Deep-Learning Based Exploitation of Eavesdropped Images

Florian Lemarchand

DOCTORAT BRETAGNE LOIRE / MATHSTIC

INSA RENNES

THÈSE DE DOCTORAT DE

L'INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE RENNES
COMUE UNIVERSITÉ BRETAGNE LOIRE

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Signal, Image, Vision

Par
Florian LEMARCHAND

Deep-Learning Based Exploitation of Eavesdropped Images

Thèse présentée et soutenue à Rennes, le 29 Septembre 2021
Unité de recherche : IETR
Thèse N° : 21ISAR 23 / D21 - 23

Rapporteurs avant soutenance :

Fan Yang	Professeure des Universités, Université de Bourgogne
Olivier Strauss	Maître de Conférences, HDR, Université de Montpellier

Composition du Jury :

Président :	William Puech	Professeur des Universités, Université de Montpellier
Examinateurs :	Fan Yang	Professeure des Universités, Université de Bourgogne
	Olivier Strauss	Maître de Conférences, HDR, Université de Montpellier
	François Berry	Professeur des Universités, Université Clermont Auvergne
	Emmanuel Cottais	Ingénieur, ANSSI
	Bart Goosens	Professeur des Universités, Ghent University
Directeur de thèse :	Maxime Pelcat	Maitre de Conférences, HDR, INSA Rennes
Encadrant :	Erwan Nogues	Ingénieur, DGA-MI

Table of Contents

Acknowledgements

Tout d'abord, je tiens à remercier mes encadrants. Merci pour la confiance et la liberté que vous m'avez accordées tout au long de ces trois années. Maxime merci d'être une mine d'idées. Cela m'a parfois donné des nœuds au cerveau mais aussi poussé à donner le meilleur. Erwan, merci de ta pertinence scientifique et de ton ouverture d'esprit. J'ai apprécié nos diverses discussions sur tous types de sujets intéressants, qu'elles soient de nature professionnelle ou non.

Merci aux membres du jury qui ont évalué mon manuscrit ainsi que ma soutenance. Merci pour votre temps et les orientations et propositions pertinentes que vous avez faites sur le travail présenté.

Merci à l'équipe IA de la DGA pour l'important travail sur ToxicAI et les discussions pertinentes sur l'interprétation des images interceptées.

Je suis reconnaissant envers Eduardo et Thomas, les stagiaires qui m'ont par leur travail aidé à développer OpenDenoising-Benchmark et NoiseBreaker, deux contributions majeures de cette thèse.

Merci à la Musique d'avoir su me proposer au jour le jour ses différentes facettes pour s'adapter à mes humeurs.

Merci aux collègues de VAADER. Ceux avec qui j'ai pu échanger, professionnellement ou personnellement. Particulièrement merci à ceux qui sont devenus des amis en partageant toutes sortes de moments que je ne saurais lister de façon exhaustive : des relectures de papiers, des pauses café, des complaintes de doctorants, des soirées ou week-end raisonnables et d'autres moins.

Merci aux occupants du "Bureau 214" qui ont rendu mes journées de travail toujours plus joyeuses avec des blagues et des surprises toutes plus rocambolesques les unes que les autres.

J'aimerais aussi ne pas remercier la Covid-19 qui a fait exploser en vol mes ambitions de collaborations internationales et de voyages scientifiques. Présenter des articles à distance fut un réel non-plaisir.

Merci à ma famille et à mes amis qui rendent ma vie heureuse à chaque instant. La réussite de cette thèse n'aurait pas été possible sans les moments ressourçant que vous me faites vivre.

Enfin, merci à Elisa qui m'a chéri tout au long de ces trois années, dans les moments de réussite comme dans ceux de doutes. Merci d'avoir été à mes côtés et des choix que tu as faits pour me permettre de réussir ce défi, souvent à tes dépens.

CHAPTER 1

Introduction

Chapter Contents

1.1 Context: Novel Challenges of Side-Channel Analysis

The recent trend of processing is to make digital data available anytime anywhere, creating new confidentiality threats. In particular, when considering highly confidential data, where printed information was kept physically protected and was accessible only to authorized persons, the data is nowadays digital. It is exchanged and consulted using Information Processing Equipments (IPEs) and their according Video Display Units (VDUs). While the main security efforts focus today on the network side of systems, there exist other security threats.

A side-channel corresponds to an unintended data path in opposition to the legacy channel. In particular, Electro Magnetic (EM) side-channels are due to fields emitted by video cables and connectors when their inner voltage changes. These side-channels are dangerous because they spread un-ciphered data outside the physical system. These emissions may be correlated to a confidential information. Therefore, an attacker receiving the signal and knowing the data encoding mechanism may access illegally the original information handled by the IPE. Under these conditions, the attacker can reconstruct the image displayed on the attacked VDU connected to the IPE. It has been shown that the content of screen can be recontructed from

tens of meters [DSV20a]. Since the pionner exploits [Van85], a lot of work has been published on the reconstruction of images from EM side-channel emanations, and this research area is still dynamic [Lav+21]. But until today, the work conducted on state of the art has mainly focused on enhancing the reconstruction from a signal processing point of view.

Recently, the image processing domain have been revolutionnized by Machine Learning (ML) and especially Deep Learning (DL). These algorithms learning tasks from data, have overpassed the performances of state of the art expert algorithms on several Computer Vision (CV) tasks. In particular, one of the tasks that have benefited from learning algorithms is the semantic classification of image content. In this task, state of the art algorithms are nowadays capable of automating interpretation of images. However, these interpretation methods are designed for natural images without corruption. Image restoration is the task concerned by removing corruptions from images. Image restoration has also benefited a lot from learning algorithms. In fact, recent algorithms outperform the former state of art expert based algorithms both on objective and subjective performances. However, the state of the art algorithms for image restoration focus on well-behaved corruptions, following parametric distribution, ruled by only a few parameters.

The images reconstructed from EM emanations are highly corrupted due to several reasons. First there is a data loss and interferences inherent to the EM emission/reception process, similarly to a radio-frequency channel in a data wireless communication. In addition, there are also defects in the reconstruction synchonization, when passing from 1D signal to an image. Finally, the defects of the hardware of the interception system introduce errors. Arise three questions that we study in this manuscript: **What is the type of corruption generated by EM emanations reconstruction? Can it be reduced to a composition of parametric distribution noises? How do current DL methods for image restoration perform on eavesdropped image?**

The audit of processing systems handling confidential data, is currently executed by experts. An expert, once the interception system in place, assesses the compromise of the audited equipment, using her/his experience. This audit protocol is time consuming and subject to human perception. Here comes another question we study in this manuscript: **Can DL be used to automate semantics retrieval from eavesdropped images?**

1.2 Objectives and Contributions of this Thesis

The main objective of this thesis is to analyze how DL techniques can be applied to eavesdropped images and if it can automate the interpretation of these images. Even though EM emanation reconstruction and DL image processing are two extensely studied domains, their concomitant use is a recent advance.

After the review of the seminal work of both eavesdropping and noisy image interpretation, we propose a set of experiments and contributions to study the feasability of automatic eavesdropping exploitation.

Three main contributions are proposed in this document. They are among the first studies of EM emanations from an image processing point of view. Accordingly, this thesis is one of the first attempt to apply DL for eavedropping image exploitation automation. The three main contributions of this thesis are briefly presented below.

1.2.1 Benchmarking of Image Restoration Algorithms

Fairly comparing denoisers has become complicated with the use of learning algorithms. In fact, algorithms may be trained and evaluated on different sets of data making the comparison unfair without retraining. This is a problem when searching for state of the art solutions for a new problem. A proposed tool, dubbed OpenDenoising, benchmarks image denoisers and aims at comparing methods on a common ground in terms of datasets, training parameters and evaluation metrics. Supporting several languages and learning frameworks, OpenDenoising is also extensible and open-source.

The second contribution of the chapter is a comparative study of image restoration in the case of a complex noise source. The experiments of that comparative study are used as a case study for the proposed benchmarking tool. Several conclusions are drawn from the comparative study. First, there is a difference in terms of performance between expert-based and learning-based methods which rises as the complexity of the noise grows. Second, the ranking of methods is strongly impacted by the nature of the noises. These results show that restoring an image from a complex noise is not universally solved by a single method and that choosing a denoiser requires automated testing.

This chapter has led to the public release of the OpenDenoising benchmark tool [1]. This work have been presented in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) in 2020 [Lem+20c].

1. https://github.com/opendenoising/opendenoising-benchmark

1.2.2 Mixture Noise Denoising Using a Gradual Strategy

Preliminary chapters will suggest that the corruption generated by the eavesdropping process is a sequential mixture of several primary corruptions. Accordingly, Chapter 5 introduces a gradual image denoising strategy called NoiseBreaker. NoiseBreaker iteratively detects the image dominating noise using a trained classifier with an accuracy of 93% and 91% for grayscale and RGB samples, respectively. Under the assumption of grayscale sequential noise mixtures, NoiseBreaker performs 0.95dB under the supervised Multi-level Wavelet Convolutional Neural Network (MWCNN) denoiser without being trained on any mixture noise. Neither the classifier nor the denoisers are exposed to mixture noise during training. NoiseBreaker operates 2dB over the gradual denoising of [LSJ20] and 5dB over the state of the art self-supervised denoiser *Noise2Void*. When using RGB samples, NoiseBreaker operates 5dB over [LSJ20] while *Noise2Void* underperforms. Moreover, this paper demonstrates that making noise analysis to guide the denoising is not only efficient on noise type, but also on noise intensity.

This manuscript has demonstrated the practicality of NoiseBreaker on six different synthetic noise mixtures. Nevertheless, the NoiseBreaker version proposed in the chapter has not permited to conclude on the efficiency of the method to restore eavesdropped images. Consequently, the hypothesis of the sequential composition of the eavesdropping corruption is not validated.

This work has lead to a presentaion in the IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP) in 2020 [Lem+20a].

1.2.3 Direct Interpretation of Eavesdropped Images

This work is presented in the last contribution chapter of the manuscript. The beginning of the manuscript studies the applicability of DL to restore eavesdropped images. This last work focuses on interpretation and studies its automation on text images. The introduction of deep learning in an EM side-channel attack is studied. The proposed method, called TxicAI, uses Mask R-CNN as denoiser and it automatically recovers more than 57% of characters, present in the test set. In comparison, the best denoising/Optical Character Recognition (OCR) pair retrieves 42% of characters. The proposal is software-based, and runs on the host computer of an off-the-shelf Software-Defined Radio (SDR) platform.

This chapter has led to the public release of two datasets of eavesdropped samples:

- a dataset of eavesdropped images made of text characters and their references [2],
- a dataset of eavesdropped natural images, based on Berkeley Segmentation Dataset (BSD), dubbed Natural Interception Dataset (NID) [3].

This work was presented in Conference on Artifical Intelligence for Defense (CAID), in 2019 [Lem+19] and in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) in 2020 [Lem+20b].

1.3 Outline

Chapter 2 introduces what is eavesdropping and in what it is a threat to the confidentiality of IPEs using VDUs. The characteristics of eavesdropping are studied. In particular, the link is made between the corruptions found in the images and their physical origin. Finally, arguments are given that motivate the study of image processing to enhance the interpretation of eavesdropped images.

Chapter 3 gives a definition of noise in an image. Main image noise distributions are detailed which opens for the introduction of more complicated compositions of these distributions. The chapter then reviews the state of the art methods for image restoration and interpretation. A distinction is made between expert and learning based algorithms. The performance step made by these latters is discussed. Evaluation and optimisation metrics as well as datasets are presented for both image quality and classification assessement. Finally, the terminology of learning algorithms, as well as discussions on their strengths and open issues for our case study, are proposed.

Chapter 4 proposes an extensible and open-source tool to benchmark fairly denoising algorithms. Then, a comparative study of state of the art denoisers is discussed. This comparative study also gives first answers on the removal of eavesdropping noise from images.

Chapter 5 presents NoiseBreaker, a gradual image denoising method that adresses the removal of sequential mixture noise. Related work is exposed before detailing the proposed method that leverages an iterative strategy. The dominant noise is detected before being removed. The method is compared to state of the art before being discussed in an ablation study.

Chapter 6 adresses the direct interpretation of eavesdropped images by proposing ToxicAI. Related work is overviewed before ToxicAI architecture is defined. The building of the open-source custom dataset of eavesdropped screens, displaying text, used to trained ToxicAI is

2. https://github.com/opendenoising/interception_dataset
3. https://github.com/opendenoising/NID

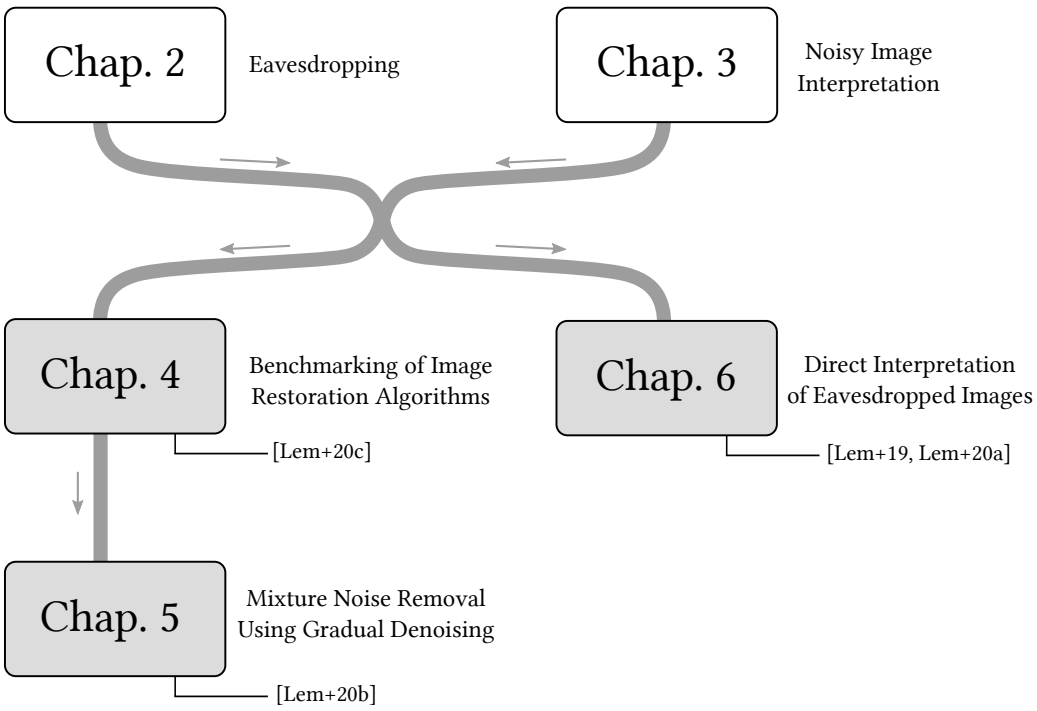


Figure 1.1 – Outline of the document structure. State-of-the art chapters are displayed in white while contribution chapters are in gray.

detailed. Then, experiments are conducted on the proposal and the results compared to the state of the art. Finally, an open-source dataset of eavesdropped natural images is proposed to extend ToxicAI.

Chapter 7 concludes the manuscript. First, the questions addressed in the document are reminded and the contributions are resumed. Opened by the principles proposed in this document, research directions for the future of eavesdropped image interpretation are proposed.

Figure 1.1 illustrates the organisation of this document. This figure highlights the links between the chapters introduced here-above.

Part I

Open Challenges in Eavesdropped Image Information Extraction

CHAPTER 2

Eavesdropping

Chapter Contents

2.1 Introduction

In the last decades, Information Processing Equipments (IPEs) have become essential in professional everyday life. This democratization has opened new threats on data security. The purpose of this chapter is to give the fundamentals of Information System Security (ISS) and its specific application to the side-channel emanations of Video Display Units (VDUs).

A standard formalization of the framework for security of IPEs is given by the Confidentiality Integrity Accessibility (CIA) triad depicted on Figure 2.1. According to the CIA model, ISS must consider three points, working together. *Confidentiality* specifies that the information is accessible only by authorized persons. *Integrity* means the system handling data should be reliable and accurate. *Availability* implies that the data is available when it is needed.

When it comes to transmit or handle sensitive data that may be received by anyone, encryption with ciphering algorithms is used to ensure the system security. This especially ap-

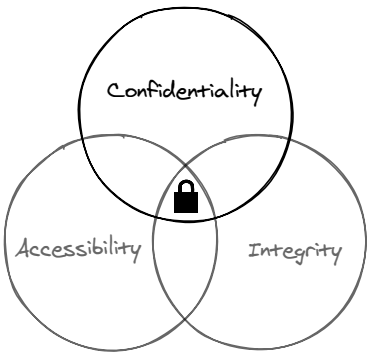


Figure 2.1 – The three components of the CIA Triad represented as an Euler diagram: *Confidentiality*, *Integrity*, *Accessibility*. Electro Magnetic (EM) side-channels compromise confidentiality.

plies to wireless communications. Therefore, system or information can be considered as vulnerable when any sensitive data (e.g. classified data) is handled before encryption or after decryption. This is particularly the case when sensitive information is handled by the end user on his device after decryption or before encryption.

Thus, the use of an encryption scheme on the *legacy channels* (see Figure 2.2) is mandatory. This makes the information non-interpretable even when eavesdropped by an attacker. Nevertheless, the same information may be emitted on a *side-channel* without encryption. The attacks then focus on any type of sensitive information restoration bypassing the protection provided by the ciphering schemes. A side-channel is defined by the presence of an information on an illegitimate channel, potentially leading to secret data being compromised. An attacker could recover the sensitive data, supposed to be transmitted by the legacy channel, using the side-channel (as depicted in Figure 2.2). The fact of listening to a side-channel is called *eavesdropping*. There exist two types of side-channels [Lav+21]. The first type, refered to as *software side-channels*, is based on hardware weaknesses. These side-channels remain into the device and require a physical access to the device to be used [Ge+17; Koc+18]. The other type, called *emanation side-channel*, is more malicious since it is *non-intrusive*. This side-channel is due to physical incidents that deviate the information of the original path to an unintended path. In particular, we are interested here in Electro Magnetic (EM) side-channel coming from screen displays. EM fields may be emited by video cables and connectors because of the voltage transitions. Such an EM field is correlated with the transmitted information, and a third-party leveraging signal processing may then recover the sensitive information.

Any electronic equipement creates emanations because of its conception and structure. These emanations must be measured and verification must be done so that no vulnerability

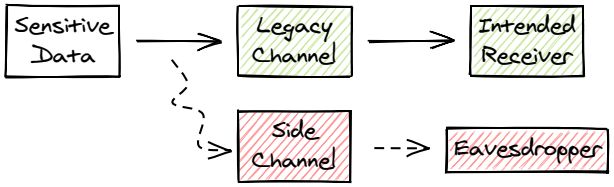


Figure 2.2 – An eavesdropper accesses sensitive data taking advantage of a side-channel.

leads to security failures. This is the area of the NACSIM report [Nat82]. In this report, the NSA defines TEMPEST and specifies the terms of *red* and *black* signals [1]. A *red signal* is an unencrypted signal that should be protected. For such a signal, protection measures should be used such as shielding or physical distancing with wires to prevent coupling. *Black signal* on the other hand requires no effort. It is supposed not to carry compromising information because of encryption that makes it unintelligible.

Countermeasures should be taken to prevent sensitive data to be intercepted using emanation side-channels. The most used countermeasure is *shielding*. In [Lav+21], Lavaud et al. detail a list of other countermeasures like changing the data-stream so that assumptions on signal properties are not respected anymore, or the use of *jamming* [SA10] to hide leakages. If such measures are not used, the last resort is *zoning*. An *air-gap* should be respected to make the interception theoretically impossible. That air gap is the minimal physical distance that makes impossible an external access to sensitive data using a side-channel. The fact of accessing information from outside an organisation is called *air gap bridging*. The definition of the air gap relies on the technology used for eavesdropping. It should be chosen according to state of the art interception methods.

The following of that chapter presents the keys that make eavesdropping images from EM side-channels possible in Section 2.2. Section 2.3 details the specificity of eavesdropped images that will be the major input data for the following of this thesis. Finally, in Section 2.4, perspectives on using image restoration to go further in the interpretation of eavesdropped images are presented.

1. See https://www.ssi.gouv.fr/uploads/IMG/pdf/II300_tempest_anssi.pdf

2.2 From Side-Channel Emanations to Image Eavesdropping

All electronic devices produce EM emanations that not only interfere with radio devices but also compromise the data handled by IPEs. A third party may perform a side-channel analysis and recover the original information, hence compromising the system privacy. This third-party obtaining access to potential sensitive data breaks the *confidentiality* aspect of the CIA triad. Screens are especially sensitive since they display information, potentially red, to users. They are often the weakest link with signal being encrypted everywhere else in the transmission pipeline. Sensitive data is exposed in a fully intelligible format and a side-channel conducted at that point could be compromising.

Pioneering work of the domain focused on Cathode Ray Tube (CRT) screens and analog signals. Van Eck *et al.* [Van85] published the first technical reports revealing how involuntary emissions originating from the electronic of VDUs can be exploited to compromise data. He mentioned that the video signal at that time does not contains synchronization information required to time the beginning of an image in the 1D eavesdropped flow. However, Van Eck proposes a simple electronic extension that fixes that synchronization issue, making the exploit easier and achievable for any electronic amateur. One would have though the transition to digital video signals to solve the issue because of smaller voltage. However, studies extend the eavesdropping exploit, using an EM side-channel attack, to digital signals and embedded circuits. Kuhn published on compromising emanations of Liquid Crystal Display (LCD) screens [Kuh13]. Other types of systems have been attacked. Vuagnoux *et al.* [VP09] extend the principle of EM side-channel attack to capture data from keyboards and, Hayashi *et al.* present interception methods based on Software-Defined Radio (SDR) targeting laptops, tablets [Hay+14] and smartphones [Hay+17].

