Feature Quantization for Defending Against Distortion of Images

Zhun Sun[1,2], Mete Ozay[1], Yan Zhang[2], Xing Liu[1], Takayuki Okatani[1,2]
[1]Tohoku University [2]RIKEN Center for AIP
{sun, mozay, zhang, ryu, okatani}@vision.is.tohoku.ac.jp

Abstract

In this work, we address the problem of improving robustness of convolutional neural networks (CNNs) to image distortion. We argue that higher moment statistics of feature distributions can be shifted due to image distortion, and the shift leads to performance decrease and cannot be reduced by ordinary normalization methods as observed in our experimental analyses. In order to mitigate this effect, we propose an approach base on feature quantization. To be specific, we propose to employ three different types of additional non-linearity in CNNs: i) a floor function with scalable resolution, ii) a power function with learnable exponents, and iii) a power function with data-dependent exponents. In the experiments, we observe that CNNs which employ the proposed methods obtain better performance in both generalization performance and robustness for various distortion types for large scale benchmark datasets. For instance, a ResNet-50 model equipped with the proposed method (+HPOW) obtains 6.95%, 5.26% and 5.61% better accuracy on the ILSVRC-12 classification tasks using images distorted with motion blur, salt and pepper and mixed distortions.

1. Introduction

Recognition of objects using distorted images is a challenge that has been studied extensively in computer vision and pattern recognition in the last decade [3, 8, 13, 28, 30, 32, 34]. While convolutional neural networks (CNNs) have achieved impressive progress for object classification and recognition in various benchmark datasets [17, 20, 42, 45], recent works [11, 12] show that their performance is severely degraded for distorted images.

In this work, we consider a collection of image distortions that are observed in real-world natural images. Specifically, we consider the following types of distortion: i) distortion caused by signal processing, ii) statistical noise and iii) occlusion (see Section 3.1 for details). Image distortions result in change of statistical properties of datasets. In other words, recognition of objects using distorted datasets can be posed as a dataset shift problem. Suppose that $\mathbf{X_{tr}}$ and $\mathbf{X_{te}}$ are sets

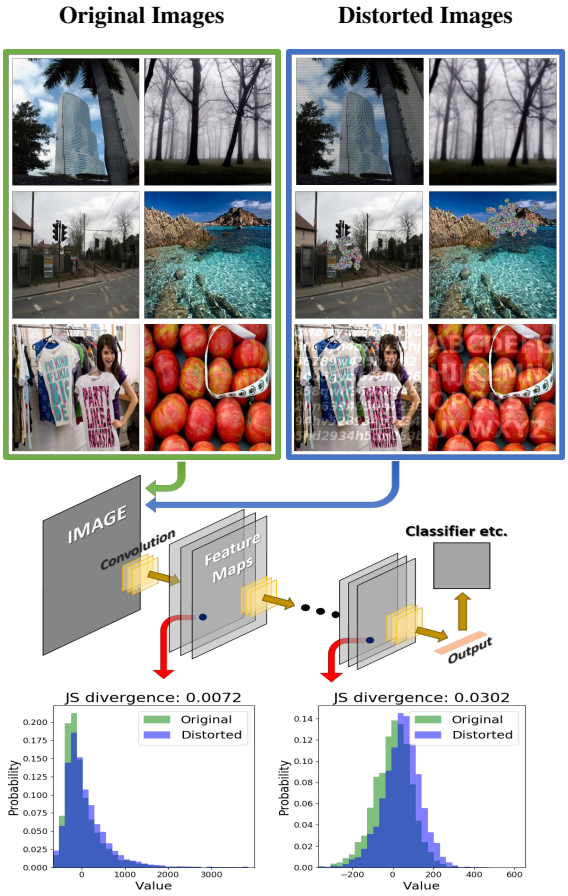


Figure 1: Divergence between distributions of neuron activities can be observed between original and distorted input images. This effect is accumulated through layers of CNNs, which can finally result in degraded performance. The probability densities are calculated using VGG-16 [42] for 5,000 original images, which belong to the validation set of ILSVRC-12, and their manually distorted versions.

of features extracted from examples belonging to datasets $\mathcal{D}_{tr}$ and $\mathcal{D}_{te}$, respectively. Moreover, suppose that a CNN model is trained using $\mathcal{D}_{tr}$, and tested using $\mathcal{D}_{te}$, a set of distorted images. Then, the distortion (such as noisy pixels)

occurred on the dataset $\mathcal{D}_{te}$ will affect extraction of feature representations learned using $\mathcal{D}_{tr}$. Consequently, feature distributions $p(\mathbf{X_{tr}})$ and $p(\mathbf{X_{te}})$ may diverge.

Feature normalization methods have been popularly used to address the data shift problems. However, normalization methods [20] do not ensure minimization of difference between higher moments, e.g. skewness and kurtosis. The relationship between moment statistics and classification performance has been studied in the last decade. For instance, the effect of moments and percentile statistics on surface reflectance properties was analyzed in [41]. On the other hand, dataset shift caused by *only* skewness and kurtosis can also decrease the performance of neural networks severely (see Section 2.1). In CNNs, the aforementioned shifts between higher moments of features extracted from clean and distorted images can be observed. In addition, the shifts are usually irregular, and their magnitude can grow from bottom to top layers (Figure 1).

In this paper, we propose an alternative approach to resolve the aforementioned problem by employing a *weak quantization operation* on features obtained at the output of convolution layers. Quantization methods are initially proposed to increase the computational performance and energy efficiency [15], while they are also helpful in eliminating minor perturbation of features under full numerical precision. We realize this approach by employing three different types of additional non-linearity in CNNs: i) a floor function with scalable resolution, ii) a power function with learnable exponents, and iii) a power function with data-dependent exponents. Our contributions can be summarized as follows:

1. We explore how a shift of higher moments of feature distributions can lead to a performance degradation. In addition, we investigate the viability of divergence reduction by using normalization methods and non-linear functions.

