Research internship report

Image restoration through real noise modeling

Thomas Michel
Research internship supervised by Thibaud Ehret

January 15, 2024

Contents

1 Introduction

Denoising is a critical task in image and video processing. Despite significant advancements in denoising techniques and imaging technology in recent years, particularly with the use of deep learning [Zhang et al., 2017, 2021, Zamir et al., 2022], new challenges continue to arise. For example, high-end cameras may still produce noisy images in dim lighting conditions, while high-speed video cameras may capture frames with low signal-to-noise ratio due to their short exposure times. The widespread use of cheaper, lower-quality sensors in devices such as mobile phones and surveillance cameras requires denoising even in well-lit scenes. These challenges are further exacerbated by the use of low-quality optics in such devices, which is often necessary due to cost and space constraints. Additionally, images are often processed directly on the device in order to conserve memory and

avoid storing large raw data files. However, this direct processing can affect the noise statistics of the output data and create complex noise distribution with spatial correlation, which can make the denoising task more difficult. Furthermore, digital images can also degrade over time, due to lossy compression when the image is uploaded or saved multiple times. This is why it is essential to have techniques for generic image restoration.

If we want for deep learning models to perform real image denoising, it is essential for it to be able to grasp the structure of the image and the noise. The diversity and the complexity of the degradation of real images makes it difficult to efficiently train such methods. Indeed, the collection of clean and noisy real images is hard and restrain the diversity of the data (controlled experiment with few devices), and their generation from synthetic noise is insufficient since the noise distributions are often unknown or complex to model. Multiple works have been done in order to address these issues, ranging from more realistic noise generation to learning noise model without datasets of noisy and clean image pairs .

In this work, we review a state-of-the-art image restoration technique proposed by Wan et al. [2022] whose goal is to restore degraded old photos using only unrelated sets of degraded photos and clean photos. We implement a part of the method that handle the denoising of images and reproduce the results. In addition, we explore some improvements to the model in order to reduce the artifacts and enhance the visual quality of the output images.

2 Previous works

Classical image denoising

Early works on image denoising are based on assumptions on the noise. One common assumption is that we can approximate the noisy image as the result of a clean image degraded with additive white Gaussian noise (AWGN). The properties of the noise are used to design filters that would remove the noise while preserving the image. For example, DCT denoising [Yu and Sapiro, 2011] is based on the assumption that the noise is AWGN. The algorithm compute the coefficients of the image in the DCT basis and noise is removed by applying a threshold on the coefficients. An inverse transformation allows recovering a denoised image. The threshold is chosen to remove the noise while preserving the image.

Other works leverage the self-similar structure of the image by assuming that similar patches are different realization of noises on the same signal. Examples of such techniques are Non-local means [Buades et al., 2005] or BM3D [Dabov et al., 2007] algorithms. Non-local means consists in selecting the patches from the image that are most similar to the patch that must be denoised and averaging the value of each pixel over all the patches. In that case, since the noise distribution is considered centered and independent for each pixel, the average is an estimate of the true value of the signal. BM3D algorithm is based on the same idea of using similar patches and push it further by performing denoising on all these patches at once, as well as using adaptive aggregation techniques to weight the importance of each patches in the final result. This last method already achieves impressive results while being relatively computationally inexpensive, which makes it become widely used.

Another approach is to estimate or learn a probabilistic model associated to the patches and then try to find the most likely original image given the noisy observation and the model. This idea can be applied either by estimating a model locally from patches of the same image, as in Non-local Bayes [Lebrun et al., 2013] or globally by learning a Gaussian mixture model of the patches from a

dataset of images, as it is the case for the EPLL method proposed by Zoran and Weiss [2011].

Neural network for image denoising

The recent success of deep learning in image processing has led to the development of several neural networks based methods for image denoising. The first works use convolutional neural networks to directly learn a mapping between the noisy image and the clean image. The network is trained with pairs of noisy and clean images. The model is then used to denoise images by applying the mapping to the noisy image.

Zhang et al. [2017] introduces DnCNN, a convolutional neural network designed to predict the residual image corresponding to the input (the difference between the noisy input and the clean underlying signal). This approach, named residual learning, extracts the noise which can then be subtracted from the noisy image to recover a cleaner one. DnCNN displays superior results compared the anterior approaches.

These kinds of techniques work well when the noise of the image corresponds to the noise used to train the network, however it does not generalize well to different types of noise. While a broader range of noise can be included in the training dataset, synthesizing realistic noise is not always possible. Self-supervised methods allow bypassing this issue, as they do not require pairs of clean and noisy images. These techniques train a network on tasks other than denoising in order to implicitly learn a model of the unknown noise. Noise-to-noise training [Lehtinen et al., 2018] uses different realizations of the noise for a same scene and asks the network to build a mapping between them. Noise-to-void [Krull et al., 2019] and noise-to-self[Batson and Royer, 2019] lift the constraint of a second noisy image by proposing blind spot networks, which implicitly exploit spatial regularity of the data. However, the performance of these two methods are noticeably lower than the noise-to-noise or noise-to-clean methods due to the loss of information caused by the blind spots at inference time. Recent works [Laine et al., 2019, Krull et al., 2020] have improved these results by incorporating information from the blind spots using Bayesian reasoning. However, these require knowledge about the noise model which is unavailable in the real noise image denoising setting.