In the meantime, one should also note that the attacker's profile is taking on a new dimension with the increased performance of SDR [Mit]. With recent advances in radio equipment, an attacker can leverage advanced signal processing to further stretch the limits of the side-channel attacks using EM emanations [Gen+18]. The use of SDR increases the surface of attack from military organizations to hackers. It also opens up new post-processing opportunities that improve attack characteristics. De Meulemeester *et al.* [De +18] leverage SDR to enhance the performance of the attack and automatically find the structure of the captured data. By retrieving the synchronization parameters of the targeted information system, the captured EM

signal can be transformed from a vector to a raster image, reconstructing the 2-dimensional sensitive visual information.

Recent works of De Meulesmeester [De 21] provide deep details on the eavesdropping process. It focuses mainly on the received radio signal and proposes several techniques to enhance the quality of the attack [DSV20a] by signal processing algorithms. Today, the state of the art research documents well the analysis of the EM spectrum to detect emanations. The reconstruction of eavesdropped screens is also documented as well as techniques to enhance their quality such as averaging.

Meanwhile, advances in Machine Learning (ML) have opened the scope of automated eavesdropped data interpretations. With the concomitant rise of powerful Graphics Processing Units (GPUs) and deep neural networks, an attacker can extract patterns or even the full structured content of the intercepted data with a high degree of confidence and a limited execution time. Previous work on the domain have mainly focused on processing the eavesdropped signal using SDRs and Central Processing Units (CPUs). In this work, we mostly use image and GPU processing.

This thesis focuses on the interpretation of eavesdropped samples from an image processing point of view. We thus present briefly the image formation pipeline and the key points that lead to the corruptions we address. We redirect the reader to the recent thesis of Pieterjan De Meulesmeester [De 21] for deeper details on the eavesdropping process.

2.3 Eavesdropped Image Characteristics

Connectors and cables are the emission antennas that lead to side-channel emanations. They connect an IPE and its VDU which constitute the *emission block*, left part of Figure 2.4. The video signal is transmitted through cable using different protocols like Video Graphics Array (VGA), High-Definition Multimedia Interface (HDMI) or Digital Visual Interface (DVI). The transmitted signal is not encrypted. It respects the protocol defined by the standards [VES15]. The voltage changes in the connector or cables generate EM emanations.

The reception block (right part of Figure 2.4) consists of a reception antenna, an SDR and a computer that hosts the signal processing required for the raster. The distance between the defective element and the reception antenna is noted d. The SDR receives an analog signal and transforms it to digital. New SDR systems also enable implementing signal processing. The raster implemented in the host computer use different processing to obtain and display an intelligible images.

Figure 2.3 – Different video connectors that may lead to compromising emanations. Images from Pierre-Michel Ricordel and Emmanuel Duponchelle [RD18].

The first step applied to the signal catched by the antenna is a demodulation at a given carrier frequency. This carrier frequency is chosen so as to maximise the quality of the restored image. Once the radio samples are received, an Amplitude Modulated (AM) detection process is performed to retrieved the compromised information as a 1-D vector. In some rare cases, a Frequency Modulated (FM) detection is done [DSV20b] to improve the restored signal quality.

As the compromised information is a video signal, several characteristics can be retrieved with an appropriate statistical analysis of the signal. The line frequency f_{line} and the frame frequency f_{frame} can be found. The next step is called *rastering*. It consists in re-arranging the 1D signal to 2D images according to the retrieved video characteristics. f_{line} and f_{frame} are directly linked to the screen resolution as well as the pixel frequency f_{pixel}.

From a signal improvement point of view, there are several techniques that can be used. The most efficient are the multi-antenna reception and the signal averaging. As the quality of the restored image is directly linked to the radiolink characteristics and the SDR receiver performance, one can use two or more antennas to produce a beamformer focused on the target [DSV20a]. On the other hand, as the target signal is a video, the same (or close) image is repeated at f_{frame} rate. Therefore, it can be averaged over time to improve the image quality. Finally, the captured signal is interpreted as a grayscale signal since all the colour components leak at the same time, summing up together. Recent work shows the trials to identify the colour components individually [DSV20c] but with no improvement of the image quality itself.

The quality of eavesdropped images highly rely on the interception conditions. Nevertheless even with perfect conditions, the images contain corruptions and do not represent directly the information displayed on the attacked screen. In the literature, the corruptions are well described but from a signal point of view [De 21]. We choose to present the corruptions from an

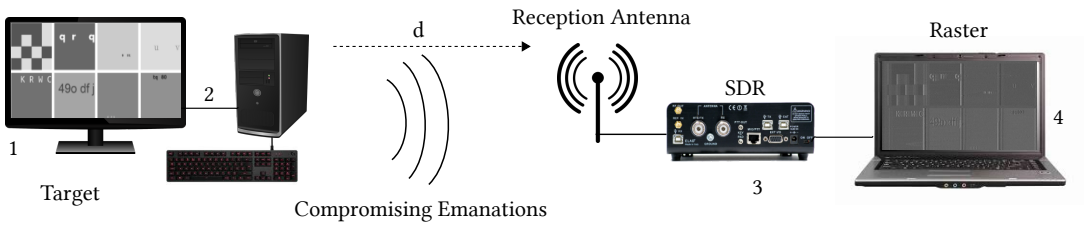


Figure 2.4 – Experimental setup: the attacked system includes an eavesdropped screen (1) displaying sensitive information. It is connected to an information system (2). An interception chain including an SDR receiver (3) sends samples to a host computer (4) that implements signal processing.

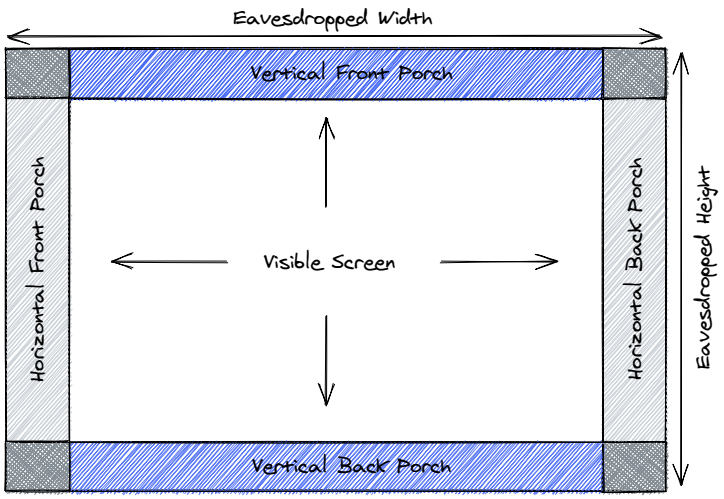


Figure 2.5 – Extra pixels are transmitted both vertically and horizontally and thus recontructed as data in eavesdropped images. Historically used to give time to CRT beam, the porch nowadays may host sound or additionnal information.

image point of view as observed at the final step being the interpretation of images and the evaluation of the compromise.

2.3.1 Image Coding

Historically, the first video communication protocol was proposed for CRT displays using the *raster scan* principle. The raster scan consists in displaying the pixels on the screen one after the other from left to right and top to bottom using an the electron beam in the case of CRTs. Due to that raster scan, protocols had to introduce extra pixels so that the beam has time to go back to the beginning of the next line or to the beginning of the next image. These undisplayed pixels are added at the end of each line and at the end of each column. Next

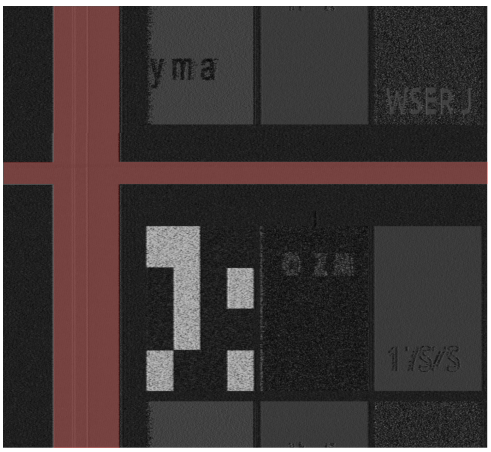


Figure 2.6 – When reconstructing images from a 1D eavesdropped signal, if the synchronization parameters are not exactly set, image appears as horizontally and vertically displaced. Also, blanked pixels (synthetically highlighted in red) are visible because contained in the raster intercepted video signal.

generations of video display still use raster scan but do not require waiting time anymore thanks to different buffering strategies. Nevertheless, the addition of extra data at the borders have been kept and its use differs depending on the standard. As an example, HDMI uses that slot to transmit sound. The timing slack offered by these undisplayed pixels is used for digital processing, e.g. for plushing pixel fifos or initializing filters for the next image/line. The reconstructed images contain the extra pixels since the borders are contained in the wired transferred video signal and thus reconstructed as image data (see Figure 2.6). These extra pixels make the intercepted image different from the one displayed on the attacked screen. In the following of that manuscript we call the extra data at the borders the *porch*. The porch is specific to the communication protocol as each of them uses the border in a different manner.

As presented above, the retrieval of synchronization parameters is essential to reconstruct the eavesdropped signal. Once the parameters are found, the 1D vector can be transformed to an image that do not drift anymore. Nevertheless, calibration has to be done so that the image is aligned with the screen in order to create proper datasets. The non-alignment with the screen is depicted in Figure 2.6 where the image should be moved up left. We propose in Chapter 6 a method that does the alignment in order to create supervised training dataset for learning algorithms.

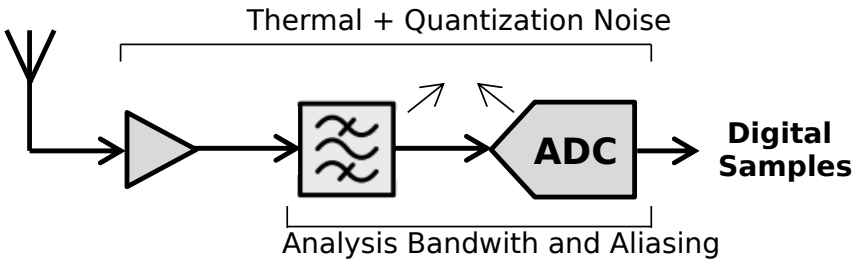


Figure 2.7 – Noise sources in the reception chain (right part of Figure 2.4).

2.3.2 Emission Defaults

Contrary to telecommunications, side-channel emission is unintentional. This leads to non controlled signals with very low Signal to Noise Ratio (SNR). Therefore, the properties of the EM signal is optimized for wireless communication.

A first interference in the eavesdropped signal is caused by video communication protocols that use several wires in the cables. When catching EM emanations, the signal of these several wires are mixed together and even interfere with each other. As an example, HDMI uses a cable for each color component of an Red Green Blue (RGB) signal as well as a wire for the synchronisation clock. Several other wires like sound or power supply exist but do not contribute to the reconstruction. However, they act as a noise sources.

The environment where the eavedropping is conducted may interfere and corrupt the reconstructed signal. Samples like the one presented in the left part of Figure 2.9 are the results of third party signal correlated with the legacy signal. Since the side-channel leakage is unintentional, it is complicated to avoid such interferences, especially in a real world experience conducted outside a laboratory.

2.3.3 Interception Impairments

Pixel Information Spreading The interception system reconstructs images from 1D EM signal. The system acquires samples of data at $f_{sampling}$. According to the Nyquist-Shannon theorem, to recover the entire signal, $f_{sampling}$ should be at least twice the maximum bandwith of the signal. However, the pixel frequency f_{pixel} may be high. As an example, f_{pixel} already reaches 125 MHz for a Full HD 1920×1080 screen display at 60 Hz (neglecting the extra pixels around the actual image). Modern receivers would allow such high $f_{sampling}$ but a trade off must be respected. Chosing an high $f_{sampling}$ which is 2 times the f_{pixel} would make the reconstruction ideal. However it brings more noise into the received bandwidth leading to a

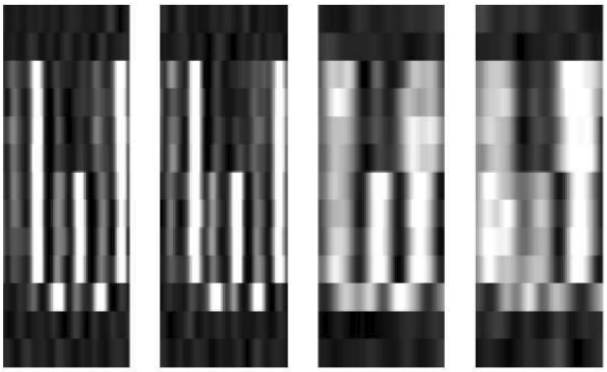


Figure 2.8 – Sampling under two times the bandwidth of the signal as stated by the Nyquist principle, results in a spreading of the information of a pixel on several neighbors. The smaller the sampling rate, the bigger the spreading. Here, the two left images are received at 200MHz while the two on the right at 50 MHz. Images from Markus Kuhn [Kuh02].

Figure 2.9 – Examples of corruptions contained in an eavesdropped image. Left: unknown interference noise, right: hybrid Gaussian (thermal) and Bernoulli (saturation) noise.

poor reconstruction. Sampling under the theoretical ideal rate leads to the loss of the horizontal scale. That loss leads itself to pixel information spreading. The fact of sampling under the pixel frequency implies that the information originally represented by a pixel is split to different pixels in the reconstructed image. This spreading is inherent to the sub sampling and cannot be avoid. The spreading results in more blurry images with less sharp edges.

Electronic noise The reception chain (right part of Figure 2.4) that carries out the interception is made of active electronic components depicted in a simplified manner by Figure 2.7. These components are sensitive to thermal noise, function of the temperature and the bandwith. A particular attention must then be paid when setting of the bandwith: higher bandwidth leads to higher noise level. Thermal noise is modeled by *Gaussian* noise. Due to the conjoint action of the amplifier and the filter, saturation may also append. This saturation can be modeled by *Bernoulli* noise, also known as salt and pepper noise. A display of the thermal and saturation noises is depicted on the right of Figure 2.9.

(a) (b)

Figure 2.10 – A reference image given to the semantic segmentation and classification framework Mask-RCNN [He+17]. (a) The rooster is detected as "bird" and relatively well segmented. (b) The eavesdropped counterpart of the reference image. Nothing is detected by the Mask-RCNN instance.

2.4 Going Further With Image Processing

When retrieving visual information from an EM signal, a non-negligible part of the original information is lost or damaged throughout the leakage/interception process. This leads to a drop of the SNR. Most related work of the literature focus on advanced processing before the image reconstruction. In [DSV20a] De Meulemeester et al. focus on fine grain dynamic synchronization to enable averaging a large number of successive samples. Doing so, they demonstrate the reconstruction of a screen content at 80 meters. However, it may be difficult to use averaging on such a large number of samples, until 400 in their experiments. This averaging requires that the synchronization of the interception is perfect to avoid pixel-wise averaging of drifting information. Also, in a context of continuous catching and interpretation, changes in the intercepted data would disturb the averaging with samples from old and actual signal being mixed.

We propose to work in the image space to benefit from the spatial properties of the addressed video signal, relax the importance of a fine synchronization and avoid averaging on large batches of images. However, due to the SNR drop caused by the corruptions evoked before, the interpretation of image may be complicated. In fact, image interception methods are generally not designed for corrupted images. As an example, in Figure 2.10 a Mask-RCNN [He+17] instance is applied to an image. Mask-RCNN is made to segment and classiffy

natural images. The algorithm succeeds in finding and segmenting the rooster. The image is then eavesdropped, which results in nothing being detected anymore by the same algorithm.

The different corruption sources presented before make the addressed problem a hybrid distortion The term *hybrid distortion* was introduced by Li et al. in [Li+20b]. The corruptions contained in their work are less aggressive than those generated by eavesdropping. This unmodeled hybrid distortion breaks the semantic priors usually leveraged by state of the art learning based algorithms to restore natural images.

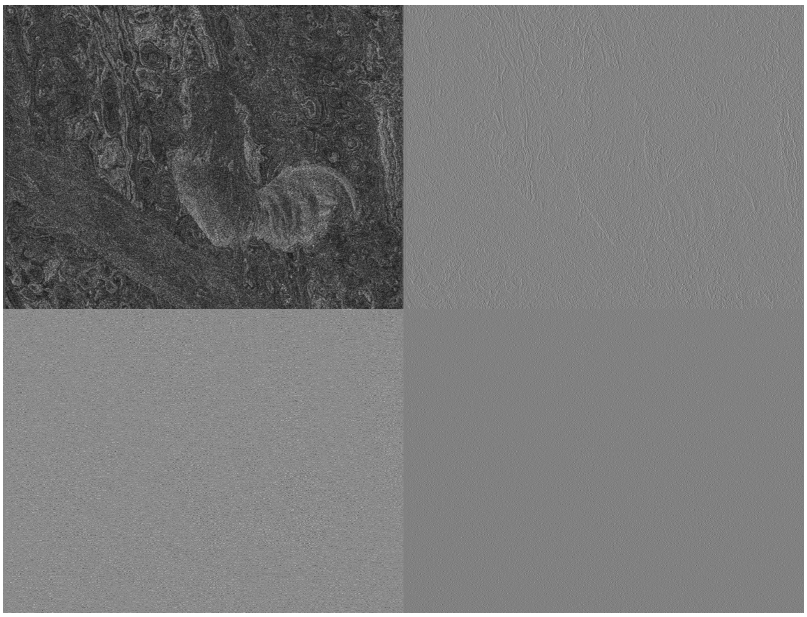
Examples of useful features when reconstructing images are gradient, edges or flat regions. We propose to use first order 2D Haar Discrete Wavelet Transform (DWT) [Dau88] as a tool [Guo+17] to highlight the consequences of the hybrid distortion generated by the eavedropping process. This transform decomposes an original image into four sub-bands that capture the average, vertical, horizontal and diagonal frequencies. In a 2D signal, the frequency represents the intensity changes, i.e. the gradients. On Figure 2.11, a DWT is applied to an image (a) and its intercepted counterpart (b). On both (a) and (b), top-left image is a downscaled version of the image to transform obtained by a $2\times$ sum-pooling. Bottom-left and top-right images relate to horizontal and vertical gradients, respectively. Finally, bottom-right quarters relate to diagonal gradients. When observing the figures, it can be observed of (b) that the transforms, contrary to (a), does not visually contain much information. This observation shows that the interception process "breaks" gradients of images.

There are two majors motivations in leveraging image processing to go further in the interpretation of intercepted images. First, interpretation of eavesdropped samples is often done by human operators. Automation of the interpretation would enable auditing systems continuously. A second motivation is to go further by enhancing the images before relying on human interpretation.

2.5 Conclusion

EM compromising emanations are a major issue when handling sensitive data on IPEs. An attacker can for example retrieve whole or part of a video signal transmitted between an IPE and its display. This is a threat to confidentiality. The images reconstructed from 1D EM compromising emanations are highly corrupted. The corruptions are diverse and come from different origins. The mis-synchronization of the interception system results in non-aligned eavesdropped and original images. The distance between the emissions and the antenna as well as the hardware defects result in a strong hybrid noising. These corruptions, due to the

(a)



(b)

Figure 2.11 – (a) Haar DWT applied to the reference image of Figure 2.10a. The edges of the rooster and the trunk are visible. (b) The transformed of the eavesdropped image of Figure 2.10b. The interception process has broken the vertical gradients. horizontal gradients still exist but are not as sharp as in the original image.

eavesdropping process itself, complicate the interpretation of eavesdropped images. Two main directions appear that motivate work on removing corruptions. First, better samples would enable better automation of the interpretation. Indeed, standard methods developed for image interpretation are designed for non-corrupted images.

Second, enhance the image quality would enable human interpretation of images with more challenging eavesdropping conditions. In particular, with a high-performance restoration of eavesdropped samples, at constant quality, the interception distance could be extended.

Next chapter covers state of the art for noisy image interpretation and particularly methods for image restoration.

CHAPTER 3

Noisy Image Interpretation

Chapter Contents

3.1 Introduction

Images retrieved from Electro Magnetic (EM) side-channel interception are highly corrupted. A first lever to obtain better samples could be to enhance the interception process. However, the interception process is highly dependent on its surrounding environment. These environmental conditions impose an upper bound to the interception quality and control. In contrast, a second lever consists in leveraging image processing to improve image denoising and interpretability. This solution acts after the image construction instead of during the interception process. With the recent progress in Machine Learning (ML), one can wonder if it is worth making the effort on improving the interception or if efforts shall be put on improv-

Figure 3.1 – Given a noisy image, depending on the final aim, different processing may be applied.

ing signal interpretation. Our work consists in studying the opportunities of post-interception image processing to better interpret eavesdropped images.

Two directions emerge when planning to go further using image processing and ML (see Figure 3.1). First, a direction consists in directly interpreting the noisy samples. Interpreting the images using an automated system pushes further the limits of side-channel attacks making it possible to monitor continuously intercepted emanations. From a protection scenario perspective, mining information from eavesdropped samples opens for assessing how compromising is an emanation, therefore, critical for sensitive information. Then, a second direction is to restore the eavesdropped samples. Restoring samples is a first automated step preparing human interpretation. That is, it becomes possible as an example to read samples intercepted in worse conditions, i.e. longer distance, noisier environment. These two directions are non exclusive. As an example it can be interesting to leverage image restoration to enhance results of interpretation.

This chapter first describes what a noisy image is in Section 3.2. Section 3.3 presents the state of the art in noisy image restoration methods. Section 3.4 reviews image interpretation methods. Popular metrics to assess restoration and interpretation methods for images are presented in Section 3.5. Section 3.6 describes popular datasets used for image restoration and interpretation problems. Section 3.7 introduces terminology and questions the strenghts and weakness of learning algorithms.

