Deep-Based Film Grain Removal and Synthesis

Zoubida Ameur, Wassim Hamidouche, Edouard Francois, Milos Radosavljevic, Daniel Menard, Claire-Hélène Demarty

Deep-based Film Grain Removal and Synthesis

Zoubida Ameur, Wassim Hamidouche, Edouard François,
Miloš Radosavljević, Daniel Menard and Claire-Hélène Demarty

***Abstract*—In this paper, deep learning-based techniques for film grain removal and synthesis that can be applied in video coding are proposed. Film grain is inherent in analog film content because of the physical process of capturing images and video on film. It can also be present in digital content where it is purposely added to reflect the era of analog film and to evoke certain emotions in the viewer or enhance the perceived quality. In the context of video coding, the random nature of film grain makes it both difficult to preserve and very expensive to compress. To better preserve it while compressing the content efficiently, film grain is removed and modeled before video encoding and then restored after video decoding. In this paper, a film grain removal model based on an encoder-decoder architecture and a film grain synthesis model based on a conditional generative adversarial network (cGAN) are proposed. Both models are trained on a large dataset of pairs of clean (grain-free) and grainy images. Quantitative and qualitative evaluations of the developed solutions were conducted and showed that the proposed film grain removal model is effective in filtering film grain at different intensity levels using two configurations: 1) a non-blind configuration where the film grain level of the grainy input is known and provided as input, 2) a blind configuration where the film grain level is unknown. As for the film grain synthesis task, the experimental results show that the proposed model is able to reproduce realistic film grain with a controllable intensity level specified as input.**

I. Introduction

Originally film grain is a characteristic of analog film. It is the result of the processes of exposing and developing silver halide crystals [1], i.e., light-sensitive crystals that when exposed to light capture an image on a film. During the development process, crystals that are exposed to sufficient light are transformed into small particles of metallic silver. Others that are not developed are removed from film, leaving tiny gaps between those which are developed. Those small particles and gaps are in fact the result of many microscopic and chemical processes that, in the final stage of printing or projecting the film, lead to the creation of images with a grainy look. Film grain appearance is therefore inevitable because of the physical nature of the process embedded in the film design itself. However, historically, it was considered as noise, and as such, technological advances have gone in the direction of its elimination. With the arrival and evolution of digital camera sensors, film grain no longer exists. Moreover, digital imaging offered many more advantages in terms of robustness, reproducibility, and above all visual quality. Yet, most professional photographers and filmmakers would rather stick with the analog aspect when it comes to producing creative and artistic content, as they find the digital content too clean and sharp, which does not necessarily capture the sought atmosphere and therefore does not evoke the desired emotions. To better portray the cinematographic aspect of an analog film, several post-processing operations are commonly applied on the digital content. Adjustment of the color palette, adjustment of the contrast and generation of film grain contribute to distinctive characteristics of an analog film within a digital content [2]. This motivation turns film grain into a visual tool and not just a by-product of chemical processing as in analog film stock.

However, within today's video distribution systems, the random nature of film grain makes its preservation difficult because of the high bitrate it requires to be encoded. Therefore, it is challenging to find a balance between perfect fidelity of film grain and efficient compression that is an integral part of any such system [3], [4]. Due to its random nature, film grain is difficult to predict by using typical prediction schemes of modern video coding standards. Thus, most of it remains in the prediction residue. Thereafter, its transformed coefficients are mainly distributed in the high frequency band, and are, consequently, more expensive to encode in the transform domain. The existence of film grain has a negative impact on the accuracy of predictions and motion estimation, which further reduces the coding efficiency in both motion estimation and spatial prediction [5]. Because of that, high bitrates are necessary to reconstitute the film grain with a sufficiently good fidelity [4]. However, such high bitrates are generally not relevant in most common video applications.

To preserve film grain while improving coding efficiency, the natural approach would be to remove film grain from the content prior to encoding in order to achieve a higher coding efficiency and synthesize it back after decoding. When it comes to modern video distribution systems, where stronger compression is an inevitable step, film grain is destroyed at the encoder by compression itself without the possibility of reconstructing it. Hence, one solution is to use a parametric model to capture film grain characteristics prior to filtering and/or encoding and synthesize it back at the decoder side with the aid of appropriate metadata. Figure 1 provides a simplified block diagram of a typical video distribution system including film grain processing steps. Given an input video sequence, film grain is first filtered and modeled in a pre-processing step. The filtered video is then encoded and transmitted to the decoder together with film grain metadata. At the decoder side, the video is decoded and passed into a post-processing step that aims at reproducing the input video look by synthesizing film grain. Thus, film grain can be recovered while the content is more efficiently compressed. Such removal and synthesis steps are already implemented in recent video coding

Zoubida Ameur, Edouard François, Miloš Radosavljević and Claire-Hélène Demarty are with InterDigital R&D, Cesson-Sévigné, France (e-mail: firstname.lastname@interdigital.com).

Wassim Hamidouche and Daniel Ménard are with Univ. Rennes, INSA Rennes, CNRS, IETR - UMR 6164, Rennes, France (e-mail: firstname.lastname@insa-rennes.fr).

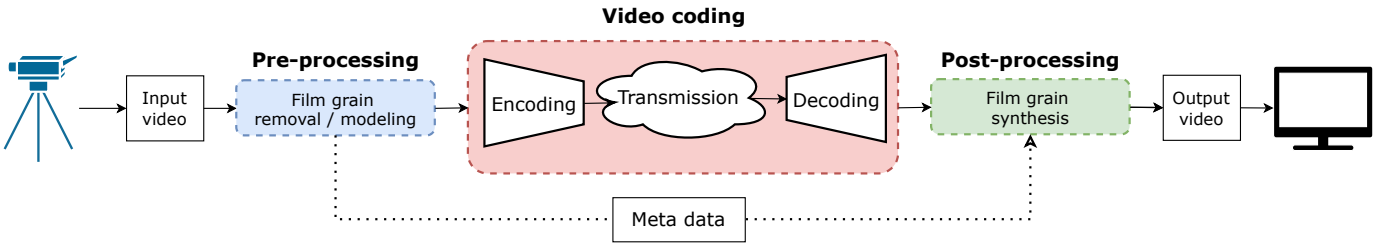


Fig. 1: A simplified framework of the video distribution system with film grain removal, modeling, and synthesis steps.

standards, *e.g.,* VVC and AV1, where bitrate savings (up to 30% in AV1) are of high interest not only for broadcast but also for streaming use cases [6].

To summarize, film grain can be used: 1) in post-production where it is added to the digital content to improve its visual appearance and to add an artistic touch; 2) after decoding the content in case film grain was filtered during pre-processing and/or by compression itself (in which case model parameters are tuned manually or automatically to match or closely approximate the original look); or 3) and this is another potential use of film grain not already discussed above, as a visual tool tasked to mask compression artifacts and restore vividness in the compressed video (in which case it does not necessarily render the original film grain look and can be added even on content that had no film grain at the first place). In the latter case, film grain helps blending the content with its underlying texture, so that there is continuity between objects in the same image. It also helps smoothing out imperfections in the content, such as compression artifacts and distortions due to transmission errors [7], [8].

Film grain is re-emerging in the age of digital content and is becoming increasingly relevant both for artistic motivations and perceptual quality enhancement. Concurrently, deep learning is nowadays applied in several computer vision and image processing tasks, with very impressive results due to the high modeling capability and advances in training and network design. To this end, we propose to leverage deep learning-based models to remove film grain before encoding and to synthesize it after decoding following the film grain encoding framework.

The rest of this paper is organized as follows. Section II describes a brief overview on film grain removal and synthesis techniques in addition to state-of-the-art image denoising methods. Section III provides a full description of the proposed solutions. In Section IV, the experimental results are presented and analyzed. Finally, Section V concludes the paper.

II. Related work

In this section, prior work related to film grain coding and processing is discussed. First, the film grain coding concept is defined as well as its standardization in different video codecs. Then, film grain coding stages are detailed, related first to film grain removal techniques as a pre-processing step, then to film grain synthesis techniques as a post-processing step. Film grain removal techniques are followed by an outline of the most interesting solutions in image denoising. Image denoising is one of the most fundamental tasks in image processing and computer vision, and de-graining is an equivalent task to denoising when film grain is considered as noise. The reader is referred to [9], [10] for a more detailed review on this topic.

A. Film grain coding

A typical modeling scheme for film grain coding was proposed by Gomila *et al.* [11], in which film grain is filtered from the original video sequence as a pre-processing step before encoding and synthesized back as a post-processing step after decoding. A similar scheme has been used for speech coding [12], where the inactive speech signal is pre-processed before encoding, and noise is added to the decoded signal for the comfort of human perception.

In the pre-processing step, film grain is further modeled and encoded in the form of a parameterized model. The model parameters are transmitted to the decoder and used to simulate back the film grain. The transmission of the parameters is accomplished by the so-called supplemental enhancement information (SEI) messages. Since the introduction of a film grain SEI message in the H.264/MPEG-4 AVC [11] standard, film grain modeling has become part of modern video coding standards. Following this, many works describing film grain parametrization and film grain synthesis within a video coding framework [13]–[15] were produced. For example, recent activity in the joint video exploration team (JVET) of ITU-T video coding experts group (VCEG) and ISO/IEC motion picture experts group (MPEG) promotes the modeling of film grain as part of the video distribution chain. The aforementioned works inherit the same syntax and semantics of the AVC film grain SEI message and apply it to the subsequent standards including H.265/HEVC and H.266/VVC. It is important to note that the SEI specification only provides the syntax to transmit parameters of the model, without the specification of the methodology for removing and synthesizing film grain or estimating model parameters.

