Article

SNPD: Semi-Supervised Neural Process Dehazing Network with Asymmetry Pseudo Labels

Fan Zhou, Xiaozhe Meng, Yuxin Feng and Zhuo Su *

Research Institute of Sun Yat-sen University in Shenzhen, School of Computer Science and Engineering, National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou 510006, China; isszf@mail.sysu.edu.cn (F.Z.); mengxzh5@mail2.sysu.edu.cn (X.M.); fengyx26@mail2.sysu.edu.cn (Y.F.)
* Correspondence: suzhuo3@mail.sysu.edu.cn

Abstract: Haze can cause a significant reduction in the contrast and brightness of images. CNN-based methods have achieved benign performance on synthetic data. However, they show weak generalization performance on real data because they are only trained on fully labeled data, ignoring the role of natural data in the network. That is, there exists distribution shift. In addition to using little real data for training image dehazing networks in the literature, few studies have designed losses to constrain the intermediate latent space and the output simultaneously. This paper presents a semi-supervised neural process dehazing network with asymmetry pseudo labels. First, we use labeled data to train a backbone network and save intermediate latent features and parameters. Then, in the latent space, the neural process maps the latent features of real data to the latent space of synthetic data to generate one pseudo label. One neural process loss is proposed here. For situations where the image may be darker after dehazing, another pseudo label is created, and one new loss is used to guide the dehazing result at the output end. We combine the two pseudo labels with designed losses to suppress the distribution shift and guide better dehazing results. Finally, the artificial and hazy natural images are tested experimentally to demonstrate the method’s effectiveness.

Keywords: image dehazing; neural process; pseudo label; distribution shift

1. Introduction

Haze can cause significant changes in the data quality of an image. Captured images in foggy weather have reduced contrast and brightness, which adversely causes difficulty for further perception and understanding for subsequent tasks. Therefore, haze removal, especially single image dehazing, is highly practical and realistic with comprehensive academic and industry value [1–4]. At present, researchers adopt a well-received physical model [5], which is formulated as as:

\[ I(x) = J(x)t(x) + A(1 - t(x)), \] (1)

where \( I(x) \) refers to the captured hazy image, \( J(x) \) is defined as a clear image (scene light), and \( A \) represents atmospheric light. \( t \) is the medium transmission map, which indicates the amount of scene light passing through the aerosol to reach the camera. \( t \) is a function of the scene depth \( d(x) \) and the scattering coefficient \( \beta \), that is, \( t(x) = e^{-\beta d(x)} \). The method of image haze removal is to estimate \( t \) and \( A \) through an acquired hazy image \( I(x) \) to calculate a haze-free image \( J(x) \).

The task of image dehazing is limited by several constraints: (1) depth map, atmospheric light intensity, and light wavelength, and (2) lack of labeled training data. Limited by the above conditions, researchers have proposed some prior methods, such as dark channel prior [6], color-line prior [7], and nonlocal color prior [8], but irregular lighting or white areas would violate these priors. To make up for the shortcomings of the prior methods, researchers use CNN methods to obtain dehazing output. They either obtain
the transmission map [9] or directly calculate the dehazing results [10–13]. Although these methods have achieved excellent performance, they are supervised networks and need numerous labeled data for training (such as the NYU Depth dataset [14] and RESIDE [15]), ignoring the role of real data in the training process. In addition, the artificial dataset contains little background and scene depth, which makes the distribution inconsistent between the artificial and the real images. Thus, algorithms based on deep learning are usually limited to synthetic training datasets and cannot be well generalized to real-world hazy images. For example, as presented in Figure 1, the atmospheric light of the synthetic data in Figure 1a is obviously greater than that of Figure 1c in red rectangles, while Figure 1b,d prove that the model trained on the artificial data does not perform well on the actual images. Another important thing is that the above CNNs either constrain latent space in middle layers [16] or introduce designed losses in the output end [10–13] to limit predicted results; they do not take both into account simultaneously.

![Figure 1. Comparison of hazy images and corresponding dehazing results. (a) One synthetic hazy image. (b) The dehazing result using a dehazing network. (c) One natural hazy image. (d) The dehazing result using the same network in (b). Some areas are highlighted by red rectangles for a better comparison and explanation.](image)

Recently, Li et al. [17] designed a semi-supervised learning dehazing algorithm, which used the $l_1$ regular dark channel loss and GAN loss to train the network during the unsupervised training stage. Rajeev et al. [16] considered the rain removal problem in function space, and they proposed a semi-supervised learning scheme that combined the synthetic and real images in the training process based on the Gaussian process (GP) [18]. As a result, they obtained good generalization performance on real-world images. However, although the above methods use real data to train the network, Li et al. [17] did not compensate for the flaws of physical priors, and the GP [16] has extremely high time complexity ($O(N^3)$) and must manually select the appropriate kernel function. In addition, [16] use GP for rain removal but not for dehazing.

We propose a semi-supervised neural process dehazing network with asymmetry pseudo labels based on the above problems and existing studies. The proposed model involves a supervised training phase on labeled data and a double pseudo label stage on unlabeled data. The network is constrained by mean square error and perceptual loss in the supervised training phase. In this stage, we train the network and save the features into a matrix to prepare for the first pseudo label generation of neural process (NP) modeling. In the asymmetry pseudo label training stage, we assume a functional relationship exists between the latent spaces of synthetic data and real data; that is, unlabeled latent features can be formulated as a weighted combination of the labeled latent features. These weights represent the randomness of the labeled features being used to express the unlabeled feature point. The NP can map the hidden feature value of real data to the hidden space of synthetic data and generate the mapping value named the pseudo latent feature label (PLFL). The PLFL is in the hidden space of synthetic data, so reducing the distance between the unlabeled data predicted value (the encoder output of unlabeled data) and the PLFL could minimize the difference between the two domains in the function space. We calculate the distance between PLFL and feature predicted values with mean square error. However, darker dehazing results may occur at the output end, so we design new loss based on
the contrast limited adaptive histogram equalization (HE) to guide the dehazing results. Specifically, we use the HE to generate the second pseudo label and propose the HE loss, enhancing the illumination and contrast.