3.2 What does it mean for an image to be noisy?

We define an image to be a 3-dimensional array of pixels. An image $\mathcal{I}$ has dimensions $[C, H, W] \in \mathbb{Z}^+$, where C is the number of channels, H the height and W the width. In this manuscript, we consider channel numbers of $C = 1$ for grayscale images and $C = 3$ for Red Green Blue (RGB) images. Images are classified by their content and many classes of image

may be defined, we consider here natural and synthetic images. Natural images are issued from photographs that represent a given scene, that may contain people, animals, landscapes, etc. In the context of this thesis, synthetic images are textual contents displayed on screens. Natural and synthetic images have different properties [TO03]. Natural images are more diverse in terms of shapes and textures. They most often content more smooth intensity transition when synthetic images are more sharp.

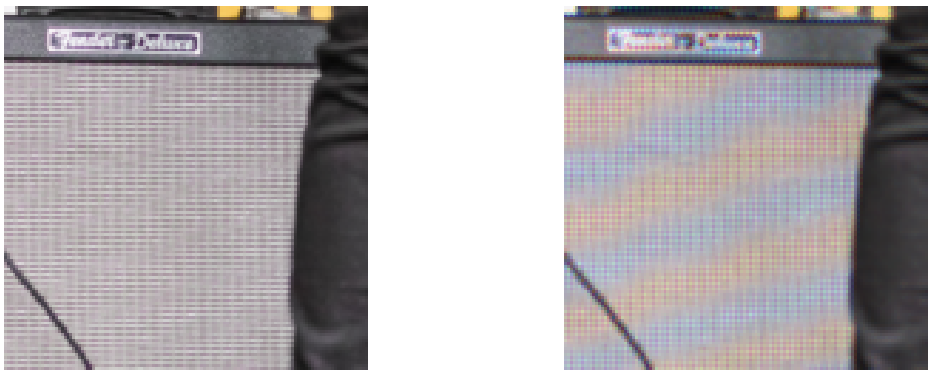
An image is said to be noisy when an unwanted signal exists jointly with the original expected content. We only consider in this manuscript corruptions that keep the dimension of the original image and are applied pixel-wise. As an example, we do not consider corruptions that result in a translation between noise free and noisy samples. There exist plenty of noise sources. The multiple factors of the hybrid noise generated by the eavedropping process is a perfect example of the numerous noise sources that exist. We present in the following several well-known noise models and real-world noises that appear in real applications.

3.2.1 Standard Noise Types

When facing an image restoration problem with pixel-wise corruption, a designer first tries to define a statistic model for the corruptions she/he tries to remove. There are well-known distributions in the literature to model noise such as the 5 following ones:

- Additive White Gaussian Noise (AWGN) is denoted $\mathcal{N}(\sigma_g)$ and applied following $p_n = p_o + \mathcal{N}(\sigma_g)$, where p_n and p_0 are the noisy and original pixel values, respectively. σ_g is the standard deviation of the Gaussian distribution. We use only centered AWGN. In other words, the mean of the distribution is 0.
- Speckle noise is denoted $\mathcal{S}(\sigma_s)$ and applied following $p_n = p_o + \mathcal{N}(\sigma_g) \times p_o$. σ_s is the standard deviation of the Gaussian distributed multiplicative factor applied to p_o.
- Uniform noise is denoted $\mathcal{U}(s)$ and applied following $p_n = p_o + \mathcal{U}(s)$. The additive corruption value is uniformly drew out of the range $[-s, s]$, i.i.d. for each pixel.
- Poisson noise, noted $\mathcal{P}$, has no parameter and is applied following $p_n = \mathcal{P}(p_o)$. The corruption for a pixel is defined following a Poisson distribution depending on the original value.
- Bernoulli noise, noted $\mathcal{B}(p)$, is an impulse noise. A pixel as probability p to be corrupted. When corrupted, the pixel is set to either 0 (min) or 255 (max) with equal probability.

The most used distribution is the AWGN as it fits many case studies and its properties are well studied. Nonetheless, some real-world corruptions, such as the ones we are interested in, do not match any of these *well-behaved* noise models.



(a) (b)

Figure 3.2 – Example of an image (a) and its Moire corrupted version (b).

3.2.2 Towards Real-World Noise Distributions

The well-behaved noises exposed in Section 3.2.1 were theorized following observation of real phenomena. We refer to these noise distributions as *primary* distributions. Primary noises have statistics distributions ruled by a known degree of freedom. However, most corruptions are more complicated and do not follow *primary* distributions. These corruptions have an unknown degree of freedom.

For some real-world image corruptions, the noising process is known or can be estimated [ALB18]. While being estimated, the noising process may not be efficiently modeled as a primary distribution or composition of primary distributions. Moire [Yua+20] is such a corruption. Moire is a sensing artifact that appears when the color array filter of a sensor interferes with high frequency patterns (see Figure 3.2). The most known Moire case is the photography of Liquid Crystal Display (LCD), Light-Emitting Diode (LED) or screens using equivalent technologies.

On the other hand, some corruptions can be modeled by composing primary noises. Composed noises, are more application specific than primary noises, but their existence constitutes an identified and important issue. Many distributions of real-world noises can be approached using noise compositions, also called *mixtures* [Zha+14]. Restoration of noise mixture corrupted images has been less addressed in the literature than that for primary noise. Furthermore, we show in Chapter 4 that the restoration methods designed for primary noises do not directly transfer to mixture noises.

In the literature, experimental noise mixtures are all created from the same few primary noises presented earlier. When modeling experimental noises, noise mixtures are either *spa-*

tially or *sequentially* composed. In a *spatially* composed noise mixture [CPM19; BR19], each pixel p of an image x is corrupted by a specific distribution $\eta(p)$. A typical example of a spatially composed mixture noise is made of 10% of uniform noise $[-s, s]$, 20% of Gaussian noise $\mathcal{N}(0, \sigma_0)$ and 70% of Gaussian noise $\mathcal{N}(0, \sigma_1)$, where the percentages refer to the amount of pixels, in the image, corrupted by the given noise. This type of spatial mixture noise has been used in the experiments of GAN-CNN based Blind Denoiser (GCBD) [Che+18] and Generated-Artificial-Noise to Generated-Artificial-Noise (G2G) [CPM19] with $s = \{15, 25, 30, 50\}$, $\sigma_0 = \{0.01, 15\}$ and $\sigma_1 = \{1, 25\}$. Real photograph noise [PR17; Abd+20] is for instance a composition of primary noises [Gow+07], generated by image sensor defects.

The mixture noise can also be *sequentially* composed as the result of applying n primary noises with distributions $\eta_i, i \in \{0..n-1\}$ to each pixel p of the image x. An example of a sequential mixture noise is the one used to test the recent Noise2Self method [BR19]. It is composed of a combination of Poisson noise, Gaussian noise with $\sigma = 80$, and Bernoulli noise with $p = 0.2$.

Despite being designed following real world scenarios, mixtures noises are also used in order to challenge the interpretation methods. Because of the plural noise sources evoked in Section 2.3, we hypothetize that the eavesdropping corruption is a mixture made of several primary noises. This is the assumption made in Chapter 5.

3.3 Overview of Image Restoration Methods

Corruptions are inherent to the entire lifespan of an image, from their acquisition to their destination being a human looking at it or a machine interpreting its content. From the beginning of the pipeline with the image sensing being possibly affected by sensor defects and poor acquisition conditions, the image undergoes corruptions. To be transmitted easily, an image is often lossy compressed. The transmission itself is a corruption source with potential fragments of the signal being lost. Image restoration, as a subset of signal processing, addresses these issues.

Image restoration is the task of estimating the original signal content of an image from a corrupted observed version. Different research areas exist within the image restoration domain. Among them, *denoising* [Tia+20], *deblurring* [WCH20] and *super-resolution* [Ha+19] are the most popular. In the recent literature most solutions propose experiments on different restoration problems. As examples, the authors of [MSY16] experiment their proposal on image denoising and super-resolution, when the authors of [Liu+18] add JPEG deblocking to

these two latters tasks. In the following of this document, for simplification, we refer to these restoration techniques as *denoising* methods.

Image denoising is an extensively studied problem [BCM05] though not yet a solved one [CM10]. The objective of a denoiser is to generate a *denoised* image $\hat{x}$ from an observation y considered to be a *noisy* or corrupted version of an original *clean* image x. y is generated by an often unknown noise function h such that $y = h(x)$ (as depicted in Figure 3.3). Most methods take into account the phenomena leading to the corruption while others completely abstract it to extend their applicability. A vast collection of noise models exists [BJ15] to represent h. Examples of frequently used models are described in Section 3.2.1. While denoisers are constantly progressing in terms of noise elimination level [Dab+07; Zha+17; Liu+18], most of the published techniques are tailored to a given *primary* noise distribution (i.e. respecting a known distribution). These methods exploit probabilistic properties of the noise they are specialised for, to distinguish noise from signal of interest.

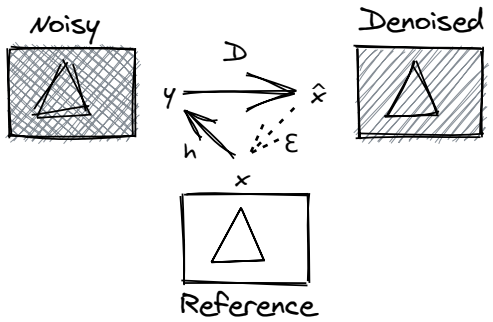


Figure 3.3 – Principle of a denoising algorithm. A *noisy* image y is given to a denoiser $\mathcal{D}$ that outputs an according *denoised* image $\hat{x}$. Depending on the noising process h, the *reference* image x is available or not. If x is available, the denoising quality ϵ can be measured (see Section 3.5).

3.3.1 Expert-Based Algorithms

We denote by *expert-based* the methods designed by experts, by opposition to learned solutions that are drawn from data. All these methods rely on assumptions about the underlying *true* signal that we want to retrieve using denoising. Expert-based denoising methods are traditionally divided into *transform-domain* and *spatial-domain* methods. Spatial-domain methods operate directly on the pixel intensities of the images. On the contrary, transform-domain methods rearrange the values to coefficients using different operations that define the *transform*. That split is also relevant for trained methods. Some of them use transform domain for

input sub-sampling in Multi-level Wavelet Convolutional Neural Network (MWCNN) [Liu+18] or directly in the network in Implicit Dual-domain Convolutional Network (IDCN) [Zhe+19]. The abstraction of feature space in neural networks can also be seen as a transform domain.

Transform-domain methods assume that the true signal is regular, which implies that it can be represented using only few coefficient in a given transform domain. In other words, the true signal is supposed to be sparsely represented in the transform domain. On the contrary, the noise being random is expected to be represented among all coefficients. Based on this assumption, it is possible to keep only few coefficients of that sparse representation, discard the others and transform back to space domain. Different orthogonal transform domains are used like Fourier, cosine or wavelet, and impact the sparsity of the representation. The act of removing some coefficients of a representation is called shrinkage and different methods can be used to do it. Among these methods, well known are soft and hard thresholding as well as adaptive algorithms that aim at remove any type of information not correlated to the initial data. These methods largely rely on the transform that can represent the true signal as sparse as possible. However, there is no orthogonal transform that works well on all interesting features (flat region, texture, edges) of an image. To counteract that issue, [FKE07] proposed a transform adaptive to salient details or homogeneous regions in an image.

Spatial-domain methods also leverage regularity properties of underlying image. Most spatial-domain solutions relate on the observation that noise is sporadic while signal is regular. However, using such paradigm, images with high frequency are poorly restored and the output image tends to be blurry. [BCM05] introduced *NL-Means* for non-local means. This method proposes to estimate a given pixel in an image with a weighted average of the pixels with a neighbourhood similar to the one of the estimated pixel. Unlike other methods, NL-means is said to be non-local as it uses information at different places in the image instead of just looking at its close neighborhood.

Well performing methods take advantage of both transform and spatial domain. Block-Matching 3D (BM3D) [Dab+07], as an example, leverages spatial information to group related image patches. A shrinkage is then done on the groups in the transform domain before coming back to spatial domain.

3.3.2 Fully-Supervised Learning Algorithms

ML are increasingly used in image denoising showing better performances than expert-based methods in fully supervised cases. First learned denoising methods were directly inspired by classification Convolutional Neural Networks (CNNs). Classification networks out-

put a prediction vector that gives in fine an information on the content of the image. In contrast authors of [JS09] were the first to propose to output an entire image given an input image.

First trained denoising methods were fully supervised, the mapping between a noisy domain and a denoised domain was learned using back-propagation. In other words, the mapping was learned using associated pairs of clean and noisy images. The difference between learned methods mainly lies in the architecture of the neural network that effectively represent the mapping between the domains. While some methods were proposed to automate the design of network architecture [SOO18], this task is today mainly done empirically by expert engineers.

Following the premises of CNN based denoisers [JS09], different strategies have been proposed such as residual learning in Denoising Convolutional Neural Network (DnCNN) [Zha+17], skip connections in Residual Encoder-Decoder Network (RED) [MSY16] or self-guidance in Self-Guided Network (SGN) [Gu+19]. Learned methods also take inspiration from the expertise gained in image processing before the rise of ML. Transform domain is used in [Liu+18] where the authors consider different wavelet decomposition to be used as sub and up-sampling operator in a multi-resolution architecture. [Zhe+19] proposes to use a dual-domain architecture that leverages complementary of spatial and transform domain corrections directly into residual branches. Some methods even introduce learnable parameters directly into expert methods like in BM3D-Net [YS18].

The weakness of discriminative denoisers is their need of large databases of independent noise realisations, including clean reference images, to learn efficiently the denoising task. To overcome this limitation, different weakly supervised denoisers have been proposed.

3.3.3 Weakly Supervised Algorithms

Weakly supervised methods apply when it is not possible to build a complete training dataset with clean references. In particular, *blind* denoisers are capable of efficiently denoising images with noise distributions not available in the training set. First studies on blind denoising have aimed at determining the level of a known noise so as to apply an adapted human-expert based denoising. Most of the recent blind denoisers focus instead on training exclusively on noisy data. In [SSR11], authors propose a noise level estimation method integrated to a deblurring method. Inspired from the latter proposal, the authors of [LTO13] propose to estimate the standard deviation of a Gaussian distribution corrupting an image to apply the accordingly configured BM3D filtering [Dab+07].

Some recent studies aim at modelling the noise distribution corrupting an image with a Generative Adversarial Networks (GAN)-based model. Once the noise distribution is modelled,

it is possible to generate independent noise realisations and train a dedicated discriminative denoiser. GCBD [Che+18] and G2G [CPM19] are examples of such denoisers.

Noise2Noise (N2N) [Leh+18] has pioneered learning-based blind denoising. Authors show that it is possible to learn a discriminative denoiser from only a pair of images representing two independent realisations of the noise to be removed. Noise2Void (N2V) [KBJ19] and Noise2Self (N2S) [BR19] are recent strategies that train a denoiser from only the image to be denoised.

[UVL20] goes a step further in the non supervision and shows that the knowledge brought by the engineering of network architecture is itself an image prior capable of denoising. Using this strategy, the authors with their method named Deep Image Prior (DIP) learn to denoise a given image using only a random initialized denoiser and the image itself.

Recently, a new family of learning algorithms called *transformers* is getting more and more interest from the community. These transformers mainly rely on internal *attention mechanisms* between patches of an image i.e. on self-contextual information. Transformers were first proposed for Natural Language Processing (NLP) [Vas+17] and adapted to image recognition tasks [Dos+20]. Image restoration counterparts were quickly proposed in [Che+20] with the Image Processing Transformer (IPT). While the first proposed methods are really data-greedy, recent publication proposes fine-tuning strategies to limit the need for large datasets [Tou+21].

3.4 Overview of Image Interpretation Methods

Image restoration is the entry point when working with noisy input images. If the images are interpreted by a human operator, the automated process can be stopped. However, in some case it is useful to automate interpretation. As an example, when auditing a system, it could be useful to monitor permanently the emanations of an Information Processing Equipment (IPE) to ensure that it does not leak compromising data.

Interpretation automation is one of the major concern of Computer Vision (CV). *Intelligent* algorithms are used to assist human operators or to fully automate the decision making process. Image interpretation groups all the tasks, that given an image return knowledge on its content. Among these tasks, well-known ones are *classification*, *segmentation* [RFB15; Min+20] or *pose estimation* [CTH20].

The objective of a classification algorithm is to identify to which of a set of *classes* a new sample belongs. Most classification framework are made of two steps. First a step named *feature extraction* is responsible for transforming the input data to a space that separates better the samples of different classes. Well known expert-based feature extractors/detectors are Scale

	Paper	Name/Acronym	Adressed Corruptions(s)
Expert	[BCM05]	NL-Means	AWGN
	[FKE07]	Pointwise SA-DCT	AWGN, JPEG Compression
	[Dab+07]	BM3D	AWGN
Fully Supervised	[JS09]	-	AWGN
	[SSR11]	-	AWGN, Blur
	[LTO13]	-	AWGN
	[MSY16]	RED	AWGN, Super-Resolution
	[Zha+17]	DnCNN	AWGN, JPEG Compression, Super-Resolution
	[YS18]	BM3D-Net	AWGN
	[Liu+18]	MWCNN	AWGN, Super-Resolution
	[Che+18]	GCBD	AWGN, Spatial Mixture Noise, Sensor Noise
	[Gu+19]	SGN	AWGN, Sensor Noise
	[Zhe+19]	IDCN	JPEG Compression
	[CPM19]	G2G	AWGN, Spatial Mixture Noise
Weakly Supervised	[Leh+18]	N2N	AWGN, Bernoulli, Poisson, Text Removal, ..
	[KBJ19]	N2V	AWGN, Microscopy Noise
	[BR19]	N2S	Mixture Noise, Sensor noise
	[UVL20]	DIP	AWGN, Super-Resolution, Inpainting
	[Che+20]	IPT	AWGN, Super-Resolution, Rain Strikes

Table 3.1 – Image restoration methods evoked in Section 3.3 and their targeted noise(s).

Invariant Feature Transform (SIFT) [Low04], Speed-Up Robust Features (SURF) [Bay+08] or Histograms of Oriented Gradients (HOG) [DT05]. Then, another block called *classifier* makes a decision on the class that better suits the features given by the extraction. The classifier must identify the feature space that belongs to each class. Typical classifiers are Support Vector machines (SVMs) [Bur98], k-Nearest Neighborss (kNNs) [Guo+03] or Naives Bayes classifiers [Mur+06].

The first neural networks to be used as classifier were Muli-Layer Perceptrons (MLPs). MLP is a scalar manipulating type of neural network in which each element of a given layer is connected to each element of the next layer. Image classification has been revolutionized by deep learning methods since LeNet-5 [LeC+98]. With the development of tailored algorithms and hardware resources, deeper and more sophisticated neural networks have emerged. The use of the convolution operator into the architecture of neural network has (among other benefits) deeply reduced the complexity of MLPs and its Fully Connected (FC) pattern by sharing parameters between the pixels in the analysed image. This complexity relief has permitted the growth of neural networks in terms of parameters and thus enhanced their modelling power. AlexNet [KSH12] has been a major advance that used a CNN to almost halve the error rate of classification state of the art on the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2012. It launched the major interest of the CV community for Deep Learning (DL). Later on, ResNet [He+16a] was released to counteract the fact that very deep networks are more difficult to train due to vanishing gradients. At the time the method was released, it was the first to be trained with as much as 150 layers, when applied to the ImageNet dataset [Den+09]. ResNet, in its deepest version, won the ILSVRC in 2015. ResNet has then been modified, using identity mappings as skip connections in residual blocks (ResNetV2 [He+16b]). With the same objective, DenseNet [Hua+17] introduces connections between layers and performs training of very deep networks. It must be noted that the advances proposed by these methods are located on the feature extraction part. Most of these methods uses FC layers for final class prediction which is nothing less than MLPs.

Most classification algorithms are designed for and trained on clean image data. Because of the changes it implies on the image, the noise disturbs the functioning of classification algorithms [HD19]. Authors of [Li+20a] propose to use Discrete Wavelet Transform (DWT) to better extract the basic object structures of input noisy image to classify. They claim that this better feature extraction leads to a classification more robust to noise.

We have seen in the last two sections that most restoration and interpretation methods are designed for well-behaved noise distributions. This manuscript focuses on eavesdropped

images that do not follow such distributions. Adaptations of state of the art methods as well as new strategies are then required and will be proposed in the following of this document.

3.5 Error Measurement and Quality Assessment

In image processing, error measurement is a full research area and a major concern. Evaluating the quality of images is required to assess the efficiency of any given method. In ML, error measurement is used not only for efficiency evaluation but also to drive optimisation processes. In fact, algorithms based on gradient descent optimisation select parameters values by minimizing errors between intended result and obtained result. In the following, we differentiate between methods used for quality assessment and methods for classification error measurement.

3.5.1 Image Quality Metrics

Mean Square Error (MSE) measures the average pixelwise squared error between two images. MSE is a metric computed between a reference image x and an evaluated image y using the following formula:

$$MSE(x,y) = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[x(i,j) - y(i,j)]^2 \tag{3.1}$$

This formula applies to grayscale images. MSE also adapts to three-dimensional RGB images or to tensors with arbitrary dimensions. In this case, the MSE scores for each dimension are averaged.

