Self-Supervised Denoising Transformer with Gaussian Process

Rajeev Yasarla  Jeya Maria Jose Valanarasu  Vishwanath S  Vishal M. Patel*
Johns Hopkins University
Department of Electrical and Computer Engineering, Baltimore, MD 21218, USA
{ryasar11, jvalana1, vpatel36}@jhu.edu

Abstract

Convolutional neural network (CNN) based methods have been the main focus of recent developments for image denoising. However, these methods lack majorly in two ways: 1) They require a large amount of labeled data to perform well. 2) They do not have a good global understanding due to convolutional inductive biases. Recent emergence of Transformers and self-supervised learning methods have focused on tackling these issues. In this work, we address both these issues for image denoising and propose a new method: Self-Supervised denoising Transformer (SST-GP) with Gaussian Process. Our novelties are two fold: First, we propose a new way of doing self-supervision by incorporating Gaussian Processes (GP). Given a noisy image, we generate multiple noisy down-sampled images with random cyclic shifts. Using GP, we formulate a joint Gaussian distribution between these down-sampled images and learn the relation between their corresponding denoising function mappings to predict the pseudo-Ground truth (pseudo-GT) for each of the down-sampled images. This enables the network to learn noise present in the down-sampled images and achieve better denoising performance by using the joint relationship between down-sampled images with help of GP. Second, we propose a new transformer architecture - Denoising Transformer (Den-T) which is tailor-made for denoising application. Den-T has two transformer encoder branches - one which focuses on extracting fine context details and another to extract coarse context details. This helps Den-T to attend to both local and global information to effectively denoise the image. Finally, we train Den-T using the proposed self-supervised strategy using GP and achieve a better performance over recent unsupervised/self-supervised denoising approaches when validated on various denoising datasets like Kodak, BSD, Set-14 and SIDD.

1. Introduction

Noise adversely affects the visual quality of images captured by camera sensor and thus has a detrimental impact on the performance of downstream computer vision tasks like classification, detection and segmentation. Hence, image denoising is an important pre-processing task in many computer vision applications. Denoising is classically formulated as follows: Given a noisy image $y$, which is a corrupted version of the clean image $x$ with known or unknown noise distribution $n$, the goal of denoising is to recover the clean image $x$ from $y$.

Denoising has been extensively studied in the literature because of its importance in several applications. Some of the early methods like BM3D [11], WNNM [16], etc. do not require clean ground-truth images. These traditional approaches are computationally efficient, do not involve any learning, and are based on natural image priors. However, they require knowledge of the noise levels making it difficult to use them in the wild. Emergence of CNNs in addressing image denoising significantly improved the quality of restored images. Many CNN methods like, RED30 [33], U-Net [39], DnCNN [55], MemNet [42], N3Net [36], and NLRN [28] address image denoising in a supervised fash-

*This work was supported by NSF CAREER award 2045489.
ion. Since these are data driven approaches, they need large amounts of paired noisy-clean images to train the network.

The In-camera Signal Processing (ISP) pipeline in modern sensors are complicated which makes the noise in the real-world difficult to model. This makes it really hard and expensive to obtain labeled pairs of noisy and corresponding ground-truth images which are essential for supervised learning based methods. Hence, most of the existing fully-supervised approaches [18, 28, 36, 55] synthetically generate the noisy images and train their network on these synthetic data pairs. However, as discussed in [20, 21], when these fully-supervised methods are tested on real-world noisy images, they tend to perform poorly because of the domain gap between synthetic and real world noise.

To overcome this problem, especially in cases where we do not have access to real-world ground-truth, Lehtinen et al. [25] applied statistical reasoning in signal reconstruction to CNNs to perform denoising. They demonstrate that it is possible to learn to restore images by using only the corrupted examples. However, they require multiple independent noisy observation of a scene to train the network. This requirement is not practical, since capturing multiple observations of the same scenes is quite challenging when there are movements in the scene. Subsequently, approaches like [21, 23, 48] were developed using a blind-spot network (BSN) structures for learning a self-supervised model. Additionally, [23, 48] employed Gaussian-Poisson noise models to further improve the performance. The main limitation for these methods is that BSN is computationally expensive and suffers from relatively low accuracy. Moran et al. [35] proposed a method that uses noiser-noisy pairs to train the network, where they assume the prior information about the noise model to obtain the denoised image. These self-supervised methods assume prior information about the noise model and although they perform well on synthetic noise, they tend to underperform on real-world noisy images. Recently, Huang et al. [20] proposed Neighbor2Neighbor, where they down-sample the noisy image into pairs to train the network. An additional regularizer is used in the loss function to account for the differences in the ground-truth of down-sampled images, and might not exploit the joint relationship between the down-sampled images. On the other hand, traditional approaches like SS-GMM [29] proposed a parametric approach to generate image prior using Gaussian mixture model (GMM) that models the relationship between patches to estimate noise characteristics like variance.