2. We propose a new approach to employ feature quantization while training CNNs. Briefly, we integrate the floor or power non-linearity function into the CNNs, such that the features from distorted images can be mapped to a new space with less divergence. Our proposed approach enables us to improve robustness of CNNs without utilizing additional training techniques such as stability training [50].

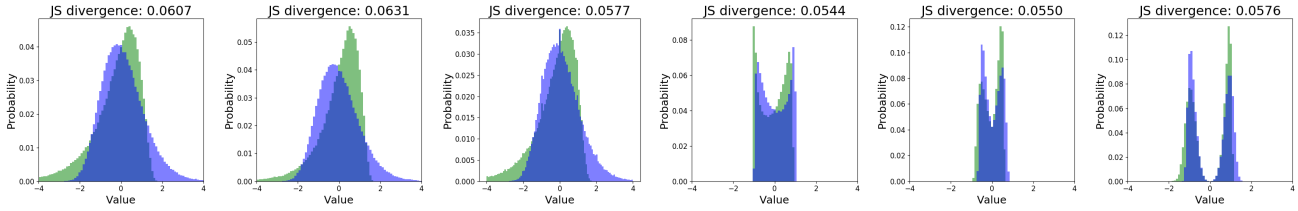
3. In experimental analyses, we demonstrate that the generalization performance of CNNs and their robustness to various types of image distortions can be improved using our proposed methods for object recognition and detection tasks using large scale datasets (e.g. the ILSVRC-12 and Pascal Voc).

Related Work

Processing Distorted Images using Deep Learning Methods: The dataset shift problem caused by image distortion has been previously tackled using several approaches [23, 43, 46]. The most widespread approach used to minimize the divergence, is to employ a generative model $p(\mathbf{z})p_\theta(\mathbf{X}|\mathbf{z})$ such that both $p(\mathbf{X_{tr}})$ and $p(\mathbf{X_{te}})$ could be inferred from a fixed distribution $p(\mathbf{z})$ which is parametrized by a set of parameters θ [19, 39, 53]. Due to the intractability of $p_\theta(\mathbf{X}|\mathbf{z})$, it could be difficult to estimate parameters θ [27]. Recent works [9, 21] have considered modeling of some specific transformation patterns such as scale and rotation by learning sub-networks with a parameter set ϕ. These sub-networks can yield a new distribution for a test dataset $q_\phi(\mathbf{X_{te}})$ that is similar to $p(\mathbf{X_{tr}})$, or a new $q_\phi(\mathbf{X_{tr}})$ that is likely to be an approximation to $p(\mathbf{X_{te}})$. Still, estimation of parameters ϕ is challenging since the transformation patterns can be very different for $\mathbf{X_{tr}}$ and $\mathbf{X_{te}}$. On the other hand, training techniques, which are used to explicitly minimize $\mathfrak{D}(\mathbf{X_{tr,I}}, \mathbf{X_{tr,I'}})$ between an input image $\mathbf{I}$ and its distorted version $\mathbf{I'}$ during training, are also shown to be helpful [50], where $\mathfrak{D}$ is a measure for distance such as the ℓ_2 distance. However, these techniques increase computational complexity of training methods. In addition, the improvement could be marginal if the prior knowledge of $p(\mathbf{X_{te}})$ used for generating distorted images is not available (see Section 3.2).

Normalization Methods used in Deep Learning: State-of-the-art normalization methods such as Batch Normalization (BN) [20] and Layer Normalization (LN) [26] are used to reduce the inherent data shift problems by fixing the mean and the variance of distribution of input features at each layer of a CNN. Concretely, if $\mathbf{X}$ (which denotes either a data matrix of training/testing samples, or a matrix of features extracted from samples) is received as input, then normalization methods output $\tilde{X} = \frac{\mathbf{X}-\mu_\mathbf{X}}{\sigma_\mathbf{X}}$ that has zero mean and unit variance. Although these normalization methods are confirmed to work well empirically, still they implicitly assume that distributions of features obtained from clean and distorted images can be parameterized according to the same function, *i.e.* Gaussian. In addition, these methods do not aim minimization of the classification loss. However this assumption does not apply to general cases as mentioned above, especially when the divergence is usually observed for higher moment statistics, i.e. skewness and kurtosis. In practice, these normalization methods are followed by a linear transformation method. This ad-hoc method slightly mitigates the problem, but still there remains large divergence between $p(\mathbf{X_{tr}})$ and $p(\mathbf{X_{te}})$.

Feature Quantization Methods used by CNNs: The effectiveness of dimension and complexity reduction of feature quantization/hashing methods have been demonstrated in the previous studies [1, 48]. Recently, various approaches have



(a) Original (b) Normalization (c) Floor (Scale=100) (d) Tanh (e) Softsign (f) Power (Exp.=0.25)

Figure 2: A demonstration of change of distributions observed by using different nonlinear functions. In the analyses, κ value of (a) original blue and green distribution was set to -0.2 and 0.5, respectively. (b)(c)(d)(e) depict distributions computed after employment of the corresponding transformations. It can be seen that distributions mapped by using additional non-linearity have smaller divergence with better fitted shapes for this task.

been proposed for quantization of weights and activations in CNNs in order to compress the networks or reduce flops during inference [37, 51, 35, 29, 52]. For instance, XNOR-Net [37] is a type of CNN that uses mostly bitwise operations to approximate convolutions, where both filters and features are binary. DoReFa-Net [51] generalizes this method and quantizes weights, activations and gradients using different widths of bits. Both methods provide accelerated training and inference, together with a reduced model size. However, these quantization methods are not proposed to gain inherent robustness against image distortion, and mainly focus on the trade-off between compression effectiveness and accuracy (although robustness can be observed against specified types of distortion, see Section 3.2).

