ image) and d is some distance (or divergence) in the feature space, e.g., the ℓ_1 loss $\|f_\psi(x) - f_\theta(\tilde{x})\|_1$. Our intuition here is that the performance of the target classifier f_θ on any quality image $\tilde{x}$ becomes comparable to that of the specialized model f_ψ on HQ images, by minimizing (3). However, in our experiments, we found that minimizing the quality loss (3) by using ℓ_1 or ℓ_2 distance is not effective for our purpose: a relevant feature-level distance metric should be carefully designed for the effective transfer.

To tackle the challenge, we assume that f_ψ is *invertible* and use its inversion $f_\psi^{-1} : \mathbb{R}^d \to \mathcal{X}$ as an *invertible decoder*. Then, we suggest using the following loss consider-

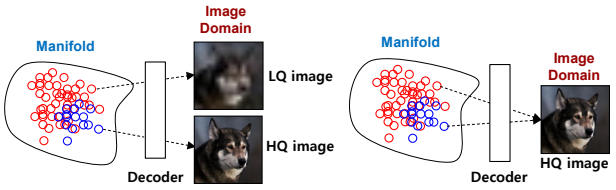


Figure 3. Illustration of (a) Injective property and (b) Non-injective property. The injective decoder ensures one-to-one mapping from features to images, such that the features are the same if their reconstructed images are identical.

ing a distance in the image domain:

$$L_{\texttt{quality}}(\theta;\tilde{\mathcal{D}}) = \frac{1}{|\tilde{\mathcal{D}}|}\sum_{(\tilde{x},y)\in\tilde{\mathcal{D}}} \|x - f_{\psi}^{-1}(f_{\theta}(\tilde{x}))\|_1, \quad (4)$$

where x is the HQ image corresponding to an any quality image $\tilde{x}$ (e.g., HQ or LQ image). Note that the decoder parameter ψ is already trained at the earlier stage and is *not updated* by (4). We highlight that all pixels have the same role and dynamic range in the image domain, whereas each feature dimension has a different dynamic range and some of the dimensions may be redundant. Therefore, considering a distance (such as ℓ_1) in the image domain is more relevant than doing it in the feature domain. This will be more discussed in Section 3.4. From the perspective of the decoder f_{ψ}^{-1}, it reconstructs HQ images only from class-aware HQ features. Hence, minimizing the loss (4) enforces the classifier f_{θ} to output class-aware HQ features even though LQ input images are given, as in Fig. 2 (c).

We remark that our invertible decoder has injective property. This property ensures 1:1 mapping from features to images as shown in Fig. 3 (a). By the property of 1:1 mapping, an HQ image can be only mapped to an HQ feature so that the target classifier is encouraged to generate an HQ-like feature from any quality input image. On the other hand, a non-injective decoder is able to map from different features to the same image (N:1 mapping) as shown in Fig. 3 (b). Hence, some features mapping to an HQ image may not be HQ-like features although one minimizes (4).

3.3. Quality-agnostic learning

We are now ready to present all the implementation details of our framework. Our method is based on the classifier-decoder architecture as shown in Fig. 1. We employ the existing backbone networks (i.e., ResNet50 [17]) as a target classifier. We adopt i-RevNet300 [25] as an invertible decoder, whose performance is comparable to that of ResNet50. To connect the classifier to the decoder, we introduce an additional 3x3 convolution. For more implementation detail, we present it in Appendix A.

To learn a quality-agnostic single classifier f_{θ}, the invertible decoder f_{ψ}^{-1} is required in advance. In the first stage,