To this end, we propose a novel self-supervised technique based on Gaussian Process (GP) (note GP is a non-parametric approach). In our proposed method, we first obtain down-sampled images from the noisy image. Then we perform random cyclical shift to these down-sampled images in order to increase the number of down-sampled images. Random cyclical shifts [10] are found to minimize artifacts in denoised images helping us to generate better quality pseudo-GTs. Further, based on the consideration that these down-sampled images have the same noise characteristics and image properties [20], we propose a pseudo-GT generation approach using a Gaussian processes (GP) to model a learnable joint distribution of the down-sampled images. Note unlike, GMM based approaches GP is non-parametric based approach that can formulate joint distribution between infinitely many random variables. Specifically, we formulate a joint Gaussian distribution between down-sampled images that learns joint relation of the denoising function mappings of the down-sampled images to generate pseudo-GT for every down-sampled image. In other words, the learnable joint distribution between down-sampled images using GP, tries to model similar properties among down-sampled images, and also accounts for the difference between down-sampled images by learning co-variance relation between the down-sampled images. Additionally, by predicting pseudo-GT for given down-sample image using other down-sampled images and their corresponding denoised clean images, GP is modelling the joint relation between the denoise function mappings of down-sampled images to learn noise properties in the noisy image. Hence, supervising the network weights using the pseudo-GT obtained by GP, helps the network to learn the joint relation between the down-sampled images and leverage the noise characteristic information from the other down-sampled images. In this way, network is trained in a self-supervised way using GP to exploit the real noise distribution, and achieve a better denoising performance.

Transformers are currently being widely adopted for various computer vision tasks [13, 17, 31, 44, 52, 58]. The major improvements of transformers come from the lack of using convolutions thus not inducing any convolutional inductive biases [38]. This enables transformers to have a global understanding of the input. Recently, transformers have also been used for many low-level vision tasks [6, 27, 46, 57]. In this work, we propose a new transformer architecture-Denoising Transformer (Den-T) tailor-made for denoising application. We note that for denoising we need a global understanding as well as attention to fine details to get the best prediction. To this end, we propose having two branches in the transformer encoder: one focusing to extract fine-context information and another to extract coarse-context information. The coarse context branch is built in a fine-to-coarse way where the feature maps are taken to a lower spatial resolution in the latent space. The fine context branch is built in a coarse-to-fine way where the feature maps are taken to a higher spatial resolution in the latent space. From our experiments, we find that this design helps in improving the denoising performance. More details on why this design works can be found in Sec 3. We train Den-T using the
proposed self-supervised technique using GP and run experiments on multiple denoising datasets like Kodak, BSD, Set-14 and SIDD where we achieve better performance than previous unsupervised/self-supervised denoising methods. Figure 1 demonstrates that with the help of multiple downsampled images and the joint distribution modeling, the proposed method is able to produce clearer and sharper outputs as compared to [20, 25].

The key contributions of this paper are as follows:

• We propose a new self-supervised image denoising approach by modelling the joint distribution between downsampled images using Gaussian processes. This helps the network to explicitly model the real noise distribution and achieve a better denoising performance.

• We propose Denoising Transformer (Den-T), a dual-branch transformer based denoising network which extracts both coarse and fine details to perform denoising.

• We demonstrate the superiority of our proposed method by conducting experiments on multiple synthetic denoising datasets generated using Kodak, BSD, Set-14, and real-world denoising dataset SIDD.

2. Related work

2.1. Supervised Denoising

Compared to the traditional approaches [7, 11, 16, 40], CNN-based methods [5, 8, 28, 33, 36, 55] have achieved superior performance for image denoising. Zhang et al. [55] was among the first CNN-based approach and they employed a residual learning mechanism for effective denoising. Later, methods like [2, 15, 18, 24, 42, 56] were proposed that introduced either efficient training or novel architectural modifications. These approaches follow a fully-supervised paradigm and require large amounts of paired noisy-clean images to train the network. However, it is extremely challenging and expensive to collect real-world paired noisy-clean images. This limits the use of supervised methods on real images with unknown noise models.

2.2. Unsupervised and Self-supervised Denoising

Over the past years, image denoising algorithms like NLM [4], BM3D [11], and WNNM [16] have been proposed which make use of local or non-local structures of the images. However, these methods require knowledge of the noise levels. Soltanayev et al. [41] proposed a image denoising method for AWGN noise models using Steins unbiased risk estimator (SURE) based method on noisy images. Zhussip et al. [59] extended SURE further by training the network using correlated pairs of noisy images.