B. Film grain removal

In the literature, algorithms specifically tailored for film grain removal were recently presented. In [16], it was proposed to use the H.264/MPEG-4 AVC video encoder for film grain removal, where film grain is estimated by subtracting the encoded picture from the original one. Campisi *et al.* [17] proposed a filter that spatially adapts to local details in the image to avoid removing them while filtering film grain. The filter belongs to the class of contrast enhancement filters [18] and its coefficients are adaptively adjusted based on

the local statistics of the image. Thus, removing film grain while avoiding adding distortions like blurring. A Bayesian approach can be incorporated into a physically motivated noise model where film grain is modeled using an inhomogenous β distribution with the variance being a function of image luminance [19]. To remove film grain, this model is combined with a recent prior model of images called fields of experts (FOE) which is a high-order Markov random field (MRF) model that captures rich structural properties of natural images.

A selective filtering using 2D spatial filters is applied only in the edge-free regions to remove film grain without blurring the edges and thus degrading the original image quality [20]. This selective filtering preserves the edges and textures of the original image, but it somewhat limits the efficiency of the coding as film grain remains in the edges or the textured regions after denoising. In [5], prior to video encoding, essential parameters of film grain are estimated such as the spatial correlation of noise and the relationship between noise variance and signal intensity. Then, a temporal filter based on multi-hypothesis motion compensated filter (MHMCF) is applied to remove film grain. MHMCF [21] is known for preserving most spatial details and edges, however, it was observed that film grain remains in the blue plane after applying the temporal denoising. Therefore, authors in [22] proposed to explore cross-color correlations to enhance denoising performance.

The aforementioned work has investigated the film grain removal task and obtained qualitative results. However, most of the proposed solutions require at least one additional step before the removal of film grain, such as film grain modeling or edge detection. As a result, the quality of the filtered outputs is highly dependent on the quality of the outputs from the previous steps. Second, some methods use hand-crafted image features as prior information for filtering, which may not be relevant to the film grain removal task. Therefore, our solution is designed to overcome these drawbacks by using an end-to-end deep learning encoder-decoder architecture that takes a grainy image as input and outputs the corresponding filtered one. Since the input and the output are renderings of the same underlying content and structure, but with a different style, the low-level information is fed directly from the encoder to the decoder via skip connections such that the relevant underlying features for the film grain removal task are learned without any hand engineering.

C. Image denoising

Image denoising is a core image processing problem that has been studied for decades [9], [10], [23]. Many studies have tackled this problem and proposed different approaches [24]–[28], each offering certain advantages and suffering from certain drawbacks making the image denoising task open and challenging. However, noise considered in most of these studies is assumed to be additive, non-correlated, Gaussian, and signal-independent, whereas film grain is signal-dependent and not necessarily Gaussian. Therefore, additional efforts had to be made to efficiently adapt state-of-the-art image denoising to the film grain removal task.

Image denoising methods can be classified into two main categories: traditional hand-crafted and deep learning-based image denoisers. Traditional image denoisers can be roughly classified into two main sub-categories: spatial domain filtering and transform domain filtering. Denoisers that operate in the spatial domain are applied directly on the image samples to suppress the unwanted variations in sample intensity values and therefore, suppress noise. On the contrary, denoisers that operate in the transform domain first map the image into corresponding transform coefficients, on which they carry out some thresholding [29]. As to learning-based image denoisers, state-of-the-art solutions are mainly based on convolutional neural networks (CNNs) which try to learn a mapping function by optimizing a loss function on a training set that contains pairs of clean references and noisy images [26], [30].

Block-matching and 3D filtering (BM3D) is the most popular hand-crafted image denoising algorithm [24]. BM3D is a non-locally collaborative filtering method in the transform domain. First, image patches similar to a given image patch are selected and grouped to form a 3D block, then, a 3D linear transform is performed on the 3D block followed by a filtering of the transform coefficients. Finally, an inverse 3D transform is applied to get back to the spatial domain and the different patches are aggregated to form the original image. This constitutes the first collaborative filtering step which is greatly improved by a second step using Wiener filtering. In the second step, instead of comparing the original patches, the filtered ones are compared and grouped to form the new 3D group which is processed by Wiener filtering instead of applying a threshold. Finally, an aggregation step is performed. A slightly different approach that utilizes temporal dimension of a video, named motion compensated temporal filter (MCTF) [31], [32] is utilized within the latest (H.266/VVC) reference software VTM. The proposed method relies on a bilateral filter [33] across neighboring pictures compensated by temporal motion. The MCTF is also used in [14] to filter out film grain from the video.

Several state-of-the-art works have addressed the problem of denoising using deep neural networks. Zhang *et al.* [26] proposed a blind CNN-based denoiser called denoising convolutional neural networks (DnCNN) that takes as input a noisy image and outputs a denoised version of it. This work demonstrated that residual learning, originated in ResNet [34], and batch normalization, derived from Inception-v2 [35], improve the denoising performance of the model. In a recent study, Zhang *et al.* proposed a non-blind CNN-based solution [30] fast and flexible denoising convolutional neural network (FFDNet) that takes as input both the noisy image and its noise level map and outputs the denoised version. They proposed two different models for grayscale and color images of 15 and 12 convolution layers, respectively. The denoising is performed on downsampled sub-images to speed-up the training and to boost the performance. DnCNN and FFDNet have comparable architectures with a collection of Convolution + Batch Normalization + ReLU layers. Another model has been proposed by the same authors in their most recent work, DRUNet [28], where residual blocks were integrated into U-Net for effective denoiser prior modeling. Like FFDNet,

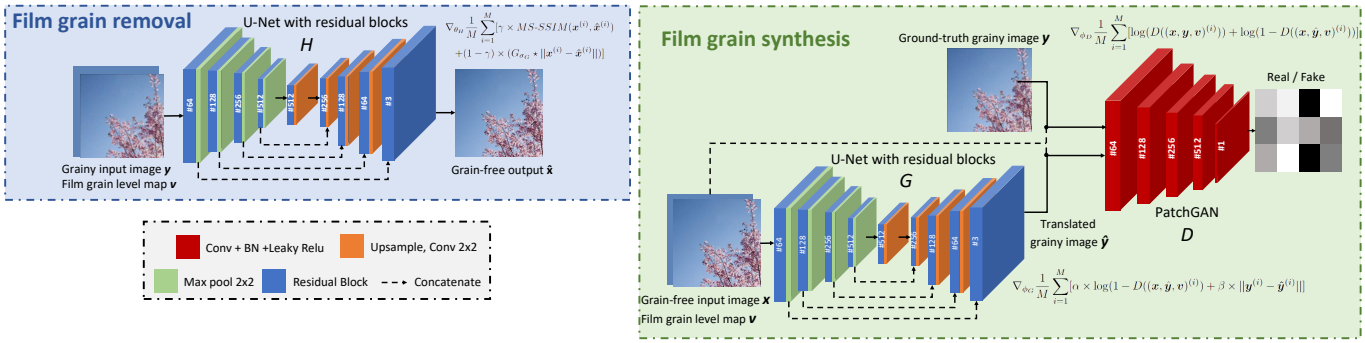


Fig. 2: The framework of our proposed film grain removal and synthesis solutions.

DRUNet can handle various noise levels via a single model. The experimental results shows that DRUNet achieves the best performance among all the state-of-the-art denoisers, both for grayscale and color images denoising.

D. Film grain synthesis

In general, viewers tend to prefer images with a certain amount of fine texture such as film grain rather than sharp images [7], [8]. Since digital video is typically noiseless and since in many cases film grain is suppressed within various filtering and/or lossy compression steps, several studies have proposed film grain synthesis solutions.

In general, film grain synthesis approaches can be classified as signal-dependent or signal-independent. Signal-independent approaches involve applying a simple addition of or multiplication by a fixed and synthesized film grain to an image, where the synthesized film grain is either a stored example of film grain obtained by scanning and digitizing examples of film grain, or by the extraction of a grain pattern from real grainy images [36]. Signal-independent approaches are simple and fast. However, their results are deterministic, hence not suitable for *synthesizing random* film grain. This results in a static film grain that can be very noticeable when applied to video sequences. However, film grain must be further blended according to the underlying signal in order to produce more realistic and pleasant visual appearance. One can classify signal-dependent film grain synthesis approaches into three main categories: mathematical-based models [37], [38], patch-based models [39] and parametric models based on texture statistics [36].

Mathematical-based models assume the presence of a pair of images, with and without film grain. In [37] and [38], higher order statistics are computed and used for noise parameter estimation and generation. However, the grain-free version of the image is not always known especially in real-world scenarios like streaming. In [39], a patch-based model was proposed. It consists of a non-parametric method for sample-wise texture synthesis, where the texture synthesis process grows a new image outward from an initial seed, one sample at a time. To synthesize a single sample, first, regions in the sample image with small perceptual distance to the single sample's neighborhood are gathered. The distance metric used to measure similarity between samples is the normalized Sum of Squared Differences (SSD). One of the regions is randomly selected and its center is used as the new synthesized sample in the context of an MRF. As for parametric models based on texture statistics, in [36] an adaptation of the parametric texture model approach [40] was adopted for film grain synthesis. First, the grain template image is decomposed into a steerable pyramid, a linear, multi-scale and multi-orientation image transform. Each scale and orientation of the pyramid are analyzed with respect to several statistical texture features including minimum and maximum gray values and correlation of sub-bands. The synthesis starts with random noise which ensures high spatio-temporal variations. The algorithm produces synthetic grain which matches the template very well while the random noise-based approach inherently provides superb spatial and temporal variations.