The proposed method uses real and synthetic data and simultaneously constrains the hidden space and output result. As a result, good generalization performance is obtained on the real haze image, and it also effectively overcomes the high complexity defect of GP. Overall, the main contributions of this research are as follows:

- We design a semi-supervised dehazing neural network using asymmetry pseudo labels based on neural process and HE, which uses synthetic and natural data information. We build functional relations between artificial data and real data in latent space from function space and project the real data into the latent space of synthetic data through neural process.
- We use the neural process and HE to generate the asymmetry pseudo labels, respectively. The neural process is employed to map the hidden features into the feature space of labeled data to generate the pseudo latent feature label. Another pseudo label of HE is to guide the dehazing results at the output end to prevent the darker results from appearing for real dehazing results.
- The proposed method combines the intermediate layer constraint and output end loss simultaneously to generate pleasing results. We demonstrate the effectiveness of the proposed algorithm, especially on actual hazy images, and achieve beautiful performance in terms of both subjective and objective assessment.

2. Related Works

This section will review and summarize some of the latest haze removal methods including prior-based, supervised, and semi-supervised haze removal methods.

2.1. Single Image Dehazing

These methods capture some physical clues as statistical priors from the clean images and then use them to calculate the transmission map. He et al. [6] found that a pixel value in every three channels in the hazy image was close to zero, so they called it dark channel prior (DCP). DCP has been successfully applied to dehazing, but it still has some limitations in white areas. Fattal [7] proposed a color-lines model from observing local pixels in an image with a linear distribution in the RGB space. Berman et al. [8] observed that the RGB pixels could be aggregated into clusters in the haze-free image, but they degenerate to a line in the corresponding hazy image. In [19], Ju et al. introduced one novel prior, i.e., gamma correction prior (GCP). They firstly acquired a virtual transformation of hazy images with GCP, and then they designed a global dehazing strategy by extracting the depth map from a hazy image and its virtual transformation. Considering brightness and contrast, Liu et al. [20] expressed the problem of haze removal as brightness reconstruction based on statistical analysis of fog-free images. Similarly, Bui et al. [21] clustered haze pixels into clusters in RGB space and used color ellipsoids to estimate transmission value. This method maximizes contrast while avoiding oversaturation. However, the aforementioned prior assumptions are often broken in some realistic scenes.

Owing to the limitation of the prior methods, learning-based techniques have been widely used to solve the dehazing problem. Zhu et al. [22] found that the color decayed with scene depth. They proposed a color decay prior, and further constructed a linear regression model of scene depth, and obtained the dehazing results through a supervised regression method. Unlike [22], more convolutional neural network (CNN) techniques have been developed. Cai et al. [9] firstly proposed an end-to-end transmission estimation CNN. Similarly, in [23], Ren et al. established a multiscale CNN (MSCNN) to remove haze by mapping the relationship between hazy images and transmission. Furthermore, [13] introduced a densely connected network with a discriminator to acquire a transmission map and atmospheric light simultaneously. The discriminator is used to ensure that the transmission map is strongly related to the dehazing results. The above methods still
obtain the transmission image first, and then obtain the dehazing images based on the atmospheric light scattering model. The authors of [12,24–26] put forward the end-to-end dehazing methods. In [27], Pang et al. designed HRGAN including a discriminator network and a generator network to achieve haze removal. Qin et al. [12] introduced attention mechanisms to channel and pixel, respectively. Dong et al. [24] established a multiscale enhanced defogging network (MSBDN) using a complex U-Net-like structure. Scholars have developed unsupervised solutions based on the atmospheric scattering model. Pan et al. [28] believed that the image restoration results should be consistent with the observed input under specific physical models, so they proposed to use the physical model to guide the specific task in the GAN framework. To better process the actual hazy images and avoid domain shift, Golts et al. [29] abandoned artificial data, such as the RESIDE dataset. They introduced completely unsupervised dehazing architecture with dark channel prior loss. Li et al. [25] regarded one hazy image as the coupling of a dehazing layer, transmission layer, and atmospheric light layer. To restore binocular hazy images, Pang et al. [30] developed a binocular image dehazing network (BidNet), which could survey the relation between binocular image pairs to improve the dehazing quality. Wu et al. [31] introduced the contrastive strategy into a CNN and employed an adaptive mixup and dynamic feature module, and they acquired very competitive performance. Furthermore, Zhang et al. [32] targeted video dehazing. They first provided a video hazy dataset and explored the temporal information with a confidential and improved network.

2.2. Semi-Supervised Image Dehazing

In recent years, some semi-supervised learning models have been developed to resolve low-level visual tasks. Wei et al. [33] used the mean absolute error loss to train a network for labeled data, and by narrowing the Kullback–Leibler (KL) divergence of rain residual distribution of the labeled and unlabeled images, the artificial rain distribution is closer to the natural rain distribution. Moreover, Rajeev et al. [16] supposed an unlabeled image could be formulated as a linear weighted combination of the labeled data in hidden space and provided a semi-supervised learning framework based on GP. However, the GP has extremely high computational time complexity and requires manual selection of a suitable kernel function.

There are some other techniques combined with semi-supervised methods, such as adversarial training [34] or pseudo labeling [35]. In these methods, unsupervised losses are based on domain-specific knowledge and cannot directly apply to image defogging. Li et al. [17] proposed a semi-supervised learning (SSL) defogging method, which first uses MSE, perceptual loss, and GAN loss to train on synthetic data, and then fine-tunes the model through DCP loss and total variation loss. Chen et al. [1] proposed a method similar to [17], first using the most advanced defogging framework (e.g., FFA, MSBDN) for labeled data, and then using prior losses for fine-tuning. Since [1,17] used prior losses, there may be cases violating the prior assumptions. Shao et al. [36] introduced the domain adaptation adversarial (DAD) method, using CycleGAN [34] and the domain adversarial method [37], to translate the synthetic data into real hazy data. However, this work needs to calculate the scene depth map, although the depth map is difficult to obtain and, moreover, DAD may cause domain mismatch. Lai et al. [38] proposed a deep network to estimate depth maps, which enhanced the smoothness of estimated depth maps by adding image alignment errors and using regularization losses. Then, a domain adversarial strategy is used to make source and target domain features indistinguishable in feature space.

