GRAM: Gradient Rescaling Attention Model for Data Uncertainty Estimation in Single Image Super Resolution

Changwoo Lee
Department of Electronic Engineering
Hanyang University
Seoul, Korea
cwl30@hanyang.ac.kr

Ki-Seok Chung[*]
Department of Electronic Engineering
Hanyang University
Seoul, Korea
kchung@hanyang.ac.kr

***Abstract*—In this paper, a new learning method to quantify data uncertainty without suffering from performance degradation in Single Image Super Resolution (SISR) is proposed. Our work is motivated by the fact that the idea of loss design for capturing uncertainty and that for solving SISR are contradictory. As to capturing data uncertainty, we often model the output of a network as a Euclidian distance divided by a predictive variance, negative log-likelihood (NLL) for the Gaussian distribution, so that images with high variance have less impact on training. On the other hand, in the SISR domain, recent works give more weights to the loss of challenging images to improve the performance by using attention models. Nonetheless, the conflict should be handled to make neural networks capable of predicting the uncertainty of a super-resolved image, without suffering from performance degradation. Therefore, we propose a method called Gradient Rescaling Attention Model (GRAM) that combines both attempts effectively. Since variance may reflect the difficulty of an image, we rescale the gradient of NLL by the degree of variance. Hence, the neural network can focus on the challenging images, similarly to attention models. We conduct performance evaluation using standard SISR benchmarks in terms of peak signal-noise ratio (PSNR) and structural similarity (SSIM). The experimental results show that the proposed gradient rescaling method generates negligible performance degradation compared to SISR outputs with the Euclidian loss, whereas NLL without attention degrades the SR quality.**

***Index Terms*—Image restoration, Neural networks, Machine learning.**

I. Introduction

Single Image Super Resolution (SISR) is a task to restore a high resolution image from a single low resolution image. Like other computer vision tasks, deep neural networks (NNs) and various optimization techniques have been adopted to improve the quality of SISR [2]–[5]. Since convolutional neural networks (CNN) have been very successful in image processing, most existing works to solve SISR have utilized CNNs, which are often referred to as Super Resolution CNNs (SRCNNs). When a machine learning technique is employed for safety-critical systems, reliability of the output of a neural network is crucial because neural networks sometimes make imprudent decisions with high confidence [6].

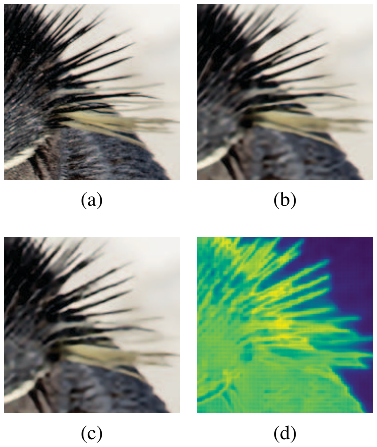


Fig. 1: Heteroscedastic uncertainty (d) can be estimated by SRCNN with the predictive variance [1], but the corresponding output (c) captures fewer details, such as edges of an object than SRCNN without uncertainty estimation (b). The original HR image is (a).

Hence, several methods to combine a neural network with a probabilistic model have been proposed in various computer vision tasks [1], [7]–[10]. These methods produce both predictions and uncertainty (or reliability) of them, by showing that the uncertainty is not only useful to make the neural network models more reliable but also improves the accuracy of the neural networks. For example, in regression tasks, one can model data uncertainty in a predictive variance, which represents how difficult it is to predict the correct answer from a given input. In this case, a neural network predicts the variance of a pixel as well as the mean. The predictive variance explicitly gets involved with the loss function—a division of the mean squared error (MSE) between the predicted mean and the target value by the predicted variance [1]. Thus, a datum with a high variance, which is challenging, has less impact during training because MSE is attenuated due to

978-1-7281-4550-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMLA.2019.00011

the high variance. Furthermore, the network trained with the uncertainty loss tends to show better accuracy than the network trained without uncertainty in several computer vision tasks such as depth estimation, pixel-wise segmentation, and image classification [1].