Based on two major and most advanced video coding standards, H.266/VVC and AV1, film grain synthesis methods have also been experimented at the decoder side, using the H.266/VVC and AV1 reference software implementations. For example, in the context of H.266/VVC, to restore the film grain in the compressed video, a frequency filtering solution to parameterize and re-synthesize film grain can be used [4], [15]. It is based on a low-pass filtering applied to the normalized Gaussian noise in the frequency domain. A film grain pattern is synthesized using a pair of cut-off frequencies, representing horizontal high cut-off frequency and vertical high cut-off frequency, which in turn characterize the film grain pattern (film grain look, shape, size, etc.). After the film grain pattern is obtained, it is scaled to the appropriate level using a stepwise scaling function which takes into account the characteristics of the underlying image. Afterwards, the film grain pattern is blended to the image by using additive blending. Likewise, in context of AV1, Norkin *et al.* [3] propose to model the film grain pattern with an autoregressive (AR) model. Since film grain strength can vary with the underlying image intensity, they have proposed to reconstruct it by multiplying two terms, the film grain pattern generated by the AR model and a piece-wise linear scaling function that scales film grain to the appropriate level before the result is added to the decoded image.

Another autoregression approach is presented in [20] where a 3D AR model is used to model film grain considering the 2D spatial correlation and the 1D spectral correlation. Instead of scaling the generated film grain pattern as in the previous

methods, the white signal used as a starting point for film grain generation is scaled. Similarly, in [5], film grain is modeled by an AR model. Newson *et al.* [2] proposed a stochastic model that approximates the physical reality of the film grain and designed a resolution-free rendering algorithm to simulate realistic film grain for any digital input image. This approach will be further detailed in subsection IV-A1.

Autoregressive models as well as frequency filtering-based methods enable the synthesis of a wide range of film grain patterns adapted to the content. In both cases, film grain is first analyzed and modeled at the encoder side with some parameters that are sent as metadata to the decoder for synthesis. The analysis stage requires a pair of samples with and without film grain and is performed only on the smooth / homogeneous regions because edges and texture can affect estimation of the film grain strength and pattern. Filtering and edge detection operations are performed for this. The synthesis step is also performed in two steps with film grain generated first and then scaled to the content using step- or piece-wise scaling functions.

Considering this and driven by the ability of generative models [41] to generate realistic images, we propose to use a cGAN which thanks to its high modeling capacity learns a mapping function between grain-free and grainy images and thanks to the conditioning on the input image, the generated film grain is content-adapted. The cGAN choice is further motivated and inspired by the work in [42] where it was used for learning distortion generation. The main contributions of our work can be summarized as follows:

- The first deep learning solutions for film grain removal and synthesis;
- Flexible deep learning-based film grain filtering and synthesis thanks to controllable intensity level;
- Content-adaptive and perceptually pleasant film grain synthesis.

III. Proposed solution

In this paper, film grain removal and synthesis methods are proposed as described in Figure 1. Deep learning models and CNNs have proven to be very powerful and to outperform traditional techniques in several computer vision tasks, we propose to address each of these two steps using deep learning-based models for their ability to process and model large amounts of data. Network architectures as well as loss functions are chosen based on the properties and objectives of each task, i.e., removal or synthesis. Note that both film grain synthesis and removal models can be derived for color or grayscale images, depending on whether the grain is present on luma only or on luma and chroma components. In each case, some YUV videos are used as input. However, when on luma only, the grayscale versions of the models are trained and used on the luma component of the videos. If the grain in present on both luma and chroma, a conversion from YUV to RGB is needed to train and use the color versions of the models, followed by a reverse conversion for the final output format. In both versions, the same model architecture and the same training details are kept, except for the adaptation to the input and output data (1 (gray) or 3 (color) channels).

A. Film grain synthesis

Film grain synthesis can be viewed as the translation of a given grain-free input image into a corresponding grainy output image while preserving the content. The goal is then to learn a mapping function from one input domain (grain-free images) $\boldsymbol{x}$ to another output domain (grainy images) $\boldsymbol{y}$.

$$\hat{\boldsymbol{y}} = G_{\Phi}(\boldsymbol{x}), \tag{1}$$

where G_{Φ} is the parametric function of the film grain generation model and Φ its training parameters.

A multitude of computer vision and image processing problems can also be modeled as image-to-image translation tasks including image synthesis [43], image segmentation [44], style transfer [45], image quality enhancement [46], [47], image compression [48]–[50], etc. In [51], a cGAN was proposed as a general-purpose solution to image-to-image translation tasks motivated by the following two insights: 1) instead of hand-engineering a loss function to be minimized during training that satisfies the learning objective of each image-to-image translation task, cGANs learn automatically a loss adapted to the task and data at hand. 2) unlike generative adversarial networks (GANs), cGANs learn the mapping by conditioning on an input and generating a corresponding output image. Since film grain is content-dependent and it is hard to manually design a loss function to be optimized for film grain synthesis, we propose to adopt a cGAN to solve the problem. Moreover, film grain synthesis has an artistic aspect, hence the use of a GAN where the goal is not to reproduce exactly the ground truth grainy images, but to generate realistic film grain while preserving the content.

1) Network architecture: Our proposed cGAN architecture is composed of a U-Net with residual blocks [52], [34] as generator and a PatchGAN as discriminator [51]. The U-Net architecture was originally designed to tackle biomedical image segmentation. However, it has not only revolutionized medical imaging segmentation, but also other related areas such as image-to-image translation tasks [28]. U-Net [52] is simply a U-shaped encoder-decoder with long skip connections between contraction and expansion levels, which represent its main feature. Residual blocks, on the other hand, were introduced as part of the ResNet architecture [34]. Thanks to the local skip connections within each residual block, deeper networks with better performance were designed without the drawbacks of deep neural networks such as gradient vanishing and explosion.

Combining the advantages of both U-Net and Residual blocks [53], our proposed generator consists of a five scale U-Net model with residual blocks. Each residual block includes batch normalization, ReLU and convolutional layers. In addition, the skip-connections within the residual block consist in a convolutional layer to resize the output of the shortcut path to be of the same dimension as that of the main path. Unlike generator models in traditional GAN architecture, our proposed generator does not take a sample point from a latent space as input since it simply learns to ignore noise [51]. To generate film grain at different intensity levels, the generator

is conditioned by both a grain-free input image $\boldsymbol{x}$ and a film grain level map $\boldsymbol{v}$.

$$\hat{\boldsymbol{y}} = G_{\Phi}(\boldsymbol{x}, \boldsymbol{v}), \tag{2}$$

The film grain level map $\boldsymbol{v}$ is a channel of the same dimensions as the input image, where all pixel values are equal to the film grain level of the corresponding target grainy image during training such that $\boldsymbol{v} : R^2 = \{0.010, 0.025, 0.050, 0.075, 0.100\}$. Thus, a single model is used to generate film grain at different intensity levels by tuning only the film grain level map.

The architecture of discriminator $\boldsymbol{D}$ is based on the PatchGAN architecture [51] with a 30×30 receptive field. The discriminator takes as input two pairs of images: 1) The grain-free input and the grainy ground truth image with its corresponding film grain level map, which it should classify as genuine. 2) The grain-free input and the grainy translated image (output by the generator) with its corresponding film grain level map, which it should classify as fake. PatchGAN tries to classify whether each 70×70 patch in an image is fake or real instead of providing a single probability for the entire input image. The use of PatchGAN limits the discriminator's attention to the local structure of the patch. Thus, it only penalizes the patch-scale structure and learns to model high frequencies. Similarly, the discriminator is trained to distinguish real ground truth grainy images from the ones translated by the generator conditioned by both grain-free input images and film grain level maps, such that it does not tolerate the generator to produce nearly the exact same output regardless of the input content nor the film grain intensity level. The detailed architecture of the proposed cGAN is illustrated in Figure 2.

2) Loss functions: During training, the generator aims to produce realistic grainy images, close to the ground truth ones, in order to fool the discriminator. Concurrently, the discriminator aims to correctly discern genuine grainy images from those translated by the generator. This leads the cGAN to model a conditional distribution of the target image $\boldsymbol{y}$ given both a grain-free input image $\boldsymbol{x}$ and a film grain level map $\boldsymbol{v}$, with the objective function $\mathcal{L}_{cGAN}(G, D)$ given by:

$$\begin{aligned}\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{v}}[\log(D(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{v}))] + \\ \mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{y}},\boldsymbol{v}}[\log(1 - D(\boldsymbol{x}, \hat{\boldsymbol{y}}, \boldsymbol{v})],\end{aligned} \tag{3}$$

where G tries to minimize this objective against an adversarial D that tries to maximize it, i.e., $G^* = \text{argmin}_G \text{max}_D \mathcal{L}_{cGAN}(G, D)$.

Several approaches have shown that combining the cGAN objective with a more traditional loss, such as ℓ_1 or ℓ_2 distance, yields better performance [51], [54]. In most computer vision tasks, ℓ_2 is the default loss function optimized during training. However, the latter penalizes high errors and tolerates small ones. Therefore, it assumes that the impact of noise is independent of the local characteristics of the image, while the human visual system (HVS) is more sensitive to luminance, contrast, and structure [55]. On the contrary, ℓ_1 does not over-penalize larger errors and has proven to be more efficient when the task involves image quality [56]. Accordingly, we combined the ℓ_1 distance with the cGAN objective loss to optimize our model. The ℓ_1 distance-based loss $\mathcal{L}_{L_1}$ represents the pixel difference between ground truth and translated images and is defined as:

$$\mathcal{L}_{L_1}(G) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{v}}\left[||\boldsymbol{y} - G(\boldsymbol{x}, \boldsymbol{v})||_1\right], \tag{4}$$

The generator G is then trained to minimize a pixel-to-pixel error with the cGAN objective loss. The cGAN objective loss represents feedback from the discriminator and reflects whether the latter has been tricked or not. This feedback helps the generator learn the mapping of the ground truth image distribution, and thus, controls the perceptual quality. On the other hand, the discriminator D is trained to distinguish between grainy ground truth and grainy translated images by maximizing the cGAN objective loss. This leads the final objective function to be defined as:

$$G^* = \text{argmin}_G \text{max}_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L_1}(G), \tag{5}$$

λ is a weighting factor that controls the contribution of the $\mathcal{L}_{L_1}$ loss in the training process of G.