These above methods have acquired nice results. However, directly calculating dehazing results using CNNs or other schemes is not an easy task because image dehazing is an ill-posed problem. Unlike most of these methods that design the loss at the output, we generate two pseudo labels in the latent space and the output, constraining the two positions at the same time.
3. Materials and Methods

Let \( D = D_l \cup D_u \) denote the training data, where \( D_l = \{x_i, y_i\}_{i=1}^n \) represents \( n \) labeled synthetic hazy images and \( D_u = \{x_i\}_{i=1}^m \) represents \( m \) unlabeled hazy images. As shown in Figure 2, the framework proposed in this paper use MSBDN as a backbone, the network contains an encoder \( H(\cdot) \) and a decoder \( G(\cdot) \), and the encoder and decoder contain 4 residual modules, respectively. The parameters of the encoder and decoder are \( \theta_{enc} \) and \( \theta_{dec} \). The training of our strategy includes two procedures. First, we fit our network on the synthetic data. Second, we use NP and HE loss to train the network on real data. Our strategy of dehazing training on unlabeled data requires establishing relationships in the feature space through the designed NP and mapping the real data features to the labeled one’s feature space.

**Figure 2.** The flow chart of the semi-supervised dehazing network with asymmetry pseudo labels in neural process regression. We use synthetic data in the supervised training phase while preserving their hidden features. In the asymmetry pseudo label stage, we use real data for training, obtain intermediate hidden features, filter the features from the synthetic data, and input them into the NP module to map the natural hidden features to the artificial data feature space.

### 3.1. Supervised Image Dehazing

We input labeled image \( x_i \) into the encoder \( H(\cdot) \) and obtain hidden features \( z_i = H(x_i, \theta_{enc}) \), and when inputting \( z_i \) into the decoder, the predicted dehazing result \( \hat{y}_i = G(z_i, \theta_{dec}) \) could be obtained. The whole process is \( \hat{y}_i = G(H(x_i, \theta_{enc}), \theta_{dec}) \). In order to fuse labeled information into unlabeled space, we store the middle feature vectors \( z_i \) of all the artificial images \( x_i \) in a matrix \( M \) that is \( M = \{z_{i,j}\}_{i=1}^n \). The dimensions of \( z_{i,j} \) are \( 1 \times 32 \times 16 \times 16 \), and it is transformed into a \( 1 \times 8,192 \) vector, then the size of the matrix \( M \) is \( n \times 8,192 \).

In this phase, the mean square error and perceptual loss are employed to constrain the supervised training process on the artificial data. The total loss of labeled data is:

\[
\mathcal{L}_{sup} = \mathcal{L}_l + \lambda_1 \mathcal{L}_p, \tag{2}
\]

where \( \lambda_1 \) is a hyper-parameter, the mean square error \( \mathcal{L}_l \) and the perceptual loss \( \mathcal{L}_p \) are defined as follows:

\[
\mathcal{L}_l = ||\hat{y}_i - y||_2, \quad \mathcal{L}_p = ||\Psi_{VGG}(\hat{y}_i) - \Psi_{VGG}(y)||_2, \tag{3}
\]

where \( \hat{y}_i \) is the predicted output, \( y \) is the ground-truth, and \( \Psi_{VGG} \) is the VGG-16 [39] network in the Pytorch warehouse. Here, the \( \lambda_1 = 0.1 \) in Equation (2).

### 3.2. Asymmetry Pseudo Label Dehazing

We transfer the model and parameters trained on the synthetic data here. We input a real hazy image \( x_{u,j} \in D_u \) into the encoder \( H \), obtain \( z_{u,j} = H(x_{u,j}, \theta_{enc}) \), \( z_{u,j} \) is the predicted
hidden feature of the encoder $H$ of the real foggy image. To fuse the information of the labeled data in this phase, $M$ could provide useful information.

Like GP, NP [40,41] is also considered from the function space, and NP combines the advantages of the GP and neural networks. Our idea is to map the predicted value $z_{u,i}$ to the latent space of the labeled data with NP, and then generate the pseudo latent feature label (PLFL). After using NP for the aforementioned extracted feature matrix $M$ and $z_{u,i}$, the PLFL is already in the latent space of synthetic hazy data, then reducing the distance between the unlabeled data predicted value (the encoder output of unlabeled data) and the PLFL could reduce the difference between the two domains.

We assume that the unlabeled feature $z_{u,i}$ of hazy data could be expressed with latent vectors $z_{l,i}$ of synthetic hazy data, then:

$$z_{u,i} = \sum_{l=1}^{m} \omega_l z_{l,i} + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2),$$  

(4)

where $\omega_l$ are coefficients, and these coefficients indicate the randomness of the artificial hazy feature points being used to express the unlabeled hazy feature point, and $\varepsilon$ is noise and follows the normal distribution $N(0, \sigma^2)$. Of course, $z_{u,i}$ may be a nonlinear combination of $z_{l,i}$, and we further suppose there exists a function distribution $F$ so that $f \sim F$, $F$ could map the features of the real hazy data to the latent space of the labeled data. One function $f$ is sampled from the distribution $F$ so that:

\[
\begin{align*}
\hat{z}_{l,i} &= f(z_{u,i}) + \varepsilon, \\
z_{l,i} &= f(z_{l,i}) + \varepsilon, \\
\varepsilon &\sim N(0, \sigma^2), \quad j = 1, \ldots, m,
\end{align*}
\]

(5)

where $\hat{z}_{l,i}$ is the PLFL obtained by $f(z_{u,i})$, $m$ is the number of natural hazy images, $f$ could map $z_{l,i}$ to itself for synthetic hazy features. According to the Bayesian rule, the joint marginal distribution of all $z_{l,i}$ and $\hat{z}_{l,i}$ is defined as:

$$p(\hat{z}_{l,1:n}, z_{l,1:n}) = \int p(f) p(\hat{z}_{l,1:n} | f, z_{l,1:n}, z_{u,i}) df,$$