Algorithm 1 Quality-Agnostic Learning

Input: Training set $\mathcal{D} = \{(x_i, y_i)\}$
Require: Invertible Classifier f_{ψ}, Class Weight w
while not converged **do**
  Sample a mini-batch $\{x_i, y_i\}_{i=1}^{n}$
  $z_i \leftarrow f_{\psi}(x_i)$
  $L_{\texttt{stage1}} \leftarrow L_{\texttt{softmax}}(z_i, y_i, w)$
  Optimize the parameters of ψ, w by $L_{\texttt{stage1}}$
end while
Input: Training set $\mathcal{D} = \{(x_i, y_i)\}$
Require: Classifier f_{θ}, Class Weight w
Require: Frozen Decoder f_{ψ}^{-1}
while not converged **do**
  Sample a mini-batch $\{x_i, y_i\}_{i=1}^{n}$
  Synthesize a corrupted mini-batch $\{\tilde{x}_i, y_i\}_{i=1}^{n}$
  $\tilde{z}_i \leftarrow f_{\theta}(\tilde{x}_i)$
  $\hat{x}_i \leftarrow f_{\psi}^{-1}(\tilde{z}_i)$
  $L_{\texttt{stage2}} \leftarrow L_{\texttt{softmax}}(\tilde{z}_i, y_i, w) + \lambda L_{\texttt{quality}}(\hat{x}_i, x_i)$
  Optimize the parameters of θ, w by $L_{\texttt{stage2}}$
end while

we obtain the decoder f_{ψ}^{-1} by training its inverse network f_{ψ} with HQ images by minimizing the following loss:

$$L_{\texttt{stage1}}(\psi, w; \mathcal{D}) = L_{\texttt{softmax}}(\psi, w; \mathcal{D}). \quad (5)$$

In the second stage, we form a classifier-decoder structure. The classifier f_{θ} is randomly initialized. The decoder f_{ψ}^{-1} is transferred from the first stage and is frozen. With the frozen decoder f_{ψ}^{-1}, we optimize a quality-agnostic classifier f_{θ} with HQ and LQ images from scratch by using the following multi-task learning loss:

$$\begin{aligned} &L_{\texttt{stage2}}(\theta, w; \tilde{\mathcal{D}}) \\ &\quad = L_{\texttt{softmax}}(\theta, w; \tilde{\mathcal{D}}) + \lambda \cdot L_{\texttt{quality}}(\theta; \tilde{\mathcal{D}}) \end{aligned} \quad (6)$$

where $\lambda > 0$ is a hyper-parameter. Although the decoder is used in the training phase, it is discarded at the evaluation phase. Hence, no additional inference cost is required.

In essence, the multi-task learning strategy may cause the conflict between the classification task and the reconstruction task due to the effect of negative transfer to each other. As the decoder is trained by the classification objective at the first stage and is no longer updated at the second stage, the reconstruction task minimizing (4) is viewed as the so-called classification task in a backward perspective. This mitigates the potential conflict on our multi-task learning at the second stage.

In summary, the proposed scheme is based on the two-stage learning framework using an invertible decoder and provides high performance on various types of quality images using a single model, comparable to that of each quality-specialized network. We describe the entire training procedure of our method in Algorithm 1.

3.4. Comparison to knowledge distillation methods

Most knowledge distillation (KD) methods including (3) are based on feature-level matching in their own ways [37, 36, 1, 43]. In the feature-based KD methods, it is hard to know which features (or channels) contribute to corruption components and how to minimize them for our purpose. Instead, we propose an image-based matching loss (4) utilizing an invertible decoder, where it is clear how pixels contribute to image corruption. Hence, this aspect allows our method to semantically minimize corruption components of individual pixels for generating HQ-like features. As shown in Fig. 2, we visualize the feature distributions on naïve augmentation, simple feature-based KD (3) and our method (4). In the feature-based KD as shown in Fig. 2 (b), HQ features appear to move toward LQ features by the constraint (3) when compared with naïve augmentation as shown in Fig. 2 (a). This results in no improvement on inter-class separability. On the other hand, LQ features converge to individual HQ features in our method as shown in Fig. 2 (c) so that inter-class separability is improved.