On the other hand, recent works in the SISR domain have modeled both neural networks and loss functions to focus on challenging images and pixels [4], [5], [11]–[13]. They actively adopt attention mechanisms to scale the loss. In other words, they have modeled neural networks and loss functions to concentrate on high variance pixels. This is based on the fact that there could be an infinite number of high resolution targets for a low resolution pixel so that upscaling the image toward the correct high resolution one is tough. Therefore, if the loss of difficult pixels were discounted by a high variance as the uncertainty loss function does, the quality of the super-resolved image cannot be guaranteed. Specifically, in the case of the data uncertainty, we observe that models trained by the previous method [1] often fail to capture high-frequency parts of the input image (see Fig. 1), unlike aforementioned vision tasks.

In this paper, to avoid such degradation, a method to evaluate uncertainty and that for attention learning are combined to minimize performance loss while making SRCNNs predict uncertainty. Our proposed method, named Gradient Rescaling Attention Model (GRAM), utilizes the predictive uncertainty as an attention mask for the gradient of NLL. The attention mask is a sigmoidal output of the predicted uncertainty, which is close to 0 if the uncertainty is low, and close to 1 if the uncertainty is high.

However, one problem still remains: we cannot directly adjust the uncertainty loss function by the attention mask because the attention mask cancels out the learning process for variance estimation. That is, the attention mask is based on the predictive variance, which can be trained when it attenuates the MSE. To this end, we rescale not the loss, but the gradient, which is a simple way to bypass the cancellation. Because our proposed method rescales the gradient, and the gradient is calculated after the loss is fixed, there is no chance for the attention mask to neutralize the uncertainty learning.

GRAM has two advantages. First, GRAM lets SRCNNs enjoy the fruit of both uncertainty and attention learning by resolving them together. Second, GRAM adds negligible overhead on training and can be easily implemented on existing machine learning frameworks [14], [15].

Performance comparison in terms of peak signal-noise ratio (PSNR) and structural similarity (SSIM) will show GRAM surpasses the naïve uncertainty loss [1], and produces high resolution images with almost the same quality as the SRCNN model that is trained only with MSE on the SISR benchmarks (Set5,Set14, BSD100, Urban100 and DIV2K [16]).

II. Background and Related Works

A. Single Image Super Resolution

Single Image Super Resolution is a problem that retrieves a high resolution (HR) image from a downsampled low resolution (LR) image. Since convolution neural networks achieve great success in image processing, several methods based on deep CNNs have tried to solve SISR. Dong et al. [2] proposed an SRCNN model to upscale LR images by competing against the non-neural network methods with a large margin. Numerous methods have been proposed with machine learning techniques such as generative adversarial networks (GAN) and attention mechanisms [3]–[5], [11]–[13].

Those methods typically train CNNs by the Euclidian distance, such as mean squared error (MSE) with a loss function to match the super-resolved image I^{SR} and the original high resolution image I^{HR}. The formal expression for training an SRCNN model is:

$$\mathcal{L}_{SR} = \frac{1}{2N}\sum_{i=1}^{N} \|I_i^{HR} - I_i^{SR}\|^2, \tag{1}$$

where $I^{SR} = \mathbf{f}(I^{LR})$ and $\mathbf{f}$ is an arbitrary upscaling function, which is commonly a neural network.

Although CNNs and other optimization techniques have made significant progress in SISR, finding the exact HR image from a lossy LR image remains unsolved because the problem is notoriously ill-posed. For instance, for a given LR image $I^{LR} \in \mathbb{R}^{H' \times W' \times 3}$ and an HR image $I^{HR} \in \mathbb{R}^{H \times W \times 3}$, the super-resolved (SR) image $I^{SR} \in \mathbb{R}^{H \times W \times 3}$ usually has an equal or a higher dimension than I^{LR} since $H' \leq H$ and $W' \leq W$. This lets us solve the seriously ill-posed inverse problem, which there can be several answers from one input—the existence eventually increases the natural variance of the SR prediction.