Real noise and blind real image denoising

Deep learning methods are capable of representing complex properties of the image and the noise, however contrary to classical single image denoising techniques, they require large datasets of pairs of noisy and clean images. To build such datasets, they often rely on synthetic training data and particularly synthetic noisy images that can be obtained easily by degrading clean images. Indeed, creating datasets with pairs of real noisy and clean images may be quite complicated and time-consuming. Even when it is possible, the noise distribution changes depending on the device and the processing the image goes through. Plotz and Roth [2017] proposes a dataset of images with real noisy and clean images obtained from long exposure. A benchmark using this dataset revealed that deep learning models that were thought to outperform classical methods such that BM3D were in fact weaker on real noise, which shows the importance of realistic training data for neural network based techniques.

An approach to overcome this issue is to generate more realistic noise in order to build datasets. Brooks et al. [2019] proposes to work with more realistic noise by simulating the whole processing pipeline and denoising directly raw images. Indeed, while the noise of processed image can be very complex due to the non-linear operations and the spatially correlated noise introduced along the

process, the noise of raw images is relatively well understood. There are two main sources of noise in raw images: the shot noise (related to photon arrival statistics) is a Poisson random variable whose mean is the true intensity of the light received by the sensor, and the read noise (electronic noise introduced while reading the value from the sensors) which is very close to a centered Gaussian random variable. Such noise can easily be reproduced and added to the clean raw image in order to obtain a more realistic noise after reprocessing of the image.

Another approach is to stop relying on noisy/clean image pairs and try to directly learn insights on the structure of the image and the noise from unpaired clean and noise photos, which are much easier to obtain. There are two main lines of research to tackle the problem of blind real image denoising for real noise. The first category [Abdelhamed et al., 2019, Jang et al., 2021, Yue et al., 2020] proposes to learn to synthesize noise using only example. This noise generator can then be used to generate noisy/clean pairs of images used to train non-blind classical models. The second category [Wan et al., 2022, Guo et al., 2021, Soh and Cho, 2021] proposes to model the latent spaces corresponding to noisy and clean images, and to learn the projection from the first space into the second. The method studied here is part of this category.

3 Method

The model proposed by Wan et al. [2022] is based on image translation. The goal is to learn a translation (mapping) between the domain of noisy images and the one of clean images. The novelty of this method is that it does not require pairs of real noisy and clean images to learn the mapping, but only unrelated sets of images of both domains. The method considers three domains represented on Figure 1 : the clean images $\mathcal{Y}$, the noisy images $\mathcal{R}$ and the synthetic domain $\mathcal{X}$ of clean images with synthetic degradation. The goal is to leverage the relation between the clean images and the images with synthetic noise to learn a mapping that will generalize well to the images with real noise.

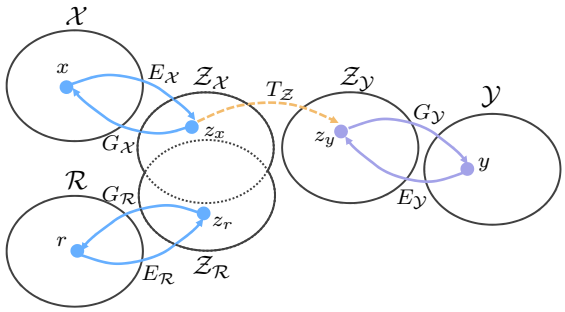


Figure 1: Translation method with three domains (from Wan et al. [2022])

First, the domains are mapped to their respective latent space. We wish for the latent representation of the synthetic images $\mathcal{Z}_{\mathcal{X}}$ and real noisy images $\mathcal{Z}_{\mathcal{R}}$ to be similar so that the mapping generalize more easily. The benefit of using latent spaces as an intermediary for the translation is to work with a more compact and meaningful representation of the image than the pixel space, resulting in an easier generalization of the model.

To define these latent spaces, we use variational autoencoders (VAE) trained for the reconstruction task. A VAE is a neural network with two parts, an encoder and a decoder. The encoder projects

the input image to a probability distribution in a latent space of smaller dimension and the decoder takes a latent representation (random variable realizing the latent distribution) and reconstruct the image. This probabilistic formulation, instead of the simpler Autoencoder architecture, allows for a smoother and regularized latent space, which improve the performance of the decoder when used for image generation. The networks are jointly trained to reconstruct the input image despite the bottleneck of the latent space, and during the process the latent representation is refined in order to retain the most important pieces information of the image.

In the model studied here, a first VAE (named VAE_1 below) is used for both real noisy images and synthetic images, with an encoder denoted as $E_{\mathcal{R},\mathcal{X}}$ and a decoder $G_{\mathcal{R},\mathcal{X}}$. An adversarial loss is added to encourage the latent space of the synthetic images to be close to the latent space of the real noisy images. Another VAE (later named VAE_2) is trained to reconstruct its input image from the domain of the clean images. The latent spaces of both VAEs constitute the latent spaces of the clean images $\mathcal{Z}_{\mathcal{Y}}$ and the noisy images $\mathcal{Z}_{\mathcal{R}} \cup \mathcal{Z}_{\mathcal{X}}$, respectively.

We then train a third network $\mathcal{T}$ to map the latent space of the noisy images to the latent space of the clean images. The full model is represented on figure 2. We use generated couples of images created from clean images degraded with synthetic noise, compute their respective latent representation and use them as target and input.