Design of Non-linearity in CNNs: Various types of non-linearity functions have been explored and proved to be beneficial for training CNNs empirically [2, 10, 18, 24, 31, 36]. However, functions endowed with power operations have been barely utilized. A former attempt [14] proposed an ℓ_p nonlinear unit with a learnable order p. An ℓ_p unit employed at a layer receives signals from a subset of units used at the previous layer, and performs ℓ_p normalization. This can be interpreted as an implicit employment of power operation for estimation and assignment of different weights to different feature activation. Their experimental results show performance improvement for some benchmark datasets. S-shaped rectified linear units (SReLU) [22] have been proposed to achieve more complicated non-linearity, and they are considered as imitation of the behavior of power or logarithm functions with performance boost.

2. Proposed Feature Quantization

Suppose that we are given a convolution kernel $\mathbf{W} \in \mathbb{R}^{C\times D\times h\times w}$ with D output channels that slides on a tensor of features $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$. Then, the output of a convolution operation $\mathbf{U} \in \mathbb{R}^{D\times \tilde{H}\times \tilde{W}}$ can be computed by

$$\mathbf{U}_d = \sum_c \mathbf{W}_{c,d} \star \mathbf{X}_c, \tag{1}$$

where $\star$ is the two dimensional convolution operation, $\mathbf{U}_d$ is the d^{th} channel of $\mathbf{U}$, $c = 1, 2, \ldots, C$ and $d = 1, 2, \ldots, D$ denotes the index of input and output channels, respectively. In our proposed approach, we consider that a distorted image $\mathbf{I}' = \mathcal{F}_{\boldsymbol{\theta}}(\mathbf{I})$, where $\mathcal{F}_{\boldsymbol{\theta}}$ is the transformation function parameterized by $\boldsymbol{\theta}$. For instance, a noisy image $\mathbf{I}'$ is obtained using additive noise $\boldsymbol{\epsilon}$ such that $\mathbf{I}' = \mathbf{I} + \boldsymbol{\epsilon}$. Various nonlinear functions $\mathcal{F}_{\boldsymbol{\theta}}$ are used to perform more complicated distortions such as occlusion and compression. Under this setting, we aim to learn features $\mathbf{X}'$ extracted from distorted image $\mathbf{I}'$, whose change is *relatively small* compared to $\mathbf{X}$ extracted from clean image $\mathbf{I}$, using feature quantization methods while training CNNs. For this purpose, we use a floor function with scalable resolution, a power function with learnable exponents and a power function with data-dependent exponents.

Floor function with scalable resolution: The floor operation can be used to remove small noise $\boldsymbol{\epsilon}$ by *quantizing* the input into a set of integers. However, a trade-off between increasing the strength of quantization and the errors occurred due to quantization should be made in order to benefit from the floor operation. Therefore, we employ scaling coefficients and compute the convolution by

$$\mathbf{U}_d = \sum_c \mathbf{W}_{c,d} \star \tau(\mathbf{X}_c, \beta_{c,d}), \tag{2}$$

where $\beta_{c,d} \in \mathbb{R}$ is a channel-wise coefficient, and τ is the element-wise floor function defined by

$$\mathbf{y} = \tau(\mathbf{x}, \beta) \triangleq \frac{\lfloor \beta \mathbf{x} \rfloor}{\beta}, \tag{3}$$

where $\lfloor x_i \rfloor = \max\{z \in \mathbb{Z} | z \leq x_i\}$ is the floor function, which is applied to each element x_i of a tensor $\mathbf{x}$.

Mathematically, floor function has zero gradient with respect to its input. In order to compute its gradient, we also employ the "straight-through estimator" method as proposed in [4, 51]. That is, we assign 1 to gradients back-propagated to lower layers during back-propagation.

Power function with a learnable exponent: Instead of explicit quantization of the input, we propose to use the power operation in the convolution operation in order to employ another non-linearity. The power function with an exponential map with range $[0, 1]$ is able to map any positive real number closer to 1. Thus, the input can be considered as $\mathbb{1}+\boldsymbol{\zeta}$, where $\mathbb{1}$ is the identity 1-tensor having the same shape that the input has. We consider this mapping as a *quasi-quantization* effect, where the smaller exponent is the heavier quantization we obtain. It is worth noting that, this can be achieved by any non-linearity functions with saturation activity, such as sigmoid, Tanh or Softsign (further discussions are given in Section 2.1). In order to append the power function into the convolution operation, we define the convolution by

$$\mathbf{U}_d = \sum_c \mathbf{W}_{c,d} \star \psi(\mathbf{X}_c, \alpha_{c,d}), \tag{4}$$

where $\alpha_{c,d} \in \mathbb{R}$ is the corresponding channel-wise exponent, and ψ is the element-wise power function defined by

$$\mathbf{y} = \psi(\mathbf{x}, \alpha) = \begin{cases} x_m^{\alpha+1}, & \text{if } x_m \geq 0 \\ -(-x_m)^{\alpha+1}, & \text{otherwise} \end{cases}, \tag{5}$$

where x_m is the m^{th} element of $\mathbf{x} \in \mathbb{R}^M$. We apply a mirror operation for negative inputs, since power function is defined on $\mathbb{R}^+$, while they are safely ignored in CNNs employing ReLU, where only positive values are propagated into the next convolution layer. The parameters α are determined to be learnable to provide an appropriate quantization strength, and they are estimated using gradients computed during back-propagation (BP) by (d and c are omitted for simplicity)

$$\frac{\partial \mathcal{L}}{\partial \alpha} = \sum_{i=1}^{m} \frac{\partial \mathcal{L}}{\partial y_i} y_i \ln |x_i|, \qquad \frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial \mathcal{L}}{\partial y_i} (\alpha + 1) \frac{y_i}{x_i}, \tag{6}$$

where $\mathcal{L}$ denotes a loss function such as a classification loss. Note that $x_i \neq 0$, otherwise we assign a 0 to the gradients. In this work, instead of providing a hard restriction to the range of α, we employ ℓ_2 and ℓ_1 (lasso) regularized terms towards α for computation of the final loss during training. Empirically this is able to stabilize the training while α may grow larger 1. Detailed analysis on the distribution of learned α as well as the effects of ℓ_2 and ℓ_1 (lasso) regularization are provided in supplementary material.