B. Modeling uncertainty in computer vision tasks

Two types of uncertainty are introduced in this section: epistemic uncertainty and heteroscedastic (aleatoric) uncertainty [1]. Epistemic uncertainty explains uncertainty within a neural network. Moreover, it can be estimated by the variance of outputs from a Bayesian Neural Network (BNN)—a neural network with stochastic weights. Given a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$, and weights of a neural network which are drawn from a prior distribution, for example $\mathbf{W} \sim \mathcal{N}(0, I)$, training BNNs is a process that a weight posterior distribution $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$ is fitted by a tractable variational distribution $q_\theta^*(\mathbf{W})$ which is parameterized by θ. Since the weights are sampled from the probability distributions, the variance of the outputs can be estimated through Monte Carlo samplings. However, quantifying epistemic uncertainty through an approximated BNN needs a large number of samples, leading to significantly increasing time for training and prediction.

On the contrary, heteroscedastic uncertainty demonstrates uncertainty within data—hence, it is also known as data uncertainty. Also, it is input-dependent so that it represents how much a given input is ambiguous, whereas homoscedastic uncertainty is invariant to inputs. That is, in SR, the heteroscedastic uncertainty goes high if a low resolution input image contains high frequency components such as edges of an object or complicated patterns. An advantage of the

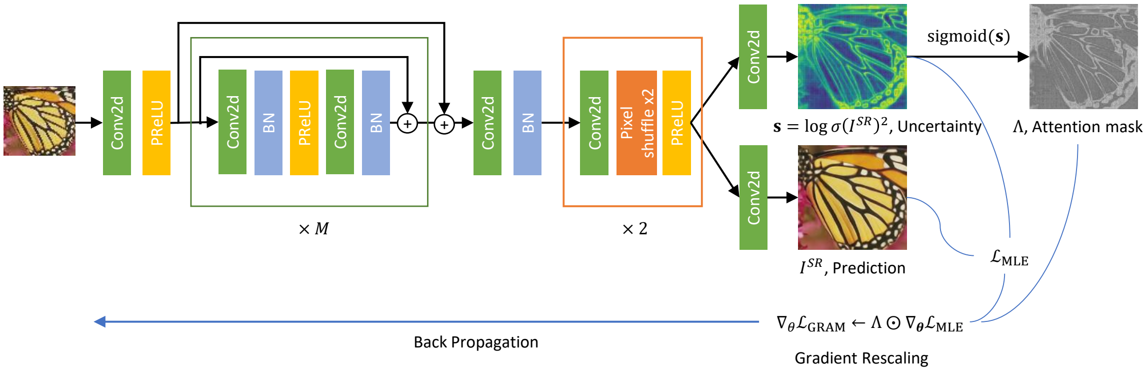


Fig. 2: SRCNN and the GRAM training scheme. In GRAM, the log variance of each pixel is transformed to attention mask $\mathrm{sigmoid}(\mathbf{s}(I_i^{SR}))$. Then the gradient of uncertainty loss $\mathcal{L}_{\mathrm{MLE}}$ (3) is rescaled by saliency matrix $\mathbf{\Lambda}$.

heteroscedastic uncertainty is that the evaluation is relatively simple compared to the epistemic uncertainty, which can be done by performing maximum likelihood estimation (MLE) on a Gaussian distribution with an input-dependent predictive variance. We no longer approximate a weight posterior distribution $p(\mathbf{W}|\mathbf{X},\mathbf{Y})$, but the variance is directly quantified from deterministic weights. Simply speaking, we add the final layer, which predicts the variance of the output, $\sigma(\mathbf{x})^2$. See Fig. 2. The loss function for training the neural network with the heteroscedastic uncertainty on MLE is:

$$\mathcal{L}_{\mathrm{MLE}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2\sigma(\mathbf{x}_i)^2}\|\mathbf{y}_i - \mathbf{f}(\theta;\mathbf{x}_i)\|^2 + \frac{1}{2}\log\sigma(\mathbf{x}_i)^2, \tag{2}$$