Peak Signal to Noise Ratio (PSNR) evaluates the ratio between the dynamic range of an image and the intensity of the corruption that affects it. PSNR is expressed in dB to narrow the range of possible outputs. PSNR between a reference image x and an evaluated image y is usually computed using MSE. The higher the value, the closer the evaluated image is with respect to the reference. PSNR is evaluated using the following formula, where d is the maximum possible value for a pixel:

$$PSNR(x,y) = 10.log_{10}\left(\frac{d^2}{MSE(x,y)}\right) \tag{3.2}$$

Structural Similarity (SSIM) [Wan+04] is an index proposed to evaluate the structural similarity between two images. The interest of this index is that human eye is sensitive to

structure changes between images. SSIM is a reference metric computed between a reference image x and an evaluated image y, like MSE and PSNR. SSIM values are in $[0, 1]$, 1 being the best value. The index is computed as follows:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3.3}$$

, with:

- μ_x, μ_y the means of x and y
- σ_x^2, σ_y^2 the variances of x and y
- σ_{xy}^2 the covariance of x and y
- $C_1 = (k_1 L)^2$ and $C_2 = (k_1 L)^2$ two variables to stabilize the division with weak denominator.
- L the dynamic range of the pixel values ($2^{bits\_per\_pixel} - 1$)
- $k_1 = 0.01$, $k_2 = 0.03$ empirical values

Throughout this manuscript we only use *objective* metrics computed on images. It is nonetheless interesting to point out that a common practice when evaluating image processing proposals is to use *subjective* metrics that reflect the human perception. Subjective testing [BT07] involve viewing sessions where human subjects are asked to rate the image quality or compare the results of different processing. A protocol must be respected to avoid mis interpretation of results. Metrics like Mean Opinion Score (MOS) is used to obtain a final numerical value out of subjective testing. Recently, researchers state that while measuring MOS is the most accurate for subjective rating, it is time consuming and expensive. An example of a recent subjective quality approximation metric is VMAF [Li+16] metric to model the subjective rating in real-time.

3.5.2 Classification Metrics

The objective of a classification algorithm is to identify to which of a set of *classes* a new sample belongs. The output of a classifier is a *prediction*. In the context of a supervised classifier, the true class called a *label* or *target* of the sample is provided to train the classifier.

A standard practice is to ask a neural network to output a probability for each of the possible classes. The softmax function is used to output a prediction vector summing to 1 as would be a probability distribution. The top prediction is then obtained passing the vector of predictions to the argmax function.

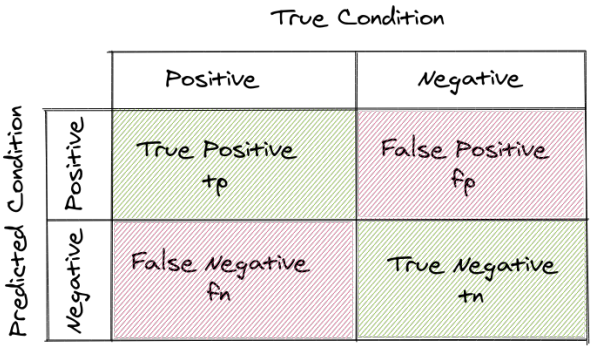


Figure 3.4 – Confusion matrix used for F-score and accuracy metrics computation.

Accuracy is the natural metric to measure the efficiency of a classification algorithm. It represents the ratio of correctly predicted samples among all predicted samples. It is often expressed as a percentage. Subclasses of accuracy are sometimes used and called Top-n accuracy. To compute these metrics, we consider a sample as well predicted if the actual class is contained in the n most probable classes as predicted by the algorithm.

$$Accuracy = \frac{\text{Number of Correctly Predicted Samples}}{\text{Total Number of Samples}} \tag{3.4}$$

However, the interpretation brought by the accuracy is limited as the problem has 2 dimensions. Accuracy only considers correct predictions and does not give a clue on the types of error done by the classifier. To look closer at results, the confusion matrix of Figure 3.4 is used. Four cases are identified, namely true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn).

Precision measures the ratio between the true positives and the predicted positives. This metric is important when false positives are costly. As an example, an application that requires an operator to look at positives should benefit from a high recall.

$$precision = \frac{tp}{tp + fp} \tag{3.5}$$

Recall measures the ratio between the true positives and the actual positives. This metric is important when false negatives are costly. As an example, in medical applications false negatives must be avoided because they mean failing to detect something.

$$recall = \frac{tp}{tp + fn} \tag{3.6}$$

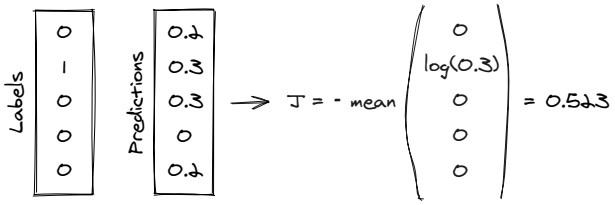


Figure 3.5 – Example of cross entropy loss computation between a prediction and a label vector for a 5-class classification problem.

F-Score is the harmonic mean of precision and recall. Using F-score contrary to accuracy emphasizes incorrectly classified cases. The use of the harmonic mean is interesting since it penalizes the extreme values.

$$\textit{F-score} = \frac{tp}{tp + \frac{1}{2}(fp + fn)} \tag{3.7}$$

Cross-Entropy, measures the difference between two probablity distributions for a given set of events. In information theory, the entropy represents the number of bits required to encode a random event of a probability distribution. In this theory, the cross-entropy would represent the number of extra bits required to represent the event of a distribution compared to another distribution. In the context of a classification, the cross-entropy measures the extra entropy of the predicted vector compared to the target vector. The minus sign makes the score decrease when the distributions get closer to each other. The cross-entropy J is calculated as follows, with y the target (label) vector, x the prediction vector and N the number of classes of the problem.

$$J = -\frac{1}{N}\sum_{n=1}^{N}\left(y_n log(x_n)\right) \tag{3.8}$$

We presented in this section different metrics that will be used in the following of the manuscript to evaluate the results of state of the art as well as proposed methods. We presented metrics for image quality assessement. MSE and PSNR are mathematicaly related and evaluate the pixelwise impairements between an image to assess and a reference. These two metrics, while evaluating the amount of corruption contained in images (with respect to a reference), do not consider the perceptual aspect of images. SSIM on contrary is designed to reflect the perceptual aspect of the image quality as perceived by the human eye. We also presented metrics used to assess the quality of classification problems. All these metrics may also be used as loss functions to drive the optimisation of learning based algorithms.

3.6 Datasets for Learning and Evaluation

Data is the keystone of learning algorithms. In this manuscript, we use supervised datasets only. Supervised datasets are made of images jointly stored with a label being a class for classification or a clean target image for restoration. We define two types of supervised dataset. The first type of dataset contains only *non-corrupted* images. We refer to these datasets as general purpose datasets. Most often these datasets have been created for classification task. To use them as image restoration datasets, the samples are corrupted using noise models, generating the noisy images while the reference image are kept as labels.

One of the most known datasets is ImageNet [Den+09]. ImageNet was first introduced as the dataset for the ILSVRC contest. The ImageNet dataset contains millions of images with their labels for classification. Berkeley Segmentation Dataset (BSD) [Mar+01] is a well-known dataset in image restoration while it was originally built for segmentation tasks. The BSD dataset is interesting due to the variety of samples it contains in terms of content semantics. BSD is traditionally splitted into two datasets of 432 and 68 images, used for training and validation/testing, respectively. The evaluation set of 68 images is refered to as *BSD68*. DIVerse 2K (DIV2K) [AT17] is a recent dataset created for the New Trends in Image Restoration (NTIRE) challenge in 2017. It contains 900 high resolution images which have at least one of their dimension that reaches the 2K (2048) dimension. Among these 900 images, 800 are identified as training samples and 100 as validation/test samples. Other large well-known datasets like CIFAR-10 or CIFAR-100 [Kri09] are used but their small resolution limits their interest.

We consider a second class of datasets applicable to real noise cases. When the noise model is not perfectly known, it is not possible to create artificial datasets. There are different methods to create such datasets. A first strategy is to acquire another sample from the same scene by a method that generates less noisy samples, thus considered as clean. In [AB18], authors capture sensor noise due to low ISO acquired images. The "noise-free" counterpart is obtained capturing the same scene with long exposure. In [ALB18], the authors constitute a dataset of noisy images captured with smartphones. Due to smartphone sensors settings, the authors cannot use long exposure. Instead, they propose a software estimate of the ground truth images.

We present in the following a method to generate a supervised dataset of eavesdropped images. This method is used to create a dataset of eavesdropped natural images and a dataset of eavesdropped textual screens in Chapter 6.

Dataset	Resolution	Number of Images
ImageNet [Den+09]	Resolution	1M +
CIFAR-10 [Kri09]	$[32, 32]$	60k
BSD [Mar+01]	$[481, 321]$	500
DIV2K [AT17]	$[2040, 1550+]$	900

Table 3.2 – Popular datasets, the resolution of their samples and the number of images they contain.

3.7 Learning Algorithms: Terminology, Strengths and Open Issues

We presented in Section 3.3 and Section 3.4 expert and learning based solutions for image restoration and interpretation. State-of-the art of both restoration and interpretation is nowadays dominated by learning methods. That domination is explained by superior performances but also comes with issues such that a high complexity and or the need for large databases representative of the problem to solve. We present in this section a brief overview of the terminology and learning principles. A reflection on the strengths and open issues of learning algorithms in our specific context is also proposed.

3.7.1 Terminology, Learning Pipeline and Architecture Specificity

When used for learning purpose, we refer to data structures with more than 2 dimensions as *tensors*. Input restoration and interpretation tensors are 4-dimensional with dimensions $[B, C, H, W] \in \mathbb{Z}^+$ (see Figure 3.6a), where B is the batch size, C the number of channels, H the height and W the width. It should be noticed that the dimension B is introduced to enable using *mini-batch learning*. Mini-batch learning consists in updating model parameters after processing only B images instead of the whole training set. Once a data tensor passes a network layer, it enters the *feature domain* and is then called a *feature map*.

While there exist different types of neural networks, in this document, we only deal with the family of *feed-forward* neural networks trained using *back-propagation*. In these networks, processing elements are arranged hierarchically in layers and the number of layers is called the *depth* of the network. Unlike Recurrent Neural Network (RNN), as an example, data flows in a unique direction without being fed back to previous layers.

We consider two modes of operation for networks, namely training and inference. At inference, input data is passed through the network. The learned function is then applied to

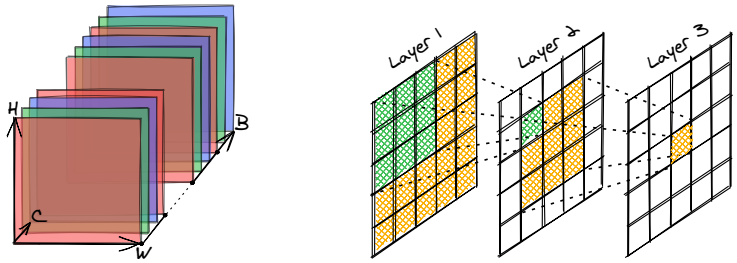
the input data. That phase is called *forward* pass. When training, an additional *backward* pass is done after the data has flowed through the network. This pass updates the parameters of the network according to their gradients computed with respect to the errors measured after the forward pass. The metric used to guide the parameters update is called the *loss function*. We call the design of the network (number of layers, size of the filters, etc..) the *architecture*. The learneable parameters that are trained and shape the final function are named alternately *parameters* or *weights* and noted θ.

	Hyperparameters	
L	$\triangleq$	Number of Layers
	Training Settings	
lr	$\triangleq$	Learning Rate
N_{epochs}	$\triangleq$	Number of Training Epochs
	Tensor Dimensions	
B	$\triangleq$	Batch Size
C	$\triangleq$	Number of Channels
H	$\triangleq$	Height
W	$\triangleq$	Width

Table 3.3 – Table of Notation of Learning Algorithms

Difference between learning Image-to-Class and Image-to-Image We previously presented restoration and interpretation as two different tasks. Using learning algorithms, these two tasks are achieved using neural networks with different architectures but with some similar blocks and principles. Both tasks are based on feature extraction out of input tensors. Image interpretation and restoration algorithms take the same tensors as input. The major difference lies in the fact that restoration is a dense estimation task. The output has the same dimension as the input (i.e. a batch of images), unlike classification that outputs classes. For that reason the networks are slightly different.

In CNNs, the size of the *receptive field* is a major concern. The receptive field of a network is the area of the input image from which a value in the network depends on (see Figure 3.6b). In other words, enlarging the receptive field brings more context information. In classification networks, large receptive fields are mainly obtained using more layers or successive down-sampling of the feature maps. Standard down-sampling strategies are *striding* and *pooling*. *Striding* consists in moving a filter by a delta of several pixels instead of moving it to the next pixel. Striding results in an output feature map spatially smaller than the input. *Pooling* directly acts on the feature maps. A sub-sampling operator is applied window-wise. The different pool-



(a) (b)

Figure 3.6 – Definition of network dimensions: (a) *Height*, *Width*, *Number of Channels* and *Batch Size*. (b) Receptive field schematics: the middle value of Layer 3 depends on all values of Layer 1.

ings differ by the size of their window and by the operation they apply. *Max* pooling, as an example, consists in replacing the window by its maximum value. The max pooling is used a lot for an historical reason being that it works well on MNIST [LeC+98]. In practice, the choice of the pooling operator depends on the data to be processed. Restoration is a dense task because it outputs a full image with the same dimension as the input (except for super-resolution where the output is bigger than the input). Being dense, restoration benefits from keeping as many features as possible. Striding and pooling are lossy sub-sampling operations since data is lost through the process. Instead of using striding or pooling, different sub-samplings have been proposed for dense tasks uses. In [Liu+18], the authors use Discrete Wavelet Transform (DWT) to decompose a feature map into 4 down-sampled feature maps in an invertible manner. The fact that DWT is invertible makes it possible to use inverse transform when up-sampling again and then avoid wasting data. No dimension is lost, the feature map are just in a transformed domain. In [Gu+19], the authors propose to use the *shuffle* sub-sampling introduced in [Shi+16]. Alike wavelet transform, the shuffling operator is completely dense and does not drop features. In fact, shuffling rearranges the feature maps dimensions by transforming spatial dimensions to channel dimensions whithout dropping any feature. [YK16] proposed to use *dilated convolutions* which, instead of skipping a step when sliding a kernel, apply the kernel in a sparse manner [1]. This solution enlarges the receptive field without sub-sampling.

For image restoration, up-sampling is mandatory to retrieve the original resolution from the deepest part of the network, when it is operated at a lower resolution. For wavelet and shuffling operators, inverse operators exist. For strided convolution, transposed convolution [DV18] is often used [1].

1. Visualisations of dilated and transposed convolutions, making their understanding easier, are proposed at: https://github.com/vdumoulin/conv_arithmetic

Pre-Processing Many datasets do not have a fixed resolution. Depending on the architecture, a neural network may not support different resolutions among images. As an example, a neural network which contains FC layers takes fixed size inputs. Resizing is often used to solve that issue. For example, a standard practice is to resize all ImageNet samples to 224×224 [KSH12] before feeding them to a classifier containing FC layers.

Several network architectures like [Gu+19] or [Liu+18] require power of 2 input image dimensions because of successive down-sampling operations. For that purpose, cropping is used instead of resizing. Doing so, only few pixels have to be removed and the other pixels are kept intact instead of being transformed by resampling.

When using small datasets in terms of number of samples like BSD or DIV2K, a common practice is to patch the images. One image is then split in $M \times N$ patches. For simplicity, patches are often squares. That patching enables using larger batch sizes and it reduces the required memory, which is often an issue when using Graphics Processing Units (GPUs). It should be noted that the patch size cannot be smaller than the receptive field size. Following the use of patches in [BSH12], the authors of [Zha+17] propose to use different patch sizes for their different experiments. Larger patch size are used for stronger corruptions to provide more information to learn.

3.7.2 Strengths of Learning Algorithms

Performance Superiority The popularity of DL methods in CV has grown because it surpasses expert methods on many tasks. When the restoration algorithm DnCNN [Zha+17] was proposed, it outperformed BM3D [Dab+07] by 0.6dB on the task of denoising AWGN with σ=50 on the BSD68 grayscale dataset. DnCNN improved the expert-based state the art from 25.62dB to 26.23dB while returning more natural looking images. The tendency is the same for classification algorithms with AlexNet [KSH12] that shortened the error rate on ILSVRC 2012 by 10% compared to the state of the art method at the time [SP11].

DL methods also have the advantage to be ran efficiently on GPUs because of their high degree of parallelism. As an example, instead of BM3D, DnCNN being a CNN, can be ran on GPUs, making around 60 times faster at inference time for a 1024×1024 image.

Modeling Capability ML algorithms are very efficient when the task is complex, i.e. when the problem is hardly invertible. In this context, it is complicated to design an expert-based method since the problem is not precisely modeled. On contrary, DL models the problem by the experience it acquires during training. Trained with data representative of the problem, complex function can be approached. This faculty is interesting for applications like medical

imaging where it is complicated to model the sensing noise. This modeling capacity seems also promising for our eavesdropping image case study. We leverage this modeling power in Chapter 6 to interpret eavesdropped images by learning on a custom noisy dataset.

3.7.3 Two Open Issues of Deep-Learning Algorithms

Training Data Dependency Learning algorithms are trained from data. In the case of supervised learning, the objective of the training procedure is to learn the function that maps the training inputs to the targets. The function modeled by the trained weights relies on the content of training input/target pairs. This is a weakness since at inference time, any input that diverges from the pairs seen at training time will no be processed well. The training data dependency complicates the comparison of state of the art methods. In fact, depending on the choice of the authors, the training and evaluation dataset may change between experiments. In that case, even if the authors publish their code and trained models, it is not possible to compare them directly. We propose in Chapter 4 a tool to ease the training and evaluation of denoising methods on equal basis to fairly compare them.

Once trained, the function modeled by the neural network is fixed. This is an issue when considering evolving problems. An an example, one that would like to add a class to a classification problem cannot do it easily. It is not convenient to retrain all the model. In [LH18], the authors evaluate several solutions to add new classes to a problem while minimizing the prediction loss on the other classes. In Chapter 5, we propose a restoration method that adresses the training data dependency and the evolving problem.

Lack of Explainability DL models have the capability to model complex problems. However, this comes at the cost of over-dimensionned networks hardly explainable. Except for very special neural networks, the architecture is fixed and considered as a hyper-parameter. The training process then tunes the trainable parameters to eventually obtain a modeling of the desired function. This modeling is obtained by the action of the parameters together with non-linear functions. Made of millions of weigths, it is complicated to explain the behaviour of a neural network. This issue is addressed by a research domain named eXplainable Artificial Intelligence (XAI) [Gun+19]. We propose in Chapter 5 a method that makes a step towards understanding of decisions by providing information about the noise classes contained in a corrupted image.

3.8 Conclusion

Noisy image restoration and interpretation are extensively addressed domains. We presented in this chapter a definition of a noisy image. A noisy image is a sample that contains unwanted extra information that bothers the interpretation of legacy information. Learning-based method have strongly enhanced the performance of noisy image restoration and interpretation compared to expert-based methods. Different classes of learning algorithms and their state of the art were exposed in the chapter. We also presented the metrics that are crucial to assess the performances of algorithms. These metrics are also useful to drive the optimisation process of learning algorithms.

The progress brought by learning comes at the cost of methods being largely reliant on data used to train the systems. In the case of image restoration, as an example, that reliance on training data heads to methods over specific to the corruption they are trained for.

This chapter has shown that learning algorithms have good performance for the task we are interested in, i.e. restoration and interpretation. It has also enlighted that these algorithms are not designed to be applied as is to the images we are interested in. The following of that manuscript evaluates how that issue can be addressed by studying the following question: to what extent learning methods can reinforce the interpretation of EM compromising information?

The next chapter studies the impact of the dominance of AWGN in the data choice for denoising architecture evaluation. In particular, a benchmark and a case study are proposed that evaluates methods developed for AWGN applied to real-world eavesdropping noise.

Part II

Contributions

CHAPTER 4

Benchmarking of Image Restoration Algorithms

Chapter Contents

4.1 Introduction

State of the art denoisers are constantly progressing in terms of noise elimination level [Dab+07; Zha+17; Liu+18] (see Section 3.3). However, most techniques are tailored for and evaluated on a given noise distribution, exploiting its probabilistic properties to distinguish it from the signal of interest. On the specific case of Additive White Gaussian Noise (AWGN), current denoisers are approaching theoretical bounds [CM10].

Besides the largely addressed *well-behaved* noise models, for which the distribution is parametric with a few parameters, image denoising is also concerned by more complex noise distributions. While these distributions are application specific, they are real-world cases directly issued from identified technical needs such as image interception in difficult conditions.

In a context where new methods are constantly appearing, it is challenging to fairly compare emerging methods to previous ones. Moreover, when a real-world noise needs to be elim-

inated, it is difficult to determine which of the existing methods is the best for the given noise characteristics. Even if most state of the art methods are evaluated on the de-facto standard databases (e.g. the 12 well-known images such as Lenna or Cameraman, BSD [Mar+01] or DIVerse 2K (DIV2K) [AT17]), methods addressing specific noises and image types have to be evaluated on tailored databases. A tool that compares performances on an equal basis is then important when designing denoising methods.

In this context, the contributions of this chapter are:

- An extensible and open-source benchmark for comparing image restoration methods.
- A comparative study of current denoisers on mixture and interception noise elimination, as a use case for the benchmark.

The chapter is organized as follows. Section 4.2 presents state of the art methods for image denoising as well as existing solutions to benchmark them. Section 4.3 describes the proposed benchmark. The comparative study, covering six restoration methods, is proposed in Section 4.4. Section 4.5 concludes the chapter.

This chapter contributions have been published in: *F. Lemarchand, E. Fernandes Montesuma, M. Pelcat, and E. Nogues, « OpenDenoising: an Extensible Benchmark for Building Comparative Studies of Image Denoisers », in 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 2648–2652.*

4.2 Related Work

This section locates the proposal in the existing work and shows the novelty of the proposed solution. First, several existing benchmarks are reviewed. Then, different state of the art image denoisers are detailed before being assessed in the comparative study of Section 4.4.