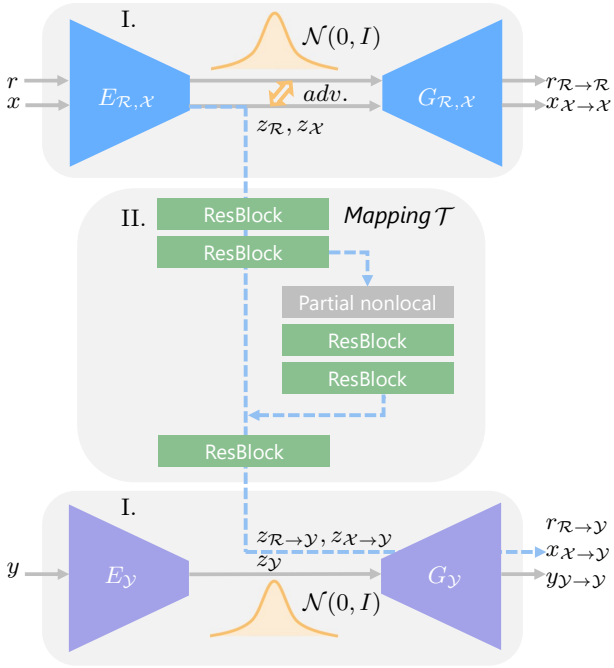


Figure 2: Architecture of the restoration network (from Wan et al. [2022]). The blue dashed line represents the processing path of a noisy image at inference time.

The method makes use of adversarial training and perceptual losses at different steps of the process in order to focus on improving the perceptual quality and to stabilize the training. The

adversarial training results from the formulation of the model as a generative adversarial network (GAN)[Goodfellow et al., 2014]. GANs are a type of generative model composed of two networks: a generator and a discriminator. The training of such network can be seen as a game in which the discriminator tries to identify if the input image is real (comes from the dataset) or fake (generated by the other network) while the generator aims at fooling the discriminator. The networks are trained simultaneously with antagonist goals and, as the accuracy of the discriminator increase, the realism of the generated image improves. Specifically, the version used in this work is inspired by LSGAN proposed by Mao et al. [2017] which adopt the mean squared error loss, perform a more stable training and generate higher quality images than the technique originally introduced by Goodfellow et al. [2014]. In the context of the image denoising method studied here, the generator is a VAE, the "real" images are the input images and the "fake" images are the one reconstructed by the VAE.

3.1 Architecture

The network is composed of three major modules. Two VAEs, onefor the noisy image domain and one for the clean domain, and a mapping network. In addition, multiple discriminator networks are used (one for each adversarial loss). The two VAEs share the same architecture but are trained completely independently with different loss functions as outlined in the next section. The VAE architecture used here is a simplified implementation that assume that the variance of the latent variables is 1. The advantages of this specific architecture are that it is easier to implement for similar results and it is fully convolutional, so it can take as input images of any size and output an image of the same size. The full model architecture is described in Table 2 (Appendix A). It is composed of the following building blocks.

1. The Convolution block is composed of a 2D convolution layer followed by an instance normalization layer and an LeakyReLU function.

2. The Deconvolution block is a transposed convolution layer, an instance normalization layer and a LeakyReLU. This block can be used to obtain a higher resolution image from a low scale representation. This block is used in the original model but it is replaced by Resize-Convolution block in the improved version we propose here. The Resize-Convolution block is formed from a Resize layer using Nearest-Neighbour upsampling, followed by a Convolution block.

3. The Residual Block is composed of two successive Convolution blocks, of which result is added to the input via a residual connection.

4. The Non-local block allows making use of features from the whole image in a flexible manner instead of being restricted to the receptive field as in a fully convolutional network. This property is important in the original model in order to restore severe structural degradation of the image such that scratches. In order to do so, a degradation mask is used with the non-local block to restrict the part of the image that can be used. In this work, we focus our implementation on the denoising part of the model so our Non-local block is simpler and directly adapted from the non-local neural network with embedded Gaussian introduced by Wang et al. [2018].

3.2 Losses and metrics

Each module of the model is trained with a combination of multiple losses. As usual for VAEs, the encoders are penalized for generating latent distributions that are too far from the prior distribution (centered Gaussian of variance 1) using KL divergence, which take a simplified form since we assume unit variance. The quality of the reconstruction generated by each module is evaluated using L_1 loss.

The model is trained in an adversarial manner, which demonstrated particularly good results for image generation in previous works [Goodfellow et al., 2014, Mao et al., 2017]. The adversarial loss $\mathcal{L}_{\text{GAN}}$ is computed from the mean squared error (MSE) between the output of a discriminator network and the target as proposed by Mao et al. [2017], however here we use a more general multiscale discriminator. To each scale corresponds a discriminator. The generated image is down scaled by a factor of two between each scale, and the final loss is an average over the losses of each level.

In addition to these contributions to the loss, a perceptual loss is used to encourage the model to generate images that are perceptually similar to the input. This concept corresponds to losses that promote realistic and perceptually pleasing results. Indeed, standard loss terms such that the L2 distance tends to encourage blurry or distorted images that minimize the loss on average but are not really satisfying for a human observer. Some perceptual losses and metrics make use of intermediary features of neural networks (the value outputted by intermediary layers of a network). The activation of these features represents higher level concepts (complex patterns or objects instead of single pixel colors), so using these features to compare images may be more relevant as an analogy to the way humans perceive images. The Learned Perceptual Image Patch Similarity (LPIPS) is a metric introduced by Zhang et al. [2018] that use a VGG network Simonyan and Zisserman [2014] pretrained for image classification. The metric takes two images as an input, pass them through the VGG network and then compute the differences of activation of a few specific layers. These differences are then combined to produce a metric that should reflect the perceptual dissimilarity between the two images (smaller LPIPS score means perceptually similar images). This metric is commonly used to evaluate image restoration models so it is also used here to complement the MSE metric.

The perceptual loss (denoted $\mathcal{L}_{\text{FM,VGG}}$ in 1) used in the method studied here is computed using a pre-trained VGG network Simonyan and Zisserman [2014], specifically the activations of 4 different layers of the network. The loss is computed as the mean squared error between the intermediary features of the input image and the ones of the generated image. The loss differs from LPIPS mainly in the layers used and the aggregation of the difference in activation of each neuron. Similarly, all the hidden layers of the discriminator D are used to compute what the original authors call the feature matching loss $\mathcal{L}_{\text{FM,D}}$.