where N is the number of batches times the number of pixels of outputs. The loss function in (2) is from the negative log-likelihood of a Gaussian distribution $\mathcal{N}(\mathbf{y}_i, \sigma(\mathbf{x}_i)^2)$. Also, (2) demonstrates that the predicted variance attenuates the Euclidian loss on each pixel, which leads (2) to a learned attenuation loss. For numerical stability, we can change the output for the predicted uncertainty from variance $\sigma(\mathbf{x})^2$ to log variance $\mathbf{s} = \log\sigma(\mathbf{x})^2$. Hence, (2) becomes:

$$\mathcal{L}_{\mathrm{MLE}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\exp(-\mathbf{s}_i)\|\mathbf{y}_i - \mathbf{f}(\theta;\mathbf{x}_i)\|^2 + \frac{1}{2}\mathbf{s}_i, \tag{3}$$

which is identical to (2).

Interestingly, the heteroscedastic uncertainty and the epistemic uncertainty are often very similar to each other so that one can model the other when it is explicitly modeled [1]. Since the heteroscedastic uncertainty is much easier to estimate and to learn than the epistemic uncertainty, we consider only the heteroscedastic uncertainty in this paper.

C. *Attention model in computer vision*

In computer vision, an attention model is often used to let the training focus on the region of interest. For example, one can utilize the edge information as an explicit attention mask to enhance the capability of SRCNN on the edge of an object [11]. The attention mask should calibrate the loss between the target and the prediction in a pixel-by-pixel manner. Also, there can be an implicit attention model that calibrates the intermediate features of neural networks by using Residual Networks [17] or channel-wise attention [18]. Recent works have used attention model in both implicit and explicit ways by Residual Networks [4], [11], [13], a second-order attention network [12] or channel-wise attention [5]. However, these architectures contain complicated paths and modules, limiting them to be deployed in practice.

III. Gradient Rescaling Attention Model

The main objective of this paper is to make the SRCNN model predict both SR images and the corresponding variance. Although previous works that utilized the predictive variance to capture uncertainty have accomplished a state-of-the-art performance in many fields of computer vision, these approaches may not provide an effective solution for the SISR problem. Therefore, in this paper, a new method to embed an ability to predict uncertainty to the SISR problem is proposed.

A. *Loss attenuation problem in SISR*

In (2) and (3), the predicted variance of a pixel attenuates the Euclidian loss with respect to its degree. A high variance of a pixel discounts the corresponding loss. Although it improves the accuracy of the model in simple regression tasks that have the same or lower dimensional outputs compared to the inputs, we observe that this learning scheme is not suitable for SISR mainly because the quality of the predicted high resolution (HR) image goes unacceptably low. In Fig. 1, the predicted HR image cannot generate patches with high variances, such as edges of an object, compared to the prediction from the network, which is trained by only MSE (without heteroscedastic uncertainty). We claim that this is mainly because the SISR problem is ill-posed, as aforementioned in Section II-A. Therefore, in the SISR problem, the most cumbersome and challenging patches contain high frequency data of which variance is relatively higher than low frequency data. The main objective of designing a good SRCNN model with an efficient training scheme is to successfully upscale high frequency and challenging image patches with rich details. This is why recent works have focused on high frequency parts by applying attention masks [4], [5], [11]–[13]. Accordingly, unlike the regression problem in which one input has an answer, pixels with high variances might not be attenuated; it should be considered with much attention, instead.

B. Turning attenuation into attention: Gradient rescaling

In this section, we propose a simple yet effective algorithm to quantify uncertainty in SISR. Unlike conventional approaches for uncertainty estimation, we suggest that a more significant weight is put on the pixels with higher uncertainty than the pixels with lower uncertainty. In addition, an attention mask $\mathbf{\Lambda}$ that scales the gradient of the individual pixel with respect to its variance is employed to switch (3) from a learned attenuation loss to learned attention. To achieve this, we reuse the log variance of pixels by putting it into a sigmoidal function:

$$\begin{aligned}\mathbf{\Lambda} = \mathrm{sigmoid}(\mathbf{s}) &= \frac{1}{1+e^{-\mathbf{s}}}\\ &= \frac{\sigma(I^{SR})^2}{1+\sigma(I^{SR})^2},\end{aligned}$$

where $\mathbf{s}_i = \log\sigma(I_i^{SR})^2$, the log variance of the predicted SR image. Each element of attention masks $\mathbf{\Lambda}$ has a value close to one or zero, if the variance of the corresponding pixel is high and low enough, respectively.