B. Film grain removal

The film grain removal task can as well be modeled as an image-to-image translation task where the goal is to learn a mapping from one input domain (grainy images) $\boldsymbol{y}$ to another output domain (grain-free images) $\boldsymbol{x}$. We propose, in this paper, two configurations of the same model, a blind version and a non-blind version. The blind version takes as input only the grainy image $\boldsymbol{y}$ while the non-blind version is provided with a grainy image $\boldsymbol{y}$ and its corresponding film grain level map $\boldsymbol{v}$ as input.

$$\begin{cases} \hat{\boldsymbol{x}} = H_{\theta_1}(\boldsymbol{y}, \boldsymbol{v}) & \text{non-blind} \\ \hat{\boldsymbol{x}} = H_{\theta_2}(\boldsymbol{y}) & \text{blind} \end{cases} \tag{6}$$

where H_{θ_1} is the parametric function of the non-blind film grain removal model and θ_1 its training parameters and H_{θ_2} is the parametric function of the blind film grain removal model and θ_2 its training parameters. $\boldsymbol{v}$ is the corresponding film grain level map of the grainy input image $\boldsymbol{y}$. Note that only the encoder-decoder architecture from the proposed cGAN is adopted to solve film grain filtering task but with different inputs and outputs as illustrated in Figure 2.

1) Loss functions: Unlike the film grain synthesis task, for which it is difficult to manually design a loss to be minimized during training, the film grain removal task can be learned by simply minimizing a pixel-to-pixel difference such as ℓ_1 and/or a perceptual quality measure such as the structural similarity index (SSIM) [55]. The film grain removal task consists in learning to properly filter film grain and restore as close as possible the grain-free ground truth image without introducing any additional distortions to the content such as: loss of detail, change in brightness or color shift. In order to fulfill all these requirements, we have opted for a weighted sum of a pixel-to-pixel loss $\mathcal{L}_{L_1}$ and a perceptual loss $\mathcal{L}_{MS-SSIM}$ as in [56] for training our model. Authors in [56] were the first to propose a mix loss function that combines ℓ_1 and the multi scale structural similarity index (MS-SSIM) [57] for training deep

learning models to solve multiple image processing problems including denoising and demosaicking, super-resolution and blocking artifacts removal. In all tasks, it has been proved that MS-SSIM helps preserve the contrast in high frequency regions while ℓ_1 helps preserve color and luminance, therefore combining them provided relatively better results both in objective and subjective evaluations. MS-SSIM is a full-reference quality metric that, for a given filtered image $\hat{\boldsymbol{x}}$ and a corresponding reference image $\boldsymbol{x}$, is defined as:

$$MS\text{-}SSIM(\boldsymbol{x},\hat{\boldsymbol{x}}) = l_M^{\alpha}(\boldsymbol{x},\hat{\boldsymbol{x}}) \prod_{j=1}^{M} cs_j^{\beta_j}(\boldsymbol{x},\hat{\boldsymbol{x}}) \tag{7}$$

$$l(\boldsymbol{x},\hat{\boldsymbol{x}}) = \frac{2\mu_{\boldsymbol{x}}\mu_{\hat{\boldsymbol{x}}} + C_1}{\mu_{\boldsymbol{x}}^2 + \mu_{\hat{\boldsymbol{x}}}^2 + C_1}, cs(\boldsymbol{x},\hat{\boldsymbol{x}}) = \frac{2\sigma_{\boldsymbol{x}\hat{\boldsymbol{x}}} + C_2}{\sigma_{\boldsymbol{x}}^2 + \sigma_{\hat{\boldsymbol{x}}}^2 + C_2}$$

α and β_j are parameters to define the relative importance of the components and are set to $\alpha = \beta_j = 1$ as in the original paper. μ_x, $\mu_{\hat{x}}$ and σ_x, $\sigma_{\hat{x}}$ are means and variances of x and $\hat{x}$ respectively. They can be viewed as estimates of the luminance and contrast of x and $\hat{x}$, while $\sigma_{x\hat{x}}$ measures the tendency of x and $\hat{x}$ to vary together, thus indicating structural similarity. $C1$ and $C2$ are used to stabilize the division and are defined as $C1 = (k_1 L^2)$, $C2 = (k_2 L^2)$ with $k_1 = 0.01$, $k_2 = 0.03$ by default and L being the dynamic range of the pixel-values. M represents the scale number at which MS-SSIM is computed.

Optimizing a model using MS-SSIM as loss function consists in maximizing the latter which is equivalent to minimizing the following equation:

$$\mathcal{L}_{MS-SSIM} = 1 - MS\text{-}SSIM(\boldsymbol{x},\hat{\boldsymbol{x}}) \tag{8}$$

Moreover, since $\mathcal{L}_{MS-SSIM}$ propagates error at a given pixel based on its contribution to $MS\text{-}SSIM$ of the central pixel according to the filter size, $\mathcal{L}_{L_1}$ is weighted by the same Gaussian filter G_{σ_G} of size 11×11 and variance $\sigma_G = 1.5$ used in $MS\text{-}SSIM$. Therefore, our film grain removal model H is optimized by minimizing the following mix loss function:

$$\mathcal{L}_H = \gamma\, \mathcal{L}_{MS\text{-}SSIM} + (1-\gamma)\,(G_{\sigma_G} \star \mathcal{L}_{L_1}) \tag{9}$$

where γ represents a weighting factor for loss functions and is set to 0.84 and $\star$ refers to the convolution operation.

IV. Experimental results

In this section, first, the dataset used to train and evaluate our proposed models is presented, as well as the details of the training. Next, the film grain synthesis task is studied through quantitative and qualitative evaluations together with an ablation study. Then, the film grain removal task is explored in the same way, in addition to an evaluation of the filtering performance on real film grain.

A. Experimental setting

1) Dataset construction: To train our proposed models for film grain removal and synthesis, a large dataset of images was collected, including 400 images from Berkeley segmentation dataset (BSD) [58], 4744 images from Waterloo Exploration Database [59], 900 images from DIV2K dataset [60], 2650 images from Flick2K dataset [61] and $140,000$ images from the Konstanz artificially distorted image quality set (KADIS-700k) [62]. Therefore, we cover a large and diverse image space which enables the model to better generalize to unseen images. From each image, the maximum number of non-overlapping patches of size 256×256 is extracted.

In order to have pairs of clean (grain-free) and grainy images to train our models, we used the publicly available code provided by Newson *et al.* which consists in an implementation of the film grain rendering algorithm proposed in [2], mentioned in Section II. To model film grain, authors employ an inhomogeneous Boolean model [63] that imitates the analog photographic process as closely as possible. The inhomogeneous Boolean model corresponds to uniformly distributed disks using a Poisson process of variable intensity λ which determines the amount of grain with respect to the local image gray level. To render film grain, they use a Monte Carlo simulation to determine the value of each output rendered pixel. The grain rendering algorithm is modeled with a single Gaussian kernel of variance σ using a Monte Carlo simulation which performs simultaneous filtering and discretization of the film grain model.

A wide range of grain types and intensities can be generated by varying the parameters of this model. The two main parameters are the average grain radius μ_r and its standard deviation σ_r. Some bigger values of these parameters accentuates the "grain" of the rendered result. The grainy image patches are obtained by adding film grain at five different intensities by varying the average grain radius μ_r in $\{0.010, 0.025, 0.050, 0.075, 0.100\}$.

To evaluate our film grain synthesis and filtering solutions on color images, we used the CBSD68 dataset composed of 68 images of size 481×324 [64], the Kodak24 dataset composed of 24 color images of size 768×512 [65] and the McMaster dataset composed of 18 images of size 500×500 [66]. For grayscale images we used the widely used Set12 dataset.

2) Training parameters: Adam algorithm [67], [68] is used to train our models by optimizing the loss functions described in Eq. (5) for film grain synthesis and Eq. (9) for film grain removal. The learning rate is fixed at $3\mathrm{e}{-4}$ for the U-Net with residual blocks for both tasks and $1\mathrm{e}{-4}$ for the PatchGAN discriminator. Batch size is equal to 1 for film grain synthesis and 16 for film grain removal.

B. Film grain synthesis results

1) Quantitative Evaluation: For the film grain synthesis task evaluation, we have adopted both the Jensen Shannon divergence - natural scene statistics (JSD-NSS) proposed by Li-Heng *et al.* in [42] and the learned perceptual image patch similarity (LPIPS) [69] metrics. The first metric is based on natural scene statistics (NSS) models in which, given a distorted and a clean image, mean-subtracted contrast-normalized (MSCN) coefficients are computed on local spatial neighborhoods of each image and their distributions are analyzed and compared. For natural images, such distributions behave normally, while distortions of different kinds perturb this regularity [70]. The second metric measures the fidelity of the synthesized images using the L1 distance between features

extracted from AlexNet pretrained on ImageNet [71] of the synthesized and the ground-truth images.

TABLE I: Mean JSD-NSS and LPIPS on CBSD68, Kodak24 and McMaster datasets for our method (on clean images and on filtered images obtained by the blind and the non-blind film grain removal models) and VVC method.