Hyper-exponent for power function: We introduce a HyperNetwork [16] approach for estimating strength of quasi-quantization effect of power function defined by

$$\boldsymbol{\alpha}_d = \mathcal{F}_d(\boldsymbol{\mu}_{\boldsymbol{X}_d}, \boldsymbol{\sigma}_{\boldsymbol{X}_d}), \tag{7}$$

where $\boldsymbol{\mu}_{\boldsymbol{X}_d}, \boldsymbol{\sigma}_{\boldsymbol{X}_d} \in \mathbb{R}^c$ stand for mean and standard deviation for all input channels, $\mathcal{F}_d$ is a mapping function and $\boldsymbol{\alpha}_d \in \mathbb{R}^c$ is an exponent computed for the output channel d.

Table 1: Averaged classification accuracy (%) of two-layer neural network models obtained using artificial datasets over 10 runs. During each run, 10,000 training and 10,000 test samples are generated, respectively. Trans. stands for the followed linear transformation proposed in [20].

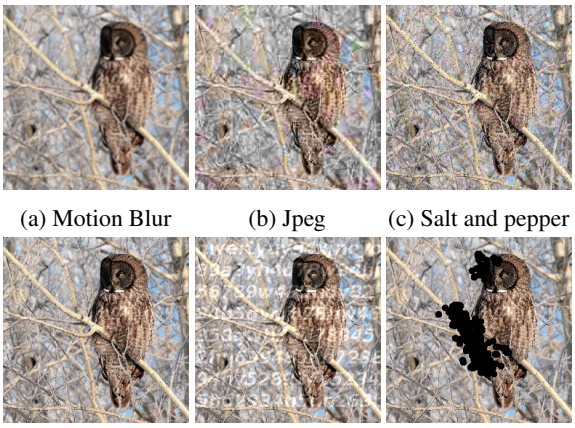
Total Features (M)	**128**			
Determinant Features (N)	1	2	4	8
Base w/o divergence	96.4	92.6	83.0	36.3
Base	93.9	87.9	73.7	26.9
Batch Norm. Only	92.1	84.9	71.5	29.3
Batch Norm. + Trans.	95.2	89.2	77.5	39.1
Base + 1 layer	93.2	85.1	70.5	27.8
Base + 2 layers	93.1	85.3	69.6	29.8
Floor	94.3	89.2	75.9	28.8
Tanh	97.4	95.1	89.4	59.5
Softsign	98.4	96.9	93.0	65.7
Power	**99.0**	**98.4**	**97.5**	**77.6**

In (2), (4) and (7), we let each output channel d own a set of parameters applied to C input channels. This method is helpful to obtain varying quantization strength in implementation of CNNs. However, c and d usually take large values in the recent CNNs. Thus, the number of parameters and computational complexity of the CNNs which employ this method may increase. Therefore, we suggest a method for sharing α among output channels. Concretely, we split D output channels into Λ portions, and all the channels within D_λ share the same set of α_{c,D_λ} ($\lambda = 1, 2, \ldots, \Lambda$), for employment of convolution with power operation. In the experimental analyses, we use $\lambda = 1$ as a default value. We employ a single β for all input and output channels to perform convolution with the floor operation.

2.1. An Analysis of Non-linearity

As discussed in Section 1, divergence caused by shifted skewness and kurtosis between feature distributions, is harmful for inference using new samples. However, minimization of this type of divergence cannot be achieved by normalization methods. Thus, we consider an alternative approach by introducing quantization non-linearity to CNNs. Our proposed approach is used to map a space of diverged distributions to a new space in which the divergence between distributions could be minimized. More precisely, the feature distributions $p(\mathbf{X_{tr}})$ and $p(\mathbf{X_{te}})$ are mapped into new distributions $\tilde{p}_\theta(\mathbf{X_{tr}})$ and $\tilde{p}_\theta(\mathbf{X_{te}})$, such that we obtain $\rho(\tilde{p}_\theta(\mathbf{X_{te}})||\tilde{p}_\theta(\mathbf{X_{tr}})) \leq \rho(p(\mathbf{X_{te}})||p(\mathbf{X_{tr}}))$, where ρ is a function which is used to measure similarity between distributions, such as Jensen–Shannon divergence.

In order to illustrate this, we design a multi-class classification experiment using an artificial dataset, where all features of samples have identical mean and variance but



(a) Motion Blur (b) Jpeg (c) Salt and pepper

(d) NGRN (e) Inpainting (f) Target occlusion

Figure 3: Samples of distorted images obtained using different types of distortion.

different skewness and kurtosis in training and testing phases. Concretely, the inputs are M-dimensional vectors $\mathbf{X} \in \mathbb{R}^M$. All features of $\mathbf{X}$ are sampled from a generalized normal distribution with zero mean and unit variance defined by

$$\frac{\Phi(y)}{1-\kappa x}, \tag{8}$$

where Φ is the standard normal probability distribution function, and y is defined by

$$y = \begin{cases} -\dfrac{\log\left[1-\kappa x\right]}{\kappa} & \text{if } \kappa \neq 0 \\ x & \text{if } \kappa = 0 \end{cases}. \tag{9}$$

We control the higher moments by randomly choosing the shape parameter from an uniform distribution $\kappa \sim \mathcal{U}(-1, 1)$. Each vector $\mathbf{X}$ consists of $N < M$ features that are used to identify class labels, and the remaining $M - N$ features are utilized as noise. The corresponding labels are defined to be $\sum_{n=0}^{N-1} 2^n \cdot \mathbf{1}(X_n \geq 0)$, where $\mathbf{1}(\cdot)$ is an indicator function that outputs 1 when the argument of the indicator is true. Thereby, we have 2^N number of classes. We choose $M = 128$, and $N \in \{1, 2, 4, 8\}$ to control the noise level.