(6)

where $p$ denotes the abstract probability distribution over all labeled and unlabeled hazy latent vectors, the $1:n$ expresses any sorted labeled latent features. If all extracted latent features are independent,

$$p(\hat{z}_{l,j}, z_{l,1:n} | f, z_{u,j}, z_{1:n}) = \prod_{i=1}^{n} N\left(z_{l,i} | f(z_{l,i}), \sigma^2\right) N\left(\hat{z}_{l,j} | f(z_{u,j}), \sigma^2\right).$$

(7)

Inserting Equation (7) into Equation (6), the above formula is specified by:

$$p(\hat{z}_{l,j}, z_{l,1:n} | f, z_{u,j}, z_{1:n}) = \int p(f) \prod_{i=1}^{n} N\left(z_{l,i} | f(z_{l,i}), \sigma^2\right) N\left(\hat{z}_{l,j} | f(z_{u,j}), \sigma^2\right) df.$$  

(8)

We suppose that mapping function distribution $F$ can be parameterized by a high-dimensional random vector $\alpha$, that is, the randomness of $F$ is determined by $\alpha$, then for the learnable function $g$, $F(z_{l,i}) = g(z_{l,i}, \alpha)$, $g$ can be implemented with an encoder. The generative model then follows from (8):

$$p(\alpha, \hat{z}_{l,j}, z_{l,1:n} | f, z_{u,j}, z_{1:n}) = p(\alpha) \prod_{i=1}^{n} N\left(z_{l,i} | g(z_{l,i}, \alpha), \sigma^2\right) N\left(\hat{z}_{l,j} | g(z_{u,j}, \alpha), \sigma^2\right),$$  

(9)

where we assume $p(\alpha)$ is a multivariate standard normal distribution obeying the idea of variational auto-encoders [42], and $g(z_{l,i}; \alpha)$ is a neural network which captures the complexities of the model. Since the decoder $g$ is nonlinear, according to the variational method, the evidence lower bound (ELBO) is directly given by:
we use the posterior $q(\alpha | z_{1:n})$ to estimate it, then:

$$
\log p\left(\hat{z}_{ij} | z_{u,j}, z_{1:n}\right) \geq E_{q(\alpha|z_{1:n},z_{u,j},\hat{z}_{ij})} \left[ \log p\left(\hat{z}_{ij} | \alpha, z_{u,j}\right) + \log \frac{p(\alpha|z_{1:n})}{q(\alpha | z_{1:n}, \hat{z}_{ij}, z_{1:n})} \right].
$$

(10)

Equation (10) gives the ELBO of the predicted latent feature. To maximize the log-likelihood, we need to maximize the ELBO. Since conditional prior $p(\alpha | z_{1:n})$ is intractable, we use the posterior $q(\alpha | z_{1:n})$ to estimate it, then:

$$
\log p\left(\hat{z}_{ij} | z_{u,j}, z_{1:n}\right) \geq E_{q(\alpha|z_{1:n},z_{u,j},\hat{z}_{ij})} \left[ \log p\left(\hat{z}_{ij} | \alpha, z_{u,j}\right) + \log \frac{q(\alpha | z_{1:n}, \hat{z}_{ij}, z_{1:n})}{q(\alpha | z_{u,j}, \hat{z}_{ij}, z_{1:n})} \right].
$$

(11)

We notice that the above formula could be transformed into another abbreviated form, i.e.,

$$
E_{q(\alpha|z_{1:n},z_{u,j},\hat{z}_{ij})} \log \frac{q(\alpha | z_{1:n}, \hat{z}_{ij}, z_{1:n})}{q(\alpha | z_{u,j}, \hat{z}_{ij}, z_{1:n})} = -KL\left[q(\alpha | z_{1:n}, \hat{z}_{ij}) \| q(\alpha | z_{1:n})\right].
$$

(12)

We find the ELBO which represents the lower bound of the conditional distribution probability of PLFL. The larger the ELBO is, the larger the likelihood function of PLFL is. Therefore, our aim is to make ELBO maximum. We define the following loss function:

$$
L_{np} = E_q (\alpha | z_{1:n}, z_{u,j}, \hat{z}_{ij}) \left[ \log p\left(\hat{z}_{ij} | \alpha, z_{u,j}\right) \right] + KL\left[q(\alpha | z_{1:n}, z_{u,j}, \hat{z}_{ij}) \| q(\alpha | z_{1:n})\right].
$$

(13)

Reducing the distance between the unlabeled data predicted value (the latent feature of unlabeled data) and the PLFL $\hat{z}_{ij}$ could minimize the distribution shift between the two domains in the function space. We calculate the difference between PLFL $\hat{z}_{ij}$ and feature predicted values $z_{u,j}$ with mean square error. Therefore, we redefine the loss $L_{np}$ as:

$$
L_{up} = -L_{np} + ||z_{u,j} - \hat{z}_{ij}||^2,
$$

(14)

where the minus sign makes the maximized ELBO turn into the minimized ELBO, $z_{u,j}$ is the latent vector obtained by feeding a natural hazy image $x_{u,j}$ through the encoder $H$.

The deduction of the formulation (10) is:

$$
\log p\left(\hat{z}_{ij} | z_{u,j}, z_{1:n}\right)
= \log \sum_{\alpha} p\left(\alpha, \hat{z}_{ij} | z_{u,j}, z_{1:n}\right)
= \log \sum_{\alpha} \frac{p(\alpha, \hat{z}_{ij}|z_{u,j}, z_{1:n}) q(\alpha | z_{u,j}, \hat{z}_{ij}, z_{1:n})}{q(\alpha|z_{1:n}, \hat{z}_{ij}, z_{1:n})}
\geq E_q(\alpha|z_{u,j}, z_{1:n}, \hat{z}_{ij}) \left[ \log \frac{p(\alpha|z_{1:n}, z_{u,j}, \hat{z}_{ij})}{q(\alpha|z_{1:n}, \hat{z}_{ij}, z_{1:n})} \right]
= E_q(\alpha|z_{u,j}, \hat{z}_{ij}, z_{1:n}) \left[ \log \frac{p(\alpha|z_{1:n}, \hat{z}_{ij})}{q(\alpha|z_{u,j}, \hat{z}_{ij}, z_{1:n})} \right]
= E_q(\alpha|z_{u,j}, \hat{z}_{ij}, z_{1:n}) \left[ \log \frac{p(\alpha|z_{1:n}, \hat{z}_{ij})}{q(\alpha|z_{u,j}, \hat{z}_{ij}, z_{1:n})} \right].
$$

(15)

A more detailed theorem and deduction of NP can be found in [40,41]. Figure 3 shows an example of NP, from which we could see that NP owns the randomness ability of GP [18] in function space.