By leveraging the fact that attention matrix $\mathbf{\Lambda}$ implies the difficulty of each pixel, we introduce a novel attention model, Gradient Rescaling Attention Model (GRAM), which rescales the gradient of $\mathcal{L}_{\mathrm{MLE}}$ with respect to $\mathbf{\Lambda}$. For a given weight metrix in a neural network θ, we define an arbitrary loss gradient $\nabla_\theta\mathcal{L}_{\mathrm{GRAM}}$. To estimate $\nabla_\theta\mathcal{L}_{\mathrm{GRAM}}$, the attention matrix is applied to the gradient of (3) by the Hadamard product ($\odot$):

$$\nabla_\theta\mathcal{L}_{\mathrm{GRAM}} \leftarrow \mathbf{\Lambda}\odot\nabla_\theta\mathcal{L}_{\mathrm{MLE}}. \tag{4}$$

The attention matrix $\mathbf{\Lambda}$ rescales gradient $\nabla_\theta\mathcal{L}_{\mathrm{MLE}}$. Therefore, the gradient of a high variance pixel has more impact on updating $\mathbf{W}$ than that of a low variance pixel. All we need to change the uncertainty loss into GRAM are sigmoid operations and multiplications; the overhead of GRAM compared to the total training procedure is negligible.

C. Why gradient rescaling?

When reusing the log variance as the attention, log variance $\mathbf{s}$ and attention matrix $\mathbf{\Lambda}$ themselves should not lose their identity. That is, if an attention mechanism affects the gradient estimates of the log variance or in the opposite way, one of two may be canceled out because $\mathbf{\Lambda} = \frac{\sigma(I^{SR})^2}{1+\sigma(I^{SR})^2}$ behaves oppositely to $\exp(-\mathbf{s}) = \frac{1}{\sigma(I^{SR})^2}$: the latter decreases while the former increases when the variance goes high. Suppose that $\mathbf{\Lambda} = \frac{\sigma(I^{SR})^2}{1+\sigma(I^{SR})^2}$ is directly scales the MLE loss (2). Then the pixel-wise loss becomes:

$$\begin{aligned}&\mathbf{\Lambda}\Big(\frac{1}{\sigma(I^{SR})^2}\|I^{HR}-I^{SR}\|^2+\log\sigma(I^{SR})^2\Big)\\ &\quad= \frac{1}{1+\sigma(I^{SR})^2}\|I^{HR}-I^{SR}\|^2+\frac{\sigma(I^{SR})^2\log\sigma(I^{SR})^2}{1+\sigma(I^{SR})^2}.\end{aligned} \tag{5}$$

Since the Euclidian loss term is rescaled by $\frac{1}{1+\sigma(I^{SR})^2}$, the attenuation becomes less significant than (2) but there is no attention mechanism. Furthermore, the biased variance estimates could make the model harder to fit the original likelihood.

However, GRAM modifies the gradient of $\mathcal{L}_{\mathrm{MLE}}$. Note that the attention mask $\mathbf{\Lambda}$ is not differentiated because it is involved in the learning process *after* the loss is differentiated. Thus, it preserves both the attention effect and variance estimates.

Moreover, GRAM can be easily implemented in existing deep learning frameworks [14], [15]. In practice, one can implement GRAM by Hadamard multiplication of $\mathbf{\Lambda}$ with $\mathcal{L}_{\mathrm{MLE}}$ (3). It implies that a deep learning framework of $\mathbf{\Lambda}$ is not going to be differentiated when calculating the gradient[1].