The full loss used for the training of VAEs can then be expressed as follows :

$$\begin{aligned}
\mathcal{L}_{\text{VAE}}(x) = &\ \text{KL}\left(E\left(x\right) \| \mathcal{N}(0, I)\right) && \text{KL-Divergence} \\
&+ \alpha \mathbb{E}_{z_x \sim E(x)}\left[\left\| G\left(z_x\right) - x \right\|_1\right] && \text{Reconstruction loss} \\
&+ \mathbb{E}_{z_x \sim E(x)} \mathcal{L}_{\text{GAN}}(x) && \text{Adversarial loss} \\
&+ \frac{1}{2} \mathbb{E}_{z_x \sim E(x)} \left(\mathcal{L}_{\text{FM,D}}(G\left(z_x\right), x) + \mathcal{L}_{\text{FM,VGG}}(G\left(z_x\right), x)\right) && \text{Perceptual loss}
\end{aligned} \tag{1}$$

Where $E\left(x\right)$ is the latent distribution returned by the encoder E for the input x, $G\left(z_x\right)$ is the image returned by the decoder G from the latent variable z_x and α is a hyperparameter.

In order to make the latent spaces of the real noisy images and the images with synthetic noise overlap, a discriminator $D_{\mathcal{R},\mathcal{X}}$ trained to classify images of these two categories from their latent representation is introduced. This produces a new adversarial loss $\mathcal{L}_{\text{VAE}_1,\text{GAN}}$ which is added to the common VAE loss $\mathcal{L}_{\text{VAE}}(x)$ of VAE_1.

$$\mathcal{L}^{\text{latent}}_{\text{VAE}_1,\text{GAN}}(x) = \begin{cases} \text{D}_{\mathcal{R},\mathcal{X}}\left(E_{\mathcal{R},\mathcal{X}}(x)\right)^2 & \text{if } x \in \mathcal{X} \\ \left(1 - D_{\mathcal{R},\mathcal{X}}\left(E_{\mathcal{R},\mathcal{X}}(r)\right)\right)^2 & \text{if } x \in \mathcal{R} \end{cases} \tag{2}$$

The mapping network is trained by first passing a clean image through VAE_2 and use the latent representation and the reconstructed image as a target, then process a synthetically degraded version of the clean image with the full denoising network (following the dashed line on Figure 2 through $E_{\mathcal{R},\mathcal{X}}$, $\mathcal{T}$ and $G_{\mathcal{Y}}$). The loss function $\mathcal{L}_{\mathcal{T}}$ of the mapping network impose constraint both on the latent space and the final image. A L1 loss $\mathcal{L}_{\mathcal{T},l_1}$ penalizes the difference between the latent representations of the images in the domain $\mathcal{Z}_{\mathcal{Y}}$ (output of the mapping network). Similarly to VAE training, an adversarial loss and a feature matching loss (perceptual loss) are computed using the final output image of the network. The final mapping loss can be written as

$$\mathcal{L}_{\mathcal{T}}(x) = \lambda_1 \mathcal{L}_{\mathcal{T},\ell_1}(x,\bar{x}) + \mathcal{L}_{\mathcal{T},\text{GAN}}(x) + \lambda_2 \mathcal{L}_{\text{FM}}(x,\bar{x}) \tag{3}$$

where λ_1 and λ_2 are hyperparameters, $\bar{x}$ is the clean image associated to the noisy image x (x is derived from $\bar{x}$ by adding synthetic noise).

For the evaluation of the model we use LPIPS metric to measure the perceptual similarity between to images, as well as the PSNR metric which is commonly used to measure the distortion of an image after a reconstruction process. This last metric is defined as

$$PSNR(x,y) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE(x,y)}}\right)$$

where $MSE(x,y)$ is the mean squared error between the images x and y and MAX_I the amplitude of the signal (typically 255 for images with pixels encoded on 8 bit).

3.3 Transposed convolution layer

When using neural networks to generate images, it is common to change the resolution of the image. For instance in VAEs, the dimension of the input image is generally decreased, so it can be represented in a low dimensional latent space, this latent representation can then be decoded to get back a high resolution image. Other networks make use of these downscaling and upscaling operations in order to exploit the structure of the image at different scales.

These operations can be performed in a variety of ways. For downscaling, strided convolutions and pooling (average-pooling, max-pooling ...) are often used. For upscaling, it is possible to use interpolation techniques such as nearest-neighbor interpolation or bilinear interpolation. Another commonly used layer for upscaling is the transposed convolution layer, also known as deconvolution.

The idea behind a transposed convolution is to use each pixel of the input image to paint a larger area in the output image based on a kernel. This operation can also be seen as applying a convolution to an image obtained from the input in which we fill the missing pixel values with zeros. For a given kernel size, this operation has the same parameters as regular convolution, namely padding and stride, that will influence the dimension of the upscaled image.

One problem with transposed convolutions is that, depending on the parameters used, the number of pixels of the input image responsible for the creation of the new pixels may not be uniform. This phenomenon is illustrated on Figure 3 for a 1D signal. While it should be possible for the networks to learn weights that correct this problem, they often have difficulties to completely compensate this effect. This phenomenon has a visible effect on the generated image, resulting in checkerboard-like patterns.