It should be pointed out that not all features in $M$ are strongly positively related to $z_{u,j}$. It takes a lot of computational power to use all the labeled features $1:n$, so we pick out the feature $z_{l,j}$ from $M$ that is most relevant to $z_{u,j}$ through the following cosine formula:

$$
\cos(z_{u,j}, z_{1:n}) = \frac{z_{u,j}^T z_{1:n}}{|z_{u,j}| \cdot |z_{1:n}|}.
$$

(16)
Figure 3. One example of NP dehazing procedure. The latent points in the figure and hazy images are in one to one correspondence and hazy data are data points in high-dimension space. Using NP to fit the curves based on the observed latent feature points in latent space, we could fit countless curves like GP. The NR encodes sampled feature points from output of the encoder \( H \) as intermediate features and establishes a multivariate normal distribution using one invariant representation. The newly unlabeled latent feature is combined with sampled feature points from one normal distribution and then we input them into one decoder to output the infinite predicted values.

We obtain \( \{c_1, c_2, \cdots, c_n\} \) by sorting the cosine values after calculation, and then pick out the \( k \) most relevant data. Here, the \( k \) is set to 32. In addition, the dehazing results of real data may be darker than raw unlabeled data, which is not very reasonable. For this phenomenon, we design HE loss. HE loss is applied to enhance the luminance and contrast. The HE is implemented as the following loss function:

\[
\mathcal{L}_{\text{HE}} = \|\hat{J} - J'\|_1, \tag{17}
\]

where \( \hat{J} \) expresses the real data predicted dehazing value of a network, and \( J' \) represents that we directly use the HE based on unlabeled data to generate pseudo GT.

We combine the supervised and asymmetry pseudo label training phase, and the total loss is defined as:

\[
\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sup}} + \lambda_{np}\mathcal{L}_{\text{np}} + \lambda_{HE}\mathcal{L}_{\text{HE}}, \tag{18}
\]

where \( \lambda_{np} \) and \( \lambda_{HE} \) are the hyper-parameters used to weigh the losses. It should be pointed out that we only hope the HE loss could enhance the illumination and contrast, but the inherent flaws of HE may negatively affect the results, so the \( \lambda_{HE} \) should be small.

3.3. Neural Process Module

The NP includes three parts: encoder \( h(\cdot, \theta) \) and \( h(\cdot, \phi) \), aggregator \( a \), and conditional decoder \( g(\cdot, \omega) \), where \( \theta, \phi \) and \( \omega \) are network parameters. Figure 4 shows the implementation step. Specifically, we input extracted feature pair \((z_{l,1:1,k}, z_{l,1:1,k})\) into the encoder \( h \) to obtain representation \( r_i = h(z_{l,1:1,k}, z_{l,1:1,k}) \). The aggregator \( a \) is responsible for determining two global first-order invariant representations \( s_c \) and \( r_c \). \( s_c \) is used to determine the parameters of the implicit distribution \( s \sim N(\mu(s_c), \sigma^2(s_c)) \), and the sample \( s \) is the key factor determining the neural network’s randomness. Another invariant representation \( r_c \) expresses the determining factor, all labeled latent components feature obey some attributes of this factor when there is a lot of data. It should be noted that randomness and deterministic factors enable NP to achieve the same function as GP, and the global representation uses the mean value operation to obtain \( r_c = a(r_i) = \frac{1}{n} \sum_{i=1}^{n} r_i \) and \( s_c = a(s_i) = \frac{1}{n} \sum_{i=1}^{n} s_i \). \( p(\hat{z}_{ij} | r_c, s_c, z_{u,j}) \) represents the prediction of the data, that is,
we sample from $N(\mu(r), I\sigma(r))$ to obtain $s$, and input random factor $s$, determined factor $r_c$, and $z_{u,j}$ into the decoder $g$ and obtain the output $\hat{z}_{l,j} = g(r_c, s, z_{u,j})$. The $\hat{z}_{l,j}$ is the PLFL.

Figure 4. The NP module includes three parts: encoder $h(\cdot, \theta)$, $h(\cdot, \phi)$, aggregator $a$, and conditional decoder $g(\cdot, w)$.

4. Experimental Results Analysis

In this section, we will prove the effectiveness of the proposed method and show the results compared with other methods.

4.1. Datasets

The current data-driven methods consume a large amount of paired data, especially some deep learning methods [12,24] that rely on indoor datasets for training. However, it is almost impossible for haze to appear indoors, so we only conduct training with outdoor data. We randomly select 5000 pairs of images from the outdoor training set (OTS) for training, and the test set uses the synthetic outdoor test set (SOTS), which contains 500 outdoor images. In the asymmetry pseudo label stage, we choose the Unannotated RealHazyImages (URHI) dataset in the subset of RESIDE to train the proposed method. The URHI contains 4807 real images of complex scenes with different haze concentrations. In the test phase, we use 4322 RTTS data provided by RESIDE and 32 real hazy images collected by Fattal [7].

4.2. Implementation Details

In the training stage, each image is randomly cropped to $256 \times 256$. The Adam optimizer is used for training, and we set the batch size as 12. The total epoch is set as 100. The initial learning rate equals $1.0 \times 10^{-4}$ and the learning rate decreases by 0.5 after every ten epochs. The super parameters $\lambda_{nf}$ and $\lambda_{HE}$ in Equation (18) are set as 1.0 and $1.0^{-1}$, respectively. All experiments are performed on an Ubuntu 18.04 system with NVIDIA GTX 2080Ti, Intel I5-7400 CPU, and PyTorch 1.2.0.