A two-layer neural network (NN) employing ReLU activation function with 128 hidden units is employed as a base model. We first generate a dataset using the same shape parameter κ for both training and test sets as a reference set (Base w/o divergence). Then, we generate diverged datasets using different κ to construct both training and test data (Base). We first compare the performance of the two-layer NN (Base) trained using both of them. Then, we employ different non-linearity (on input vectors), and test the performance using diverged datasets. The results are given in Table 1. It can be seen that, if the distributions of features are shifted by higher moment statistics, then the performance of base model (Base) is degraded notably in all cases. While BN seems to be helpful, normalization without using a linear transformation even performs worse than the Base, except the cases where the number of classes is large. In Figure 2, we can see that the divergence is even larger for the normalized data compared to the original data. The results indicate that the linear transformation contributes to improvement of the robustness to divergence more than normalization.

Next, we examine the change of performance using the proposed scalable floor function (defined in (3)) and power function with trainable exponents (defined in (5)), together with two reference functions Tanh [25] and Softsign [5]. We observed that the floor function improved the performance of the base model by 0.4% to 1.8%, only by employing weak quantization operation that decreases the precision of input values. Meanwhile, the networks overcome the shift of distributions by a large margin using the non-linearity function which reshapes the distributions. We emphasize that employment of such non-linearity is not targeting at removing skewness and kurtosis totally, but rather mapping them into less diverged distributions (Figure 2). Meanwhile, Tanh and Softsign have been replaced by rectified non-linearity [33] in state-of-the-art CNNs due to the vanishing gradient problem.

3. Experimental Results

3.1. A Brief Analysis of Image Distortions

In this subsection, we define the distortion methods employed in our analyses. We consider three types of distortion, namely statistical noise, signal processing loss, and occlusion. Samples of distorted images obtained using different distortion methods are given in Figure 3. For each type of distortion, we employ four sets of hyper-parameters to generate samples at different distortion strength.

Signal Processing Loss: We consider this type of distortion as information loss occurred during acquisition or processing of 2D images. We choose three cases for generating distorted images; *Motion/Defocus blurring* [**M./D. Blur**]: Blurring an image attenuates the image's high-frequency components, hence the information in the corresponding frequency is lost. We convolve the image with 2D blurring kernels of different sizes to generate blurred test images. *Jpeg compression* [**Jpeg/Jpeg2K**]: The Jpeg compression [47] is a popularly used image compression method that offers a selectable trade-off between storage size and image quality. Encoding steps of Jpeg compression such as down-sampling and quantization will result in certain loss of information. Especially, when a large compression ratio is employed, severe high frequency loss can be observed. *Chromatic aberration* [**Aber.**]: Chromatic aberration is observed, when a lens cannot bring all color wavelengths to the same focal plane due to dispersion. Then, colored edges can be observed around objects in the images. We simply shift RGB channels of an image towards different directions to reproduce this phenomena.

Statistical Noise: We consider the following noise types;

Table 2: Classification accuracy (Top-5 accuracy(%)) obtained using distorted images.

Models	Clean	M.Blur	D.Blur	Jpeg	Jpeg2K	Aber.	S. & P.	NGRN	Y+$\mathcal{N}$	CC+$\mathcal{N}$	Inp.	Occ.	Mix.
ResNet18	90.29	55.08	**79.81**	59.47	74.13	83.28	53.84	71.81	**63.00**	59.96	63.26	54.26	59.15
+SF-100	**90.38**	56.67	79.35	**62.92**	73.94	**84.35**	51.90	73.67	62.16	61.50	64.23	52.80	54.29
+POW-1	90.26	57.72	79.29	57.65	73.49	83.23	57.27	74.46	60.99	**67.70**	**66.48**	54.19	60.57
+HPOW	89.80	58.28	77.89	58.20	**74.97**	83.09	**58.57**	72.35	62.17	64.22	65.36	51.21	**62.30**
+SF-POW	90.35	**59.56**	79.41	62.83	74.32	84.03	57.83	**74.94**	61.20	64.20	64.73	**54.83**	55.51
+DoReFa [51][b]	84.14	51.99	78.97	59.47	72.52	69.60	35.24	58.53	53.33	26.00	45.34	43.78	43.41
+Stability [50]	89.61	49.67	75.73	61.07	71.16	82.19	50.66	68.31	56.70	61.58	62.71	41.21	54.37
ResNet50	93.40	63.29	84.44	79.80	80.93	88.94	69.94	81.27	72.72	73.71	69.06	59.31	65.35
+SF-100	93.48	65.28	84.08	79.98	80.94	87.91	67.24	82.82	73.30	74.94	62.41	60.17	59.88
+POW-1	93.59	66.15	83.73	**80.19**	81.42	88.91	71.66	82.91	73.33	76.91	63.65	**61.33**	65.31
+HPOW	**93.70**	**70.24**	84.80	78.26	**82.16**	**89.19**	**75.20**	**84.24**	**77.65**	**78.49**	**71.20**	60.38	**70.96**
+SF-POW	93.38	64.20	**84.87**	79.14	80.55	88.40	69.66	83.23	72.39	77.29	69.61	59.98	67.07
+DoReFa [51][b]	86.37	48.37	72.03	57.14	71.11	76.19	41.40	59.71	46.92	49.21	59.74	45.54	48.86
+Stability [50]	92.85	57.48	81.82	71.62	80.26	87.65	60.09	76.60	67.12	75.68	67.14	57.14	50.44

[a] Top-1 accuracy is reported.
[b] We employ (W,A,G) = (1,4,32) for configuration as suggested in [51].