Lehtinen et al. [25] proposed a self-supervised solution which avoids paired noisy-clean data, and instead uses paired noisy-noisy images of the same scene to train the network. Thereafter, in the self-supervised image denoising, Noise2Void (N2V) [21], Noise2Self [3], Noise2Same [30], Self2Self [37] and Noisier2Noise [35] are proposed that uses only one noisy image per scene to train the network. Methods like Probabilistic N2V [22], Laine et al. [23], and MWCNN [48] propose an elegant way of modeling noise and probabilistic inference to further improve the denoising performance. Noise-as-clean (NAC) [51] addressed the image denoising task by focusing on the cases where noise is weak. Huang et al. [20] down-sampled the noisy image into neighboring pairs of down-sampled images, and used them to train the network, where the proposed loss accounts for the difference in the ground-truth of the neighboring downsampled images.

2.3. Transformers for low-level vision


3. Preliminaries

Problem setting. Given a set of only noisy images \( D = \{ y_i \}_{i=1}^M \), our objective is to train Den-T \( f_{\theta}(\cdot) \) and learn the network weights \( \theta \) to perform image denoising. We follow Huang et al. [20] where only noisy images are used to train the network in a self-supervised fashion. Given a noisy image \( y \in \mathcal{D} \), we generate down-sampled images with cell-size \( 2 \times 2 \) (for more details about down-sampling please refer [20]) and randomly shift them to obtain more downsampled images for \( y \). Finally, using the proposed method we compute pseudo-GT’s for these down-sampled images, and use them for training the network.

Motivation for Self-supervision with GP: Just minimizing L2-Norm between noisy image pairs (in case of N2N [25]) or minimizing L2-Norm between down-sampled images with additional regularizer (in case of Neighbor2Neighbor [20]) might not be beneficial for network in learning the noise model. The additional regularizer [20] accounts for the difference in the ground-truth of downsampled images but doesn’t help the network learn the relationship between the down-sampled images or the noise model. In contrast to [20], we believe that learning joint relation between the down-sampled images is beneficial for a self-supervised method to achieve better performance, since the joint relationship between the down-sampled images leverages the noise information present in the downsampled images. In other words, formulating joint relationship between the denoising function \( f(\cdot) \) mappings of down-sampled images using GP, we can learn the noise...
Figure 2. **Overview of the proposed method SST-GP.** Given a noisy image $y$, we generate down-sampled images $\{y^d_i\}_{i=1}^N (= Y)$, and pass them through Den-T to obtain $Z$ and $X$. Later, we model joint distribution between down-sampled images $(Y)$ using GP to compute pseudo-GTs for each of the down-sampled image $y^d_i$. We then train SST-GP using the proposed loss $L_{GP}$ and $L_M$. $L_2$ represents L2-norm. Down-Sampler represents the down-sampling technique used in [20]. Blue arrow denotes the path network denoised image prediction ($\hat{x}_{i,pseudo}^d$), and grey arrow denotes the path for pseudo-GT ($\hat{x}_{i,pseudo}^d$) prediction using Gaussian process.

information present in denoised images. To this end, we propose a self-supervised technique based on Gaussian process (GP) to learn pseudo-GT for each down-sampled image while not requiring any paired noisy or clean images to update the network weights.

Let $y$ and $s$ be two independent noisy images conditioned on $x$, such that $E_{z|x}(y) = x$ and $E_{z|x}(s) = x + \varepsilon$ where $\varepsilon \neq 0$ and small. Thus, $y = x + n_1$, $s = x + \varepsilon + n_2$, where $n_1$ and $n_2$ are additive zero mean noises with variance $\sigma_y^2$ and $\sigma_s^2$. If we approximate $\varepsilon$ with a Gaussian distribution, i.e. $\varepsilon \sim \mathcal{N}(0, \sigma_s^2)$. Let $\hat{n}_2 = n_2 + \varepsilon$, then,

\[
P(n_1, \hat{n}_2) = \mathcal{N}(0, \Sigma^2), \quad \Sigma^2 = \begin{bmatrix} \sigma_s^2 & 0 \\ 0 & \sigma_y^2 + \sigma_s^2 \end{bmatrix}
\]

\[
P(y - x - n_1, s - x - \hat{n}_2) = \mathcal{N}(0, \Sigma^2) \quad \Rightarrow \quad P(y - f(y), s - f(s)) = \mathcal{N}(0, \Sigma^2)
\]

Since in optimal (ideal denoise network) case $x \approx f(y)$, where $f(\cdot)$ represents the function for denoising network. This allows us to formulate learnable joint distribution between function mappings ($f(\cdot)$) of $y$ and $s$ with help of GP where in learning this relation between $f(y)$ and $f(s)$, GP learns the noise information present in $y$ and $s$. By conditioning this joint distribution between $f(y)$ and $f(s)$ (in Eq. 1) with $f(s)$ we can predict the denoised image for $y$ as $\mu_y$. We can define $\mu_y$ in Eq. 1 as pseudo-GT for $y$ and learn the networks weights $\theta$ by minimizing the negative log-likelihood of the conditional distribution as follows,

\[
L_{GP} = -\log P \left( \mu_y - f(y) \mid s, f(s) \right)
\]  

(2)