Dataset	Intensity level	JSD-NSS				LPIPS			
		clean	non-blind	blind	VVC	clean	non-blind	blind	VVC
CBSD68	0.010	0.0003	0.0005	0.0004	0.0090	0.008	0.036	0.036	0.212
	0.025	0.0003	0.0005	0.0004	0.0086	0.012	0.039	0.038	0.211
	0.050	0.0003	0.0005	0.0005	0.0083	0.021	0.046	0.046	0.215
	0.075	0.0003	0.0005	0.0005	0.0082	0.033	0.057	0.057	0.232
	0.100	0.0003	0.0006	0.0005	0.0076	0.047	0.069	0.070	0.256
Kodak24	0.010	0.0003	0.0011	0.0007	0.0053	0.012	0.011	0.011	0.205
	0.025	0.0003	0.0011	0.0009	0.0052	0.015	0.014	0.014	0.201
	0.050	0.0003	0.0009	0.0008	0.0052	0.026	0.024	0.024	0.204
	0.075	0.0004	0.0010	0.0008	0.0049	0.041	0.038	0.038	0.227
	0.100	0.0005	0.0010	0.0009	0.0048	0.056	0.053	0.053	0.264
McMaster	0.010	0.0004	0.0008	0.0005	0.0054	0.006	0.006	0.006	0.142
	0.025	0.0003	0.0007	0.0004	0.0050	0.008	0.007	0.007	0.141
	0.050	0.0003	0.0007	0.0004	0.0050	0.014	0.013	0.013	0.014
	0.075	0.0003	0.0006	0.0003	0.0046	0.022	0.021	0.021	0.141
	0.100	0.0002	0.0004	0.0002	0.0040	0.033	0.030	0.031	0.148

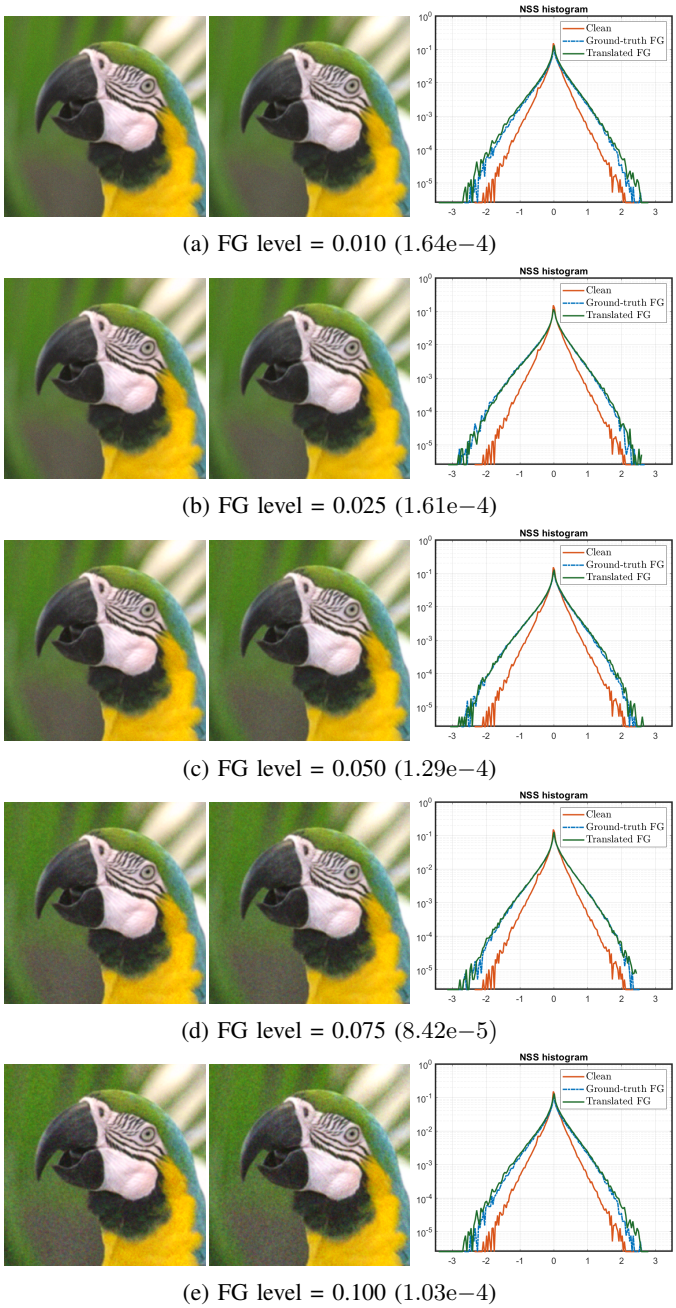
Depending on the use-case, film grain synthesis is either performed on original clean images (artistic content creation) or on filtered/decoded images (video compression). Therefore, we evaluate the results of our synthesis model on both clean and filtered images (as output by our proposed grain removal models for both blind and non-blind configurations). When using filtered images as input, it allows to approximate the full compression framework (excluding the encoding/decoding steps). In this case, we also compare the performances of our proposed solution with the implementation in VVC [4] (again excluding the encoding/decoding steps, and using the clean image as input, for a fair comparison). Results on all test-sets at all intensity levels are reported in Table I.

Mean JSD-NSS values observed for our method on clean images are very small, around 3e−4, which means that the compared distributions are close and therefore the compared images contain the same distortion type, i.e., film grain. The synthesis performed on filtered images (blind or non blind) has slightly higher scores of JSD-NSS. However they reflect the performance of both the filtering model and the synthesis model altogether, *i.e.,* any distortion introduced by the film grain removal model would negatively affect the score. Likewise, the difference between the synthesis on original images or on filtered images is also captured by LPIPS. The higher LPIPS scores observed when synthesizing grain on filtered images assess once again both the similarity of the synthesized grain as well as the perceptual quality.

One way to improve the scores when filtered images are used as input would be to further refine the models through some joint training. Separate trainings of the film grain removal and synthesis models with the ground-truth image pairs (clean and grainy) ensures that each model learns its task accurately. However, an additional joint training, where the

TABLE II: Mean JSD-NSS and LPIPS (in the form mean JSD-NSS / LPIPS) comparison between ground truth and translated grainy images in terms of intensity levels on CBSD68, Kodak24 and McMaster datasets.

Dataset	Generated FG level	Ground truth FG level		
		0.010	0.050	0.100
CBSD68	0.010	**0.0003 / 0.009**	0.0003 / 0.025	0.0008 / 0.097
	0.050	0.0004 / 0.023	**0.0003 / 0.021**	0.0006 / 0.068
	0.100	0.0007 / 0.087	0.0004 / 0.079	**0.0003 / 0.059**
Kodak24	0.010	**0.0003 / 0.012**	0.0003 / 0.036	0.0008 / 0.138
	0.050	0.0005 / 0.033	**0.0003 / 0.026**	0.0006 / 0.091
	0.100	0.0010 / 0.126	0.0007 / 0.079	**0.0005 / 0.056**
McMaster	0.010	**0.0004 / 0.006**	0.0005 / 0.018	0.0016 / 0.071
	0.050	0.0004 / 0.018	**0.0003 / 0.014**	0.0100 / 0.048
	0.100	0.0009 / 0.067	0.0005 / 0.044	**0.0002 / 0.033**



(a) FG level = 0.010 (1.64e−4)

(b) FG level = 0.025 (1.61e−4)

(c) FG level = 0.050 (1.29e−4)

(d) FG level = 0.075 (8.42e−5)

(e) FG level = 0.100 (1.03e−4)

Fig. 3: Color film grain synthesis results on image "kodim23" from Kodak24 dataset. Ground truth grainy images on left, translated grainy images on right, with corresponding NSS histograms comparison. JSD-NSS values between parenthesis.

(a) FG level = 0.010 (5.24e−4) (b) FG level = 0.025 (5.34e−4) (c) FG level = 0.050 (3.91e−4) (d) FG level = 0.075 (6.42e−4) (e) FG level = 0.100 (8.09e−4)

Fig. 4: Gray film grain synthesis results at different intensity levels on image "01" from Set12 dataset.

output of the film grain removal model serves as input to the film grain synthesis model, would allow to refine both models and get closer to the real-world scenario where the grain coding process is end-to-end.

Our method surpasses the VVC method in terms of both metrics. Note that VVC's method fails to correctly synthesize the correct grain pattern and intensity due to its analysis step. Indeed, the method of Newson *et al.* [2] generates accurate grain in the images but also tends to blur the resulting grainy image. As the grain analysis from VVC is conducted on the difference image between the grainy and the clean version of the image, which therefore not only contains the grain but also edges removed due to blurring, this leads to incorrect grain parameters and consequently a poor grain synthesis. This bias of our dataset penalized the estimated grain parameters of the VVC's method and should be investigated and corrected in the future. However, it still remains the only publicly available large dataset so far. It is also worth noting that as the intensity level increases, the difference between our model's output and the original grain became more noticeable in terms of LPIPS, although perceptually this decrease in performance is not noticable.

To further investigate the model outputs in terms of intensity level, mean JSD-NSS and LPIPS values are computed between ground truth grainy images and translated ones at three different intensity levels including 0.010, 0.050 and 0.100, where the smallest values should be observed when ground truth and generated film grain levels match. Results are summarized in Table II, where, in fact, the diagonal values are the smallest, confirming that distributions of generated and ground truth film grain at corresponding intensity level are the most similar. This demonstrates that our proposed model does not simply add some film grain but respects the film grain level specified at the input, based on which it controls the generated film grain intensity. In addition, the adopted metrics are well suited to distinguishing between intensity levels.

2) Qualitative Evaluation: Figure 3 visually compares color ground truth and translated grainy images at different intensity levels (for the color version of the synthesis model). Qualitative evaluation shows that the film grain map is not ignored but properly considered for controlling the intensity of the generated film grain. For each intensity level, NSS histograms are presented to compare distributions used for calculating the mean JSD-NSS metric. Clean, ground truth and translated grainy images distributions are plot, from which we can observe that generated film grain distribution is closer and more similar to that of the ground truth grainy image than to the clean image one, hence the small JSD-NSS values obtained in Table I. Of course, generated film grain is not the exact same one as that of the ground truth, but they are hardly distinguishable perceptually. One can also see that distributions are wider for low intensity levels and narrower for higher film grain levels.

Considering the grayscale version of the model, Figure 4 illustrates the resulting grainy images from our model at different intensity levels where from the lowest to the highest intensity level, film grain is more pronounced and accentuated.

Figure 5 illustrates the visual comparison of the synthesized images from the three different inputs (original grain-free image and blind and non-bling filtered versions from the removal model) with the ground-truth. One could observe similar grain pattern and intensity across the versions with no visible perceptual difference, proving that both the synthesis and the end-to-end version of the framework give qualitatively good performances.