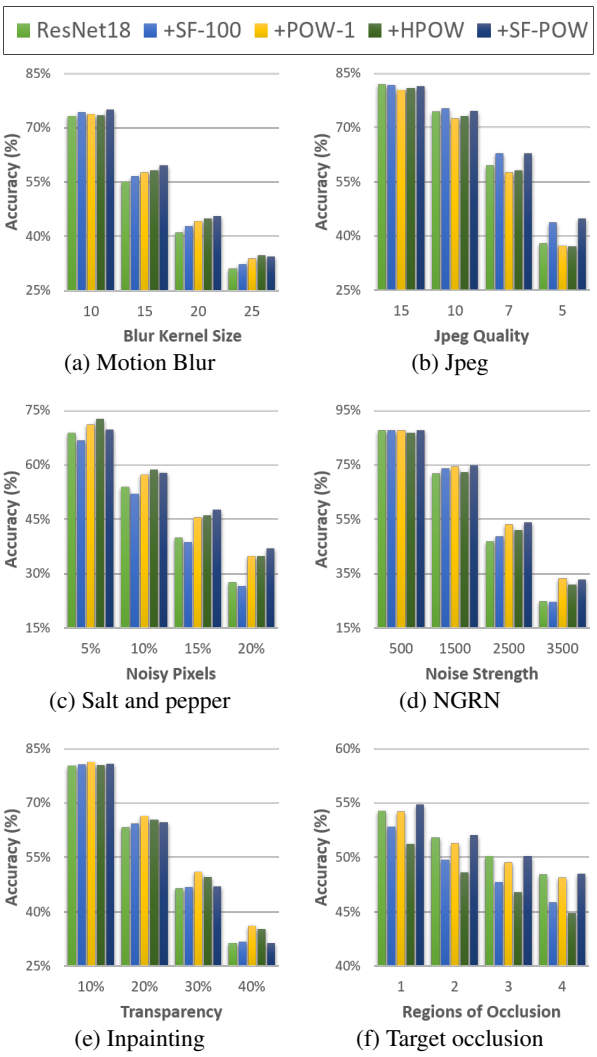
Salt and pepper noise (impulse valued noise) [**S. & P.**]: Salt and pepper noise [7] randomly drops original values (or maximize the values) of some pixels in an image, instead of corrupting the whole image. We randomly select pixels in the image according to a uniform distribution, and set their values to 0 or 255. *Non-Gaussian random noise* [**NGRN**]: Random noise is characterized by intensity and color fluctuations above and below actual image intensity. We employ Fast Fourier Transform (FFT) and inverse FFT to obtain the representations in different domains, and employ noise sampled from a Gaussian distribution with zero mean and different standard deviations. *Additive Gaussian noise* [**Y/CC**+$\mathcal{N}$]: Furthermore, we transform the images into YCbCr color space, and employ additive Gaussian noise to the Y channel (the luma component) and the CbCr channels (blue-difference and red-difference chroma components).

Occlusions: We consider two different artificial occlusion methods; *Inpainting* [**Inp.**]: Inpainting [6] confuses CNNs in a similar way as semantic occlusions do, *i.e*. features extracted from regions covered by translucent in-painting may appear to be from other classes due to the shift in statistics. We employ randomly generated strings with different transparencies to generate inpaintings. *Targeted occlusion* [**Occ.**]: Attention targeted occlusion [44] is designed to obliterate the information important for recognition of a target class [44]. We employ gradient methods to obtain a saliency map that records pixel-wise classification scores. Then, we occlude some clusters of pixels that contribute most to the final classification score with black masks. We employ a pre-trained Plain-18 [17] network to compute the saliency map. The strength of this type of occlusion can be increased by increasing the number of clusters occluded. *Mixed Noise* [**Mix.**]: Image distortions in real-world scenario are often more complicated, therefore we introduce a mixture of various types of synthetic distortions (Additive Gaussian noise in CbCr channels, Salt and pepper, Inpainting and Jpeg compression) to simulate the real-world distortions.

3.2. Experimental Analyses of Classification Performance using the ILSVRC-12 Dataset

We performed a standard object classification task using the ILSVRC-12 [40] dataset to investigate the robustness of our proposed method. We employ two different models learned using ResNet-18 and ResNet-50 as base models, and modify them with our proposed methods. We employ the training scheme and data augmentation methods described in [17] for training, and a single crop of size 256×256 for validation. The proposed models are employed as follows: i) ResNet quantized by floor with scale β=100.0 (+SF-100), ii) ResNet equipped with power non-linearity using one set of learnable exponents (+POW-1), iii) ResNet equipped with a HyperNetwork for estimating the exponents of power function (+HPOW), iv) ResNet equipped with both power and floor functions (scale β=100 and split Λ=1, +SF-POW). For ResNet-50, the proposed methods are only employed before the convolution layers with 3×3 kernels.

Moreover, we introduce two different models for reference: i) DoReFa Networks [51] (+DoReFa) that employ 1, 4, and 32 bit widths for weights, activations and gradients, respectively, as suggested in their paper. It is worth noting that, the models equipped with scalable floor non-linearity is closely related to DoReFa models with full precision weights, gradients and activations with low bit widths, where the range of values is restricted within the present ability of the low bit widths (e.g. $[0, 1]$). However, the proposed floor method does not have this limitation by employing



(a) Motion Blur (b) Jpeg (c) Salt and pepper (d) NGRN (e) Inpainting (f) Target occlusion

Figure 4: Classification accuracy (Top-5 accuracy(%)) obtained using images with different strength of distortions.

full precision of floating numbers. ii) Models optimized through a stability training method proposed in [50]. Briefly, we fine-tune the last fully-connected layer of the learned models by regularizing the divergence between classification score $p(\boldsymbol{y}|\boldsymbol{I})$ and $p(\boldsymbol{y'}|\boldsymbol{I'})$ inferred from the image $\boldsymbol{I}$ and $\boldsymbol{I'} = \boldsymbol{I} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\sigma})$. We use the hyper-parameters that are employed for classification tasks in [50], where $\boldsymbol{\sigma} = 0.04$, and the regularization coefficient is 0.01.