4.3. Results Comparison

To effectively evaluate the effectiveness of our method, this section presents the results of different methods for the outdoor synthetic haze images. We carry out qualitative and quantitative evaluations on the comparative experiments, respectively. Then, common SSIM and PSNR are used for quantitative comparison. In the comparative experiment, we choose 11 advanced methods to compare with ours. The comparison methods include prior-based methods DCP [6] and NLD [8], the fully supervised methods including AOD [10], EPDN [26], PDN [43], GDN [11], FFA [12], and MSBDN [24], unsupervised methods
including ZID [25], and domain adaptation methods SSL [17], DAD [36], and PSD [1], and these learning-based methods were all trained on SOTS.

The outdoor dehazing results for the synthetic data are shown in Figure 5. We can see that the color of the sky area derived from DCP and NLD is chaotic, and the image of DCP is dark. Although ZID is an unsupervised method, it avoids the problems caused by distribution shift, and it uses dark channel loss, so the sky area has the same problem as DCP. There are apparent fog residues in the AOD results, the color in the EPDN results is inconsistent with GT, and the color turns yellow. FFA and MSBDN have achieved higher scores on both PSNR and SSIM, but their performance degrades rapidly on real data in Figure 6. Except for the PSD and our results, the other results are a bit darker. Compared with the ground truth, the result of PSD is very bright because it uses the method of image enhancement and bright channel prior simultaneously to increase the illuminance. Our proposed method is very pleasing in subjective visual perception.

![Figure 5](image)

**Figure 5.** The proposed method compares the results of defogging on SOTS images subjectively with those of different methods. GT stands for ground truth. Some areas are emphasized with red. Yellow rectangles are zoomed-in areas and are suggested for a better visualization and comparison.

Figures 6–9 present the subjective experimental results of the proposed method and other methods for Fattal’s data and the RTTS dataset, respectively. All supervised haze removal methods, including FFA and MSBDN, perform well on synthetic data but they are not satisfactory on real data. The supervised dehazing results have haze residue and are even ineffective, such as FFA and MSBDN failing with real data dehazing, which again proves a domain gap between the synthesized and the real image. The results of ZID are more dim than raw input. In the domain adaptation methods, the SSL dehazing result appears dark for foreground and background. The results of DAD show color variations in content and chaos at the edges. The result of PSD is too bright, and the saturation is too high, making it uncomfortable to look at for a long time. Thus, the PSD method did not find a domain invariant space combined with results in Figures 5–8. Combined with Figure 8, the dehazing results of DAD are more in line with subjective perception than SSL and PSD on RTTS. Combined with the pixel statistics graph in Figure 6, DAD and our dehazing results are similar to pixel statistics, but our results are smoother. The SSL and DAD in Figure 8 are too dim, and results of SSL show more residual haze. The PSD shows the same problem in Figure 6. Figures 7 and 9 present more dehazing results including Fattal and RTTS, respectively. Our results are consistent with human subjective perception. Our
results do not have the above problems, and our results generally achieve a nice dehazing effect, which shows that the proposed method has found a better domain invariant space.

![Figure 6](image)

**Figure 6.** Our method is compared with the dehazing results and corresponding histogram graphs of other methods on Fattal’s images. The proposed method has moderate saturation and no confusion in colors. Red rectangles are used for a better visualization and comparison.

![Figure 7](image)

**Figure 7.** The proposed method results on Fattal’s dataset. It can be seen from the results that our method removes the haze without oversaturation and color confusion, and the contrast and illumination of dehazing results look comfortable.

Figure 10 shows the atmospheric light and contrast changes before and after dehazing. We randomly selected 100 images from RTTS to calculate the average contrast and atmospheric light changes before and after dehazing, verifying that our method can effectively avoid the darker conditions. In addition, we use four well-known no-reference image quality assessment indexes: NIQE [44], BRISQUE [45], BlurMetric [46], and NIMA [47]. All these metrics are evaluated on RTTS, and the results are listed in Table 1.
Figure 8. The proposed method is compared with the dehazing results of semi-supervised and domain adaptation methods on RTTS images. The proposed method has moderate saturation and brightness, and does not produce color abnormality. For better comparison, red rectangles are enlarged views.

Figure 9. The proposed method results on RTTS dataset. It can be seen that our method removes the haze without oversaturation and color confusion, and the dehazing results are comfortable.
Figure 10. The image contrast and atmospheric light change before and after image dehazing; it can be seen that the proposed method enhances the contrast and brightness of the image and avoids the darker dehazing result.

Table 1. Quantitative results of different methods are obtained on RTTS using NR-IQA metrics. Red indicates the best and blue represents the second best.

| Methods  | NIQE ↓ | BRISQUE ↓ | BlurMetric ↓ | NIMA ↑ |
|----------|--------|-----------|--------------|-------|
| Hazy     | 3.583  | 37.011    | 0.3138       | 4.3250|
| SSLD     | 3.489  | 32.428    | 0.2964       | 4.2132|
| DAD      | 3.672  | 32.727    | 0.3713       | 4.0055|
| MSBDN    | 3.154  | 28.743    | 0.3060       | 4.1401|
| PSD      | 3.202  | 25.239    | 0.2989       | 4.3459|
| Ours     | 3.414  | 17.911    | 0.2896       | 4.6448|

4.4. Ablation Experiment

In order to better comprehend the function of every module of our method, we progressively increase the proposed NP and HE to the backbone network and compare the SSIM and PSNR after adding modules. Specifically, we performed an ablation experiment involving four control groups, the backbone network without NP and HE, the baseline model with NP, the baseline model with HE, and the proposed model. We train the four models with SOTS and URHI datasets and then pick out some comparison results randomly. Figure 11 lists the experimental results on SOTS, Fattal’s data, and RTTS. We can see that the backbone model has little effect on the natural data, and even black holes appear. The model trained with HE results in residual fog at the edge, but it looks bright. The model trained only with NP can remove the haze, but the result is darker than the hazy input. Our proposed method combines the advantages of HE and the NP and achieves better results, which demonstrate the dehazing effect of the NP and the guiding function of the HE. Table 2 shows the PSNR and SSIM computed on RESIDE and tested on SOTS. Again, our model achieves good performance. From Table 2, the designed method achieves the best performance.