(a) ground-truth (b) clean (c) non-blind (d) blind

Fig. 5: Visual comparison of film grain synthesis given different inputs: (b) a clean original image, (c) a filtered image using the non-blind film grain removal model, (d) a filtered image using the blind model.

3) Ablation Study: For a more extensive analysis of our film grain synthesis solution, an ablation study was conducted to evaluate the contribution of the different components of the model including the role of the residual blocks in the generator and the role of the discriminator. To this end, three

TABLE III: Average peak signal-to-noise ratio (PSNR) (dB) / SSIM results of different film grain removal methods with different intensity levels manually chosen to match the intensity levels of our methods, on CBSD68 and Set12 datasets.

	Level	BM3D [24], [72]	DnCNN [26]	IRCNN [28]	FFDNet [30]	DRUNet [28]	FFDNet-FG	Ours	Ours (Blind)
CBSD68	0.010	28.89 / 0.858	28.92 / 0.848	28.91 / 0.848	28.68 / 0.834	28.74 / 0.840	32.82 / 0.924	33.44 / 0.931	33.36 / 0.930
	0.025	28.87 / 0.858	28.88 / 0.847	28.18 / 0.811	28.67 / 0.834	28.74 / 0.840	32.64 / 0.920	33.25 / 0.927	33.17 / 0.927
	0.050	28.31 / 0.829	28.70 / 0.839	28.10 / 0.808	27.88 / 0.791	28.13 / 0.807	32.07 / 0.908	32.65 / 0.917	32.57 / 0.916
	0.075	27.81 / 0.804	28.40 / 0.826	27.98 / 0.804	27.82 / 0.791	27.66 / 0.781	31.35 / 0.892	31.92 / 0.903	31.85 / 0.902
	0.100	27.40 / 0.783	28.05 / 0.809	27.50 / 0.777	27.29 / 0.762	27.61 / 0.782	30.67 / 0.875	31.25 / 0.888	31.18 / 0.887
Set12	0.010	29.29 / 0.854	29.36 / 0.872	29.39 / 0.874	29.10 / 0.862	29.20 / 0.866	-	33.37 / 0.919	33.36 / 0.919
	0.025	29.25 / 0.852	29.30 / 0.870	28.33 / 0.833	29.09 / 0.861	29.17 / 0.866	-	33.06 / 0.914	33.04 / 0.914
	0.050	28.66 / 0.830	29.03 / 0.854	28.27 / 0.832	28.10 / 0.825	28.53 / 0.839	-	32.42 / 0.904	32.39 / 0.903
	0.075	28.14 / 0.811	28.48 / 0.817	28.17 / 0.828	28.04 / 0.825	27.98 / 0.815	-	31.63 / 0.890	31.62 / 0.889
	0.100	27.65 / 0.795	27.74 / 0.766	27.64 / 0.805	27.39 / 0.800	27.90 / 0.817	-	30.92 / 0.876	30.87 / 0.875

different configurations are evaluated. Config. 1: a basic U-Net architecture is optimized using an $\mathcal{L}_{L_1}$ loss. Config. 2: a cGAN based on a basic U-Net as generator and a PatchGAN as discriminator, optimized with a weighted sum of an $\mathcal{L}_{L_1}$ loss and an adversarial loss is used. Config. 3: a cGAN based on a U-Net with residual blocks as generator and a PatchGAN as discriminator, optimized with the same weighted sum as in Config. 2 is used. Config. 3 corresponds to our proposal.

TABLE IV: Comparison of mean JSD-NSS results between the three different configurations considered in the ablation study for film gain synthesis on CBSD68 dataset.

Dataset	FG level	Config. 1	Config. 2	Config. 3
CBSD68	0.010	0.0625	0.0054	**0.0003**
	0.025	0.0626	0.0055	**0.0003**
	0.050	0.0626	0.0059	**0.0003**
	0.075	0.0640	0.0063	**0.0003**
	0.100	0.0656	0.0071	**0.0003**

Quantitative and qualitative evaluations of the three configurations considered in the ablation study are reported in Table IV and Figure 6. Table IV reports mean JSD-NSS values obtained with the different configurations on CBSD68 dataset. Similar conclusions can be drawn on Kodak24 and McMaster datasets. One can observe that the highest values are obtained with Config. 1, lower values by adding a discriminator in Config. 2, and much smaller values by adopting the residual blocks in the U-Net architecture in Config. 3. These values are well interpreted by the qualitative evaluation in Figure 6, which shows the output of each configuration at intensity level 0.010. Config. 1. produces blurred results and adds no film grain; hence the large JSD-NSS values observed in Table IV and that is mainly due to the exclusive use of the $\mathcal{L}_{L_1}$ loss in the training and optimization process. Config. 2 produces grainy images thanks to the discriminator but with an unpleasant appearance due to the lack of modeling capability. While Config. 3 produces realistic film grain thanks to the use of residual blocks. As it has already been proved theoretically and in various applications of ResNets, deeper and more efficient networks can be built with residual blocks resulting in better performance and rendering in film grain synthesis.

4) Computational complexity: Since the film grain synthesis model is deployed at the decoder side, i.e., on resource-constrained devices, we measured the computational complexity of our proposed solution in terms of number of model parameters and number of floating-point operations (flops). Both numbers (32 million parameters and 114 GFLOPs for an input image of size 256x256) show that the complexity is high and that the solution needs to be optimized to be deployed on a device with limited resources. However our proposed model still remains the first attempt to solve the film grain synthesis task using deep learning, and no other deep learning models could be used for comparison in terms of complexity.[1]

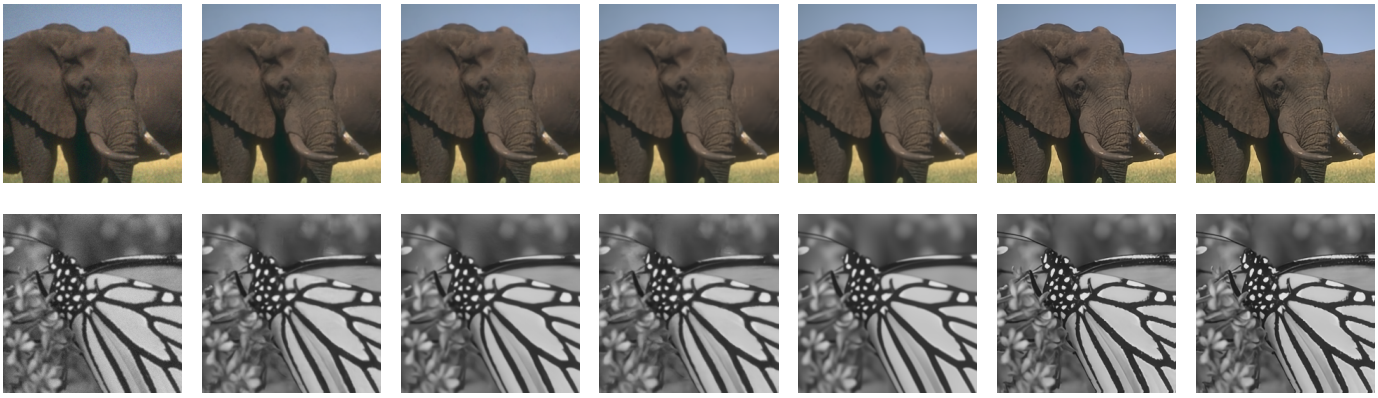
(a) Config. 1 (b) Config. 2 (c) Config. 3

Fig. 6: Color film grain synthesis with the three different configurations considered in the ablation study on image "0058" from CBSD68 dataset.

C. Film grain removal results

1) Quantitative Evaluation: For the film grain removal task, we compared our two proposed film grain removal solutions (non-blind and blind) with several state-of-the-art denoising methods, including one representative model-based method named BM3D [24] for grayscale images and CBM3D [72] for color images and four CNN-based methods that handle a wide range of noise levels, DnCNN [26], IRCNN [28], FFDNet [30] and DRUNet [28]. The blind version of DnCNN was evaluated on our test set, whereas for all other methods (BM3D, IRCNN, FFDNet and DRUNet), we manually and independently mapped the noise intensity levels they were trained on to the five grain intensity levels present in our test set. Indeed, each technique in the state-of-the-art did have its own intensity levels. The noise intensity levels adopted for each of them are as follows: BM3D: [5, 5, 10, 15, 20], IRCNN: [5, 15, 15, 15, 20], FFDNet: [5, 5, 10, 10, 15] and DRUNet: [5, 5, 10, 15, 15]. We assume that the selected noise levels

[1]These python packages [73], [74] were used to measure the computational complexity of the model.



(a) Grainy (b) DnCNN (c) IRCNN (d) FFDNet (e) DRUNet (f) Ours (g) Ours (blind)

Fig. 7: Film grain removal results of different methods on image "0060" from CBSD68 dataset (top) and image "05" from Set12 dataset (bottom) with film grain level 0.010 for our solutions and corresponding intensity levels for each method in the literature.