4. Experiments

4.1. Experimental setup

Common corruptions and evaluations. A common corruption evaluation dataset (e.g., ImageNet-C [19]) has been released to assess robustness of neural networks in image classification [19]. The ImageNet-C consists of 15 corruption types (gaussian/shot/impulse noise, glass/motion/defocus/zoom blur, contrast, elastic, JPEG, pixelate, frost, fog, snow, and brightness) as test corruptions and 4 corruption types (speckle noise, gaussian blur, spatter, and saturate) as holdout corruptions. Each corruption type contains five severity levels. Each corruption error (CE) is computed over five severity levels, that is $\texttt{CE}_c = \sum_{s=1}^{5} E_{c,s} / \sum_{s=1}^{5} E_{c,s}^{\texttt{AlexNet}}$. It represents a relative performance of the target model compared with AlexNet model. To measure total corruption errors, Mean Corruption Error (mCE) (%) is typically used. The mCE is the average of 15 test corruption errors (CEs), that is $\texttt{mCE} = \frac{1}{15} \sum_{c=1}^{15} \texttt{CE}_c$. For image classification tasks, we compute the mCE (%) as a metric of seen corruptions. Meanwhile, we calculate each average accuracy (%) of holdout corruptions over five severity levels as a metric of unseen corruptions. On the other hand, the assessment of face recognition tasks is done by measuring accuracy (%) based on the results calculated by cosine distance between given positive/negative features.

Image classification. We train several network architectures such as ResNet18 [17], ResNet50 [17] and ResNeXt101 [48] with ImageNet-1K [5]. Since our method can be adopted to those backbone networks, we refer to our methods as QualNet18, QualNet50 and QualNeXt101, respectively. To produce better clean accuracy of QualNet, we make a fine-tuning version of our methods as QualNet18* and QualNet50*, where QualNet18* and QualNet50* are fine-tuned from a pre-trained clean classifier[2] (unlike QualNet that are trained from scratch). We make a 200-class version of ImageNet-1K train/validation set and ImageNet-C for ablation study. Here, 200 classes of ImageNet-200 are drawn from those of Tiny-ImageNet [46]. We train with ImageNet-1K up to 90 epochs. The initial learning rate is set to 0.1 and it is divided by 10 at the $30_{\texttt{th}}$ and $60_{\texttt{th}}$ epochs. The batch size is set to 256.

Face recognition. In face recognition tasks, it would be challenging to learn a *large-margin discriminative* features via an *invertible* classifier in the first stage. To tackle the challenge, we suggest using the encoder-decoder structure in both stages as shown in Fig. D.1 in Appendix. This structure allows for training the classifier with large angular-margin and the decoder that reconstructs only from the large-margin features (see Appendix D for more detail). This modified architecture is referred to as QualNet50-LM. We train LResNet50A-IR [6] for CASIA-Webface training dataset [12]. The CASIA-Webface contains 0.49M face images collected from 10K subjects. The training dataset is cropped to 112x112 by a face detector, MTCNN [49]. We use CosFace [45] as an angular-margin softmax loss since it gives the best performance in our experiments. The initial learning rate is set to 0.1 and it is divided by 10 at the $18\texttt{K}_{\texttt{th}}$, $28\texttt{K}_{\texttt{th}}$, $36\texttt{K}_{\texttt{th}}$ and $44\texttt{K}_{\texttt{th}}$ iterations. The training is complete at the $47\texttt{K}_{\texttt{th}}$ iteration. The batch size is set to 256.

Methods. Naïve augmentation is a technique that synthesizes corrupted/low-resolution images from the original ones, and then trains with original and augmented ones. A specialized network is a method that trains and tests a certain corruption/resolution type data under i.i.d condition, which gives the performance guideline of a certain type [14]. For the target classifier, we remove max-pooling and replace the first convolution of stride 1 behind max-pooling with the convolution of stride 2, which produces slightly better performance on corrupted images. We use the modified architectures as default. We use 50% clean/high-resolution images and 50% corrupted/low-resolution images during the training unless otherwise specified. The corrupted images are uniformly sampled from 15 corruption types with five severity levels. For resolution-agnostic face recognition, the low-resolution images are also uniformly sampled from three resolution types. We use the hyperparameter of QualNet as $\lambda = 0.1$ and CosFace [45] as $s = 30, m = 0.25$ unless otherwise specified. For the knowledge distillation methods [21, 37, 36, 1, 43], the reported hyperparameters are used. In ablation study, we use the hyperparameter of QualNet as $\lambda = 0.6$.