Table 2. Objective evaluation after elimination of different modules. We can see that the designed method obtains the best performance.

| Models            | PSNR  | SSIM  |
|-------------------|-------|-------|
| Backbone          | 21.11 | 0.9020|
| Backbone + NP     | 26.49 | 0.9237|
| Backbone + HE     | 26.27 | 0.9171|
| Backbone + NP + HE| 27.18 | 0.9409|
Figure 11. Gradually increasing different modules on backbone, and output of related results. The dehazing results comparison of backbone with different modules. Red boxes are highlighted for a better visualization comparison. The best subjective perception could be found in our proposed method.

Figure 12 presents the dehazing results obtained directly using HE. The dehazing images obtained directly using HE suffer from color shift and color confusion, and its results keep illumination and contrast. However, there is no color clutter with our method.

4.5. NP and GP

Noticeably, the NP has the same function as GP [18]. Therefore, we could use GP to map the features of the real data to the synthetic feature space as in [16]. However, as shown in Figure 13, the dehazing results of GP have serious problems; some areas in its results turn white and lose information, which is intolerable and not present in our result. In addition, the time complexity of GP is higher ($O(N^3)$); we train the GP and our model for 30 epochs, the time spent is shown in Table 3. Comparing the training time of using NP and GP, the time complexity of NP is much lower than that of GP.
**Figure 12.** Dehazing results using the HE directly without network. We can see color shift and color confusion in HE results, while our results do not have these problems. Red rectangles are highlighted for a better visualization and comparison.

**Figure 13.** The GP, NP, and our full model dehazing results on RTTS dataset. The dehazing results of GP have serious problems, some areas in its results turn white and lose information, which is not present in our results.

**Table 3.** NP and GP run time comparison. The NP and GP are function space methods, NP could clearly reduce run time.

| Regression Methods | Time (Seconds) |
|--------------------|----------------|
| GP                 | 359,916.9      |
| NP                 | 67,912.5       |
5. Conclusions

Aiming at the problem that the dehazing effect of synthetic data defogging training models is unsatisfactory on real data, we show a semi-supervised neural process dehazing network with asymmetry pseudo labels. This method starts with the backbone network pre-trained on artificial data and uses natural images to retrain the network with designed losses. We assume unlabeled features could be represented as a weighted combination of the labeled features in latent space. These weights express the randomness of the labeled data points being employed to represent the unlabeled data point. The NP can map the hidden feature value of real data to the hidden space of synthetic data and generate the first pseudo latent feature label. Reducing the distance between the real data predicted value and the pseudo value could minimize the difference between the two domains in the function space. Dim dehazing results may occur at the output end, so we adopt the HE to generate the second pseudo label and propose the HE loss, enhancing the illumination and contrast. Numerous experiments have proved that our proposed method achieves good generalization performance in real-world dehazing.

Author Contributions: Conceptualization, F.Z. and X.M.; methodology, F.Z. and X.M.; software, X.M.; validation, X.M., Y.F. and F.Z.; formal analysis, Z.S.; investigation, X.M.; resources, X.M.; data curation, X.M. and Y.F.; writing—original draft preparation, F.Z. and X.M.; writing—review and editing, Z.S.; visualization, X.M.; supervision, Z.S.; project administration, F.Z.; funding acquisition, F.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Shenzhen Science and Technology Program (No. JCYJ20200109142612234), the Guangdong Basic and Applied Basic Research Foundation (No. 2021A1515012313), and the Key-Area Research and Development Program of Guangdong Province (No. 2020B111350003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within this article.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Chen, Z.; Wang, Y.; Yang, Y.; Liu, D. PSD: Principled synthetic-to-real dehazing guided by physical priors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7180–7189.
2. Hassan, H.; Mishra, P.; Ahmad, M.; Bashir, A.K.; Huang, B.; Luo, B. Effects of haze and dehazing on deep learning-based vision models. Appl. Intell. 2022, 1–19. [CrossRef]
3. Zhao, W.; Zhao, Y.; Feng, L.; Tang, J. Attention Optimized Deep Generative Adversarial Network for Removing Uneven Dense Haze. Symmetry 2022, 14, 1. [CrossRef]
4. Zhu, H.; Peng, X.; Chandrasekhar, V.; Li, L.; Lim, J.H. DehazeGAN: When image dehazing meets differential programming. In Proceedings of the International Joint Conferences on Artificial Intelligence Organization, Stockholm, Sweden, 13–19 July 2018; pp. 1234–1240.
5. Narasimhan, S.G.; Nayar, S.K. Vision and the atmosphere. Int. J. Comput. Vis. 2002, 48, 233–254. [CrossRef]
6. He, K.; Sun, J.; Tang, X. Single image haze removal using dark channel prior. IEEE Trans. Pattern Anal. Mach. Intell. 2010, 33, 2341–2353.
7. Fattal, R. Dehazing using color-lines. ACM Trans. Graph. 2014, 34, 1–14. [CrossRef]
8. Berman, D.; Treibitz, T.; Avidan, S. Non-local image dehazing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1674–1682.
9. Cai, B.; Xu, X.; Jia, K.; Qin, C.; Tao, D. Dehazenet: An end-to-end system for single image haze removal. IEEE Trans. Image Process. 2016, 25, 5187–5198. [CrossRef]
10. Li, B.; Peng, X.; Wang, Z.; Xu, J.; Feng, D. AOD-net: All-in-one dehazing network. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4770–4778.
11. Liu, X.; Ma, Y.; Shi, Z.; Chen, J. Griddehazenet: Attention-based multi-scale network for image dehazing. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 7314–7323.
12. Qin, X.; Wang, Z.; Bai, Y.; Xie, X.; Jia, H. FFA-net: Feature fusion attention network for single image dehazing. In Proceedings of the AAAI Conference on Artificial Intelligence, 2020, New York, NY, USA, 2–12 February 2020; pp. 11908–11915.
13. Zhang, H.; Vishal, M.P. Densely connected pyramid dehazing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3194–3203.

14. Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor segmentation and support inference from RGBD images. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Volume 7576, pp. 746–760.

15. Li, B.; Ren, W.; Fu, D.; Tao, D.; Feng, D. Benchmarking single-image dehazing and beyond. IEEE Trans. Image Process. 2018, 28, 492–505. [CrossRef]

16. Yasarla, R.; Sindagi, V.A.; Patel, V.M. Syn2Real transfer learning for image deraining using Gaussian processes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 2726–2736.