In this way, we can learn the joint relation in $y$ and $s$ using GP with help of learnable kernel functions which is beneficial in modelling the similar properties $y$ and $s$ and account also for differences between them. Updating the network weights using $L_{GP}$ using $\mu_y$ helps the network to leverage noise present in $s$. We can extend this to multiple noisy observations $\{y_i\}$ (where $E_{zi|x}(y_i) = x + \varepsilon_i$, and $\varepsilon_i$’s are small), and formulate joint Gaussian distribution using GP to leverage noise information in $\{y_i\}$’s and update the network weights using following optimization:

\[
P (f(y_i) \mid \{y_j\}_{j\neq i}, \{f(y_j)\}_{j\neq i}) = \mathcal{N} (\mu_{y_i}, \Sigma_i^2)
\]

\[
L_{GP} = -\log P (\mu_{y_i} - f(y_i) \mid \{y_j\}_{j\neq i}, \{f(y_j)\}_{j\neq i})
\]

(3)

4. Proposed Method

Given a noisy image $y$, following Huang et al. [20] we obtain neighboring down-sampled images. Then we perform cyclical random shifts to these down-sampled images in order to obtain more down-sampled images for $y$. Note that [10] explained that random cyclical shifts minimizes the artifacts and aliasing effects introduced during down-sampling. Thus, for noisy image $y$, we obtain a set of $N$ down-sampled cyclically-shifted images, $\{y^d_1, y^d_2, \ldots, y^d_N\}$. Next, we forward these down-sampled images, $\{y^d_1, y^d_2, \ldots, y^d_N\}$ through the denoising network and inverse-shift them to obtain the corresponding denoised down-sampled images, $\{\hat{x}_1^d, \hat{x}_2^d, \ldots, \hat{x}_N^d\}$. Figure 2 gives an overview of the proposed method where each down-sampled image $y^d_i$ is passed through the encoder to obtain intermediate vector $z_i^d = g(y_i^d, \theta_g)$. The vector $z_i^d$ is then forwarded to a decoder followed by a inverse-cyclical shift to obtain the corresponding denoised down-sampled image, i.e $\hat{x}_i^d = Inv (h(z_i^d, \theta_h))$. Here, $Inv(\cdot)$ represents inverse-cyclical shift function. SST-GP is trained with two losses: (i) $L_M$, (minimizing the L2-norm between down-sampled images, and (ii) $L_{GP}$. The latter loss is constructed based on pseudo-GT predicted by the joint distribution modeled with $\{\hat{x}_1^d, \hat{x}_2^d, \ldots, \hat{x}_N^d\}$ using Gaussian processes. First, we explain the details of our transformer network Den-T and then explain how we train it using our proposed GP based self-supervised approach.

4.1. Denoising Transformer (Den-T)

We use a dual branch transformer based encoder and a convolutional decoder for Den-T. The two branches of our encoder are: 1) Fine Context Transformer Branch (FTB) and 2) Coarse Context Transformer Branch (CTB).
is the input. Similar to the original self-attention network block, follows: positional information for transformers as shown in [49].

- **Coarse Context Transformer Branch:** To extract fine-detailed information from the input image, CNN-based methods like [45, 53] project the features to a high spatial resolution. Inspired by these works, we apply the same process on the self-attention features to extract fine-details. We use three transformer blocks in this branch with upsampling in between every transformer block. Performing self attention in a high spatial resolution latent space helps in attending to smaller information as the feature space. Upsampling here is done using bilinear interpolation.

- **Transformer Block:** Each transformer block is equipped with multi-head self-attention layers and feed forward networks, i.e.,

\[
\text{Attn}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d}} \right) V,
\]

where \(d\) represents the dimensionality. We use multiple attention heads in each transformer block and that number is a hyper-parameter which we vary across each stage in the transformer encoder. More details regarding the hyper-parameter settings can be found in the supplementary document. The self-attention features are then passed to a FFN block. In the FFN block, we use depth-wise convolution to MLP inspired from [26, 47, 49]. Using depth-wise convolution here helps bring locality information and provides positional information for transformers as shown in [49]. The computation in the FFN block can be summarized as follows:

\[
\text{FFN}(A) = \text{MLP}(\text{GELU}(\text{DW C}(\text{MLP}(A)))) + A,
\]

where \(A\) corresponds to the self-attention features, \(\text{DW C}\) is depth-wise convolution [9], \(\text{GELU}\) is Gaussian error linear units [19], and \(\text{MLP}\) is multi-layer perceptron.

- **Decoder:** We use a convolutional decoder with a series of convolutional and upsampling layers to output the denoised image. An overview of Den-T can be found in Figure 3.