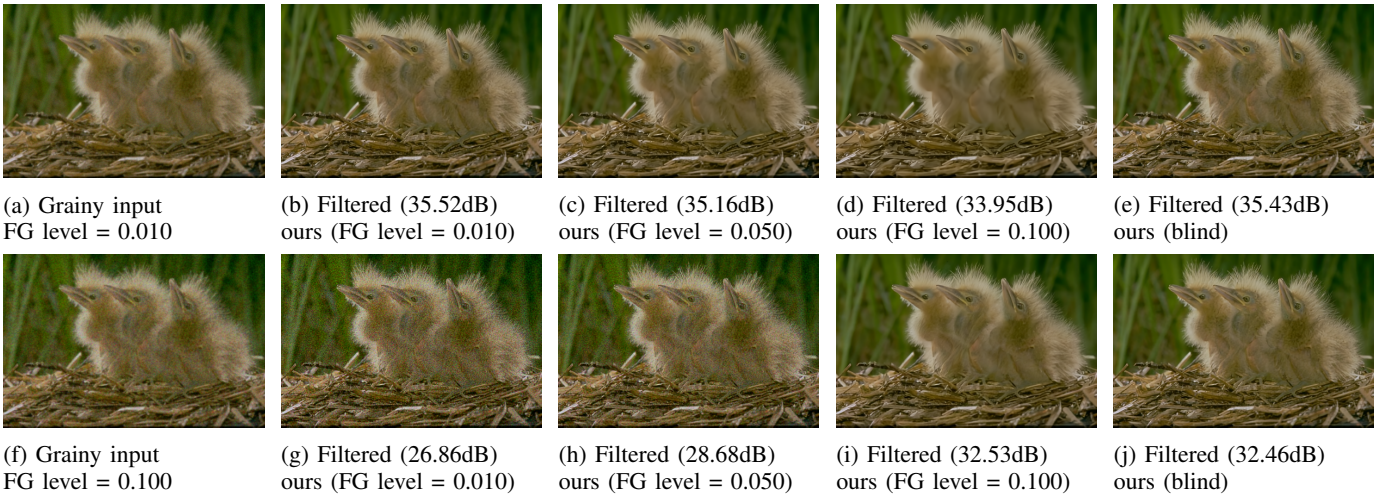
for each method represent the five grain intensity levels in our test set. Table III reports the average PSNR and SSIM results on CBSD68 and Set12 datasets at all intensity levels.

The CNN-based methods achieve almost equal performance, with DnCNN in the lead with a stable performance across all intensity levels. The latter was evaluated using its blind version, i.e., no intensity level is specified as input. This demonstrates the blind model's adaptability to the different levels and highlights the advantage of designing blind denoisers that can effectively handle test sets with varying intensity values and ranges. Slightly better results are provided by DRUnet and IRCNN thanks to the residual blocks in their architecture. Residual blocks have also been proven to provide better performance in our ablation study for both film grain removal and synthesis tasks (Tables IV and V). In addition, BM3D provides comparable performance to deep learning models.

Note that the state-of-the-art denoisers were initially trained for noise removal rather than film grain removal. Additionally, grainy images in our dataset also exhibit some slight blur due to the method in [2] used to build it. Therefore, for a fair comparison, we retrained FFDNet for the film grain removal task on our training dataset which we named FFDNet - FG. Better PSNR and SSIM results are observed with the adjusted model since it is evaluated on the same task it was trained for, but also because it is trained and tested on sets coming from the same distribution. However, it is still not as efficient as our proposed solutions. Globally, our proposed solutions outperform all tested methods on all test sets at all intensity levels. Besides, we observe that the performance of both versions (blind and non-blind) tends to deteriorate at higher intensity levels, meaning that it is more difficult to restore details in the presence of strong film grain, as it is the case with state-of-the-art denoising models. Still, the values of PSNR and SSIM are above 30 dB and close to 0.900, respectively, which is quite acceptable. We can also notice that the blind and non-blind models achieve almost similar performance, implying that both learnt to correctly match inputs to outputs.

2) Qualitative Evaluation: Figure 7 visually illustrates the results for film grain removal on grayscale and color images. For color images, all state-of-the-art denoising methods are efficient in removing film grain even if they were designed a priori for Gaussian noise removal. Nevertheless, each method introduces some loss of details and sharpness. Note that, for each method, the noise intensity level specified as input was manually set to best meet the film grain intensity level of 0.010 in our test set. As for the two versions of our proposed solution, they both recover most of the details with a slight color shift noticed with the blind configuration. For grayscale images, obviously, all methods successfully filter film grain, but some tend to introduce some distortions in flat regions including DnCNN and FFDNet. Despite our attempts to match the noise levels to our grain levels, the best trade-off between grain removal and detail preservation is not achieved by state-of-the-art denoisers. Note that the bias in our dataset may prompt our film grain removal models to learn not only de-graining, but also deblurring and edge enhancement for a better reconstruction of the original clean image. Hence, the sharper outputs.

In order to investigate the relevance of the film grain level map in the non-blind configuration, the latter performance is evaluated by providing it with a film grain level map that is less than, equal to, and greater than the film grain level of a given grainy input image. In Figure 8, a same grainy image but with different input grain level (0.010 for first row; 0.100 for second row) is used as input to both the blind and non-blind models. For the latter, 3 different level maps are considered. In each row, we present the input grainy image (first column), 3 successive outputs of the non-blind model for different level maps (columns 2 to 4) and the output of the blind model (last column). In the first row, when the film grain level map matches the input image grain level, the best PSNR is obtained with a very well filtered output as shown in Fig. 8(b). Whereas for a higher film grain level map, a lower PSNR is obtained with a tremendous loss of details and sharpness in the filtered outputs (see Fig. 8(c) and (d)). In the second row, when the



(a) Grainy input FG level = 0.010 (b) Filtered (35.52dB) ours (FG level = 0.010) (c) Filtered (35.16dB) ours (FG level = 0.050) (d) Filtered (33.95dB) ours (FG level = 0.100) (e) Filtered (35.43dB) ours (blind)

(f) Grainy input FG level = 0.100 (g) Filtered (26.86dB) ours (FG level = 0.010) (h) Filtered (28.68dB) ours (FG level = 0.050) (i) Filtered (32.53dB) ours (FG level = 0.100) (j) Filtered (32.46dB) ours (blind)

Fig. 8: Color film grain removal using blind vs. non-blind versions of our proposed solution with variable film grain level maps on image "0031" from CBSD68 dataset: first row, input film grain level = 0.01. Second row: input film grain level = 0.1.

film grain level map matches the input image grain level, the best PSNR is obtained along with the best perceptual quality as shown in Fig. 8(i). On the other hand, for a lower film grain level map, a very low PSNR is obtained, and the model is not able to filter film grain properly (see Fig. 8(g) and (h)). This proves that the film grain level map is not ignored and is indeed considered by the non-blind model. On the other hand, the blind model performs well for both high and low film grain levels of grainy inputs without any information other than the grainy image itself.

3) Ablation study: For further analysis of our film grain removal solution, an ablation study was conducted to evaluate the contribution of the different components of the network including the role of residual blocks and the mix loss function. Three different scenarii are investigated. In Config. 1: a basic U-Net architecture optimized with an $\mathcal{L}_{L_1}$ loss is proposed. In Config. 2: the same configuration is used, but with residual blocks in U-Net. Config. 3 corresponds to our proposed model: a U-Net with residual blocks optimized with a mix loss of $\mathcal{L}_{L_1}$ and $\mathcal{L}_{MS-SSIM}$ losses.

TABLE V: Average PSNR (dB) / SSIM results comparison with the three different configurations considered in the ablation study of film grain removal task on CBSD68 dataset.

	FG level	Config. 1	Config. 2	Config. 3
CBSD68	0.010	33.19 / 0.926	33.42 / 0.929	**33.44 / 0.931**
	0.025	33.01 / 0.923	33.23 / 0.926	**33.25 / 0.927**
	0.050	32.42 / 0.912	32.62 / 0.915	**32.65 / 0.917**
	0.075	31.69 / 0.896	31.88 / 0.900	**31.92 / 0.903**
	0.100	31.02 / 0.880	31.20 / 0.884	**31.25 / 0.888**

Quantitative evaluation of the three configurations considered on CBSD68 dataset is reported in Table V, from which we can observe that U-Net with residual blocks in Config. 2 achieves better performance than basic U-Net in Config. 1 thanks to the modeling capacity provided by residual blocks. Moreover, the model optimized using a mix loss function of $\mathcal{L}_{MS-SSIM}$ and $\mathcal{L}_{L_1}$ in Config. 3 achieves higher PSNR and SSIM than models optimized with only $\mathcal{L}_{L_1}$ in Config. 1 and 2. This is inline with the conclusion reached in [75] which demonstrates that quality of the results can be improved significantly with better loss functions considering the same network architecture. Moreover, combining MS-SSIM and $\mathcal{L}_{L_1}$ improved not only SSIM but also PSNR.

D. Generalization to unseen film grain

As our film grain removal model was trained on synthetic film grain generated using the same model, it is very important to test its limits on film grain coming from different sources. To do so, a frame from the CrowdRun test sequence that already contains film grain was selected to be filtered using the blind version of our solution. Figure 9 shows a frame from the sequence and some grainy cropped patches and their corresponding filtered versions output by our blind model. One can see that film grain is properly filtered with no noticed loss in details or sharpness or blurriness. This test underlines the importance of developing a blind model that allows to filter film grain coming from different sources where the scale used to measure the level is probably not the same or simply unknown.

V. Conclusion

In this paper, we are the first to propose to solve the film grain coding problem using deep learning. We have trained flexible and efficient deep learning models for film grain removal and synthesis tasks. We have shown that an encoder-decoder architecture is effective in removing film grain and provides better performance in terms of quality metrics when optimized using a combination of a pixel-to-pixel loss with a perceptual loss. As for film grain synthesis, GANs have proved to be effective in generating and synthesizing film grain with same statistics and similar look as the one in the training set. Controllable film grain intensity generation is supported

Fig. 9: Visual results of the blind film grain removal model on unseen film grain from CrowdRun sequence. On the left, the original frame with film grain and on the right, the cropped grainy patches (top) and cropped filtered patches (bottom).

by a conditional GAN, conditioned both on the input image and the intensity level, whereas flexible film grain removal is supported by a blind and a non-blind encoder-decoder architectures with competitive results. As future work, we seek to further investigate an end-to-end workflow with some joint learning of the removal and synthesis steps, and possibly with the encoding/decoding steps included. A measure of bit-saving achieved in this context, as well as subjective tests to evaluate the quality of the results would be of high interest. An additional perspective could be to explore ways of reducing the complexity of the proposed solution.