[2]The pre-trained clean classifier is trained with clean images using the frozen invertible decoder.

Methods	Architecture	Gaussian Noise ↑ (%)	Gaussian Blur ↑ (%)
DeepCorrect [2]	ResNet18	60.33	58.21
QualNet18 (ours)		**60.52**	**61.05**

Methods	Architecture	Clean ↑ (%)	mCE ↓ (%)
AuxTraining [50]	ResNet18	69.94	78.86
QualNet18 (ours)		68.44	74.98
QualNet18* (ours)		**70.16**	**73.75**
Naïve Augmentation	ResNet50	74.88	52.35
Original KD [21]		73.92	56.73
KD FitNet [37]		74.75	52.73
KD PKT [36]		75.29	53.33
KD VID [1]		74.85	51.29
KD SP [43]		74.93	53.34
QualNet50 (ours)		**75.30**	**50.60**

Table 1. Comparison results against related works. We train with ImageNet-1K [5] and evaluate ImageNet-1K validation set and ImageNet-C [19]. "Clean" indicates Top-1 clean accuracy (%). "mCE" shows the performance (%) over 15 corruption types (less is better). Note that gaussian blur and noise are the average accuracy (%). "*" denotes the fine-tuning version of our method. The best results are indicated in bold.

4.2. Quality-agnostic image classification

Comparison to quality-aware methods. This experiment aims to show the performance improvement compared to the quality-aware methods. The baselines are chosen as DeepCorrect [2] and AuxTraining [50]. To fairly compare with DeepCorrect and AuxTraining, we follow the experimental setup of their literature. To compare with DeepCorrect, we follow the evaluation protocol of its literature where accuracy (%) of gaussian noise and blur is computed by averaging the accuracy of seven levels (including clean). We measure a Top-1 clean accuracy (%) and mCE (%) over 15 corruption types of ImageNet-C [19] in order to compare against AuxTraining. As shown in Table 1, our method provides better performance than the quality-aware methods. In particular, our method improves the accuracy of gaussian noise from 60.33% to 60.52%, and gaussian blur from 58.21% to 61.05%, compared to DeepCorrect. Furthermore, our method can be extended to a version of 15 corruption types without any additional modules, whereas those quality-aware methods require an additional module per corruption type. Even if they can add modules, they do not guarantee performance improvement over diverse corruption types.

Comparison to knowledge distillation methods. Since our method can be interpreted as a knowledge distillation method in the image domain, we investigate whether our method yields better performance than other knowledge distillation methods [21, 37, 36, 1, 43] under the same knowledge. Namely, we use our invertible classifier trained at the first stage as a teacher network. Note that KD FitNet [37] is implemented by using (3). We measure a Top-1 clean accuracy (%) and mCE (%) over 15 corruption types of ImageNet-C [19]. As shown in Table 1, we found that the knowledge distillation methods are not beneficial to improve performance when compared to the naïve augmentation. As discussed in Section 3.4, most knowledge distillation methods are not effective to generate HQ-like features as in Figure 2. They rely on feature-level matching in different ways and their HQ features are prone to become LQ features, which is opposed to our purpose. Therefore, they are limited to improve performance as reported in Table 1.

Methods	Architecture	Clean ↑ (%)	mCE ↓ (%)
Vanilla	ResNet18	**70.33**	87.10
Naïve Augmentation		67.88	63.66
QualNet18 (ours)		68.42	61.66
QualNet18* (ours)		69.91	**60.31**
Vanilla	ResNet50	76.28	78.20
Naïve Augmentation		74.88	52.35
QualNet50 (ours)		75.30	**50.60**
QualNet50* (ours)		**76.74**	51.33
Vanilla	ResNeXt101 -32x8d	79.64	69.84
Naïve Augmentation		79.55	43.38
QualNeXt101 (ours)		**79.84**	**42.50**

Table 2. Comparison results over several network architectures. We train with ImageNet-1K [5] and evaluate ImageNet-1K validation set and ImageNet-C [19]. "Clean" indicates Top-1 clean accuracy (%). "mCE" shows the performance (%) over 15 corruption types (less is better). "*" denotes the fine-tuning version of our method. The best results are indicated in bold.