### 4.2. Self-Supervision using GP

As we do not have the corresponding ground-truths for the down-sampled images \(\{y_1^d, y_2^d, \ldots, y_N^d\}\), we use GP to model the noise information between the noisy down-sampled images. Specifically, we use GP to generate the pseudo-GT’s and use them for supervision. The primary intuition behind the pseudo-GT generation is to formulate a joint relation between \(\{y_1^d, y_2^d, \ldots, y_N^d\}\), as they share same image properties and the corresponding input down-sampled images share the same noise distribution. This motivates us to formulate a learnable joint Gaussian distribution between \(\{y_1^d\}_{i=1}^N\), and predict pseudo-GT for every down-sampled image \(y_i^d\) using the denoised images of other down-sampled images \(\{\hat{x}_j^d\}_{j\neq i=1}^N\). In this way, we are learning a covariance relation and also noise present in the down-sampled images \(\{\hat{y}_i^d\}_{i=1}^N\), to train the denoising network in a self-supervised fashion.

- **Pseudo-GT:** Given \(\{y_1^d, y_2^d, \ldots, y_N^d\}\), we forward them through Den-T to obtain the corresponding intermediate vectors \(\{z_1^d, z_2^d, \ldots, z_N^d\}\). These intermediate vectors are then passed through a decoder network and inverse-shifted to obtain the corresponding denoised images \(\{\hat{x}_1^d, \hat{x}_2^d, \ldots, \hat{x}_N^d\}\). The denoising function mappings between \(y_i^d\) and \(\hat{x}_i^d\), i.e., \(\hat{x}_i^d = f(y_i^d)\), \(\forall i = 1, 2, 3, \ldots, N\) can be modelled using GP by formulating a joint Gaussian distribution between these function mappings of down-sampled images. Assuming these function mapping \(f(.)\) form a Gaussian process (GP) which is an infinite collection of functions of which any finite subset of these function mappings form a jointly Gaussian distribution. Then joint Gaussian distribution for function \(f(.)\) mappings of down-sampled images is formulated as follows:

\[
\begin{bmatrix}
 f_1(\cdot)
 f_2(\cdot)
 \vdots
 f_N(\cdot)
\end{bmatrix} \sim \mathcal{GP}(\mu^d, K(Z^d, Z^d) + \sigma^2_Z I),
\]

where, \(\mu^d\) function value obtained using GP, and \(K(\cdot, \cdot)\) is the learnable kernel function that defines the covariance relation among down-sampled images. \(K(\cdot, \cdot)\) is Rational quadratic (RQ.) based kernel function defined as follows,

\[
K(Z^d, Z^d)_{p,q} = \kappa(z_p^d, z_q^d) = \alpha^2 \left( 1 + \frac{\|z_p^d - z_q^d\|^2}{\beta^2} \right)^{-0.5}
\]

Note that \(\alpha\), \(\beta\), and \(\sigma_z\) are learnable parameters which help in learning the covariance relation among the down-sampled images.

Here, \(Z\) is constructed using the intermediate latent vectors, i.e., \(Z = \{z_i^d\}_{i=1}^N\). We use \(Z\) in order to compute
covariance since intermediate latent vectors $z^d_i$'s are more informative than $y^d_i$'s. Let, $Y$ be a set of all down-sampled images generated from $y$, i.e. $Y = \{y^d_i\}_{i=1}^N$, and $X$ be a set of the corresponding function values, i.e $X = \{\hat{x}^d_i\}_{i=1}^N$. We define $Y_c$ as a set of all down-sampled image excluding $y^d_j$, i.e $Y_c = \{y^d_i : i = [1, N]$ and $i \neq j \}$, similarly $\hat{X}_c = \{\hat{x}^d_i : i = [1, N]$ and $i \neq j \}$. Likewise, we define $Z_c$ as a set of all intermediate vectors of the down-sampled images excluding $z^d_j$, i.e $Z_c = \{z^d_i : i = [1, N]$ and $i \neq j \}$. Using the joint distribution in Eq. 6, we can obtain conditional distribution for $f(y^d_j)$ as the following Gaussian distribution given $Y$, $Z$ and $\hat{X}_c$,

$$P(f(y^d_j)|Y, Z, \hat{X}_c) = \mathcal{N}(\mu_j, \Sigma_j),$$

where

$$\mu_j = K (z^d_j, Z_j) [K (Z_j, Z_c) + \sigma^2 I]^{-1} \hat{X}_c,$$

$$\Sigma_j = K (z^d_j, z^d_j) [K (Z_j, Z_c) + \sigma^2 I]^{-1} K (Z_j, z^d_j) + \sigma^2 I.$$  

(9)