4.2.1 Related Work on Benchmarks of Image Denoisers

An active research domain in complex noise restoration is photograph restoration. This domain aims at removing a noise introduced by sensor hardware defects. Supervised datasets can be built by calibrating a sensor and hence obtaining pairs of clean and noisy samples. Darmstadt [PR17] and PolyU [Xu+18] are such datasets. Authors propose to use their datasets as a means for benchmarking denoising algorithms. This work is complementary to our proposed benchmark that can adapt to different datasets.

Open-source projects have been created to benchmark denoising methods. The University of Toronto proposes a benchmark [1] to provide reproducibility for a method proposed in [EFJ09]. This benchmark is tailored to the solution and not built to be extended. Another unpublished benchmark exists that implements denoising as well as other restoration algorithms such as super-resolution or colorisation [2]. The benchmark is limited to learning-based, Python-implemented and pre-trained methods. The latter limitation drastically reduces the use of such benchmark for complex noises. Indeed, most state of the art methods, when delivered trained, are trained on well-behaved noise. This chapter proposes a benchmark extensible in several aspects. Indeed, the user can introduce his datasets, denoising methods, and metrics.

4.2.2 Chosen Image Denoisers for Benchmarking

Image denoising techniques are as old as image sensors whose defects they counteract. Current denoising solutions are either expert-based denoisers, human crafted based on an expertise of artifacts or of statistical noise properties, or learning-based denoisers leveraging on latent image priors extracted from data (see Section 3.3). We prioritise in our tests the following methods either for their state of the art performance on well-behaved noise or for their potential to denoise eavesdropped images.

Block-Matching 3D (BM3D) [Dab+07] is a state-of-the-art expert-based method for AWGN removal. BM3D performs block matching to find patches with similar content in the image and uses collaborative filtering, thresholding and Wiener filtering into the transform domain to restore the image.

Authors of [Vin+10] were the firsts to propose an encoding/decoding model with denoising objective. Their proposal named "stacked auto-encoder" learns to map the noisy image to a latent space (encoding) and projects back the latent representation to the input space (decoding) to produce the denoised image. More recent auto-encoders have been proposed such as RED [MSY16] that adds convolutional layers and uses skip connections to better keep image priors throughout the encoding/decoding pipeline.

Following these premises of denoising auto-encoders, several Convolutional Neural Network (CNN) methods have emerged such as Denoising Convolutional Neural Network (DnCNN) [Zha+17]. DnCNN is inspired by the well-known VGG [SZ15]. It exploits residual learning, i.e. it learns to isolate the noise $\mathbf{h}$ from the corrupted sample to later remove this noise instead of directly recovering the latent clean signal. DnCNN in its "blind" version

1. www.cs.utoronto.ca/~strider/Denoise/Benchmark/
2. https://github.com/titsitits/open-image-restoration

demonstrates its ability to handle different noise levels. That makes it a potential candidate for eavesdropping corruption removal. Multi-level Wavelet Convolutional Neural Network (MWCNN) [Liu+18] is also CNN-based. Its novelty lies in the symmetrical use of wavelet and inverse wavelet transforms into the contracting and expanding parts of a U-Net [RFB15] architecture. The use of wavelet enables *safe* subsampling with no information loss providing a better recovering of textures and sharp structures. This faculty of texture and sharp structures recovering seems promising for eavesdropping corruption removal.

The most recent learning-based methods are less supervised, i.e. they require less noisy/-clean image pairs to train. In Noise2Noise (N2N) [Leh+18] and Noise2Void (N2V) [KBJ19] , authors propose a tactic to train a denoising model using a single noise realisation. Authors introduce the idea of *blind-spot masking* during training. They claim that the essential advantage of that strategy is to avoid to learn the identity due to the masking of the central value of the receptive field.

These methods are evaluated on well-behaved noises (typically AWGN) for the tests to be easily reproducible and comparable to state of the art. Only Noise2Void is evaluated on medical images subject to complex noise. In the following, our open benchmark is proposed to assess fairly the quality of denoisers.

4.3 Proposed Benchmark

Considering the above discussed issues with existing benchmarks, we propose the OpenDenoising benchmark illustrated in Figure 4.1. It is an open-source tool with tutorials and documentation [3] released under a CeCILL-C license. OpenDenoising is implemented in Python and has been designed for extensions. Adding a new denoiser to the benchmark is a matter of minutes following a tutorial and opens for comparison with the built-in methods evoked in Section 4.4. For learning-based methods, the application is compatible and tested with most major frameworks (Tensorflow, Keras, Pytorch, Matlab). For learning-based training and evaluation, it is possible to use one or several datasets either supervised or not. Any scalar metric being coded in Python can be used in the benchmark. Several pre-processing functions, e.g. for data augmentation, are provided, and custom functions can be introduced.

The user chooses whether a training is required for a method and in that case selects training parameters. Once the training is launched, monitorings can be output by the benchmark to observe the learning phase. When trained models are available, evaluation is launched with

3. https://github.com/opendenoising/benchmark

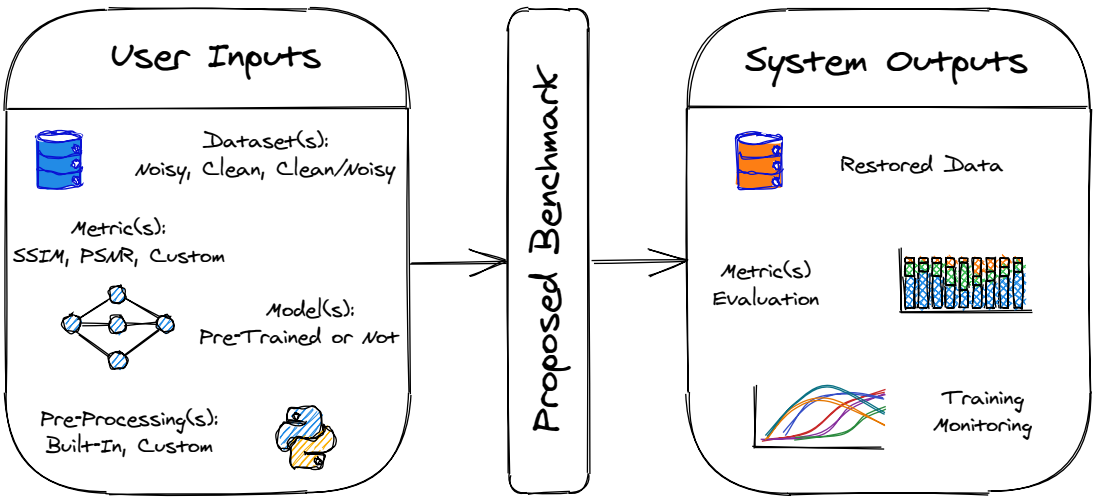


Figure 4.1 – OpenDenoising block diagram. Users can tune datasets, metrics, denoising models and evaluation functions. OpenDenoising produces denoised samples as well as performance metrics.

custom or built-in metrics. The results are outlined using custom or built-in plots and/or stored as images or csv summaries.

As an example of OpenDenoising versatility, it is possible to extend the benchmark to classification methods only implementing custom evaluations metrics. Other potential usages of OpenDenoising include: study the extensibility of methods to new applications (see Section 4.4), study the strategies for re-training off-the-shelf methods (from scratch or with fine-tuning), and tune hyper-parameters. The experimental results presented in the next section exploit OpenDenoising to build a comparative study of state of the art denoisers on different types of noise.

4.4 A Comparative Study of Denoisers

In this section, we apply top-ranking denoisers to images with various noises. For comparison fairness of training-based methods, no data augmentation is made and the same training datasets are used for all methods. Apart from this setup, methods are trained (when applicable) using original publications parameters and training strategies. Four noise types with increasing complexity are exploited to observe the behavior of the studied denoisers. Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) are used to respectively evaluate the point to point and structural quality of the denoised image.

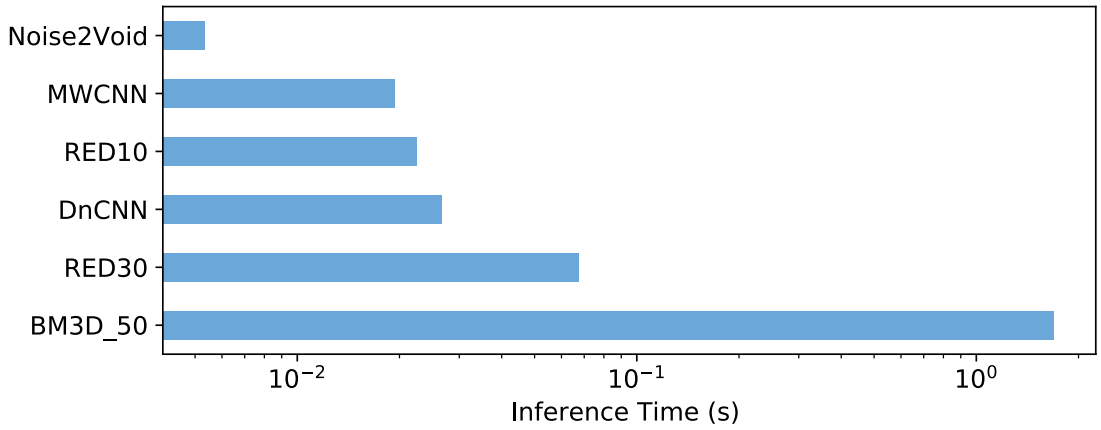


Figure 4.2 – Inference time (log scale) for different denoisers. Image resolution is 256×256. Noise2Void is the fastest method by almost 10 folds. MWCNN, RED10, DnCNN and RED30 are close to each other. BM3D is the slowest with an inference time over one second. Setup: Intel Xeon W-2125 CPU and Nvidia GTX1080 Ti GPU. Note that BM3D runs on Central Processing Unit (CPU) only while the other methods run on both CPU and Graphics Processing Unit (GPU).

	Dataset	No Denoising	BM3D	RED10	RED30	DnCNN-B	MWCNN	Noise2Void
PSNR	Gaussian	14.96	23.90	25.52	**25.82**	25.67	25.49	23.41
	Mixture	10.58	18.29	24.25	**24.58**	24.50	24.30	19.93
	Interception-Like	17.16	22.04	51.56	**52.08**	51.66	51.16	21.70
	Interception	9.46	9.61	22.59	23.46	23.04	**23.66**	9.46
SSIM	Gaussian	0.24	0.67	0.71	**0.73**	0.72	0.72	0.62
	Mixture	0.12	0.31	0.67	**0.68**	**0.68**	**0.68**	0.51
	Interception-Like	0.11	0.98	**0.99**	**0.99**	**0.99**	**0.99**	0.98
	Interception	0.32	0.73	0.94	**0.96**	0.95	**0.96**	0.47

Table 4.1 – Average evaluation of PSNR and SSIM metrics on test sets for, from top to bottom row: AWGN with noise level $\sigma = 50$; Mixture noise made of AWGN with noise level $\sigma = 50$ and Bernoulli noise with $p = 0.2$; Interception-Like noise being interception reference samples noised with AWGN with noise level $\sigma = 50$; Interception noise. The test set for each noise is made of 200 samples excluded from training set.

4.4.1 Gaussian Noise

First, Gaussian noise is used to test the methods in their original conditions. Denoisers are evaluated on a common noisy dataset corrupted with AWGN. The underlying data is made of 10k natural images extracted from the ImageNet [Den+09] evaluation set. Average PSNR and SSIM are shown in Table 4.1 and example images displayed on Figure 4.4. Figure 4.3 shows in boxplots the 10th, 25th, 75th and 90th percentiles of PSNR results as well as the median. To focus on difficult noises, the maximum noise level commonly found in papers is picked, namely $\sigma = 50$. Experimental results, shown on the first line of Table 4.1, are coherent with the published ones, though slightly under because no data augmentation is applied. On Gaussian noise, RED30 outperforms other methods (by a limited 0.15dB ΔPSNR and 1% ΔSSIM) but it is also the most costly Deep Learning (DL) solution in terms of number of parameters.

4.4.2 Mixture Noise

Complicating the denoising task, a mixture noise is then studied. This mixture noise is constructed through the successive corruption of the samples by the previously used AWGN ($\sigma = 50$) and an additional Bernoulli corruption (20% of corrupted pixels, half 0, half maximum). This noise mixture roughly models the behaviour of an image sensor introducing Gaussian noise because of its hardware non-uniformity and Bernoulli noise due to pixel defects.

Figure 4.3 shows that learning-based methods perform consistently better than BM3D. BM3D is here used out of its original objective (i.e. Gaussian denoising) and thus performs poorly. Another information brought by mixture noise is that Noise2Void clearly underperforms compared to other learning-based methods. This is not surprising considering the addition of Bernoulli noise that damages the spatial coherence used as a hypothesis in the Noise2Void strategy. RED10, RED30, DnCNN and MWCNN have close performances with a narrow victory for RED30 (0.08dB ΔPSNR).

4.4.3 Interception Noise

A real-world complex noise is now studied, generated by intercepting images from Electro Magnetic (EM) emanations. Electronic devices produce EM emanations that not only interfere with radio devices but also compromise the data they handle. A third party performing a side-channel analysis can recover internal information from both analog [Van85], and digital circuits [Kuh13]. Following an eavesdropping procedure, it is possible to build a supervised

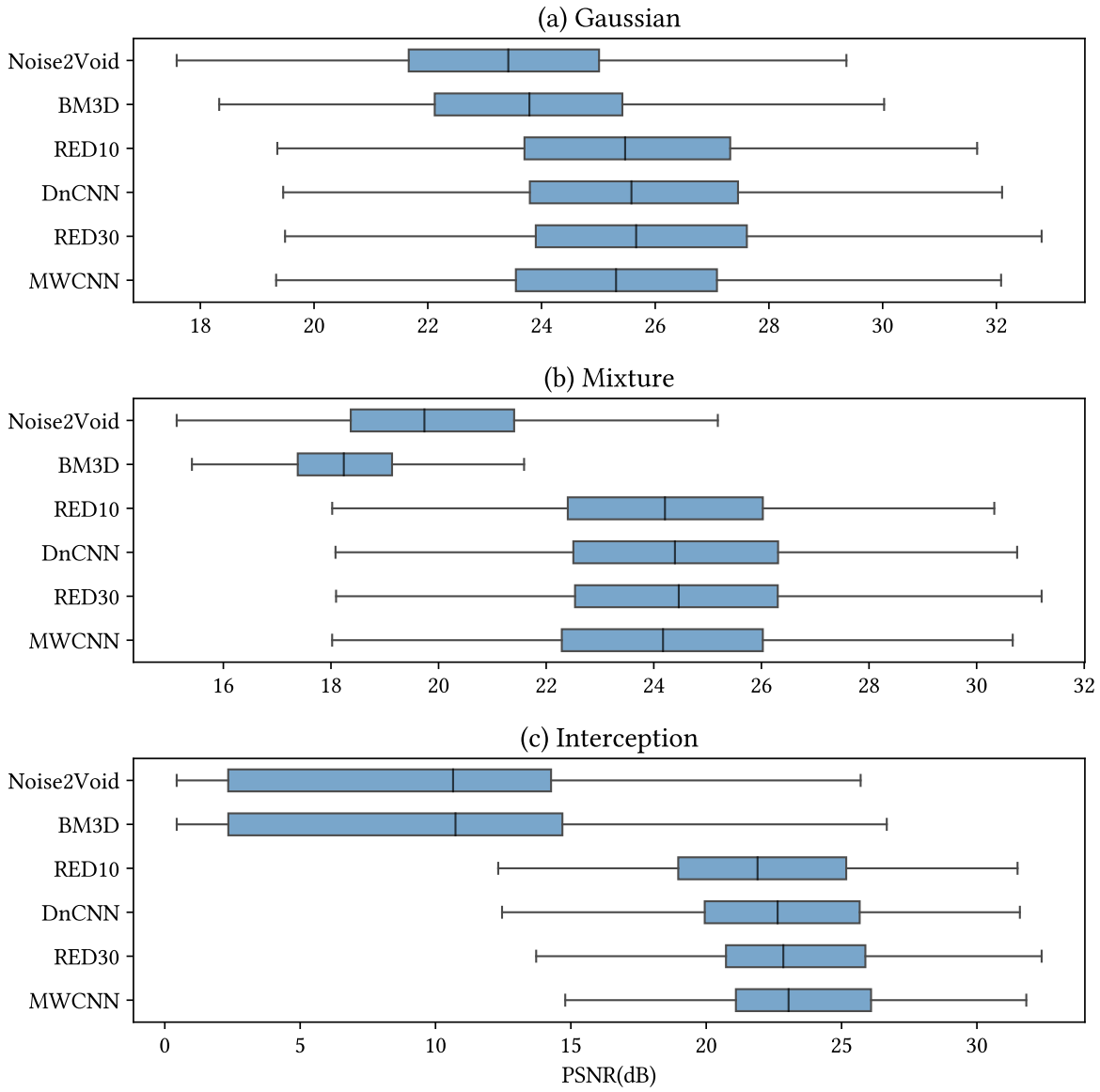


Figure 4.3 – Peak Signal to Noise Ratio (PSNR) of denoised images on (a) Gaussian noise, (b) Mixture and (c) Interception noise. Outliers are not displayed.

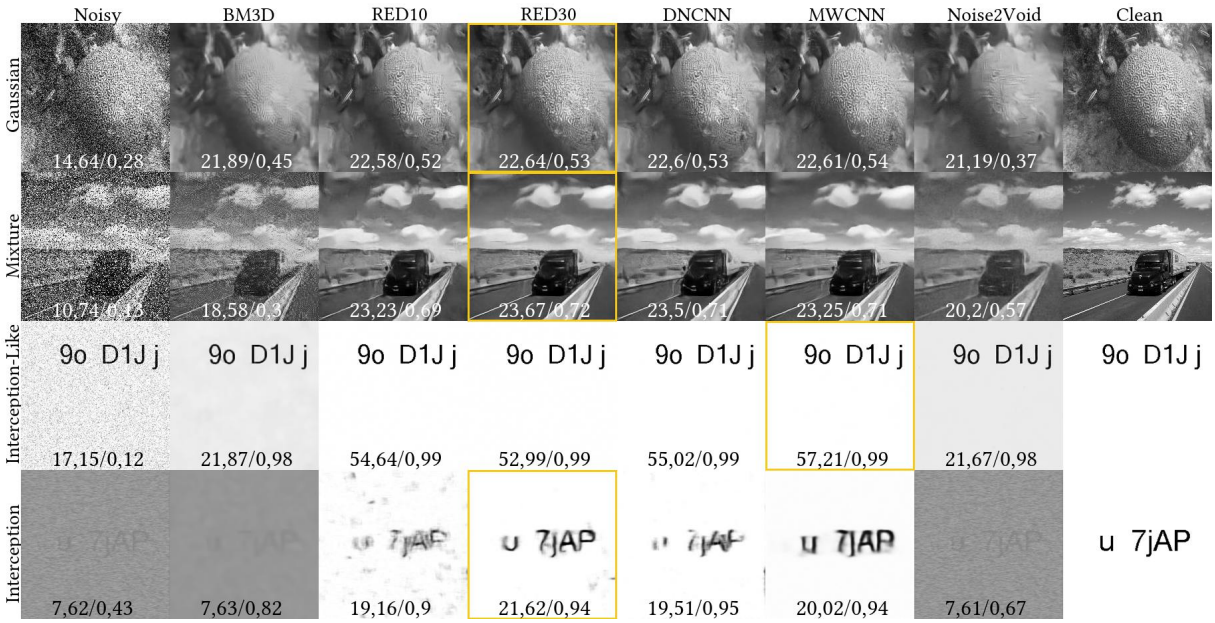


Figure 4.4 – From top to bottom, one sample per dataset is shown noisy (left), denoised with different denoisers (middle) and clean (right). PSNR/SSIM are displayed for each sample. Images with the best compromise between PSNR and SSIM metrics are yellow-boxed.

dataset made of pairs of reference images, originally displayed on a screen, and their intercepted noisy versions. Interception strongly damages the images and denoising is necessary to interpret their content. For reproducibility, we released the dataset used for this study [4]. It contains more than 120k samples.

To study the noise complexity, the intercepted clean samples are also artificially corrupted using AWGN with $\sigma = 50$. The resulting samples are called interception-like. As shown in Table 4.1, most methods perform well on that denoising task. The clean content of the intercepted samples contains black characters printed on a white background. The latent clean distribution of samples is thus not an issue to denoise with learning-based methods, Noise2Void excluded. Only BM3D and Noise2Void have problems with background restoration (see Figure 4.4). This phenomena is due to the correlation of the samples content and the noising process. The background of the clean samples is fully white. When applying AWGN, half of the noise coefficients are negative and samples are clipped to an integer format. Thus, the assumption of a Gaussian distribution does not hold, leading to poor restoration results with non-supervised methods, unable to adapt.

4. https://github.com/opendenoising/interception_dataset

Table 4.1, Figure 4.3 and Figure 4.4 show the results of denoising methods applied to interception noise. Metrics drop for all methods on this complex noise. Noise2Void does not manage to denoise at all. As explained in the original paper, Noise2Void has difficulties with noise correlated between several pixels which is here the case. BM3D is built for AWGN and is not trainable, hence the poor results in that case. Others learned methods like RED10 and DnCNN produce interesting denoising but perceptual results of Figure 4.4 show some hardly interpretable samples, not revealed by SSIM. RED30 and MWCNN are the best-performing methods for interception noise removal but still with some remaining artifacts.

4.4.4 Discussion

Different conclusions can be drawn from the above experiments using OpenDenoising and 4 different datasets.