IV. Experiments

A. Datasets

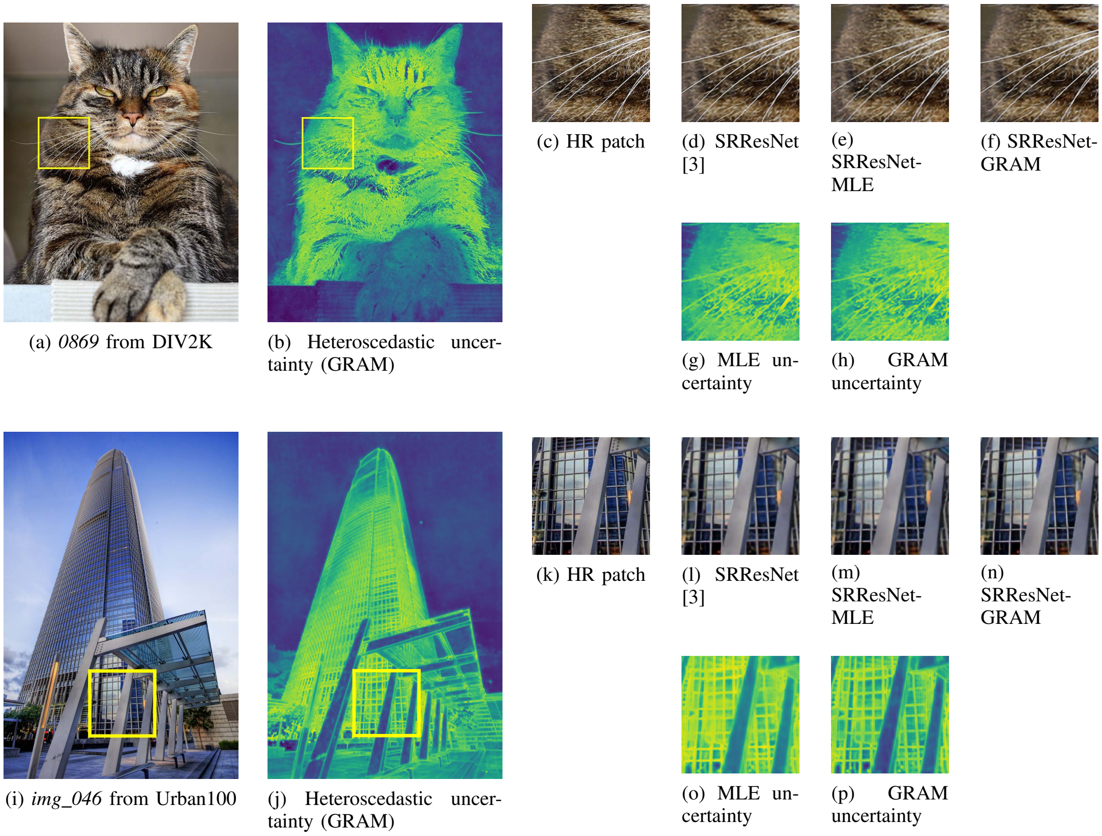
We use four standard benchmark datasets (Set5, Set14, BSD100, and Urban100) and a newly proposed high resolution image dataset (DIV2K) for experiments. We train SRCNNs based on the SRResNet [3] network with the DIV2K training set, and then test them with the five benchmarks. We use the validation set on the DIV2K dataset because the test set is not available.

B. Implementation details

We use a similar architecture of SRResNet in [3], except for the last convolution layer. The last layer of our SRCNN is duplicated: one for the SR image and the other for the log variance. Fig. 2 illustrates the overall configuration. Specifically, we use a kernel size of 9 for non-residual convolution layers and 3 for residual blocks. The Leaky ReLU function is assigned for activations. We also use the pixel shuffling used in [3] to upscale the intermediate feature map. We set the number of residual blocks M as 16 and the number of channels of the intermediate feature map as 64. SRResNet is designed for predicting a $\times 4$ upscaled SR image. We implement SRResNet on our framework for fair comparisons.

Further, for all experiments, we use 16 randomly cropped 96×96 RGB images from the DIV2K training dataset as the target HR image minibatch during training. Then, we produce 16 24×24 LR patches with the bicubic interpolation. During preprocessing, we scale HR patches from -1 to 1 and LR patches from 0 to 1. All training minibatches are randomly flipped and rotated in 90, 180, and 270 degrees for data

[1] Commonly used deep learning frameworks provide this function. For example, in Tensorflow [14], the notification function is $\mathrm{tf.stop\_gradient}()$.