Quality-agnostic models. We investigate our method to show the consistency of performance improvement across several network architectures such as ResNet18 [17], ResNet50 [17] and ResNeXt101-32x8d [48]. Once our decoder is trained at the first stage, it can easily be connected to any classifier architectures at the second stage. In the experiments, we choose a vanilla model (trained with clean images only) and naïve data augmentation model (trained with clean and corrupted images) as baselines. As reported in Table 2, our method provides further performance improvement in both clean and corruption types over various network architectures, compared with its counterpart (naïve augmentation). In particular, our method achieves the mCE from 63.66% to 61.66% and clean accuracy from 67.88% to 68.42%, compared to the naïve augmentation for ResNet18. Nevertheless, the QualNet18/50 underperform the corresponding vanilla models with respect to the clean accuracy. We believe that the network capacity of ResNet18/50 may be insufficient to train diverse quality types from scratch, but sufficient to train with clean images only (vanilla model). As a result, QualNet18/50 may learn a deficient model on clean type. On the other hand, the performance of 79.84% (clean accuracy) and 42.50% (mCE) for QualNeXt101 is better than that of the corresponding vanilla (79.64%) and naïve augmentation (43.38%) since

Methods	Top-1 Accuracy (%)			
	Spatter	Saturate	Speckle Noise	Gaussian Blur
Vanilla	58.11	65.92	44.88	44.75
Naïve Augmentation	61.28	68.15	63.13	55.73
QualNeXt101 (ours)	**62.48**	**69.19**	**64.21**	**57.24**
AugMix [20]	53.27	61.48	50.61	47.22
ANT [38]	52.41	61.30	58.15	43.10
QualNet50 (ours)	**54.04**	**62.28**	**63.13**	**50.54**

Table 3. Accuracy (%) on unseen corruption types. We train ResNeXt101-32x8d [48] for the upper three methods. The lower three methods are based on ResNet50 [17]. We use ImageNet-1K [5] as a training set. The best results are indicated in bold.

the network capacity is large enough. To further improve clean accuracy for QualNet18 and QualNet50, we conduct our fine-tuning version (QualNet18* and QualNet50*) as described in Section 4.1. We achieve the clean accuracy of QualNet50* (76.74%) even better than that of the corresponding vanilla models (76.28%), but not in the case of QualNet18* (see Table 2). This is due to the small network capacity of ResNet18 to learn diverse quality images in a single model.

4.3. Model robustness

Robustness to unforeseen corruptions. Adversarial Noise Training (ANT) [38] demonstrated that noise corruptions help increase the robustness to other common corruption types. This means that we can achieve better generalization to unforeseen corruption types by augmenting diverse quality images during the training as in [20, 38, 18]. Basically, our framework encourages a model to be effectively learned with various quality images, such that we can also expect that it achieves further performance improvement with respect to unforeseen corruption types. To confirm this, we evaluate performance on unforeseen corruption types such as spatter, saturate, speckle noise, and gaussian blur in ImageNet-C benchmark [19]. We use ResNeXt101-32x8d as a classifier in this experiment. As shown in the upper side of Table 3, the naïve data augmentation results in better robustness than the vanilla model (trained with clean only). This implies that data augmentation with various corruption types helps improve the robustness. We also observe that our framework improves accuracy from 61.28% to 62.48% on spatter, 68.15% to 69.19% on saturate, 63.13% to 64.21% on speckle noise, and 55.73% to 57.24% on gaussian blur, compared to the naïve augmentation. As a result, our framework contributes not only a quality-agnostic (seen) model as discussed in Section 4.2 but also a well-generalized (unseen) model.