Robustness is evaluated by Top-5 classification accuracy for distorted images, except for the target occlusion, where Top-1 is employed since this type of distortion is only targeted for the ground truth label. We analyze robustness of performance of the models to the proposed distortions in Table 2. We also report the detailed results obtained for different models and distortion strength in Figure 4 (ResNet-18, results for ResNet-50 are provided in the supp. mat.). In Table 2, we observe that most of the proposed methods have similar performance compared to reference models on clean or slightly distorted images, while the stability training employed in [50] decreases performance. For instance, the +HPOW model boosted the standard classification performance of the ResNet-50 by 0.30%. The only notable performance decrease (0.49%) is observed in ResNet-18+HPOW model, which can be attributed to increasing complexity of the baseline ResNet-18 by employment of HyperNetworks at all its convolution layers. However, if stability training is used, then the performance is decreased by 0.68% and 0.55% for ResNet-18 and and ResNet-50, respectively.

In addition, models that employ quantization with either a floor or power function perform better than the original model under most of the conditions. Notably, when ResNet-18 is used as a reference, the +SF-100 model provided the robustness against Jpeg compression by 3.45%. The +POW-1 model provided 7.74% and 3.22% performance boost against CC+$\mathcal{N}$ and Inpainting, respectively. Meanwhile, the +HPOW models boost the robustness of ResNet-50 against most of the distortions. For instance, 6.95%, 5.26%, 4.97% and 5.61% performance boost were observed against Motion blur, Salt and pepper, Y+$\mathcal{N}$ and mixed distortions, respectively. On the other hand, there also exist risks of performance decrease in some special cases. For instance, the ResNet-50+SF-100 model is disrupted by inpainting, while +POW1 and +HPOW models are weak or neutral against Jpeg compression. Moreover, the integrated model +SF-POW is able to dodge this risk and boost the performance for most types of distortion. These improvements can be further observed in Figure 4, where the proposed method boosts the robustness of the base model under both minor and heavy distortion in most cases.

Furthermore, we observed that the DoReFa models behave similar to +SF-100 models, which perform well against Jpeg compression. However, their overall performance is underwhelming, and decreases heavily with respect to statistical noise and occlusion. The models optimized with stability training also gain decent robustness against Jpeg compression as reported in [50]. However, their performance is severely degraded for other types of distortion. We argue that, although better robustness is observed for Jpeg compression empirically, the Gaussian prior is still not a viable choice for numerous types of distortion. Hence, employment of the prior knowledge on distortion is necessary to carry out stability training, which could be difficult in practice.

3.3. Analyses of Detection Results using the Pascal Voc 2007 Dataset

In this section, we examine the performance of our proposed method for an object detection task using the Pascal Voc 2007 dataset. We employ Faster-RCNN [38] for detection, and Zeiler and Fergus (ZF) model [49] as the baseline CNN model. We append our proposed power function to

Figure 5: Examples of object detection results obtained using distorted images. Text given in green color indicates the class of objects with confidence. Rows from top to bottom: Original images, images with minor distortion (Inpainting, Random noise), images with heavy distortion (Inpainting, Random noise, Salt & Pepper noise, blurring, Jpeg compression). Left and right images are selected from *training* and validation set, respectively.

the last three convolution layers of the ZF model (ZF-POW-1), and evaluate the change in performance. We implement the both models for training using the ILSVRC-2012 with random initialization to ensure the fairness for evaluation of detection. We manually select snapshots of both models that provide the same classification accuracy (Top-1/5 58.6%/81.7%). Then, we train both models using the Pascal Voc 2007 training dataset, and we manually select a snapshot of the ZF-POW-1 model which provides the detection performance (58.7% mAP) that is same to that of the fully trained ZF model. We employ the distortion patterns given in Section 3.1, and two datasets (Mix.Light and Mix.Heavy) that employ mixed patterns of distortion (see Figure 5). The results given in Table 3 show that, although both models have the same detection performance in the original validation set, the model equipped with power convolution gains $2.2\% - 3.3\%$ mAP under different distortions.

Table 3: Detection performance (% mAP) for the distorted Pascal Voc 2007 validation set using different patterns.

Models	Original	Mix. Minor	Mix. Heavy
ZF [49, 38]	**58.7**	50.0	14.3
ZF+POW-1	**58.7**	52.2	17.6

4. Conclusions and Discussions

In this work, we propose a feature quantization approach to enhance the robustness of CNNs to image distortion for popular object recognition and detection problems. We consider this challenge as a dataset shift problem, where the higher moment statistics of feature distributions shift due to distortion. In order to attenuate this effect, we apply non-linearity by integrating a floor or power function into the convolution operation in CNNs. We give insights into the efficiency of our proposed method in dealing with the dataset shift problem, compared to other different types of non-linearity. The experimental results obtained using benchmark datasets indicate a substantial boost of robustness of feature representations to various types of distortions. We believe that this approach can be beneficial for training of CNNs in various computer vision tasks, where distortions may impair the performance, such as object identification and detection, image retrieval and restoration.