We use $\mu_j$ computed using GP in Eq. 9 as pseudo-GT ($\hat{x}^{d,\text{pseudo}}_j$) for the down-sampled image $y^d_j$. For every down-sampled image generated using input image $y$, we compute network’s denoised down-sampled image $\hat{x}^{d,\text{pred}}_i = \text{Inv}(h(g(y^d_i, \theta_c, \theta_d))) = f(y^d_i, \theta)$ and pseudo-GT ($\hat{x}^{d,\text{pseudo}}_i$) computed using GP (here $\text{Inv}(\cdot)$ represents the inverse-cyclical shifting function). Finally, we minimize the L2-error between $\hat{x}^{d,\text{pred}}_i$ and $\hat{x}^{d,\text{pseudo}}_i$ to update the network weights ($\theta$), hence incorporating the modeled joint distribution between down-sampled images that helps learning the noise information to perform image denoising. Further, we gate the L2-error between $\hat{x}^{d,\text{pred}}_i$ and $\hat{x}^{d,\text{pseudo}}_i$ with the inverse of the computed variance $\Sigma_j$ in order to obtain more accurate predictions. This gating ensures that lesser importance is given to the uncertain predictions while learning the network weights. Additionally, we minimize the variance that helps GP model to learn the joint distribution more accurately, and obtain accurate pseudo-GT labels. The proposed GP based loss on the down-sampled images is as follows,

$$\mathcal{L}_{\text{GP}} = -\log P(\mu_j, f(y^d_j) | Y, Z, \hat{X}_c) = \sum_{i=1}^N \frac{1}{2} \left[ \| \hat{x}^{d,\text{pred}}_i - \text{Inv}(g_i) \|_2^2 + \log(2\pi) \right].$$

(10)

L2-norm loss: Motivated by the loss proposed in Noise2Noise [25] and Haug et al. [20], we use the following objective function $\mathcal{L}_M$ to exploit the down-sampled image pairs:

$$\mathcal{L}_M = \frac{1}{M} \sum_{i=1}^N \sum_{j \neq i \neq j} \| (\hat{x}^{d,\text{pred}}_i - \text{Inv}(g_i), \hat{x}^{d,\text{pred}}_j - \text{Inv}(g_j)) \|_2^2$$

(11)

here, $\text{Inv}(\cdot)$ represents inverse-cyclical shift function.

Total loss: The overall loss function used for training the SST-GP is defined as follows,

$$\mathcal{L}_{\text{total}} = \mathcal{L}_M + \lambda_{\text{GP}} \mathcal{L}_{\text{GP}},$$

(12)

where $\lambda_{\text{GP}}$ is a predefined weight that is set equal to 0.03. We provide an ablation study for $\lambda_{\text{GP}}$ in the supplementary document. In our experiments, we use values for $\lambda_{M}$ in the order of $10^{-3}$ and the values of $\mathcal{L}_{GP}$ in the order of $10^{-1}$.

4.3. Implementation details

We train our SST-GP network using $\mathcal{L}_{\text{total}}$ with Den-T as base denoising network. We use Adam optimizer with a learning rate of 0.0002 and batch-size of 4 to train SST-GP for a total of 60 epochs. We decrease the learning rate by a factor of 0.5 for every 25 epochs. During training, the images are randomly cropped to the size of $256 \times 256$. We set $\lambda_{\text{GP}} = 0.03$, cell size $k = 2$ in generating down-sampled images using [20]. We shift each down-sampled cyclical for 4 times, so $N = 8$ for every noisy image $y$. Pseudo algorithm for training the SST-GP are provided in the supplementary document.

5. Experiments and Results

In this section, we provide the results of various experiments conducted to demonstrate the effectiveness of the proposed approach. In addition, we also provide a comparison of the proposed method with existing methods on both synthetic and real-world noisy datasets.

5.1. Dataset details

**Synthetic datasets:** For training SST-GP to perform experiments using synthetic sRGB space, we use 50k clean images from the validation dataset of ImageNet [12]. Crops of $256 \times 256$ are obtained from these 50k clean images and used to generate noisy images by adding the following 4 different noise levels: (i) Gaussian noise with fixed standard deviation $\sigma = 25$, (ii) Gaussian noise with varied noise level, $\sigma = [5, 50]$, (iii) Poisson noise with fixed $\lambda = 30$, and (iv) Poisson noise with $\lambda = [5, 50]$. Note that these $\sigma$, $\lambda$ values correspond to pixel intensities in the range of [0, 255]. Synthetic test sets are created using the clean images from Kodak [14], BSD [34], and Set-14 [54] datasets.

**Real datasets:** Authors of SIDD [1] collected real-world noisy images of 10 static scenes using 5 smart phone cameras in different lighting conditions. The authors grouped the collected images into SIDD Medium Dataset for training, and use SIDD Validation and Benchmark Dataset in RAW formats. Following the same protocol, we use the SIDD Medium training Dataset to train SST-GP, and use the Validation and Benchmark Datasets for evaluation and comparisons.