Comparison to other data augmentation methods. In this experiment, we compare our method with the state-of-the-art data augmentation methods such as AugMix [20] and Adversarial Noise Training (ANT) [38] regarding the robustness of unseen corruptions. We train our method (QualNet50) with 15 corruption types and download the released models (ResNet50) for AugMix and ANT methods to evaluate performance on unforeseen corruption types such as spatter, saturate, speckle noise, and gaussian blur in ImageNet-C benchmark [19]. As shown in the lower side of Table 3, our method provides better accuracy of all unseen corruption types than the other data augmentation methods. Specifically, our method improves the performance from 50.61% to 63.13% on speckle noise, and 47.22% to 50.54% on gaussian blur, 53.27% to 54.04% on spatter, and 61.48% to 62.28% on saturate, compared to AugMix.

4.4. Resolution-agnostic face recognition

Comparison to specialized networks. The goal of this experiment is to verify the performance of our method that converges to those of the resolution-specialized models in face recognition tasks. The baselines are chosen as resolution-specialized networks and naïve data augmentation. In the training, we augment low-resolution images such as 56x56, 28x28, and 14x14 from the original images 112x112 by using the nearest interpolation and then up-sample those low-resolution images to the original size. For evaluation on low-resolution face benchmarks, we generate low-resolution versions of CFP-FP [39] and AgeDB-30 [34]. AgeDB-30 includes $6,000$ positive and negative pairs of face images in age variations. CFP-FP contains $7,000$ positive and negative pairs of face images in the frontal-profile configuration. Each specialized network offers the performance guideline for a certain resolution [14], which is referred to as target performances of our method. As shown in Table 4, our method mostly achieves the individual performance of the specialized models. Furthermore, some of the results show the performance of our method even better than that of the specialized networks. This is valid to the condition that the model is well-trained with augmented data since data augmentation helps improve performance. On the other hand, the naïve augmentation model appears to be an underfitted model that discourages the performance on some resolution types.

Robustness to realistic low-quality images. We have demonstrated that our method is robust to seen corrupted images and unseen corrupted images (both synthesized). In practice, it is more important that the model is robust to realistic low-quality images. As discussed in Section 4.3, when the model considers a diversity of known corruption types in the training phase, one can naturally expect that it can be also robust against unknown corruption types (not only synthetic, but also realistic). To confirm this, we evaluate a realistic low-resolution test dataset, e.g., TinyFace [3]. This dataset is drawn from realistic low-resolution faces, not synthesized by artificial down-sampling of high-resolution images. To evaluate TinyFace test set, we use the mod-

Methods	Training Resolutions	Accuracy (%) on Test Resolutions							
		112x112		56x56		28x28		14x14	
		AgeDB-30	CFP-FP	AgeDB-30	CFP-FP	AgeDB-30	CFP-FP	AgeDB-30	CFP-FP
Specialized Network (Target Performance)	112x112	92.37	95.71	-	-	-	-	-	-
	56x56	-	-	91.25	95.34	-	-	-	-
	28x28	-	-	-	-	86.00	92.67	-	-
	14x14	-	-	-	-	-	-	74.20	84.80
Naïve Augmentation	All	90.32 (-2.05)	95.09 (-0.62)	89.82 (-1.43)	94.99 (-0.35)	85.85 (-0.15)	92.49 (-0.18)	72.48 (-1.75)	**84.64 (-0.16)**
QualNet50-LM (ours)		**92.22 (-0.15)**	**96.24 (+0.53)**	**91.27 (+0.02)**	**95.81 (+0.47)**	**87.05 (+1.05)**	**93.03 (+0.36)**	**73.20 (-1.00)**	84.51 (-0.29)

Table 4. Accuracy (%) on face benchmarks such as AgeDB-30 [34] and CFP-FP [39] in four resolution types. We train LResNet50A-IR [6] with CASIA-Webface dataset [12] for all methods. We use the hyperparameter ($\lambda = 1.0$) of our method in this experiment. Each specialized network is trained and tested with a certain resolution type, which provides the performance guideline on a certain resolution type. Naïve augmentation model and our method are trained with those resolution types in a single network. "QualNet50-LM" is different from QualNet, which is described in Section 4.1. The performance difference between naïve augmentation/ours and the corresponding specialized network is denoted in brackets. The best results are indicated in bold.