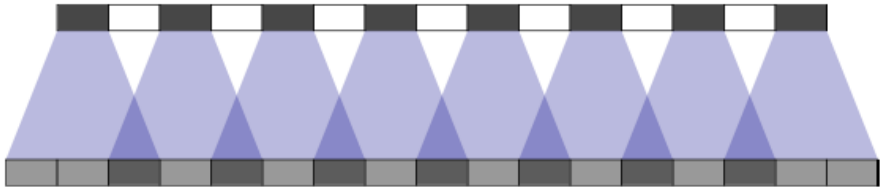


Figure 3: Overlap pattern of a transposed convolution on a 1D signal. Kernel size = 3. Stride = 2 (from Odena et al. [2016])

In particular, transposed convolution layers are used in the decoders of both VAEs in the original model. We can, as expected, observe checkerboard artifacts on the results of the denoising model as we demonstrate in the experiments (cf. Figure 5). We propose to replace these layers by the combination of an upsampling layer and a convolution layer, as this association is less prone to produce undesirable checkerboard effects [Odena et al., 2016].

4 Experiments

4.1 Training

The models used in this study were implemented from scratch based on the model described by Wan et al. [2022] and the code provided by the authors. We noticed differences between the description and the actual implementation. The model described by the paper defines the one referred below as the model without perceptual loss. The main differences are that both VAEs were trained without the use of a perceptual loss and with a single-scale discriminator. A second model, that we refer to as the original model, replicates the architecture of the provided implementation. Finally, an improved model is derived from the original by replacing transposed convolution layers by Resize-Convolution layers. The training is similar to the one announced in the original paper. We use the ADAM optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is set to 0.0002. We use randomly cropped images of size 256×256 pixels. The parameters of the loss are $\alpha = 10$,$\lambda_1 = 60$ and $\lambda_2 = 10$.

4.2 Dataset

Two training datasets were used in this study. Flickr500 provides 500 clean images for our experiment, however the results using this dataset did not seem as good as the ones announced by Wan et al. [2022] for the original model. We think that the reasons that could explain this difference in visual quality could be the lack of diversity of training examples as well as the aliased aspect of the images,

which is difficult to reproduce with this architecture and may have penalized the training as well as the evaluation of the final model.

We then used the PascalVOC dataset for our final evaluations of the model. We observed a faster training as well as higher score for both PSNR and LPIPS metrics after the training. This dataset contains more diversity and less altered images, however the overall quality of the pictures seems inferior to the Flickr500 dataset and all the samples are provided in the JPEG format, which could make it harder to remove this type of compression artifacts.

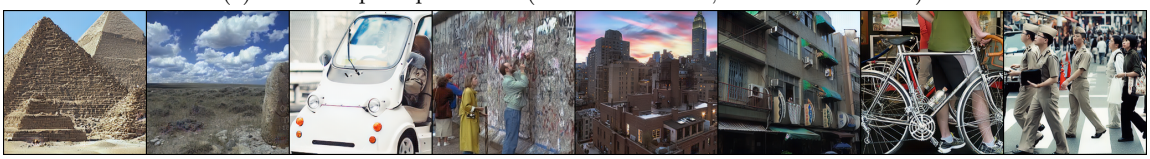
The images with synthetic degradation are obtained by applying any combination of the following transformations in a random order to a clean image from the dataset (either Flickr500 or PascalVOC):

- Gaussian blur with a standard deviation between 1 and 5 and a kernel size between 3 and 7. This effect is obtained by convolution of the image with a Gaussian kernel.
- Additive white Gaussian noise : An additive noise sampled from a centered Gaussian distribution of standard deviation between 40 and 100 is added to each pixel independently.
- JPEG compression with a quality factor between 40% and 100%.
- Color jitter : Uniform shift of the values of the pixels in an image.

(a) Input image

(b) Original model (LPIPS = 0.0905, PSNR = 25.11dB)

(c) Without perceptual loss (LPIPS = 0.2662, PSNR = 22.84dB)



(d) Replacing transpose convolution layer by resize-convolution (LPIPS = 0.0964, PSNR = 24.01dB)

Figure 4: Image reconstruction with VAE_2 trained on Flickr500 (Figure best seen zoomed).

Figure 5: Reconstruction results from VAE_2 demonstrating checkerboard artifacts (details from left to right: Input image, Original model results, Without perceptual loss, With Deconvolution blocks replaced by Resize-Convolution blocks)

4.3 Results

The figure 4 compares the reconstruction results obtained with different variations of VAE_2 networks. We can observe that without perceptual loss during training, the images obtained are smoothed out and a lot of fine details are lost. In particular, the phenomenon is apparent with the human faces that lack recognizable features.

Checkerboard artifacts can be observed on the images reconstructed by VAE_2 and the images denoised by the full model, as demonstrated on Figure 5. These artifacts are often visible in originally flat areas for the two models that use transposed convolutions, but it seems to be reduced by the introduction of a perceptual loss during training. Replacing that layer with a resizing layer followed by a standard convolution seems to completely solve this issue. However, we observe a decrease of accuracy of the reconstruction according to both PSNR and LPIPS scores. Best performances were achieved with nearest neighbor upscaling and 5×5 convolutions which slightly increase the number of parameters of the new model (4×4 transposed convolution were used for the original model). Figure 4 displays the results of all the models trained on Flickr500 dataset for comparison. Better

reconstruction results were achieved using PascalVOC dataset. The PSNR increased to 28.8 dB for the original model and 27.4 dB for the one without transposed convolutions.