5.2. Comparisons on synthetic test data

We use PSNR and SSIM to compare SST-GP against the state-of-the-art (SOTA) methods. We train all the networks using ImageNet [12] following the steps mentioned in the respective SOTA methods. We denote Laine19 [23] with probabilistic post-processing as Laine-pme, and without as Laine-mu. Table 1 shows comparisons on synthetic Gaussian noise test sets, where our proposed method significantly outperforms the previous methods. Table 2 shows
along with proposed GP loss in a fully-supervised manner with pairs noisy-clean images. It can be observed that the results of our method are more clearer and sharper compared to the predictions of other methods. In contrast to other methods [20, 23, 25, 39], we used down-sampled images and modelled joint distribution using GP, that helped the proposed SST-GP outperform the other methods by a significant margin and it is able to produce sharper images than the other methods. Note that in Table 3, we also present the oracle performance i.e. when Den-T trained in a fully-supervised manner using GP loss \( \mathcal{L}_{GP} \). In Table 4, we can see that using SST-GP significantly improved the performance of both U-Net and Den-T by \( \sim 0.4 \)dB while trained in a fully-supervised manner. The main reason for this improvement is that proposed pseudo-GT based GP approach learns the relation between the down-sampled images and updates the networks using \( \mathcal{L}_{GP} \). In Table 4, it can be observed that using \( \mathcal{L}_{GP} \) in a fully-supervised way using the pairs of noisy-clean images with same losses (\( L_2 \) and proposed GP based loss \( \mathcal{L}_{GP} \)). In Table 4, we can see that Den-T outperforms U-Net even while trained in a similar fully-supervised fashion with comparably less number of parameters. Additionally in Table 4, we compare computational complexity of Den-T using Giga Multiply Accumulate(GMacs) operations per second.

**Impact of \( \mathcal{L}_{GP} \):** In Table 4, it can be observed that using \( \mathcal{L}_{GP} \) significantly improved the performance of both U-Net and Den-T by \( \sim 0.4 \)dB while trained in a fully-supervised manner with and without FTB and CTB branches to understand the contributions of individual branches. From Table 5, we can observe that using both branches together help us get a better performance. Additionally, we compare the performance of Den-T with existing state-of-the-art transformer based denoising networks like SwinIR [27], and Uformer [46]. In Table 5, we can observe that Den-T outperforms Swin-IR [27], and Uformer [46].

### 5.5. Limitations

Training time of SST-GP with \( \mathcal{L}_{GP} \) is 1.5 times slower when compared to training time of Den-T with \( L_2 - norm \), since \( \mathcal{L}_{GP} \) involves matrix multiplication for computing \( \mu \) and \( \Sigma \) (refer Eq. 9). Table 6 shows that Den-T w/ GP requires higher memory during training, this is due to two reasons: (i) matrix multiplication for computing \( \mu \) and \( \Sigma \) in a fully-supervised manner using the pairs of noisy-clean images with same losses (\( L_2 \) and the proposed GP based loss \( \mathcal{L}_{GP} \)).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Noise Type</th>
<th>( \mu )</th>
<th>( \Sigma )</th>
<th>( \mathcal{L}_{GP} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Den-T w/ GP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Den-T w/o GP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 5.5. Limitations

Training time of SST-GP with \( \mathcal{L}_{GP} \) is 1.5 times slower when compared to training time of Den-T with \( L_2 - norm \), since \( \mathcal{L}_{GP} \) involves matrix multiplication for computing \( \mu \) and \( \Sigma \) (refer Eq. 9). Table 6 shows that Den-T w/ GP requires higher memory during training, this is due to two reasons: (i) matrix multiplication for computing \( \mu \) and \( \Sigma \) in a fully-supervised manner using the pairs of noisy-clean images with same losses (\( L_2 \) and the proposed GP based loss \( \mathcal{L}_{GP} \)).

In Table 4, we can see that Den-T outperforms U-Net even while trained in a similar fully-supervised fashion with comparably less number of parameters. Additionally in Table 4, we compare computational complexity of Den-T using Giga Multiply Accumulate(GMacs) operations per second.

**Impact of \( \mathcal{L}_{GP} \):** In Table 4, it can be observed that using \( \mathcal{L}_{GP} \) significantly improved the performance of both U-Net and Den-T by \( \sim 0.4 \)dB while trained in a fully-supervised manner. The main reason for this improvement is that proposed pseudo-GT based GP approach learns the relation between the down-sampled images and updates the networks using \( \mathcal{L}_{GP} \). Additionally, we compare the performance of Den-T with existing state-of-the-art transformer based denoising networks like SwinIR [27], and Uformer [46]. In Table 5, we can observe that Den-T outperforms Swin-IR [27], and Uformer [46].

### 5.5. Limitations

Training time of SST-GP with \( \mathcal{L}_{GP} \) is 1.5 times slower when compared to training time of Den-T with \( L_2 - norm \), since \( \mathcal{L}_{GP} \) involves matrix multiplication for computing \( \mu \) and \( \Sigma \) (refer Eq. 9). Table 6 shows that Den-T w/ GP requires higher memory during training, this is due to two reasons: (i) matrix multiplication for computing \( \mu \) and \( \Sigma \) in a fully-supervised manner using the pairs of noisy-clean images with same losses (\( L_2 \) and the proposed GP based loss \( \mathcal{L}_{GP} \)).