VI. Acknowledgement

This work is supported by the PHC Maghreb project Eco-VVC funded by the France Campus and the 3EMS project funded by the the “Region Bretagne” and “Images et Réseaux”, France.

References

[1] N. Salvaggio, *Basic photographic materials and processes*. Routledge, 2013.

[2] A. Newson, J. Delon, and B. Galerne, “A stochastic film grain model for resolution-independent rendering,” in *Computer Graphics Forum*, vol. 36, pp. 684–699, Wiley Online Library, 2017.

[3] A. Norkin and N. Birkbeck, “Film grain synthesis for av1 video codec,” in *2018 Data Compression Conference*, pp. 3–12, IEEE, 2018.

[4] M. Radosavljevic, E. François, E. Reinhard, W. Hamidouche, and T. Amestoy, “Implementation of film-grain technology within vvc,” in *Applications of Digital Image Processing XLIV*, vol. 11842, p. 118420D, International Society for Optics and Photonics, 2021.

[5] J. Dai, O. C. Au, C. Pang, W. Yang, and F. Zou, “Film grain noise removal and synthesis in video coding,” in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 890–893, 2010.

[6] “Netflix assumes 30% bitrate saving using film grain synthesis.” https://waveletbeam.com/index.php/news/48-netflix-film-grain-synthesis.

[7] T. Kurihara, Y. Manabe, N. Aoki, and H. Kobayashi, “Digital image improvement by adding noise: An example by a professional photographer,” *Journal of Imaging Science and Technology*, vol. 55, no. 3, pp. 30503–1, 2011.

[8] X. Wan, H. Kobayashi, and N. Aoki, “Improvement in perception of image sharpness through the addition of noise and its relationship with memory texture,” in *Human Vision and Electronic Imaging XX*, vol. 9394, p. 93941B, International Society for Optics and Photonics, 2015.

[9] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, “Deep learning on image denoising: An overview,” *Neural networks: the official journal of the International Neural Network Society*, vol. 131, pp. 251–275, 2020.

[10] B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, “Image denoising review: From classical to state-of-the-art approaches,” *Information fusion*, vol. 55, pp. 220–244, 2020.

[11] C. Gomila and A. Kobilansky, “Sei message for film grain encoding,” *ISO/IEC JTC1/SC29/WG11, ITU-T SG16 Q.6 document JVT-H022*, May 2003.

[12] A. Benyassine, E. Shlomot, H.-Y. Su, D. Massaloux, C. Lamblin, and J.-P. Petit, “Itu-t recommendation g. 729 annex b: a silence compression scheme for use with g. 729 optimized for v. 70 digital simultaneous voice and data applications,” *IEEE Communications Magazine*, vol. 35, no. 9, pp. 64–73, 1997.

[13] S. McCarthy, F. Pu, T. Lu, P. Yin, W. Husak, and T. Chen, “Illustration of the film grain characteristics sei message in hevc,” *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCTVC-AM0023*, teleconference, April 2020.

[14] M. Radosavljević, E. François, W. Hamidouche, T. Amestoy, and G. Gautier, “Enhancement of film grain parameter estimation for different intensity intervals,” *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JVET-W0072*, teleconference, July 2021.

[15] M. Sean, Y. Peng, H. Walt, P. Fangjun, L. Taoran, C. Tao, M. Radosavljević, E. François, G. R. Vijayakumar, P. Kaustubh, and K. Shireesh, “Fixed-point grain blending process for film grain characteristics sei message,” *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, document JVET-W0095*, teleconference, July 2021.

[16] M. Schlockermann, S. Wittmann, T. Wedi, and S. Kadono, “Film grain noise coding in h. 264/avc,” in *JVT 9th meeting, Doc. JVT-I034, San Diego*, 2003.

[17] P. Campisi, J. Yan, and D. Hatzinakos, “Signal-dependent film grain noise generation using homomorphic adaptive filtering,” *IEE Proceedings-Vision, Image and Signal Processing*, vol. 147, no. 3, pp. 283–287, 2000.

[18] J.-S. Lee, “Digital image enhancement and noise filtering by use of local statistics,” *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 165–168, 1980.

[19] T. M. Moldovan, S. Roth, and M. J. Black, “Denoising archival films using a learned bayesian model,” in *2006 International Conference on Image Processing*, pp. 2641–2644, IEEE, 2006.

[20] B. T. Oh, C.-C. J. Kuo, S. Sun, and S. Lei, “Film grain noise modeling in advanced video coding,” in *Visual Communications and Image Processing 2007*, vol. 6508, p. 650811, International Society for Optics and Photonics, 2007.

[21] L. Guo, O. C. Au, M. Ma, Z. Liang, and C. K. Yuk, “A multihypothesis motion-compensated temporal filter for video denoising,” in *2006 International Conference on Image Processing*, pp. 1417–1420, IEEE, 2006.

[22] I. Hwang, J. Jeong, J. Choi, and Y. Choe, "Enhanced film grain noise removal for high fidelity video coding," in *2013 International Conference on Information Science and Cloud Computing Companion*, pp. 668–674, IEEE, 2013.

[23] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," *Visual Computing for Industry, Biomedicine, and Art*, vol. 2, no. 1, pp. 1–12, 2019.

[24] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[25] P. Jain and V. Tyagi, "Spatial and frequency domain filters for restoration of noisy images," *IETE Journal of Education*, vol. 54, no. 2, pp. 108–116, 2013.

[26] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE transactions on image processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[27] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, "Attention-guided cnn for image denoising," *Neural Networks*, vol. 124, pp. 117–129, 2020.

[28] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3929–3938, 2017.

[29] J. Hou, *Research on image denoising approach based on wavelet and its statistical characteristics*. PhD thesis, Dissertation, Huazhong University of Science and Technology, 2007.

[30] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.

[31] J. Enhorn, R. Sjöberg, and P. Wennersten, "A temporal pre-filter for video coding based on bilateral filtering," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 1161–1165, IEEE, 2020.

[32] W. Per, H. Christopher, and J. Ström, "Gop-based temporal filter improvements," *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, document JVET-V0056*, teleconference, April 2021.

[33] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pp. 839–846, 1998.

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

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, pp. 448–456, PMLR, 2015.

[36] P. Schallauer and R. Mörzinger, "Film grain synthesis and its application to re-graining," in *Image Quality and System Performance III*, vol. 6059, p. 60590Z, International Society for Optics and Photonics, 2006.

[37] J. C. K. Yan and D. Hatzinakos, "Signal-dependent film grain noise removal and generation based on higher-order statistics," in *Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics*, pp. 77–81, IEEE, 1997.

[38] J. C. K. Yan, P. Campisi, and D. Hatzinakos, "Film grain noise removal and generation for color images," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, vol. 5, pp. 2957–2960, IEEE, 1998.

[39] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1033–1038, IEEE, 1999.

[40] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *International journal of computer vision*, vol. 40, no. 1, pp. 49–70, 2000.

[41] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, (Cambridge, MA, USA), p. 2672–2680, MIT Press, 2014.

[42] L.-H. Chen, C. G. Bampis, Z. Li, and A. C. Bovik, "Learning to distort images using generative adversarial networks," *IEEE Signal Processing Letters*, vol. 27, pp. 2144–2148, 2020.

[43] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *International Conference on Machine Learning*, pp. 1060–1069, PMLR, 2016.

[44] D. Li, J. Yang, K. Kreis, A. Torralba, and S. Fidler, "Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8300–8311, 2021.

[45] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

[46] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, "Deep universal generative adversarial compression artifact removal," *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 2131–2145, 2019.

[47] F. Vaccaro, M. Bertini, T. Uricchio, and A. Del Bimbo, "Fast video visual quality and resolution improvement using sr-unet," in *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1221–1229, 2021.

[48] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. V. Gool, "Generative adversarial networks for extreme learned image compression," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 221–231, 2019.

[49] B. Kang, S. Tripathi, and T. Q. Nguyen, "Generating images in compressed domain using generative adversarial networks," *IEEE Access*, vol. 8, pp. 180977–180991, 2020.

[50] T. Kaneko and T. Harada, "Blur, noise, and compression robust generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13579–13589, 2021.

[51] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[52] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[53] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[54] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

[55] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[56] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on computational imaging*, vol. 3, no. 1, pp. 47–57, 2016.

[57] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, vol. 2, pp. 1398–1402 Vol.2, 2003.

[58] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, pp. 416–423, IEEE, 2001.

[59] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2016.

[60] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[61] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 136–144, 2017.

[62] H. Lin, V. Hosu, and D. Saupe, "Kadid-10k: A large-scale artificially distorted iqa database," in *2019 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–3, IEEE, 2019.

[63] D. Stoyan, W. S. Kendall, S. N. Chiu, and J. Mecke, *Stochastic geometry and its applications*. John Wiley & Sons, 2013.

[64] S. Roth and M. J. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, p. 205, 2009.

[65] R. Franzen, "Kodak lossless true color image suite," *source: http://r0k.us/graphics/kodak*, vol. 4, no. 2, 1999.

[66] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *Journal of Electronic imaging*, vol. 20, no. 2, p. 023016, 2011.

[67] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[68] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *arXiv preprint arXiv:1904.09237*, 2019.
[69] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.
[70] A. Galdran, T. Araújo, A. M. Mendonça, and A. Campilho, "Retinal image quality assessment by mean-subtracted contrast-normalized coefficients," in *European Congress on Computational Methods in Applied Sciences and Engineering*, pp. 844–853, Springer, 2017.
[71] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.
[72] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space," in *2007 IEEE International Conference on Image Processing*, vol. 1, pp. I–313, IEEE, 2007.
[73] "Keras model profiler." https://pypi.org/project/model-profiler/.
[74] "Keras flops." https://pypi.org/project/keras-flops/.
[75] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.