In all our experiments, we observe areas in which the colors are significantly darker than the original image, generally completely black or with a contrasted checkerboard pattern. The effect is visible when asking the VAE to perform a reconstruction of an image that is not in the training set and remains after training the complete model. This kind of artifact is rather rare, however it could already be observed in the original model as shown on Figure 6. We believe that this effect may be due to the fact that the VAE is trained on images that are very different from the images that are used for testing. The removal of the transposed convolution layers seems to improve the results, however it is still present (but without the periodic checkerboard pattern). We observe a decrease of the effect with more training. In order to further improve the results, we decided to train the model on a bigger dataset, which reduced the occurrence of these artifacts as well as sped up the training (as in better metrics for a given number of iterations) and increased both LPIPS and PSNR scores. In addition, the denoised images generated by the network differ depending the dataset both in sharpness of edges and the average color. The difference in results can be observed in Figure **??**.

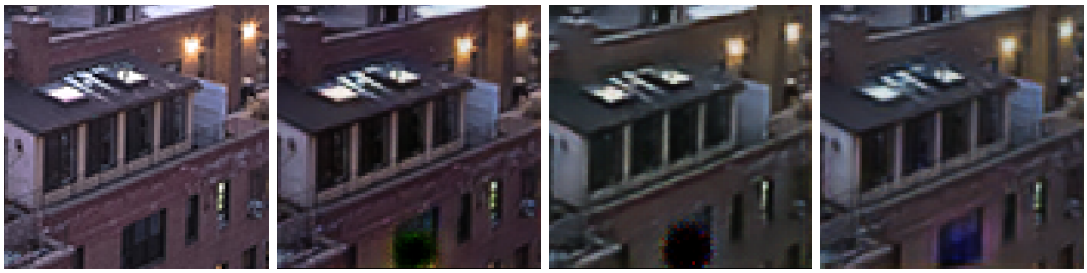


Figure 6: Example of a "black artifact" in the window at the bottom-center of the image (extracted from Figure 4).

Examples of denoising results are displayed in Figure 7. We compare our results (on the right) to the one of the original implementation provided by the authors (in the middle). Our version was trained on PascalVOC dataset to match the original model. We can observe that the checkerboard effect, which was well visible on the whole denoised photo when zoomed, is completely removed in our version. However, we can observe that some wavelike patterns appear in the normally flat background of our result (first row). The origin of this degradation may be either a structural defect in the original image or the artifacts introduced by JPEG compression of an already noisy image. Overall, the original model seems to be able to better remove this type of artifacts. One additional drawback is the presence of the strangely colored areas that correspond to the black artifacts described above. They seem to already be present in the original model but their occurrence rate is higher in our model. From our observations during this study we hypothesize that the model can be further trained in order to reduce this issue, however we are not sure of its true origin.

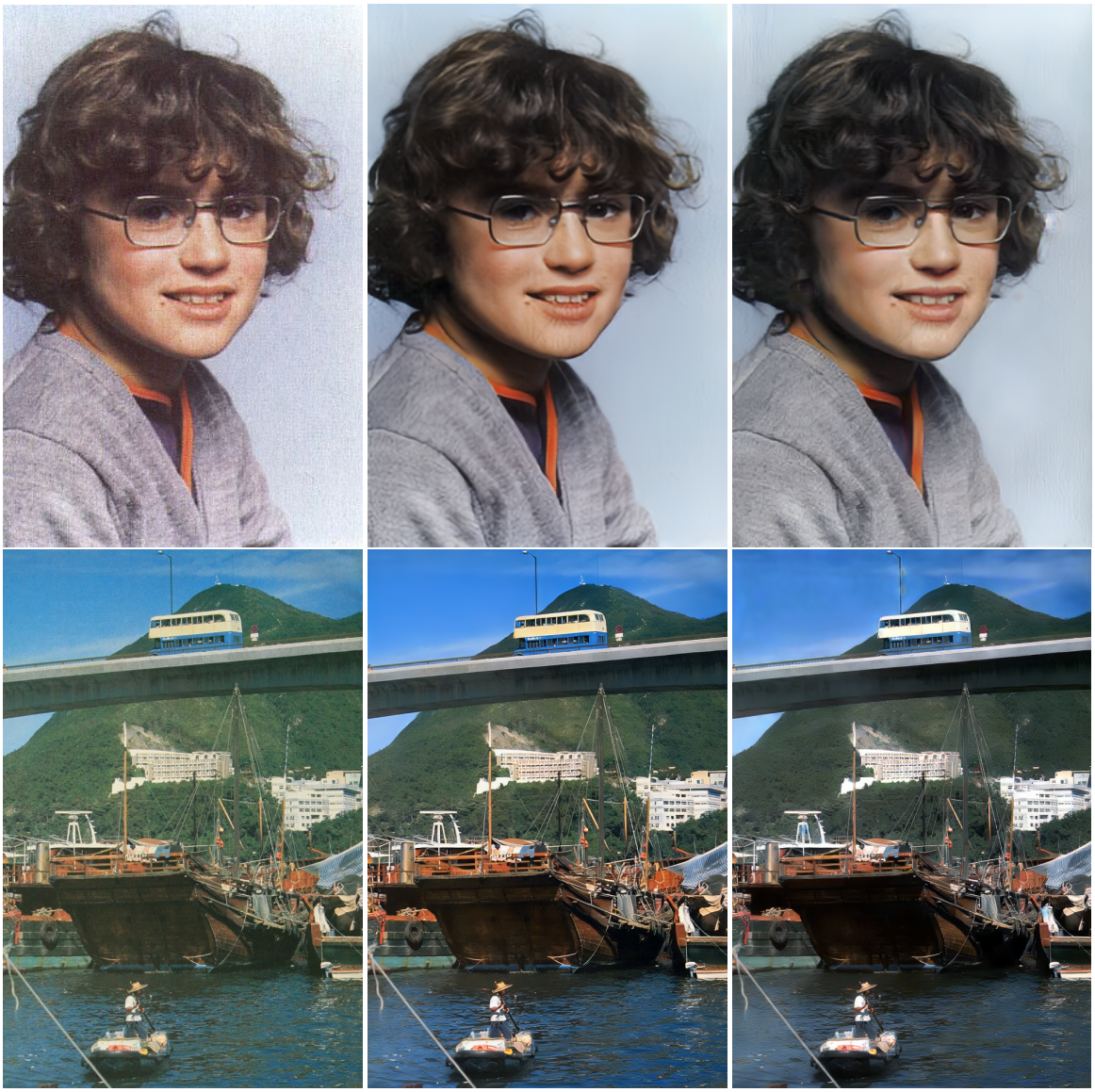


Figure 7: Comparison of the denoising results of different models (From left to right: Input image, original model, Without transposed convolution) (Image best seem zoomed)