Acknowledgments

This work was partly supported by Council for Science, Technology and Innovation (CSTI), Cross-ministerial Strategic Innovation Promotion Program (Infrastructure Maintenance, Renovation and Management), JST CREST Grant Number JPMJCR14D1, and the ImPACT Program "Tough Robotics Challenge" of the Council for Science, Technology, and Innovation (Cabinet Office, Government of Japan).

References

[1] D. Achlioptas. Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 274–281. ACM, 2001. 2

[2] F. Agostinelli, M. Hoffman, P. J. Sadowski, and P. Baldi. Learning activation functions to improve deep neural networks. *CoRR*, abs/1412.6830, 2014. 3

[3] H. Bay, B. Fasel, and L. V. Gool. Interactive museum guide: Fast and robust recognition of museum objects. In *Proceedings of the First International Workshop on Mobile Vision*, 2006. 1

[4] Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 3

[5] J. Bergstra, G. Desjardins, P. Lamblin, and Y. Bengio. Quadratic polynomials learn better image features. Technical Report 1337, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Apr. 2009. 5

[6] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 6

[7] A. K. Boyat and B. K. Joshi. A review paper: Noise models in digital image processing. *arXiv preprint arXiv:1505.03489*, 2015. 6

[8] G. Chechik, G. Heitz, G. Elidan, P. Abbeel, and D. Koller. Max-margin classification of data with absent features. *Journal of Machine Learning Research*, 9(Jan):1–21, 2008. 1

[9] G. Cheng, P. Zhou, and J. Han. Rifd-cnn: Rotation-invariant and fisher discriminative convolutional neural networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[10] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015. 3

[11] G. B. P. da Costa, W. A. Contato, T. S. Nazare, J. E. Neto, and M. Ponti. An empirical study on the effects of different types of noise in image classification tasks. *arXiv preprint arXiv:1609.02781*, 2016. 1

[12] S. Dodge and L. Karam. Understanding how image quality affects deep neural networks. *arXiv preprint arXiv:1604.04004*, 2016. 1

[13] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282, 2010. 1

[14] C. Gulcehre, K. Cho, R. Pascanu, and Y. Bengio. Learned-norm pooling for deep feedforward and recurrent neural networks. In *ECML PKDD*, pages 530–546, 2014. 3

[15] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan. Deep learning with limited numerical precision. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1737–1746, 2015. 2

[16] D. Ha, A. Dai, and Q. V. Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016. 4

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1, 6

[18] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015. 3

[19] G. E. Hinton, A. Krizhevsky, and S. D. Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011. 2

[20] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 448–456, 2015. 1, 2, 4

[21] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015. 2

[22] X. Jin, C. Xu, J. Feng, Y. Wei, J. Xiong, and S. Yan. Deep learning with s-shaped rectified linear activation units. *CoRR*, abs/1512.07030, 2015. 3

[23] A. Kanazawa, A. Sharma, and D. Jacobs. Locally scale-invariant convolutional neural networks. *arXiv preprint arXiv:1412.5104*, 2014. 2

[24] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter. Self-normalizing neural networks. *arXiv preprint arXiv:1706.02515*, 2017. 3

[25] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012. 5

[26] J. Lei Ba, J. R. Kiros, and G. E. Hinton. Layer Normalization. *ArXiv e-prints*, July 2016. 2

[27] K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 991–999, 2015. 2

[28] A. Leonardis and H. Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 78(1):99–118, 2000. 1

[29] F. Li, B. Zhang, and B. Liu. Ternary weight networks. *arXiv preprint arXiv:1605.04711*, 2016. 3

[30] F. Lindner, U. Kressel, and S. Kaelberer. Robust recognition of traffic signals. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 49–53. IEEE, 2004. 1

[31] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013. 3

[32] B. M. Marlin. *Missing data problems in machine learning*. PhD thesis, University of Toronto, 2008. 1

[33] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010. 5

[34] R. O'Reilly, D. Wyatte, S. Herd, B. Mingus, and D. Jilk. Recurrent processing during object recognition. *Frontiers in Psychology*, 4:124, 2013. 1

[35] E. Park, J. Ahn, and S. Yoo. Weighted-entropy-based quantization for deep neural networks. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 5456–5464, 2017. 3

[36] P. Ramachandran, B. Zoph, and Q. V. Le. Swish: a self-gated activation function. *arXiv preprint arXiv:1710.05941*, 2017. 3

[37] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer, 2016. 3

[38] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 7, 8

[39] S. Rifai, X. Muller, X. Glorot, G. Mesnil, Y. Bengio, and P. Vincent. Learning invariant features through local space contraction. *ArXiv e-prints*, Apr. 2011. 2

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 6

[41] L. Sharan, Y. Li, I. Motoyoshi, S. Nishida, and E. H. Adelson. Image statistics for surface reflectance perception. *Journal of the Optical Society of America A*, 25(4):846–865, 2008. 2

[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[43] K. Sohn and H. Lee. Learning invariant representations with local transformations. *arXiv preprint arXiv:1206.6418*, 2012. 2

[44] Z. Sun, M. Ozay, and T. Okatani. Design of kernels in convolutional neural networks for image classification. *arXiv preprint arXiv:1511.09231*, 2015. 6

[45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 1

[46] Y. Tang and C. Eliasmith. Deep networks for robust visual recognition. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1055–1062, 2010. 2

[47] G. K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992. 5

[48] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1113–1120. ACM, 2009. 2

[49] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 7, 8

[50] S. Zheng, Y. Song, T. Leung, and I. Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4480–4488, 2016. 2, 6, 7

[51] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016. 3, 6

[52] C. Zhu, S. Han, H. Mao, and W. J. Dally. Trained ternary quantization. *arXiv preprint arXiv:1612.01064*, 2016. 3

[53] W. Zou, S. Zhu, K. Yu, and A. Y. Ng. Deep learning of invariant features via simulated fixations in video. In *Advances in Neural Information Processing Systems*, pages 3212–3220, 2012. 2