In Table 4, we can see that Den-T outperforms U-Net even while trained in a similar fully-supervised fashion with comparably less number of parameters. Additionally in Table 4, we compare computational complexity of Den-T using Giga Multiply Accumulate(GMacs) operations per second.

**Impact of \( \mathcal{L}_{GP} \):** In Table 4, it can be observed that using \( \mathcal{L}_{GP} \) significantly improved the performance of both U-Net and Den-T by \( \sim 0.4 \)dB while trained in a fully-supervised manner. The main reason for this improvement is that proposed pseudo-GT based GP approach learns the relation between the down-sampled images and updates the networks using \( \mathcal{L}_{GP} \). Additionally, we compare the performance of Den-T with existing state-of-the-art transformer based denoising networks like SwinIR [27], and Uformer [46]. In Table 5, we can observe that Den-T outperforms Swin-IR [27], and Uformer [46].

### 5.5. Limitations

Training time of SST-GP with \( \mathcal{L}_{GP} \) is 1.5 times slower when compared to training time of Den-T with \( L_2 - norm \), since \( \mathcal{L}_{GP} \) involves matrix multiplication for computing \( \mu \) and \( \Sigma \) (refer Eq. 9). Table 6 shows that Den-T w/ GP requires higher memory during training, this is due to two reasons: (i) matrix multiplication for computing \( \mu \) and \( \Sigma \) in a fully-supervised manner using the pairs of noisy-clean images with same losses (\( L_2 \) and the proposed GP based loss \( \mathcal{L}_{GP} \)).

In Table 4, we can see that Den-T outperforms U-Net even while trained in a similar fully-supervised fashion with comparably less number of parameters. Additionally in Table 4, we compare computational complexity of Den-T using Giga Multiply Accumulate(GMacs) operations per second.

**Impact of \( \mathcal{L}_{GP} \):** In Table 4, it can be observed that using \( \mathcal{L}_{GP} \) significantly improved the performance of both U-Net and Den-T by \( \sim 0.4 \)dB while trained in a fully-supervised manner. The main reason for this improvement is that proposed pseudo-GT based GP approach learns the relation between the down-sampled images and updates the networks using \( \mathcal{L}_{GP} \). Additionally, we compare the performance of Den-T with existing state-of-the-art transformer based denoising networks like SwinIR [27], and Uformer [46]. In Table 5, we can observe that Den-T outperforms Swin-IR [27], and Uformer [46].

### 5.5. Limitations

Training time of SST-GP with \( \mathcal{L}_{GP} \) is 1.5 times slower when compared to training time of Den-T with \( L_2 - norm \), since \( \mathcal{L}_{GP} \) involves matrix multiplication for computing \( \mu \) and \( \Sigma \) (refer Eq. 9). Table 6 shows that Den-T w/ GP requires higher memory during training, this is due to two reasons: (i) matrix multiplication for computing \( \mu \) and \( \Sigma \) in a fully-supervised manner using the pairs of noisy-clean images with same losses (\( L_2 \) and the proposed GP based loss \( \mathcal{L}_{GP} \)).

In Table 4, we can see that Den-T outperforms U-Net even while trained in a similar fully-supervised fashion with comparably less number of parameters. Additionally in Table 4, we compare computational complexity of Den-T using Giga Multiply Accumulate(GMacs) operations per second.
6. Conclusion

In this work, we proposed a new method: Self-Supervised Transformer with Gaussian Process (SST-GP) for image denoising. We proposed a new self-supervised technique where given a noisy image, we generate multiple cyclically shifted noisy down-sampled images and model a joint distribution between them using GP. We also introduced a denoising transformer (Den-T) which is a dual-branch network architecture to extract both coarse and fine details to perform denoising.

Table 5. PSNR/SSIM comparisons for ablation study of Den-T using Kodak testset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Fully-supervised</th>
<th>Self-supervised</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSD</td>
<td>Gaussian σ = 25</td>
<td>30.96/0.878</td>
<td>31.22/0.881</td>
</tr>
<tr>
<td></td>
<td>Poisson σ = 30</td>
<td>30.35/0.868</td>
<td>30.84/0.887</td>
</tr>
<tr>
<td>Parameters (Million)</td>
<td>31</td>
<td>31</td>
<td>24</td>
</tr>
<tr>
<td>GMacs(Million)</td>
<td>35.8</td>
<td>61.6</td>
<td>16.0</td>
</tr>
</tbody>
</table>

Table 6. GMacs comparison for image size $256 \times 256$. 

<table>
<thead>
<tr>
<th>Method</th>
<th>U-Net</th>
<th>U-Net w/GP</th>
<th>Den-T</th>
<th>Den-T w/GP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMacs</td>
<td>9.38</td>
<td>12.75</td>
<td>16.02</td>
<td>20.49</td>
</tr>
</tbody>
</table>
References


deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017. 1, 2, 3