4.4 Evaluation without reference

It is difficult to evaluate the model on real noisy images since we do not normally have the corresponding clean images. We decided to use the method proposed by Talebi and Milanfar [2018] which makes use of a neural network to predict the quality of an image. It provides a score between 0 and 10 that should be comparable to the scores given by human observers. The advantage of this

kind of evaluation is that it does not require a reference image, which is often unavailable when denoising real images. We use the implementation provided by Lennan et al. [2018].

We evaluate the model on a small sample of old photos (two of them are displayed in Figure 4) in order to compare the effect of each method on quality assessment. Two models are offered with the implementation [Lennan et al., 2018], one for aesthetic assessment and one for technical assessment. They differ in the data used for the training (the score associated to the images either refers to aesthetic or technique).

Using the technical assessment model, the average quality of the images does not improve on average from using our version of the denoising method. However, the assessment of each image does change after denoising, some improve and some get a lower score. The original method demonstrates a slight improvement of the technical score.

Using the aesthetic assessment model, the score obtained by the images is clearly improved by the denoising models. Most images obtain a better score with both method, with an advantage for our version.

These results, presented in Table 1, could be explained by the data the assessment network were trained with. The "aesthetic" assessment may be better correlated to the tasks the denoising network is trying to perform. In this work we put an emphasis on old photo restoration with unknown noise model, however the results show that, in addition to denoising, the network performs other kinds of restoration such that debluring and color adjustment, which may have a bigger impact on the perceived aesthetic quality than on the technical quality.

	Technical	Aesthetic
Input	5.53	4.70
Original model	5.74	5.02
Without transposed convolution (PascalVOC)	5.52	5.10
Without transposed convolution (Flickr500)	5.54	5.33
Input + SR	5.68	5.30
Original model + SR	5.84	5.49
Without transposed convolution (PascalVOC) + SR	5.67	5.45
Without transposed convolution (Flickr500) + SR	5.60	5.60

Table 1: Average quality assessment score (computed using models from Talebi and Milanfar [2018])

4.5 Combining image restoration and Super-Resolution

We studied the effect of combining the denoising model with a Super-Resolution technique. Real-ESRGAN is a GAN based architecture for Super-Resolution proposed by Wang et al. [2021] in which the generator takes an image as input and return an image of higher resolution (up to $\times 4$ higher with the version we used). The experiment consists in first denoising the photo and then upscaling it by a factor of 3.5 using the pretrained model provided by the authors and without fine-tuning. The aesthetic and technical scores are presented in Table 1 and some example images are shown in Figure 8.

We can observe that the technical scores are marginally increased by this extra step. As for the aesthetic score, the results are greatly improved, with a maximum reached by combining our model without Deconvolution block with Real-ESRGAN. Figure 8 illustrate how each model influence the final result. Our version without transposed convolutions allows to recover more realistic textures

than the original version or Real-ESRGAN only. We can observe that Super-Resolution amplifies and hallucinate some of the artifacts present on low resolution images. Finally, Real-ESRGAN alone provides particularly detailed reconstruction of the faces compared to the model studied here.

Overall, our version of the model combined with Real-ESRGAN provides promising results since the super resolution network is not well-fitted to specifically remove old photo degradation. However, in some cases the two network do not interact well with each other, which leads to unrealistic texture on the high resolution results. This issue may be improved by fine-tuning the Super-Resolution network with results from our model.

It should be noted that Real-ESRGAN was trained on a different dataset and only with synthetically degraded images (low resolution noisy images) so the combination does not introduce weights learned from clean/noisy real image pairs.

5 Conclusion

In this report, we reviewed the model proposed by Wan et al. [2022] for old image restoration. This model is able to learn image restoration from unrelated set of noisy and clean real images. We implemented the core part of the method (except scratch restoration and face enhancement) and we were able to produce results close to those announced in the paper. We were able to identify and document necessary parts of the implementation that were missing in the paper, such as the generalized use of perceptual losses and the multi-layer discriminator.

One of the main flaws of the original model is the production of checkerboard-like artifacts on the images, creating unrealistic periodic patterns that degrade their visual quality. We were able to address this issue by replacing the transposed convolution layers with resize-convolution blocks. We nevertheless noted decreased performance of the VAE components of the methods according to LPIPS and PSNR metric.

We tried to evaluate the denoising without reference using the technique proposed by Talebi and Milanfar [2018] with mild success. While the aesthetic assessment of the image is clearly improved by the method, the technical score seems to weakly correlate with the denoising process of the networks. Finding a relevant metric for image assessment without reference will be an important challenge for the future development of image generation techniques.

The experiments with the combination of the denoising network studied with a super resolution network shows satisfying results. The two techniques share some functionalities, but they can still mutually improve the quality of their output. However, the tasks performed by the networks are partly redundant so future researches may combine the advantages of both techniques in a single and more efficient network.