First, Figure 4.3 shows that the performance ranking between methods strongly depends on noise type, training dataset and evaluation dataset. When calculating *Kendall's Tau correlation coefficient* [JIP10], a value of 0.6 is obtained between Gaussian noise ranking and Interception noise ranking, ranking based on the mean PSNR. This correlation coefficient, while being high - Kendall's Tau is in $[-1; 1]$; -1 and 1 respectively meaning fully discordant and fully concordant rankings -, shows the need for a benchmark such as the proposed one. It automates the comparison process and the selection of a given method for a denoising problem. As an example, it would be a wrong choice to pick RED30 instead of MWCNN for interception restoration based on the original paper evaluations (e.g. Gaussian evaluation). MWCNN is indeed both more efficient and less computationally intensive than RED30, as shown in Figure 4.2.

Results of Table 4.1, Figure 4.3 and Figure 4.4 show a growing gap between expert-based and learning-based method as the complexity of the denoising increases. This can be explained by the flexibility of learning-based models and the advanced information brought by a supervised training. This is evidenced by the low performance of the non-supervised Noise2Void on Mixture and Interception noises.

As stated in several studies, PSNR and SSIM do not suit well the assessment of interception restoration. Figure 4.4 shows that while evaluation metrics on interception noise are reasonably good (PSNR/SSIM values around 20dB/0.9), the perceptual quality (human looking) is poor. The explanation lies in the latent content of the samples, made of black characters on a white background. A good background restoration is sufficient to raise good evaluation metrics. This issue is evoked in [JAF16] where authors propose a different evaluation metric to

overcome the problem. As will be shown in Chapter 6, in that specific case, character recognition rate can be used to assess the denoising performance.

In order to go further on interception noise understanding, we propose in Chapter 6 an open dataset of eavesdropped natural images dubbed Natural Interception Dataset (NID).

4.5 Conclusion

In this chapter, the OpenDenoising tool has been proposed. OpenDenoising benchmarks image denoisers and aims at comparing methods on a common ground in terms of datasets, training parameters and evaluation metrics. Supporting several languages and learning frameworks, OpenDenoising is also extensible and open-source. At the time writing this manuscript the github repository of the OpenDenoising tool has received 16 stars. The second contribution of the chapter is a comparative study of image restoration in the case of a complex noise source.

Three major conclusions arise from the comparative study. First, the difference in terms of performance between expert-based and learning-based methods rises as the complexity of the noise grows and eavesdropping noise is clearly of high complexity, higher than mixture noise. Second, the ranking of methods is strongly impacted by the nature of the noises. Finally, MWCNN proves to be the best method for the considered real-world interception restoration task. It slightly outperforms DnCNN and RED30 while being substantially faster. These results show that restoring an image from a complex noise is not universally solved by a single method and that choosing a denoiser requires automated testing.

Next chapter proposes a method that adresses the removal of sequential mixture noises following the idea that eavesdropping noise may be approached by such artificial distributions.

CHAPTER 5

Mixture Noise Denoising Using a Gradual Strategy

Chapter Contents

5.1 Introduction

Composed noises are more application specific than primary noises, but their removal constitutes an identified and important issue. Real photograph noise [PR17] is for instance a sequential composition of primary noises [Gow+07], generated by image sensor defects. Many distributions of real-world noises can be approached using noise compositions, also called *mixtures*. Noise mixture removal has been less studied in the literature than primary noise removal. When modelling experimental noises, noise mixture are either *spatially* or *sequentially* composed. In a *spatially* composed noise mixture [CPM19; BR19], each pixel p_n of an image is corrupted by a specific distribution $\eta(p_n)$ such that h is composed of the set $\{\eta(p_n), p_n \in Dom(x)\}$. The mixture noise can also be *sequentially* composed as the result of

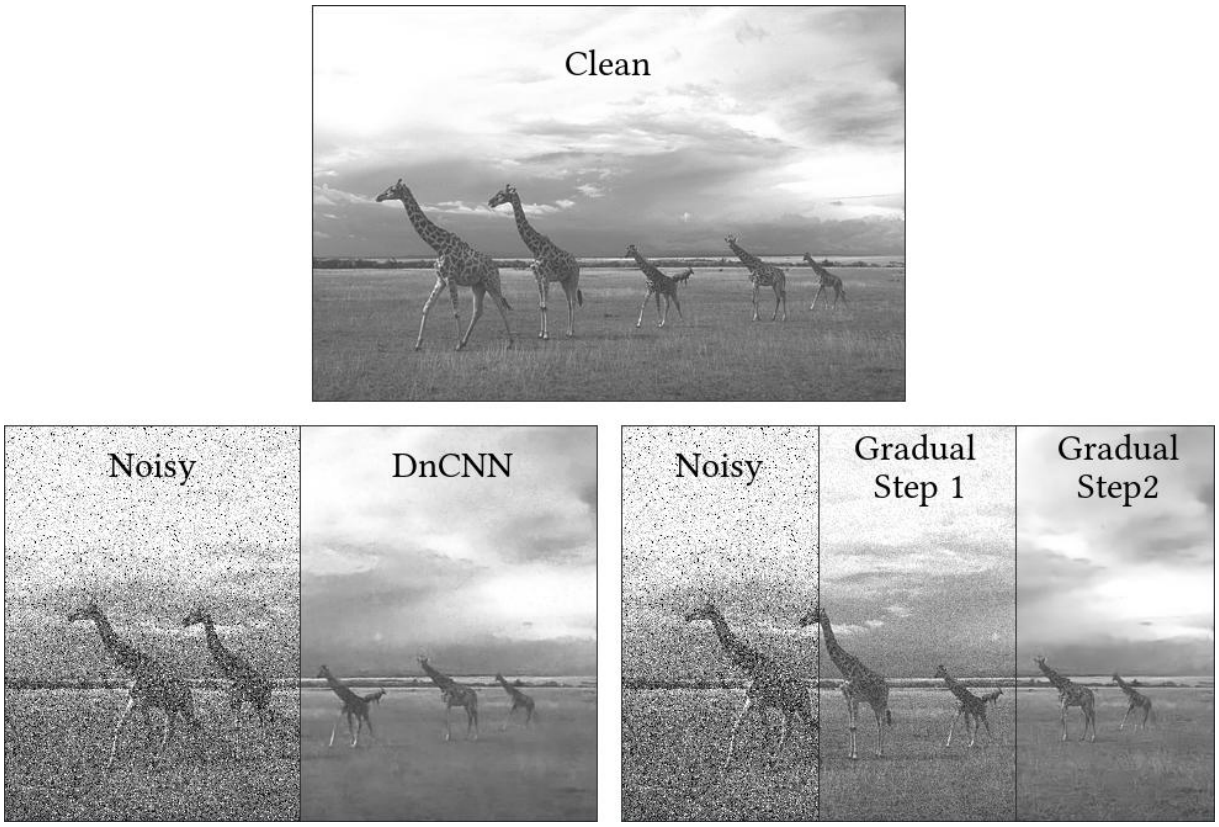


Figure 5.1 – Comparison between a traditional end-to-end denoiser, removing noise mixture at once, and gradual denoising removing primary noises one after the other. Top: clean image; bottom-left: traditional denoising with noisy and denoised using Denoising Convolutional Neural Network (DnCNN) trained on a noise mixture dataset; bottom-right: NoiseBreaker trained on primary noises.

applying n primary noises with distributions $\eta_i, i \in \{0..n-1\}$ to each pixel ρ of the image x. This chapter focuses on sequential noise mixture removal.

Real-world noises, when not generated by a precisely known random process, are difficult to restore with a discriminative denoiser that requires a set of (y, x) pairs of observed and clean images. Some processes can be approached and emulated to generate supervised databases [Yua+20; Abd+20]. For other applications, the lack of clean images makes it difficult to build such supervised databases [PR17]. *Blind denoising* addresses this lack of supervised dataset by learning denoising strategies without exploiting clean data. Blind denoisers are capable of training without clean data but operate on average 5dB under supervised denoisers on noise mixtures (Section 5.4).

This chapter introduces NoiseBreaker, an image denoiser that recursively detects the dominating noise type in an image as well as its noise level, and removes the corresponding noise. The resulting step-by-step gradual strategy is driven by a noise analysis of the image to be restored. The solution leverages a pool of denoisers trained for primary noise distributions and applied sequentially following the prediction of a noise classifier. Different versions of NoiseBreaker are introduced in Section 5.3 and their performances are evaluated on two databases in Section 5.4. An ablation study is presented in Section 5.5.

Additionally to a denoised image, NoiseBreaker also produces a classification of the dominating noises in the image. Having this information has several advantages. First, by decomposing the mixture denoising problem into primary ones, a library of standard denoisers can be built to answer any noise removal problem. This first point is central to NoiseBreaker. Secondly, a description of the image noise content helps to identify the physical source of data corruption. Finally, under the assumption of sequential noise composition, it is possible to identify the noising pipeline from the identified noise distribution. The noise distribution being known, a generation of large training databases becomes feasible to feed fully supervised methods.

The main contributions of this work are:

- The NoiseBreaker gradual image restoration strategy, recovering step by step an image corrupted by a sequential mixture of different noise types and intensity. The method operates without prior knowledge on the mixture composition.
- Qualitative and quantitative results on two datasets of images corrupted with strong noise mixtures, in order to compare NoiseBreaker with state of the art methods.
- A detailed ablation study to assess and validate the choices adopted in the architecture of NoiseBreaker.

The remaining of this chapter is organized as follows. Section 5.2 presents related work on image noise analysis and image denoising. Section 5.3 details the proposed solution. Section 5.4 evaluates the proposal on synthetic noise mixture and situates among state of the art solutions. Section 5.5 conducts an ablation study to assess the relevance of core features of NoiseBreaker. Section 6.5 concludes the chapter and gives future perspectives.

This chapter is based on our following work *F. Lemarchand, T. Findeli, E. Nogues, and M. Pelcat, « Noisebreaker: Gradual image denoising guided by noise analysis », in 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), 2020, pp. 1–6.*

5.2 Related Work

NoiseBreaker can be considered a weakly supervised method, as denoisers are trained on primary noises and used on mixture noises. The covered related work includes not only weakly supervised methods such as blind denoisers, but also fully supervised denoisers. Indeed, fully supervised denoisers provide an upper bound for the performance obtained by weakly supervised methods. Finally, the most commonly used noise mixtures are evoked, as well as the classification-based methods that address these noise mixtures.

5.2.1 Blind Denoising

Fully supervised deep learning denoisers forge a restoration model from pairs of *clean* and *noisy* images. Following the premises of Convolutional Neural Network (CNN) based denoisers [JS09], different strategies have been proposed such as residual learning [Zha+17; Led+17], skip connections [MSY16], the use of transform domain [Liu+18] or self-guidance for fast denoising [Gu+19]. The weakness of supervised denoisers is their need of large databases with clean images.

To overcome this limitation, different weakly supervised denoisers have been proposed. In particular, *blind* denoisers are capable of removing noise without clean data as reference. The first studies on blind denoising have aimed at determining the level of a known noise in order to apply an adapted human-expert based denoising (e.g. filtering). Noise2Noise (N2N) [Leh+18] has pioneered learning-based blind denoising, using only noisy data. It demonstrates the feasibility of learning a discriminative denoiser from only one pair of images representing two independent realisations of the noise to be removed. Inspired by this work, Noise2Void (N2V) is a recent method that trains a denoiser from only noisy data.

5.2.2 Noise Mixtures

To challenge denoisers and approach real-world cases, different types of noise mixtures have been proposed. Mixtures are created from the set of primary noises. A typical example of a spatially composed noise mixture [Zha+14] is made of 10% of uniform noise $[-s, s]$, 20% of Gaussian noise $\mathcal{N}(0, \sigma_0)$ and 70% of Gaussian noise $\mathcal{N}(0, \sigma_1)$. These percentages refer to the amount of pixels in the image corrupted by the given noise. This type of spatial noise mixture has e.g. been used in the experiments of Generated-Artificial-Noise to Generated-Artificial-Noise (G2G) [CPM19]. An example of sequential noise mixture is used to test the recent Noise2Self method [BR19]. It is composed of a combination of Poisson noise, Gaussian noise, and Bernoulli noise. In Chapter 4, denoising methods designed for Additive White Gaussian Noise (AWGN) removal are compared when retrained and evaluated on sequential mixtures of Gaussian and Bernoulli distributions. Experimental results show that denoising performances severely drop on complex noises even when using fully supervised learning methods such as DnCNN. This observation motivates the current study and the chosen sequential noise mixture.

5.2.3 Classification-Based Denoising

The image classification domain has been revolutionized by deep learning methods since LeNet-5 [LeC+98]. With the development of tailored algorithms and hardware resources, deeper and more sophisticated neural networks have emerged. ResNet [He+16a] was released to counteract the fact that very deep networks are more difficult to train. At the time the method was released, it was the first to be trained with as much as 150 layers, when applied to the ImageNet dataset [Den+09]. ResNet, in its deepest version, won the ImageNet Large Scale Visual Recognition Competition (ILSVRC) classification challenge in 2015. ResNet has then been modified, using identity mappings as skip connections in residual blocks (ResNetV2 [He+16b]). With the same objective, DenseNet [Hua+17] introduces connections between layers and performs training of very deep networks. Seeking a good trade-off between classification efficiency and hardware resources, MobileNets [How+17] is a particularly versatile family of classifiers. MobileNetV2 has become a standard for resource aware classification. In our study, we use MobileNetV2 network pre-trained on ImageNet and fine-tuned [Taj+16] for noise classification.

In [SSR11], authors propose a noise level estimation method integrated to a deblurring method. Inspired from this proposal, authors of [LTO13] estimate the standard deviation

of a Gaussian distribution corrupting an image to apply the accordingly configured Block-Matching 3D (BM3D) filtering [Dab+07]. This can be interpreted as a noise characterization, used to set parameters of a following dedicated denoising process.

Recent studies have proposed classification-based solutions to the image denoising problem [SDC19; LSJ20]. Sil et al. [SDC19] denoise by choosing one primary noise denoiser out of a pool, based on a classification result. NoiseBreaker goes further by considering mixture noises, sequentially extracted from the noisy image using a sequence of classify/denoise phases. In [LSJ20], authors adopt a strategy close to NoiseBreaker. However, NoiseBreaker differentiates from that proposal by refining the noise classes into smaller ranges of noise levels. To the best of our knowledge, the present study is the first to use noise type and intensity classification for denoising purposes. We demonstrate in the results Section 5.4 that NoiseBreaker outperforms [LSJ20]. The main reason for NoiseBreaker to outperform the results of [LSJ20] is that the denoising pipeline is less constrained. As an example, the first step of [LSJ20] is to detect if the corruption is a mixture. If it is a mixture and the mixture contains Gaussian noise then the Gaussian noise is removed first. Our proposal does not compel the denoising process to follow a predefined order and lets the classifier drive the denoising strategy to be conducted.

In this chapter, we propose to tackle the denoising of sequential noise mixtures via an iterative and joint classification/denoising strategy. Our solution goes further than previous work by separating the denoising problem into simpler steps, optimized separately.

5.3 Gradual Denoising Guided by Noise Analysis

NoiseBreaker is qualified as gradual because it denoises the input image step-by-step, alternating between noise detection and removal. NoiseBreaker leverages a classifier acting as a noise analyser and guiding a pool of denoisers specialized to *primary noise* distributions. Both the noise analyser and the gradual denoising strategy are detailed hereunder. NoiseBreaker handles numerous noise mixtures at inference time without information on the composition of the mixture. Neither the classifier nor the denoisers are exposed to mixture noise during training.

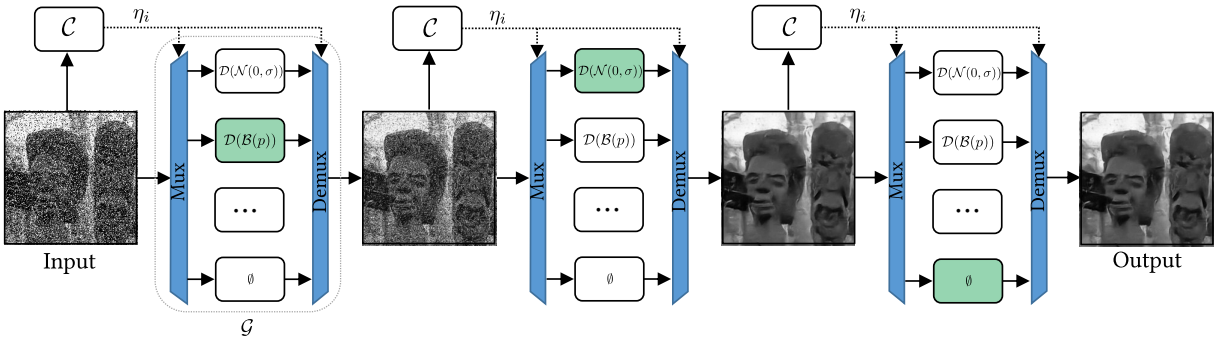


Figure 5.2 – Example of NoiseBreaker gradual denoising. A noisy input image is fed to the classifier $\mathcal{C}$ which outputs a prediction η_i. This prediction drives the gradual denoising block $\mathcal{G}$ that selects the primary denoiser $\mathcal{D}(\eta_i)$ to be applied. The process runs for two steps until no noise is detected by $\mathcal{C}$.

5.3.1 Noise Analysis

The objective of the noise classifier $\mathcal{C}$ is to separate images into n noise classes. A noise class is defined by a noise type and a range of parameter values. A class is denoted using $\eta_{i,j}$ with i an index among a list H of noise types and j an index for the different ranges of a given noise type. When no parameter exists for a noise type or an only range is used, the class is denoted using η_i. η_i (or $\eta_{i,j}$) is referred to as a *primary noise*. One may note that one class does not refer to any noise and serves to identify *clean* images.

The architecture of the classifier is composed of a feature extractor, called *backbone*, followed by two Fully Connected (FC) layers, called *head*. Section 5.5 experiments versions of NoiseBreaker with MobileNetV2 [How+17], DenseNet121 [Hua+17] and ResNet51V2 [He+16b] backbones. From these results, in NoiseBreaker, the backbone of MobileNetV2 is responsible for extracting features out of the images. The two head FC layers have respectively 1024 and n units, where n is the number of classes of the classification problem. The input image resolution is chosen to be 224×224 as in the original MobileNetV2 implementation. The first FC layer has ReLu activations while the second uses softmax activations to obtain outputs in $[0, 1]$, seen as confidence levels. The output of this second FC layer, passed through an argmax function, gives the label with the highest confidence level.

5.3.2 Gradual Denoising

A noisy image is given as an input to the classifier $\mathcal{C}$ trained to differentiate noise classes. $\mathcal{C}$ supplies a prediction η_i to $\mathcal{G}$, the gradual denoising block. $\mathcal{G}$ selects the corresponding denoiser

Class	Noise Type	Parameters
η_0	Gaussian ($\mathcal{N}$)	$\sigma_g = [0, 55]$
η_1	Speckle ($\mathcal{S}$)	$\sigma_s = [0, 55]$
η_2	Uniform ($\mathcal{U}$)	$s = [-50, 50]$
η_3	Bernoulli ($\mathcal{B}$)	$p = [0, 0.4]$
η_4	Poisson ($\mathcal{P}$)	$\emptyset$
η_5	Clean ($\emptyset$)	$\emptyset$

Table 5.1 – List of classes for NBreaker-N and the noise type and level they represent.

$\mathcal{D}(\eta_i)$ that restores the image. $\mathcal{D}(\eta_i)$ is a *primary denoiser*, specialized for the denoising of the η_i noise class. A primary denoiser is a denoiser trained with pairs of clean and synthetic noisy images from the η_i class. The process ($\mathcal{C}$ followed by $\mathcal{G}$) iterates n times until $\mathcal{C}$ detects the class *clean*. The architectures of $\mathcal{C}$ and $\mathcal{G}$ are linked because they share the same noise classes and operate together. An important property of NoiseBreaker is that it can be extended by adding a class to $\mathcal{C}$ and the corresponding denoiser to $\mathcal{G}$. An example of gradual denoising is given in Figure 5.2 where two noise classes are successively detected and treated.

5.3.3 Noise Classes and Primary Denoisers

The definition of the classes is important in NoiseBreaker. The classes are selected to fit a large set of mixture noises. The primary noises used in these classes represent the most used noise distributions in the literature. The five primary noises presented in Section 3.2.1 are used in NoiseBreaker. For the following experiments, the noise parameters σ_g, σ_s, s and p are randomly drawn out of the considered ranges to prove the adaptability of the method.

Unlike [LSJ20], with NoiseBreaker it is possible to use class refinement to differentiate primary noises with same types but different noise levels. The efficiency of class refinement is assessed in Section 5.5. Three versions of NoiseBreaker are mentioned in the following. First, NBreaker-N (for NoiseBreaker-Naive) is an implementation where each noise type is associated to a unique class that covers the entire noise level range. The description of the classes considered for NBreaker-N is given by Table 5.1.

A second, main version, simply called NoiseBreaker, uses the same noise types as NBreaker-N but makes use of class refinement, the noise level ranges are split into smaller ones. Table 5.2 describes the classes of NoiseBreaker. refining the classes fosters more tailored primary denoisers. On the other hand, refinements increase the classification problem complexity, as well as the number of primary denoisers to be trained. A third version, called

Class	Noise Type	Parameters	Denoiser
$\eta_{0,0}$	Gaussian ($\mathcal{N}$)	$\sigma_g = [0, 15]$	MWCNN [Liu+18]
$\eta_{0,1}$		$\sigma_g =]15, 35]$	
$\eta_{0,2}$		$\sigma_g =]35, 55]$	
$\eta_{1,0}$	Speckle ($\mathcal{S}$)	$\sigma_s = [0, 15]$	SGN [Gu+19]
$\eta_{1,1}$		$\sigma_s =]15, 35]$	
$\eta_{1,2}$		$\sigma_s =]35, 55]$	
$\eta_{2,0}$	Uniform ($\mathcal{U}$)	$s = [-10, 10]$	SRResNet [Led+17]
$\eta_{2,1}$		$s = [-50, 50]$	
η_3	Bernoulli ($\mathcal{B}$)	$p = [0, 0.4]$	SRResNet [Led+17]
η_4	Poisson ($\mathcal{P}$)	$\emptyset$	SRResNet [Led+17]
η_5	Clean ($\emptyset$)	$\emptyset$	$\emptyset$

Table 5.2 – List of classes for NoiseBreaker, the noise type and level they represent. The denoiser related to a class is mentioned, according to Table 5.3.

