Supplementary Materials: Model-blind Video Denoising Via Frame-to-frame Training

Thibaud Ehret Axel Davy Jean-Michel Morel
Gabriele Facciolo Pablo Arias
CMLA, ENS Cachan, CNRS
Université Paris-Saclay, 94235 Cachan, France
thibaud.ehret@ens-cachan.fr

To demonstrate the flexibility of the proposed model blind video denoising we apply it to sequences obtained from very different sources.

1. Examples with grayscale videos

A challenging test case of our blind denoising is old digitalized films. The difficulty with this data is that the film quality degraded gradually with time and can be phisically damaged during its manipulation and reproduction, creating several type of artifacts. The two examples shown in Figures 1, 2, and 3 are examples from footage of World War I [1]. In addition to the noise and the damaged parts of the film, there's also a strong compression that has been applied after digitalization. All of this makes modelling the noise very difficult. Yet, the proposed frame-to-frame fine-tuning strategy is still able to learn to denoise these sequences. The blind denoiser is able to remove most of what can be consider as noise while retaining most details. In Figure 2 we show a comparison with the pre-trained network (starting point of the fine-tuning) and the result of VBM3D. VBM3D receives as input the noise level σ. We tested several noise levels and chose then one the seemed best. As one can see in Figure 2 the blind denoiser keeps more details in the fields, the building or the airplane than the pre-trained network and VBM3D.

2. Examples with color videos

Our last experiment is with a video shot with a Samsung Galaxy S7. The video is shot in a low light, and processed by the camera pipeline. This means that it has been demosaicked, denoised (by a fast method running directly on the phone), among other quality enhancement algorithms, and finally compressed. The remaining noise is therefore completely distorted, being colored and non-stationary. We used a pretrained network which is a color DnCNN trained for Gaussian noise with standard deviation $\sigma = 25$. We use these hyper-parameters for the fine tuning: a learning rate of 1.10^{-4} and $N = 10$ iterations of the Adam optimizer. Figure 4 presents a crop of the video. Here again, blind denoising largely removes the artifacts left by the phone's pipeline and therefore improves the overall visual quality of the video.

[1] https://www.army.mil/

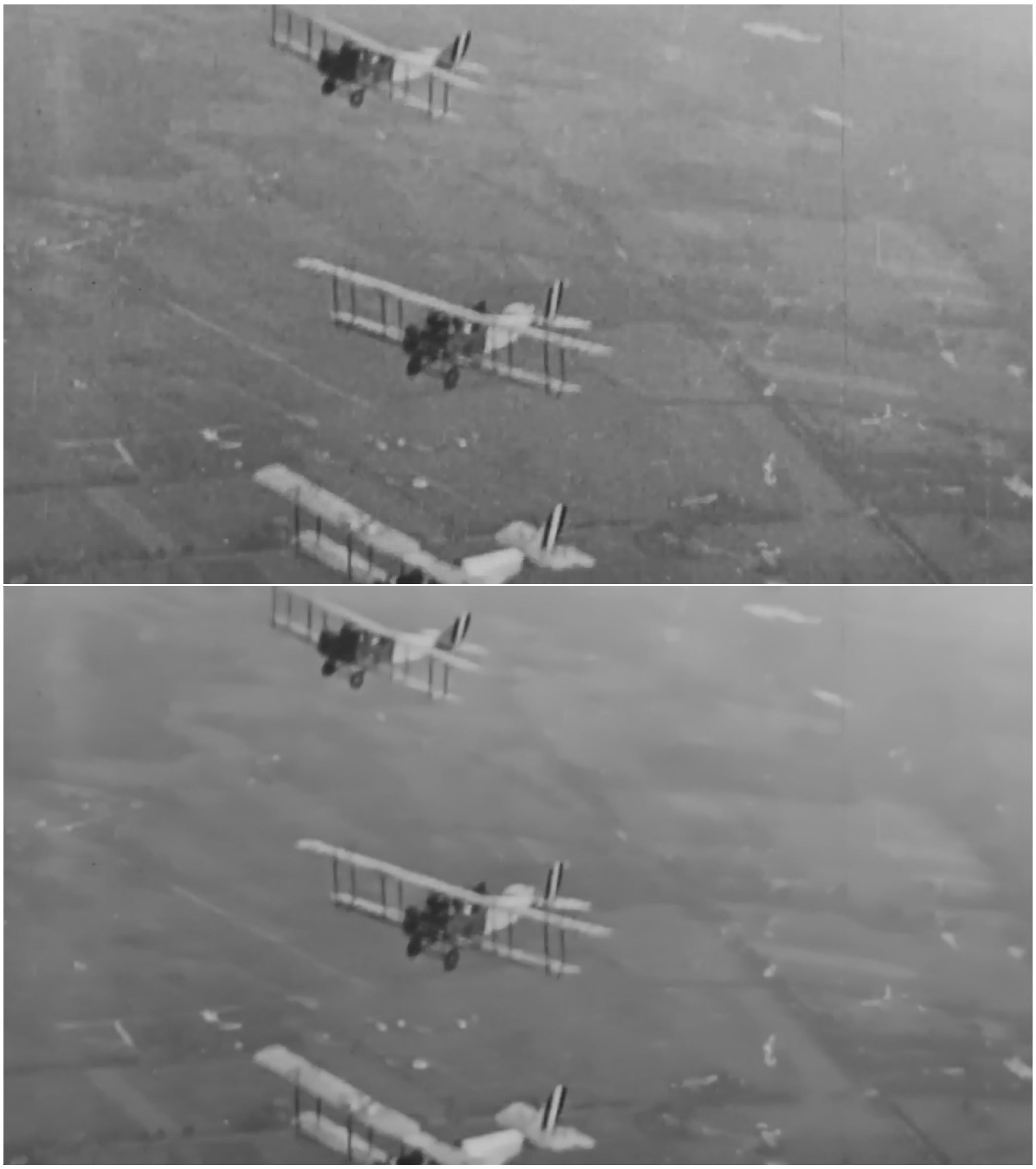

Figure 1: Blind denoising (on the right) of images coming from a video taken during World War I (original on the left). Blind denoising is useful in this case because it would be nearly impossible to recreate this type of noise to train a network.

Figure 2: Blind denoising better preserves details than methods for a predefined noise. From top to bottom, left to right: original, blind denoising, denoising with the pre-trained network and VBM3D with hand-tuned noise parameter.

Figure 3: Blind denoising (on the right) of images coming from a video taken during World War I (original on the left). Blind denoising is useful in this case because it would be nearly impossible to recreate this type of noise to train a network.

Figure 4: Example of denoised image (bottom) coming from a mobile phone (top). The results is more natural and pleasing to the eye as it doesn't have all these ugly artifacts.