Finally, while the technique proposed by Wan et al. [2022] provides impressive results, it still relies on the generalization of the network trained from synthetic noisy images. The noise model is simple, which may make the training and the generalization harder. Including a more realistic noise model to this technique could be a way to facilitate generalization and further improve the results.

References

Abdelrahman Abdelhamed, Marcus A Brubaker, and Michael S Brown. Noise flow: Noise modeling with conditional normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3165–3173, 2019.

Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *International Conference on Machine Learning*, pages 524–533. PMLR, 2019.

Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11036–11045, 2019.

Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 2, pages 60–65. Ieee, 2005.

Kostadin Dabov, Alessandro Foi, and Karen Egiazarian. Video denoising by sparse 3d transform-domain collaborative filtering. In *2007 15th European Signal Processing Conference*, pages 145–149. IEEE, 2007.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL `https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf`.

Lanqing Guo, Siyu Huang, Haosen Liu, and Bihan Wen. Fino: Flow-based joint image and noise model. *arXiv preprint arXiv:2111.06031*, 2021.

Geonwoon Jang, Wooseok Lee, Sanghyun Son, and Kyoung Mu Lee. C2n: Practical generative noise modeling for real-world denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2350–2359, 2021.

Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2129–2137, 2019.

Alexander Krull, Tomáš Vičar, Mangal Prakash, Manan Lalit, and Florian Jug. Probabilistic noise2void: Unsupervised content-aware denoising. *Frontiers in Computer Science*, 2:5, 2020.

Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. *Advances in Neural Information Processing Systems*, 32, 2019.

Marc Lebrun, Antoni Buades, and Jean-Michel Morel. A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013.

Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*, 2018.

Christopher Lennan, Hao Nguyen, and Dat Tran. Image quality assessment. `https://github.com/idealo/image-quality-assessment`, 2018.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. doi: 10.23915/distill.00003. URL `http://distill.pub/2016/deconv-checkerboard`.

Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1586–1595, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Jae Woong Soh and Nam Ik Cho. Deep universal blind image denoising. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 747–754. IEEE, 2021.

Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE transactions on image processing*, 27(8):3998–4011, 2018.

Ziyu Wan, Bo Zhang, Dongdong Chen, Pan Zhang, Dong Chen, Fang Wen, and Jing Liao. Old photo restoration via deep latent space translation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1905–1914, 2021.

Guoshen Yu and Guillermo Sapiro. Dct image denoising: a simple and effective image denoising algorithm. *Image Processing On Line*, 1:292–296, 2011.

Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. Dual adversarial network: Toward real-world noise removal and noise generation. In *European Conference on Computer Vision*, pages 41–58. Springer, 2020.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5728–5739, 2022.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 international conference on computer vision*, pages 479–486. IEEE, 2011.

Appendices

A Architecture of the network

Module	Layer	Kernel size / stride	Output size
Encoder E	Conv	$7 \times 7/1$	$256 \times 256 \times 64$
	Conv	$4 \times 4/2$	$128 \times 128 \times 64$
	Conv	$4 \times 4/2$	$64 \times 64 \times 64$
	ResBlock×4	$3 \times 3/1$	$64 \times 64 \times 64$
Decoder G	ResBlock×4	$3 \times 3/1$	$64 \times 64 \times 64$
	Deconv	$4 \times 4/2$	$128 \times 128 \times 64$
	Deconv	$4 \times 4/2$	$256 \times 256 \times 64$
	Conv	$7 \times 7/1$	$256 \times 256 \times 3$
	Tanh	/	$256 \times 256 \times 3$
Discriminator scale 1	Conv	$4 \times 4/2$	$128 \times 128 \times 64$
	Conv	$4 \times 4/2$	$64 \times 64 \times 128$
	Conv	$4 \times 4/2$	$32 \times 32 \times 256$
	Conv	$4 \times 4/1$	$32 \times 32 \times 512$
	Conv	$4 \times 4/1$	$32 \times 32 \times 1$
Discriminator scale 1/2	Conv	$4 \times 4/2$	$64 \times 64 \times 64$
	Conv	$4 \times 4/2$	$32 \times 32 \times 128$
	Conv	$4 \times 4/2$	$16 \times 16 \times 256$
	Conv	$4 \times 4/1$	$16 \times 16 \times 512$
	Conv	$4 \times 4/1$	$16 \times 16 \times 1$
Mapping $\mathcal{T}$	Conv	$3 \times 3/1$	$64 \times 64 \times 128$
	Conv	$3 \times 3/1$	$64 \times 64 \times 256$
	Conv	$3 \times 3/1$	$64 \times 64 \times 512$
	Nonlocal	$1 \times 1/1$	$64 \times 64 \times 512$
	Resblock×2	$3 \times 3/1$	$64 \times 64 \times 512$
	ResBlock×6	$3 \times 3/1$	$64 \times 64 \times 512$
	Conv	$3 \times 3/1$	$64 \times 64 \times 256$
	Conv	$3 \times 3/1$	$64 \times 64 \times 128$
	Conv	$3 \times 3/1$	$64 \times 64 \times 64$

Table 2: Detailed network structure.

B Combining old photo denoising and Super-Resolution : Examples

(a) From left to right: Input image after super resolution, zoom on the first image, Original model + super resolution, our model + super resolution. Our model decreases the unrealistic periodical pattern of the two other methods.

(b) From left to right: Input image, denoised with our model, Super resolution without denoising, Super resolution after denoising. Super resolution alone seems more realistic in that case even if the realistic brick texture are likely hallucinated

(c) Face reconstruction using different methods (From left to right: Input, Original model, Without transposed convolution, Downscaled result from Real-ESRGAN)

Figure 8: Combined results of Real-ESRGAN and old photo denoising