NBreaker-S (for NoiseBreaker-Same), is proposed to study the architecture distinction between primary denoisers. For this version, all primary denoisers use Multi-level Wavelet Convolutional Neural Network (MWCNN) architectures contrary to NoiseBreaker that authorizes different architectures to be used for different classes. Lastly, a version named NBreaker-I (for NBreaker-Inverse) is introduced for further assessment of the proposed gradual denoising. NBreaker-I uses exactly the same classes as NoiseBreaker but denoises the samples in the exact inverse order of corruption, without using the decision of the classifier. In this version, the noise mixture composition is considered as known and the denoisers are manually employed to test their performance independently from the performance of the classifier.

The choice of the primary denoisers is also of primary concern. NoiseBreaker authorizes different denoising architectures for different noise classes. For each class, the best denoising architecture is selected through a benchmark study. The effectiveness of authorizing different denoising architectures for different noise types and the benchmark study used for selection of primary denoisers are discussed in Section 5.4. The benchmarking study has been performed using the OpenDenoising benchmark tool. The results show that for a given noise type, the same denoising architecture can be used for all classes. Following the benchmark study of Table 5.3, NoiseBreaker uses MWCNN [Liu+18] for Gaussian noise type, Self-Guided Network (SGN) [Gu+19] for Speckle and Super-Resolution Residual Network (SRResNet) [Led+17] for Bernoulli, Poisson and Uniform, as summarized in Table 5.2.

Class	MWCNN [Liu+18]	SGN [Gu+19]	SRResNet [Led+17]	DnCNN [Zha+17]
$\eta_{0,0}$	**34.74**	34.20	32.20	30.57
$\eta_{0,1}$	**28.45**	28.15	28.26	26.84
$\eta_{0,2}$	**26.70**	25.57	25.58	24.90
$\eta_{1,0}$	39.53	**39.97**	39.19	38.54
$\eta_{1,1}$	31.73	**32.28**	31.38	29.39
$\eta_{1,2}$	28.70	**29.10**	28.15	27.32
$\eta_{2,0}$	35.03	35.12	**37.75**	27.47
$\eta_{2,1}$	27.63	27.86	**28.18**	25.82
η_3	31.37	31.28	**33.23**	29.82
η_4	31.13	29.00	**31.21**	30.59

Table 5.3 – Benchmark study between MWCNN, SGN, SRResNet and DnCNN denoising architectures for each noise class of NBreaker (Table 5.2). Evaluation dataset is made of ImageNet samples unseen in the training. Results suggest that the ranking between architectures differs depending on the noise content. Also, a unique architecture can be used for all noise levels of a given noise type.

5.4 Experiments: Noise Mixture Removal

This section presents the evaluation of NoiseBreaker. Results are compared to the human-expert method BM3D, N2V as a non-supervised method, [LSJ20] as state of the art of classification-based mixture denoising, and finally to MWCNN as a fully-supervised end-to-end denoiser. Denoising results are shown on two datasets including the recent high resolution DIV2K [AT17]. Data and experimental settings for this section are exposed first followed by results. Noise analysis and gradual denoising are evaluated separately. Finally, discussions on error cases are conducted.

5.4.1 Data and Experimental Settings

Noise Analysis The noise classifier $\mathcal{C}$ is fine-tuned using a subset of ImageNet [Den+09]. The first 10000 images of the ImageNet evaluation set are extracted, among which 9600 serve for training, 200 for validation and 200 for evaluation. To create the classes, the images are first re-scaled to 224×224 to fit the fixed input shape. Images are then noised according to their destination class, described in Table 5.2. The training data (ImageNet samples) is chosen to keep a similar underlying content in the images, with respect to those of the backbone pre-training. Similar content with corruption variations enable to concentrate the classification on the noise and not on the semantic content. To avoid fine-tuning with the same images as the

Dataset	Denoiser	C_0	C_1	C_2	C_3	C_4	C_5
BSD68 Grayscale	Noisy	12.09/0.19	16.98/0.36	18.21/0.42	14.05/0.28	13.21/0.24	24.96/0.73
	BM3D [Dab+07]	21.49/0.54	24.00/0.61	24.28/0.62	22.30/0.56	22.05/0.56	24.95/0.65
	Noise2Void [KBJ19]	22.13/0.60	20.47/0.36	20.55/0.35	24.06/0.68	23.70/0.66	25.08/0.66
	Liu et al. [LSJ20]	21.04/0.52	25.96/0.74	27.17/0.82	27.11/0.80	26.83/0.77	27.52/0.83
	NoiseBreaker (Ours)	**23.68/0.68**	**26.33/0.82**	**27.19/0.84**	**29.94/0.90**	**29.70/0.91**	**30.85/0.92**
BSD68 RGB	Noisy	11.71/0.18	16.98/0.36	18.05/0.40	13.00/0.24	13.01/0.24	25.15/0.74
	BM3D	21.24/0.57	24.72/0.66	24.88/0.66	21.96/0.59	22.00/0.59	25.73/0.70
	Noise2Void	13.34/0.17	17.60/0.31	18.30/0.34	15.45/0.24	15.63/0.25	25.27/0.66
	Liu et al.	21.02/0.60	23.56/0.68	24.15/0.69	18.84/0.51	19.23/0.53	20.13/0.54
	NoiseBreaker (Ours)	**21.88/0.71**	**26.81/0.82**	**26.58/0.82**	**25.45/0.81**	**25.20/0.80**	**29.77/0.88**
DIV2K Grayscale	Noisy	11.69/0.15	16.80/0.32	18.18/0.37	12.36/0.18	12.95/0.22	24.47/0.70
	BM3D	22.05/**0.64**	25.36/0.71	26.23/0.71	22.36/0.65	22.75/0.65	27.20/0.76
	Noise2Void	22.24/**0.64**	21.01/0.40	21.08/0.39	22.87/0.66	24.78/0.71	26.47/0.69
	NoiseBreaker (Ours)	**22.88**/0.61	**26.81/0.83**	**28.35/0.87**	**25.13/0.73**	**35.99/0.97**	**32.23/0.94**
DIV2K RGB	Noisy	11.33/0.14	17.14/0.33	19.02/0.40	12.93/0.21	13.07/0.22	25.32/0.72
	BM3D	21.36/0.63	26.05/0.73	26.85/0.74	22.49/0.67	22.74/0.67	27.99/0.79
	Noise2Void	13.14/0.12	17.80/0.26	19.14/0.31	15.02/0.19	15.78/0.22	25.45/0.63
	NoiseBreaker (Ours)	**22.35/0.71**	**27.57/0.84**	**27.68/0.83**	**25.62/0.81**	**26.48/0.83**	**29.82/0.87**

Table 5.4 – Average PSNR(dB)/SSIM results of the proposed and competing methods for grayscale and RGB denoising with the noise mixtures of Table 5.5 on BSD68 and DIV2K. Bold value indicates the best performance.

	Noise 1	Noise 2
C_0	$\mathcal{N}([0, 55])$	$\mathcal{B}([0, 0.4])$
C_1	$\mathcal{N}([0, 55])$	$\mathcal{S}([0, 55])$
C_2	$\mathcal{N}([0, 55])$	$\mathcal{P}$
C_3	$\mathcal{B}([0, 0.4])$	$\mathcal{S}([0, 55])$
C_4	$\mathcal{B}([0, 0.4])$	$\mathcal{P}$
C_5	$\mathcal{S}([0, 55])$	$\mathcal{P}$

Table 5.5 – Definition of the noise mixtures used for evaluation. Noise 1 is applied first on the sample followed by Noise 2.

pre-training, the ImageNet evaluation set is taken. The weights for the backbone initialisation, pre-trained on ImageNet, are taken from the official Keras MobileNetV2 implementation. In this version, NoiseBreaker contains 11 classes. Thus, the second layer of the head has accordingly 11 units. The classifier is trained for 200 epochs with a batch size of 64. Optimisation is performed through an Adam optimizer with learning rate 5.10^{-5} and default settings for other parameters [KB14]. The optimisation is driven by a categorical cross-entropy loss. A step scheduler halves the learning rate every 50 epochs.

Gradual Denoising For primary denoisers training, the first 9600 images of the ImageNet evaluation set are extracted and corrupted according to the classes mentioned in Table 5.2. For evaluation, the 68 images of the BSD68 [Mar+01] benchmark are used as well as the 100

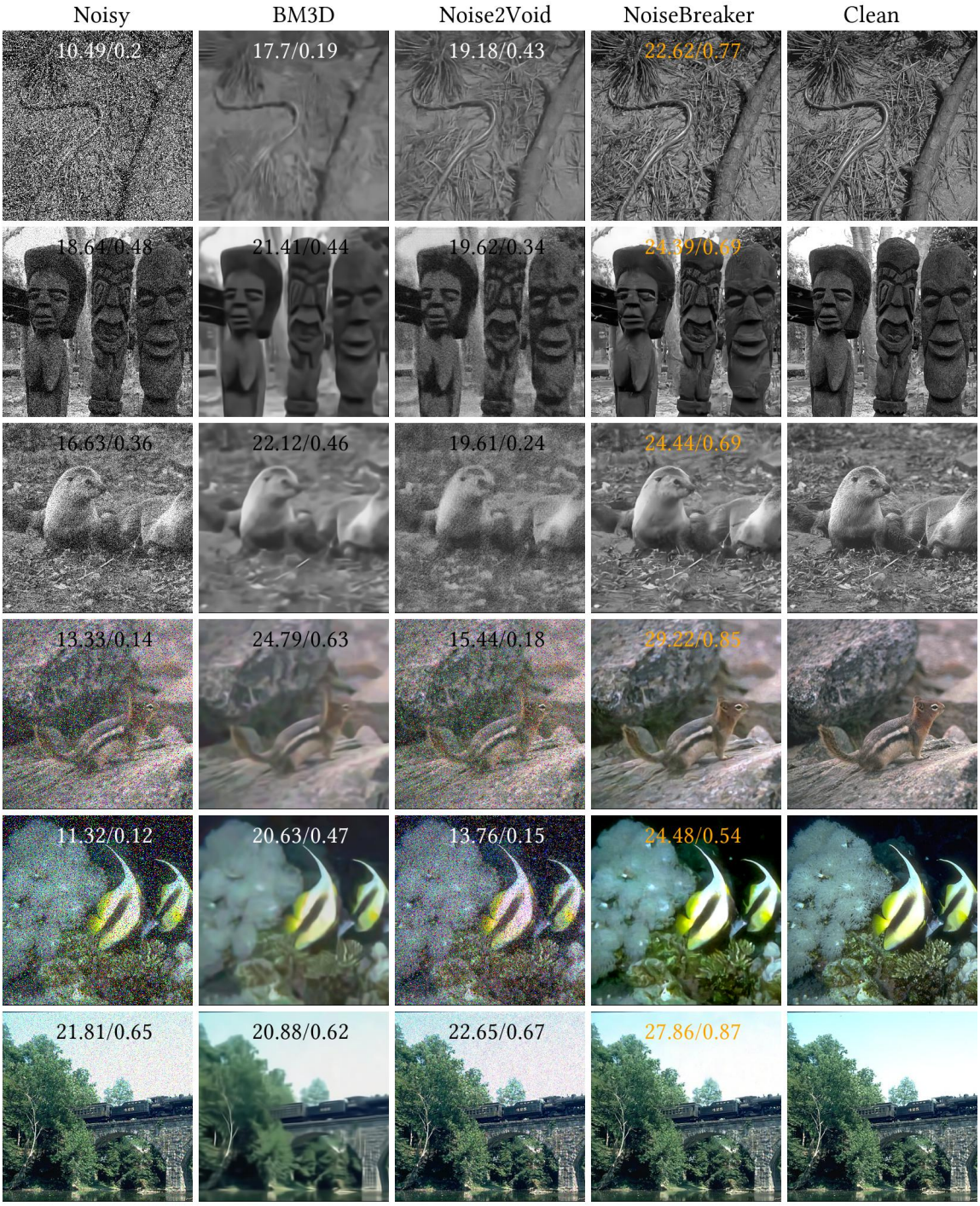
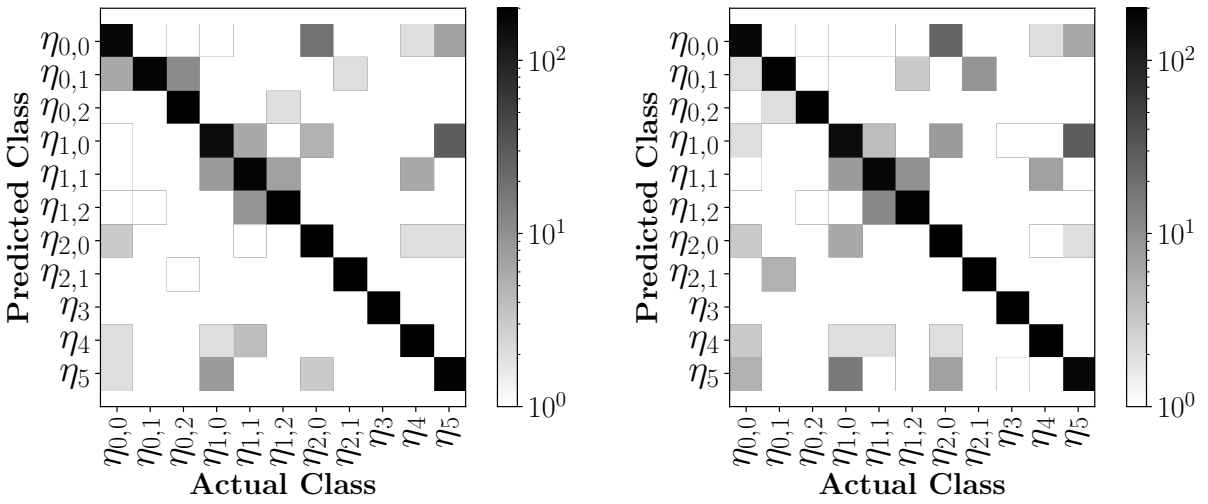


Figure 5.3 – Qualitative results on BSD68 dataset. Samples from the left column are corrupted with mixture C_0 to C_6, respectively. Images are tagged with Peak Signal to Noise Ratio (PSNR)/Structural Similarity (SSIM) values and the best PSNR value for an image is yellow colored. Note that samples are chosen to be representative of the average PSNR of their corresponding classes. [LSJ20] not displayed because no available code. **Better viewed on screen.**

evaluation images of DIV2K [AT17]. Six different sequential mixtures corrupt these images. For comparison purposes, the noise types of [LSJ20] are selected. Noise levels are shown in Table 5.5. The primary noises are either AWGN with $\sigma_g \in [0, 55]$, Bernoulli noise with $p \in [0, 0.4]$, white Speckle noise with $\sigma_s \in [0, 55]$ or Poisson noise. σ_g, σ_s and p values are randomly picked. This random draw is used to prove the adaptability of our method to variable noise levels. The size of BSD68 samples is either 321×481 or 481×321 and the DIV2K samples have a dimension larger than 2040×1550. When evaluating gradual denoising, $\mathcal{C}$ predicts the noise class using a patch of size 224×224 cropped from the image to be denoised. Because of their architectures, the input resolution of MWCNN and SGN are constrained. To bypass this issue, all samples are cropped to the closest resolution that satisfies the constraints. The training of $\mathcal{G}$ comes down to the training of its primary denoisers. NoiseBreaker uses off-the-shelf architectures for primary denoisers (Table 5.2) selected from results of a benchmark study displayed in Table 5.3. Results show that a unique architecture can be used for the different classes of a given noise type. From results of Table 5.3, the primary denoiser architectures are MWCNN [Liu+18] for Gaussian noise, SGN [Gu+19] for Speckle and SRResNet [Led+17] for Bernoulli, Poisson and Uniform. These denoisers are trained with the parameters mentioned in their original papers. Only the training data differ since it is made of the corresponding primary noise (according to Table 5.2).

Compared methods Our solution is evaluated in comparison with BM3D [Dab+07], N2V [KBJ19], MWCNN [Liu+18] and Liu et al. [LSJ20]. Although a potential competitor, G2G [CPM19] is evaluated on other noise mixtures and no code is publicly available yet. BM3D is a human-expert method. It is not trained but requires σ, the standard deviation of the noise distribution. $\sigma = 50$ is chosen since it performs the best over the range of noise mixtures used for evaluation. N2V is a self-supervised denoiser. Training is carried out with the publicly available code and the original paper strategy, and the data is corrupted with the synthetic evaluation mixture. For [LSJ20], results are extracted from the paper tables and given only for BSD68 as no code is publicly available. MWCNN is used as a reference supervised method, trained on the noise mixtures themselves. Comparison to supervised learning is unfair to NoiseBreaker because NoiseBreaker is never exposed to noise mixtures during training. NoiseBreaker discovers the mixtures only when inferring. MWCNN is chosen as the supervised reference because it performs the best on average over the classes of the benchmark study (Table 5.3). It is worth mentioning that N2V and MWCNN models are trained for each evaluation noise mixture while NoiseBreaker handles all evaluation classes with the same configuration and neither the classifier nor the denoisers are exposed to mixture noise dur-



(a) NBreaker-Grayscale

(b) NBreaker-RGB

Figure 5.4 – Log scale confusion matrices of noise classification. Classes content is described in Table 5.2. (a) and (b) are the results for grayscale and RGB classification, respectively.

ing training. The comparison is done based on values of SSIM and PSNR in dB and shown in Table 5.4. Qualitative results are given in Figure 5.3.

5.4.2 Results

Noise Analysis Figure 5.4 presents the results of NoiseBreaker classifiers through confusion matrices in log scale. The evaluation on 2200 images unseen during training (200 for each class) gives an accuracy score of 93% for grayscale images and 91% for RGB images.

The most recurrent error (29% of all the errors for grayscale, 41% for RGB) is the misclassification of low noise intensity images, classified as clean (η_5) or as other low intensity noise ($\eta_{0,0}$, $\eta_{1,0}$, $\eta_{2,0}$). These effects can be observed in Figure 5.4 (a) and (b) at ($\eta_{0,0}$, $\eta_{1,0}$), ($\eta_{2,0}$, $\eta_{0,0}$) or ($\eta_{1,0}$, η_5), where the first and second indexes represent the actual class and the predicted class, respectively. Clean images are sometimes classified as having low intensity noise (26% of all the errors for grayscale, 21% for RGB). Such errors can be seen at (η_5, $\eta_{0,0}$) and (η_5, $\eta_{1,0}$). The impacts of such misclassification are evoked in Section 5.4.3. Confusions also occur between different noise levels within a unique noise type, e.g. ($\eta_{1,1}$, $\eta_{1,2}$). They represent 33% of all errors for grayscale and 22% for RGB. These latter errors have low impact on the final aim, namely an efficient denoising. Indeed, this type of misclassification is caused by a noise level at the edge between two classes. The selected denoiser is then not optimal for the actual noise

	C_0	C_1	C_2	C_3	C_4	C_5	Avg.
NoiseBreaker	23.68	26.33	**27.19**	29.94	29.70	30.85	27.95
MWCNN	**26.22**	**26.84**	26.77	**31.28**	**30.59**	**31.62**	**28.89**

Table 5.6 – Average PSNR in dB over the noise mixtures of Table 5.5 applied to grayscale BSD68. Denoisers are NoiseBreaker and fully-supervised MWCNNs trained on the mixtures. Note that a unique instance of NoiseBreaker handles all mixtures while a model of MWCNN is trained independently for each mixture.

level but it addresses the correct noise type. Next paragraph evaluates the performance of the classification when associated to the gradual denoiser.

Gradual Denoising Table 5.4 compares denoising performance of NoiseBreaker BM3D and N2V, and [LSJ20] on the noise mixtures of Table 5.5. Methods are evaluated on BSD68 and DIV2K both in grayscale and RGB. Scores for noisy input images are given as baseline.

When evaluating the methods on BSD68 grayscale samples, NoiseBreaker operates 2dB higher in PSNR than the competing method of [LSJ20], on average over the six mixtures. BM3D and N2V suffer from being applied to noise mixtures far from Gaussian distributions and show average PSNRs 5dB under NoiseBreaker. Note that NoiseBreaker, without previous contact with the noise mixtures, outperforms N2V that is trained on each mixture. PSNR scores on DIV2K grayscale match with results on BSD68, with NoiseBreaker outperforming N2V by 5dB. For SSIM scores, NoiseBreaker leads on BSD68 with a score of 0.85, 0.13 higher than [LSJ20] and 0.54 higher than BM3D and N2V. On DIV2K, NoiseBreaker has an average SSIM score of 0.83, 0.23 higher than BM3D. These quantitative results are confirmed by qualitative results of the first three rows of Figure 5.3.

When considering RGB denoising, a first observation is that N2V does not denoise as expected using the code and recommendations made available by authors. Indeed, N2V on BSD produces an average PSNR only 1.3dB higher than the noisy samples. Another observation is that for C_5 on BSD68, the authors of [LSJ20] give, in their paper, a score 5dB under the PSNR of noisy samples. NoiseBreaker, with an average PSNR of 25.95dB over the six mixtures on BSD68, operates 4.8dB higher than [LSJ20]. In terms of SSIM, NoiseBreaker shows an average score of 0.81, a 0.38 increase over [LSJ20]. The results follow the same trend on DIV2K with NoiseBreaker reaching an average PSNR of 26.59dB and SSIM score of 0.82, respectively 2dB and 15% higher than BM3D.