(a) *0869* from DIV2K

(b) Heteroscedastic uncertainty (GRAM)

(c) HR patch

(d) SRResNet [3]

(e) SRResNet-MLE

(f) SRResNet-GRAM

(g) MLE uncertainty

(h) GRAM uncertainty

(i) *img_046* from Urban100

(j) Heteroscedastic uncertainty (GRAM)

(k) HR patch

(l) SRResNet [3]

(m) SRResNet-MLE

(n) SRResNet-GRAM

(o) MLE uncertainty

(p) GRAM uncertainty

Fig. 3: Comparison of ×4 results on DIV2K (a-h) and Urban100 (i-p) from SRResNet [3], `SRResNet-MLE` and `SRResNet-GRAM`. In the uncertainty map, the brighter, the higher uncertainty. Best viewed in zoom and color.

TABLE I: PSNR and SSIM evaluation. Presented in the form of PSNR(dB)/SSIM and the performance difference from SRResNet. SRResNet* refers to our implementation of the baseline SRResNet [3].

Dataset	Bicubic	SRResNet*	SRResNet-MLE	**SRResNet-GRAM** (Ours)
Set5	28.42/0.8099	31.87/0.8885	31.43/0.8834 (-0.44/-0.0051)	31.82/0.8880 (**-0.05/-0.0005**)
Set14	26.00/0.7025	28.31/0.7749	27.98/0.7680 (-0.33/-0.0069)	28.24/0.7734 (**-0.07/-0.0015**)
BSD100	25.96/0.6692	27.48/0.7332	27.25/0.7260 (-0.23/-0.0072)	27.45/0.7313 (**-0.03/-0.0019**)
Urban100	23.14/0.6583	25.86/0.7766	25.21/0.7555 (-0.65/-0.0211)	25.68/0.7709 (**-0.18/-0.0057**)
DIV2K [16]	28.09/0.7740	30.26/0.8335	29.87/0.8252 (-0.39/-0.0083)	30.15/0.8312 (**-0.09/-0.0023**)

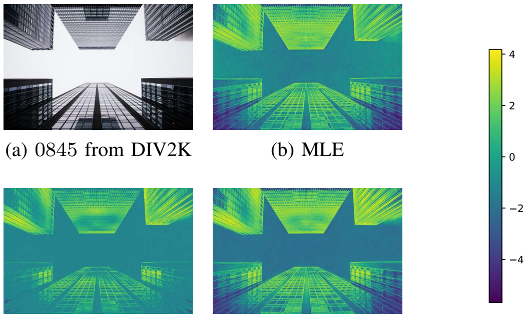
augmentation. An adaptive learning rate method called `Adam Optimizer` is chosen with $\beta_1 = 0.9, \beta_2 = 0.99$. We train all networks with a learning rate of 0.0001 and 2×10^5 iterations and additional 10^5 iterations with a learning rate of 0.00001.

C. Quantitative comparison

We conduct the evaluation of PSNR and SSIM on five datasets: Set5, Set14, Urban100, BSD100 and DIV2K [16]. The baseline for the quantitative comparison is SRResNet, which is trained by the MSE loss $\mathcal{L}_{SR}$ (1). We will refer `SRResNet-MLE` as the SRResNet which is trained by uncertainty loss $\mathcal{L}_{\text{MLE}}$ (3), and `SRResNet-GRAM` as the SRResNet which is trained by the GRAM loss (4). First, we estimate how much performance is degraded by (3). For Set5, the mean PSNR and SSIM values of the baseline SRResNet are 31.87 dB and 0.8885, respectively. However, the respective mean PSNR and SSIM values of `SRResNet-MLE` are 31.43 dB and 0.8834. Similarly, `SRResNet-MLE` shows the worst performances for all benchmarks. We suppose that the uncertainty loss makes the neural network learn less from challenging data. Strikingly, the performance metrics of `SRResNet-GRAM` are the as same level as those of SRResNet—the performance differences are negligible compared to other models in all benchmarks. The results confirm that our method does not suffer from performance loss while predicting the heteroscedastic uncertainty, and are summarized in Table I.