Methods	Resol.	Rank-1	Rank-5	Rank-10
Specialized Network	112x112	24.46	32.64	36.48
Naïve Augmentation	All	32.97	40.91	44.31
QualNet50-LM (ours)	All	**35.54**	**44.45**	**47.42**

Table 5. Open-set face identification results (%) on realistic corruption types. We evaluate a realistic low-quality dataset (TinyFace [3]) with the models trained for Table 4. "Resol." denotes the training resolution. "QualNet50-LM" is described in Section 4.1. The best results are indicated in bold.

els trained for Table 4 where the models are trained with CASIA-WebFace [12]. The result is given in Table 5. Although we use *synthetic* data augmentation during the training for naïve augmentation model, it results in more robust model to *realistic* corruptions (24.46% to 32.97% for Rank-1 accuracy). Furthermore, we observe that QualNet50-LM provides additional improvement from 32.97% to 35.54% for Rank-1 accuracy. This accounts for the importance of data augmentation for diverse image qualities as well as the importance of its effective learning framework (ours).

4.5. Ablation study

In this section, we decompose our method into three main modules: *decoder*, *two-stage learning strategy*, and *invertible attribute*. We investigate what module significantly contributes to performance improvement. According to Table 6, no module means the naïve data augmentation. Adding a decoder represents the classifier-decoder architecture and we train it with HQ and LQ images in an end-to-end manner. Adding a decoder and two-stage learning means that the decoder is obtained by training an encoder-decoder architecture with HQ images at the first stage, and then we train a new classifier-decoder (transferred and frozen) architecture with HQ and LQ images at the second stage. Adding a decoder, two-stage learning and invertible property represent our method (QualNet) where the invertible decoder is obtained by training its inverse network (i.e., invertible classifier) with HQ images at the first stage, and then we train a new classifier-decoder (frozen) architecture with HQ and LQ images at the second stage. For all models,

With Decoder		✓	✓	✓
Two-Stage Learning			✓	✓
With Invertible Attribute				✓
Clean ↑ (%)	80.45	78.68	81.81	**82.33**
Corruptions ↑ (%)	70.82	69.91	71.64	**72.30**

Table 6. Ablation study on our modules. We train ResNet50 [17] (classifier) with ImageNet-200 and evaluate 200-class versions of ImageNet-1K validation set and ImageNet-C [19]. "Clean" and "Corruptions" indicate Top-1 clean accuracy (%) and average accuracy on 15 corruption types, respectively. The best results are indicated in bold.

we augment 15 corruption types as described in Section 4.1 during the training. The results are shown in Table 6. Simply introducing the decoder to a classifier leads to lower performance than its counterpart (naïve augmentation) since the additional reconstruction objective may obstruct to learn class-aware features which is necessary for classification, i.e., two tasks are in conflict. This explains the necessity of the elaborated learning strategy under multi-task learning. Namely, applying a two-stage learning strategy on top of classifier-decoder architecture improves the accuracy from 78.68% to 81.81% for clean and 69.91% to 71.64% for corruptions as shown in Table 6. Furthermore, we observe that adding the invertible attribute, i.e., using the inverse of the invertible classifier as a decoder (QualNet), achieves the best clean accuracy of 82.33% and corruption accuracy of 72.30%. For more insights on our method, we perform more ablation studies in Appendix C.

5. Conclusion

In this paper, we propose a new training framework to produce high performance on any quality images. We train our method with diverse corruption types and multiple resolution types, and demonstrate its effectiveness on various benchmarks of image classification and face recognition. We hope that our method will be extended to other applications such as adversarial robustness, masked face recognition and low-level vision tasks.

References

[1] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9155–9163, 2019.

[2] Tejas S Borkar and Lina J Karam. Deepcorrect: Correcting dnn models against image distortions. *IEEE Transactions on Image Processing*, 28(12):6022–6034, 2019.