17. Li, L.; Dong, Y.; Ren, W.; Pan, J.; Gao, C.; Sang, N.; Yang, M.H. Semi-supervised image dehazing. IEEE Trans. Image Process. 2019, 29, 2766–2779. [CrossRef]

18. Rasmussen, C.E.; Nickisch, H. Gaussian processes for machine learning (GPML) toolbox. J. Mach. Learn. Res. 2010, 11, 3011–3015.

19. Liu, P.J.; Horng, S.J.; Lin, J.S.; Li, T. Contrast in haze removal: Configurable contrast enhancement model based on dark channel prior. IEEE Trans. Image Process. 2018, 28, 2212–2227. [CrossRef] [PubMed]

20. Liu, T.M.; Kim, W. Single image dehazing using color ellipsoid prior. IEEE Trans. Image Process. 2017, 27, 999–1009. [CrossRef] [PubMed]

21. Bui, T.M.; Kim, W. Single image dehazing using color ellipsoid prior. IEEE Trans. Image Process. 2017, 27, 999–1009. [CrossRef] [PubMed]

22. Pan, J.; Dong, J.; Liu, Y.; Zhang, J.; Ren, J.; Tai, Y.W.; Yang, M.H. Physics-based generative adversarial models for image restoration and beyond. IEEE Trans. Pattern Anal. Mach. Intell. 2020, 43, 2449–2462. [CrossRef]

23. Ren, W.; Liu, S.; Zhang, H.; Pan, J.; Cao, X.; Yang, M.H. Single image dehazing via multi-scale convolutional neural networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 154–169.

24. Dong, H.; Pan, J.; Xiang, L.; Hu, Z.; Zhang, X. Multi-scale boosted dehazing network with dense feature fusion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 2157–2167.

25. Li, B.; Gou, Y.; Liu, J.Z.; Zhu, H.; Zhou, J.T.; Peng, X. Zero-shot image dehazing. IEEE Trans. Image Process. 2020, 29, 8457–8466. [CrossRef] [PubMed]

26. Golts, A.; Freedman, D.; Elad, M. Unsupervised single image dehazing using dark channel prior loss. IEEE Trans. Image Process. 2019, 28, 2692–2701. [CrossRef]

27. Pan, J.; Dong, J.; Liu, Y.; Zhang, J.; Ren, J.; Tang, J.; Tai, Y.W.; Yang, M.H. Physics-based generative adversarial models for image restoration and beyond. IEEE Trans. Pattern Anal. Mach. Intell. 2020, 43, 2449–2462. [CrossRef]

28. Pan, J.; Dong, J.; Liu, Y.; Zhang, J.; Ren, J.; Tang, J.; Tai, Y.W.; Yang, M.H. Physics-based generative adversarial models for image restoration and beyond. IEEE Trans. Pattern Anal. Mach. Intell. 2020, 43, 2449–2462. [CrossRef]

29. Golts, A.; Freedman, D.; Elad, M. Unsupervised single image dehazing using dark channel prior loss. IEEE Trans. Image Process. 2019, 28, 3211–3221. [CrossRef]

30. Pang, Y.; Nie, J.; Xie, Y.; Han, J.; Li, X. Visual haze removal by a unified generative adversarial network. IEEE Trans. Circuits Syst. Video Technol. 2018, 29, 3211–3221. [CrossRef]

31. Pang, Y.; Nie, J.; Xie, Y.; Han, J.; Li, X. BidNet: Binocular image dehazing without explicit disparity estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 5931–5940.

32. Wu, H.; Qu, Y.; Lin, S.; Zhou, J.; Qiao, R.; Zhang, Z.; Xie, Y.; Ma, L. Contrastive learning for compact single image dehazing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–15 June 2021; pp. 10551–10560.

33. Zhang, H.; Vishal, M.P. Densely connected pyramid dehazing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 8160–8168.

34. Qu, Y.; Chen, Y.; Huang, J.; Xie, Y. Enhanced pix2pix dehazing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 8160–8168.

35. Li, L.; Dong, Y.; Ren, W.; Pan, J.; Gao, C.; Sang, N.; Yang, M.H. Semi-supervised image dehazing. IEEE Trans. Image Process. 2019, 28, 3104–3118. [CrossRef]

36. Pang, Y.; Meng, D.; Zhao, Q.; Xu, Z.; Wu, Y. Semi-supervised transfer learning for image rain removal. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3877–3886.

37. Zhu, Q.; Rosenbaum, D.; Maddison, C.; Ramalho, T.; Saxton, D.; Shanahan, M.; Teh, Y.W.; Rezende, D.; Eslami, S.A. Conditional neural processes. In Proceedings of the International Conference on Machine Learning PMLR, Stockholm Sweden, 10–15 July 2018; pp. 1704–1713.
41. Garnelo, M.; Schwarz, J.; Rosenbaum, D.; Viola, F.; Rezende, D.J.; Eslami, S.; Teh, Y.W. Neural processes. *arXiv* 2018, arXiv:1807.01622.

42. Blei, D.M.; Kucukelbir, A.; McAuliffe, J.D. Variational Inference: A Review for Statisticians. *J. Am. Stat. Assoc.* 2017, 112, 859–877. [CrossRef]

43. Yang, D.; Sun, J. Proximal dehaze-net: A prior learning-based deep network for single image dehazing. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 702–717.

44. Mittal, A.; Soundararajan, R.; Bovik, A.C. Making a completely blind image quality analyzer. *IEEE Signal Process. Lett.* 2012, 20, 209–212. [CrossRef]

45. Mittal, A.; Moorthy, A.K.; Bovik, A.C. No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.* 2012, 21, 4695–4708. [CrossRef]

46. Crete, F.; Dolmire, T.; Ladret, P.; Nicolas, M. The blur effect: Perception and estimation with a new no-reference perceptual blur metric. In Proceedings of the Human Vision and Electronic Imaging XII SPIE, San Jose, CA, USA, 29 January–1 February 2007; Volume 6492, pp. 196–206.

47. Talebi, H.; Milanfar, P. NIMA: Neural image assessment. *IEEE Trans. Image Process.* 2018, 27, 3998–4011. [CrossRef] [PubMed]