D. Qualitative comparison

Visual comparison of the predicted mean of ×4 SR results shows much insight—`SRResNet-MLE` has difficulty in upscaling high frequency patches. In Fig. 3 (e), the upscaled cat's mustaches are distorted. Also, Fig. 3 (m) exposes hazy check



(a) 0845 from DIV2K (b) MLE

(c) Loss rescaling (d) Gradient rescaling

Fig. 4: Log variance of (a) 0845 from DIV2K, (b) predicted by `SRResNet-MLE`, (c) loss rescaling attention, and (d) `SRResNet-GRAM`.

patterns compared to Fig. 3 (l). However, `SRResNet-GRAM` preserves those challenging regions at the same level as SRResNet—Fig. 3 (f) and (n) contain sharper details than Fig. 3 (e) and (m), respectively. These confirm that the $\nabla_\theta \mathcal{L}_{\text{GRAM}}$ can produce a better model than the conventional attenuation loss.

Further, we have investigated how `SRResNet-MLE` and `SRResNet-GRAM` capture uncertainty in SISR. Both models predict the variance of burdensome pixels as high while predicting the variance of more accessible parts of images such as backgrounds or smooth textures as low. For instance, the predicted variance of the cat's front feet in Fig. 3 (b) is lower than the cat's body and face because the feet are out of focus and blurred than other parts. For this reason, GRAM can capture the heteroscedastic uncertainty while it uses uncertainty as the attention.

We also compare GRAM to the loss rescaling attention (5) to investigate what if attention mask Λ is directly involved in the training loss. As we expected in III-C, the SRResNet model which is trained by the loss rescaling attention captures uncertainty imprecisely. In Fig. 4, (b) and (d) are akin to each other, where the result from the loss rescaling attention shows unnecessarily higher uncertainty on low variance image patches (the bottom left and the bottom right corners of (a)). Assuming that the uncertainty presented in 4 (b) is accurate, our proposed gradient rescaling method predicts data uncertainty much better than the loss rescaling.

V. Conclusion

In this paper, we proposed a learning method to quantify data uncertainty without suffering from performance degradation in Single Image Super Resolution (SISR). Since the conventional uncertainty loss function is designed to attenuate the Euclidian distance of pixels with high predictive uncertainty, we claimed that the primary source of the problem is the attenuation mechanism that is not appropriate for ill-posed inverse problems such as SISR. To resolve the issue, we proposed a new attention model that rescales the gradient of the conventional uncertainty loss. The experimental results verify that our method maintains the quality of SISR results compared to the loss function without uncertainty while confirming that the uncertainty loss without attention deteriorates the SISR quality.

Acknowledgment

This research was funded and conducted under 「the Competency Development Program for Industry Specialists」 of the Korean Ministry of Trade, Industry and Energy (MOTIE), operated by Korea Institute for Advancement of Technology (KIAT). (No. N0001883, HRD program for Intelligent semiconductor Industry)

References

[1] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[2] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[3] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[4] J.-H. Kim, J.-H. Choi, M. Cheon, and J.-S. Lee, "Ram: Residual attention module for single image super-resolution," *arXiv preprint arXiv:1811.12043*, 2018.

[5] W.-Y. Lee, P.-Y. Chuang, and Y.-C. F. Wang, "Perceptual quality preserving image super-resolution via channel attention," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 1737–1741.

[6] S. G. Finlayson, H. W. Chung, I. S. Kohane, and A. L. Beam, "Adversarial attacks against medical deep learning systems," *arXiv preprint arXiv:1804.05296*, 2018.

[7] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, 2017, pp. 6402–6413.

[8] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018.

[9] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 7047–7058.

[10] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," *arXiv preprint arXiv:1711.09325*, 2017.

[11] Y. Liu *et al.*, "An attention-based approach for single image super resolution," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 2777–2784.

[12] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 065–11 074.

[13] Y. Zhang *et al.*, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 286–301.

[14] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[15] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.

[16] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 114–125.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.