[3] Zhiyi Cheng, Xiatian Zhu, and Shaogang Gong. Low-resolution face recognition. *The Asian Conference on Computer Vision (ACCV)*, 2018.

[4] Chollet and Francois. Xception: Deep learning with depthwise separable convolutions. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *arXiv preprint arXiv:1801.07698v1*, 2018.

[7] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[8] Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, et al. On robustness and transferability of convolutional neural networks. *arXiv preprint arXiv:2007.08558*, 2020.

[9] Samuel Dodge and Lina Karam. Understanding how image quality affects deep neural networks. *International Conference on Quality of Multimedia Experience (QoMEX)*, 2016.

[10] Samuel F. Dodge and Lina J. Karam. A study and comparison of human and deep learning recognition performance under visual distortions. *International Conference on Computer Communications and Networks (ICCCN)*, 2017.

[11] Samuel F Dodge and Lina J Karam. Quality robust mixtures of deep neural networks. *IEEE Transactions on Image Processing*, 27(11):5553–5562, 2018.

[12] Shengcai Liao Dong Yi, Zhen Lei and Stan Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[13] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *International Conference on Learning Representations (ICLR)*, 2018.

[14] Robert Geirhos, Carlos R. M. Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. Generalisation in humans and deep neural networks. *Advances in Neural Information Processing Systems (NIPS)*, pages 7538–7550, 2018.

[15] Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. The reversible residual network: Backpropagation without storing activations. *Advances in Neural Information Processing Systems (NIPS)*, pages 2214–2224, 2017.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[18] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

[19] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations (ICLR)*, 2019.

[20] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *International Conference on Learning Representations (ICLR)*, 2020.

[21] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *Advances in Neural Information Processing Systems Workshops (NIPSW)*, 2014.

[22] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[23] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

[25] Jörn-Henrik Jacobsen, Arnold W.M. Smeulders, and Edouard Oyallon. i-revnet: Deep invertible networks. *International Conference on Learning Representations (ICLR)*, 2018.

[26] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *The European Conference on Computer Vision (ECCV)*, pages 694–711, 2016.

[27] Insoo Kim, Seungju Han, Seong-Jin Park, Ji-Won Baek, Jinwoo Shin, Jae-Joon Han, and Changkyu Choi. Discface: Minimum discrepancy learning for deep face recognition. *The Asian Conference on Computer Vision (ACCV)*, 2020.

[28] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)*, 2014.

[29] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[31] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[32] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[33] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D. Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019.

[34] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: the first manually collected, in-the-wild age database. *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 51–59, 2017.

[35] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. *International Conference on Machine Learning (ICML)*, 2010.

[36] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. *The European Conference on Computer Vision (ECCV)*, 2018.

[37] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *International Conference on Learning Representations (ICLR)*, 2015.

[38] Evgenia Rusak, Lukas Schott, Roland S. Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. *The European Conference on Computer Vision (ECCV)*, 2020.

[39] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[40] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016.

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015.

[42] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[43] Fred Tung and Greg Mori. Similarity-preserving knowledge distillation. *The IEEE International Conference on Computer Vision (ICCV)*, pages 1365–1374, 2019.

[44] Igor Vasiljevic, Ayan Chakrabarti, and Gregory Shakhnarovich. Examining the impact of blur on recognition by convolutional networks. *arXiv preprint arXiv:1611.05760*, 2016.

[45] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[46] Jiayu Wu, Qixiang Zhang, and Guoxi Xu. Tiny imagenet challenge. Technical report.

[47] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. Adversarial examples improve image recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[48] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[49] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, pages 1499–1503, 2016.

[50] Linfeng Zhang, Muzhou Yu, Tong Chen, Zuoqiang Shi, Chenglong Bao, and Kaisheng Ma. Auxiliary training: Towards accurate and robust models. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[51] Yiren Zhou, Sibo Song, and Ngai-Man Cheung. On classification of distorted images with deep convolutional neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.