To further assess the results of NoiseBreaker, these are compared to the results of a fully supervised method that removes the noise mixtures as a whole. MWCNN architecture is chosen as the reference supervised denoiser and an independent model is trained for each evaluated

mixture. Table 5.6 presents the average PSNR over the evaluation noise mixtures for NoiseBreaker and MWCNNs models. On average, NoiseBreaker operates 0.95dB under the fully supervised MWCNNs. The result for $\mathcal{C}_0$ show a 2.54dB loss over fully supervised MWCNN. This mixture is the most challenging for a denoiser that does not have supervision on the entire mixture. Indeed, it contains strong corruption that makes complicated a gradual approach. This is further studied in Section 5.4.3. For classes $\mathcal{C}_1$ to $\mathcal{C}_5$, NoiseBreaker keeps up with supervised denoisers with an average loss of 0.6dB and a better score on $\mathcal{C}_2$. Note that in case of an unknown mixture, MWCNN fully supervised training would not be possible. On the contrary, NoiseBreaker handles the six evaluation noise mixtures with an only configuration and adapts to new mixture compositions. Figure 5.3 shows subjective results both for grayscale and RGB samples. This figure confirms the fact that N2V underperforms on RGB images and the blurring effect of BM3D. On the other hand, NoiseBreaker produces samples with relatively low noise levels, clear edges and contrasts.

As a conclusion on noise mixture removal experiments, we show that NoiseBreaker keeps up with the supervised denoiser MWCNN and outperforms the state of the art N2V method and the related proposal of Liu et al [LSJ20].

5.4.3 Errors and Limitations

The following evaluates errors and limitations of NoiseBreaker.

When considering noise mixtures, NoiseBreaker first tackles the dominating noise detected by the classifier. When a primary noise is particularly stronger than the others, the first denoiser is tailored to remove a large corruption and its output image strongly differs from its input. Thus, the intermediate restored image deviates in terms of noise distribution from what is expected to be the second noise in the mixture. When such deviation happens, two problematic behaviors are observed. In the first case, the second noise classification fails to detect the noise distribution and chooses a wrong denoiser that degrades the image. This failure results in a major quality loss (Figure 5.5a). In the second case, the classifier predicts the sample as clean and no further denoising is performed, leaving some noise in the image (Figure 5.5b). In the best case, the first denoiser alone efficiently removes the two primary noises (Figure 5.5c). As a general observation, NoiseBreaker operates with difficulty on heavy degradations such as the one of evaluation class C_0. This constitutes a promising research direction for improving NoiseBreaker.

(a) (b) (c)

Figure 5.5 – Examples of BSD68 grayscale samples for which the gradual denoising diverges from the expected inverse corruption order. (a) and (b) are corrupted by C_0, and (c) by C_5. In (a), a second wrong denoiser is applied and damages the sample. In (b), the *clean* class has been detected but a low strength Gaussian noise is still present. In (c), the first Speckle noise removal has also removed most of the Poisson Noise component of the mixture.

5.5 Experiments: Ablation Study

In this section, experiments are conducted separately on core features of NoiseBreaker. Experiments evaluate the respective impact on the results of NoiseBreaker of the noise classification and of the primary denoisers. For the ablation study, only grayscale images are used. The training dataset is the same as in Section 5.4.

5.5.1 Impact of Classification on NoiseBreaker

NoiseBreaker performance depends on the performance of its noise classifier and on the capacity of each primary denoiser to remove its primary noise distribution. The following evaluates the impact of noise classification on the performance of NoiseBreaker.

5.5.1.1 Backbone Choice

Three backbones are compared to justify the choice of MobileNetV2 as the NoiseBreaker backbone. The compared backbones are MobileNetV2, DenseNet121 and Resnet50V2. These three backbones are chosen for their limited complexity and good performance on the ILSVRC validation set. Table 5.7 shows that the performances of the three backbones are close to each other. Resnet50V2 and DenseNet121 reach an accuracy of 0.94 while MobileNetV2 scores 0.93. While being close in accuracy, MobileNetV2 has respectively 10 and 3 times less parameters than Resnet50V2 and DenseNet121.

Backbone	Number of Parameters	Accuracy
ResNet50V2 [He+16b]	$23,564,800$	**0.94**
DenseNet121 [Hua+17]	$7,037,504$	**0.94**
MobileNetV2 [How+17]	$\mathbf{2,257,984}$	0.93

Table 5.7 – Comparison of backbones as noise classifier of NoiseBreaker. MobileNetV2 is chosen because it is the least complex while having competitive accuracy on the noise analysis evaluation dataset of Section 5.4.1.

Backbone	C_0	C_1	C_2	C_3	C_4	C_5	Avg.
ResNet50V2	**23.89**	**26.48**	**27.32**	29.81	29.55	**30.85**	**27.98**
MobileNetV2	23.68	26.33	27.19	**29.94**	**29.70**	**30.85**	27.95

Table 5.8 – Average PSNR in dB over the classes of Table 5.5 for two backbones to be used in the noise classifier of NoiseBreaker. MobileNetV2 is chosen for its good performance-to-cost ratio.

Table 5.8 compares NoiseBreaker performances using ResNet50V2 and MobileNetV2 as classifier backbones. On average over the six evaluation classes, NoiseBreaker with MobileNetV2 backbone operates only 0.03dB under the version with ResNet50V2. NoiseBreaker with MobileNetV2 backbone performs better for the last three mixtures and is only 0.21dB below on the others. The classifier is used at each iteration of the gradual denoising process of NoiseBreaker. These results validate the use of MobileNetV2 as backbone since its performances is only 0.03dB under the one of ResNet50V2 while being ten times lighter.

5.5.1.2 Classification Order

NBreaker-I is an *ideal* version of NoiseBreaker without a classifier for noise analysis. The gradual denoising is conducted in the exact inverse order of corruption by primary noises. For instance, when using NBreaker-I on the evaluation mixture C_4, the primary denoiser for the class η_4 is applied, followed by the primary denoiser for the class η_3. The performance of NBreaker-I is shown in Table 5.9. It should be noted that removing the noises in the exact inverse order of corruption does not improve the results obtained by NoiseBreaker. NoiseBreaker gives better results for four of the six evaluation classes, and operates 0.75dB over NBreaker-I on average over the six classes. These results demonstrate that NoiseBreaker is robust to the wrong decisions of its classifier. Depending on the noise mixture composition and primary denoisers capacities, it happens that the second corruption is removed by the first denoiser. In such a case, the NoiseBreaker classifier predicts the *clean* class and no further processing

is conducted. NBreaker-I performance particularly show that an additional denoising step in this specific case damages the image.

Denoiser	C_0	C_1	C_2	C_3	C_4	C_5	Avg.
NBreaker-I	**23.96**	25.02	25.05	**30.07**	28.77	30.35	27.20
NoiseBreaker	23.68	**26.33**	**27.19**	29.94	**29.70**	**30.85**	**27.95**

Table 5.9 – Average PSNR in dB over the classes of Table 5.5 for NBreaker-I and NoiseBreaker.

5.5.2 Impact of Primary Denoisers

NoiseBreaker employs primary denoisers depending on the noise classifier prediction. The primary denoisers are the actuators of the image restoration. The following study evaluates two points that distinguish the proposed method from state of the art, namely class refinement and denoising architecture tailoring to each noise type.

5.5.2.1 Noise Class Refinement

NBreaker-N is a version of NoiseBreaker that does not use class refinement. Each noise type is represented by a single class that includes the entire range of parameter values, when applicable. The results of the comparison between NoiseBreaker and NBreaker-N are shown in Table 5.10. On average over the six evaluation classes, NoiseBreaker operates 0.11dB higher than NBreaker-N and presents the best results on four classes. This gain, while moderate, shows that class refinement does improve denoising performance.

Denoiser	C_0	C_1	C_2	C_3	C_4	C_5	Avg.
NBreaker-N	22.80	**26.86**	**27.59**	29.86	29.65	30.29	27.84
NoiseBreaker	**23.68**	26.33	27.19	**29.94**	**29.70**	**30.85**	**27.95**

Table 5.10 – Average PSNR in dB over the classes of Table 5.5 for NBreaker-N and NBreaker.

5.5.2.2 Architecture Distinction

Table 5.11 presents the comparison between NoiseBreaker and NBreaker-S. In NBreaker-S, all the primary denoisers use the MWCNN architecture. MWCNN is chosen here as the reference architecture since it proves to perform better on average over the classes of the benchmark study of Table 5.3. The results of Table 5.11 show a significant gain, as NoiseBreaker

operates on average 0.21dB over NBreaker-S. The denoiser distinction makes NoiseBreaker perform better on four of the six evaluation classes.

Denoiser	C_0	C_1	C_2	C_3	C_4	C_5
NBreaker-S	22.77	**26.65**	**27.54**	29.50	29.43	30.54
NoiseBreaker	**23.68**	26.33	27.19	**29.94**	**29.70**	**30.85**

Table 5.11 – Average PSNR in dB over the classes of Table 5.5 for NoiseBreaker using different denoising architectures for each noise type and a version of NoiseBreaker called NBreaker-S using MWCNN for all primary denoisers.

5.6 Conclusion

This chapter has introduced a gradual image denoising strategy called NoiseBreaker. NoiseBreaker iteratively detects the image dominating noise using a trained classifier with an accuracy of 93% and 91% for grayscale and RGB samples, respectively. Under the assumption of grayscale sequential noise mixtures, NoiseBreaker performs 0.95dB under the supervised MWCNN denoiser without being trained on any mixture noise. Neither the classifier nor the denoisers are exposed to mixture noise during training. NoiseBreaker operates 2dB over the gradual denoising of [LSJ20] and 5dB over the state of the art self-supervised denoiser *Noise2Void*. When applied to Red Green Blue (RGB) samples, NoiseBreaker operates 5dB over [LSJ20] while *Noise2Void* underperforms. Moreover, this chapter has demonstrated that making noise analysis to guide the denoising is not only efficient on noise type, but also on noise intensity.

This chapter has showed the practicality of NoiseBreaker on six different synthetic noise mixtures. Future works include the application of NoiseBreaker to noisy images corrupted with deeper noise mixtures, i.e. made of more than two primary noises. While the hypothesis has been made in Chapter 2 that the eavesdropping noise is a sequential mixture, NoiseBreaker has not proven to be efficient on its removal.

Next chapter proposes to automate the interpretation of eavesdropped samples using a Deep Learning (DL)-based textual retrieval and en according custom metric. A dataset crafted for the task is also introduced.

CHAPTER 6

Direct Interpretation of Eavesdropped Images

Chapter Contents

6.1 Introduction

As introduced in Chapter 2, all electronic devices produce Electro Magnetic (EM) emanations that not only interfere with radio devices but also compromise the data handled by the information system. A third party may perform a side-channel analysis and recover the original information, hence compromising the system privacy. While pioneering work of the domain focused on analog signals [Van85], recent studies extend the eavesdropping exploit using an EM side-channel attack to digital signals and embedded circuits [Kuh13]. The attacker's profile is also taking on a new dimension with the increased performance of Software-Defined Radio (SDR). With recent advances in radio equipment, an attacker can leverage advanced signal processing to further stretch the limits of the side-channel attack using EM emanations [Gen+18]. With the fast evolution of deep neural networks, an attacker can extract patterns or even the

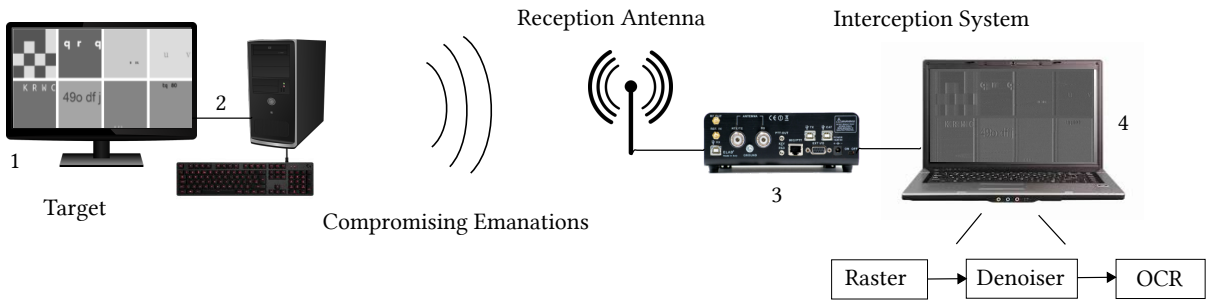


Figure 6.1 – Experimental setup: the attacked system includes an eavesdropped screen (1) displaying sensitive information. It is connected to an information system (2). An interception chain including an SDR receiver (3) sends samples to a host computer (4) that implements our proposed automated interpretation.

full structured content of the intercepted data with a high degree of confidence and a limited execution time.

In this chapter, a learning-based method is proposed to not only denoise eavesdropped images but also interpret them. To reduce the scope, the method focuses on textual images. In fact, we consider that confidential documents mainly contain text. The method is based on the specialization of Mask R-CNN [He+17] as a denoiser and classifier. A complete system is demonstrated, embedding SDR and deep-learning, that detects and recovers leaked information at a distance of several tens of meters. It provides an automated solution where the data is interpreted directly. The solution is compared to other system setups.

The chapter is organized as follows. **??** presents existing methods to recover information from EM emanations. Section 6.2 describes the proposed method for automatic character retrieval. Experimental results and detailed performances are exposed in Section 6.3. Section 6.4 introduces an open dataset of eavesdropped natural images and proposes some experiments to characterize the corruptions the dataset contains. Section 6.5 concludes the chapter.

This chapter contributions have been published in : *F. Lemarchand, C. Marlin, F. Montreuil, E. Nogues, and M. Pelcat, « Electro-Magnetic Side-Channel Attack Through Learned Denoising and Classification », in 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, p. 2882–2886.*

6.2 Proposed Side-Channel Attack

6.2.1 System Description

Figure 6.1 shows the proposed end-to-end solution. The system is the same as the one proposed in Chapter 2 but the host computer implements our automated interpretation method. The interception system automatically reconstructs leaked visual information from compromising emanations. The setup is composed of two main elements. At first the antenna and SDR processing capture in the Radio Frequency (RF) domain the leaked information originating from the displayed video. Then, the demodulated signal is processed by a host computer, recovering a noisy version of the original image [Kuh13] leaving room for advanced image processing techniques. On top of proposing an end-to-end solution from capturing to the data itself, the system includes machine interpretation. It captures compromising signals and recognizes automatically the leaked data assuming textual information. A first step based on a Mask R-CNN (Mask R-CNN) architecture embeds the following: denoising, segmentation, character detection/localization, and character recognition. A second step post-processes the Mask R-CNN output. A Hough transform is done for text line detection and a Bitap algorithm [Mye99] is applied to approximate match information. This setup detects several forms of compromising emanations (analog or digital) and automatically triggers an alarm if critical information is leaking. Next sections detail how the method is trained and integrated.

6.2.2 Dataset Construction

A substantial effort has been made on building a process that semi-automatically generates and labels datasets for supervised training. Each sample image is made up of a uniform background on which varied characters are printed. Using that process, an open data corpus of 123.610 labeled samples, specific to the problem at hand, has been created to further be used as training, validation and test datasets. This dataset is available online [1] to train denoiser architectures in difficult conditions.

The proposed setup, to be trained, denoises the intercepted sample images and extracts their content, i.e. the detected characters and their positions. The input space that should be covered by the training dataset is large and three main types of interception variability can be observed. Firstly, interception induces an important loss of the information originally existing in the intercepted data. The noise level is directly linked to the distance between the

1. https://github.com/opendenoising/interception_dataset

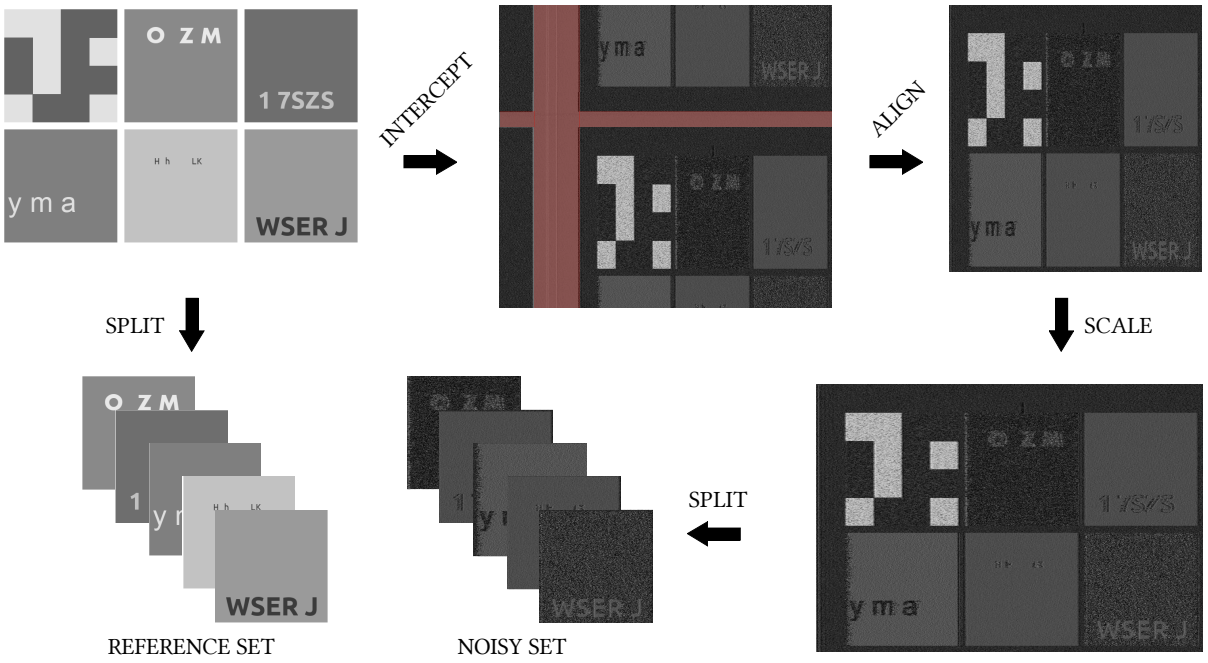


Figure 6.2 – A reference sample is displayed on the target screen (top-left). The interception module outputs uncalibrated samples. Vertical and horizontal porchs (red) helps alignment and porch withdrawal (top-right). Samples are rescaled and split into patches to obtain the same layout than the reference set.

antenna and the target. Several noise levels are generated by adding RF attenuation after the antenna. That loss itself causes inconsistencies in the rasterizing stage. Secondly, EM emanations can come from different sources, using different technologies, implying in turn different intercepted samples for the same reference image. The dataset covers Video Graphics Array (VGA), Display Port (DP)-to-Digital Visual Interface (DVI) and High-Definition Multimedia Interface (HDMI) cables and connectors. Besides this unwanted variability, a synthetic third type of variability is introduced to solve the character retrieval problem. Many different characters are introduced in the corpus to be displayed on the attacked screen. They range from 11 to 70 points in size and they are both digits and letters, and letters are both upper and lower cases. Varied fonts, character colors and background colors, as well as varied character positions in the sample are used. Considering these different sources of variability, the dataset is built trying to get an equi-representation of the different interception conditions.

The choice has been made to display on the target screen a sample containing patches of size 256×256 pixels (top-left image of Figure 6.2). For building the dataset, having multiple patches speeds the process up because smaller samples can be derived from a single screen interception and more variability can be introduced in the dataset. The main challenge when

creating the dataset lies in the sample acquisition itself. Indeed, once intercepted, the samples are not directly usable. The interception process outputs samples such as the one of Figure 6.2 (middle-top) where intercepted characters are not aligned (temporally and spatially) with respective reference samples. An automated method is introduced that uses the porches, artificially colored in red in Figure 6.2 (middle-top), to align spatially samples. Porches are detected using brute-force search of large horizontal and vertical gradients (to find vertical and horizontal porches, respectively). A validation step ensures the temporal alignment, based on the insertion of a QRCode in the upper-left patch. If the QRCode is similar between the reference and the intercepted image, the image patches are introduced in the dataset.

Data augmentation [MG18] is used to enhance the dataset coverage area. It is done onto patches to add variability into the dataset and reinforce its learning capacity. Conventional methods are applied to raw samples to linearly transform them (Gaussian and median blur, salt and pepper noise, color inversion and contrast normalization).

6.2.3 Implemented Solution to Catch Compromising Data

In order to automate the interception of compromising data, the Mask R-CNN has been turned into a denoiser and classifier. The implementation is based on the one proposed by W. Abdulla [2]. Other learning-based and expert-based signal processing methods, discussed in Section 6.3.2, are also implemented to assess the quality of the proposed framework. Mask R-CNN is a framework adapted from the previous Faster R-CNN [Ren+17]. The network consists of two stages. The first stage, also known as *backbone* network, is a *ResNet101* convolutional network [He+16a] extracting features out of the input samples. Based on the extracted features, a Region Proposal Network (RPN) proposes Region of Interests (RoIs). RoIs are regions in the sample where information deserves greater attention. The second stage, called *head* network, classifies the content and returns bounding box coordinates for each of the RoIs. The main difference between Faster R-CNN and Mask R-CNN lies in an additional Fully Convolutional Network (FCN) branch [SLD17] running in parallel with the classification and extracting a binary mask for each RoI to provide a more accurate localization of the object of interest.

Mask R-CNN is not originally designed to be used for denoising but rather for instance segmentation. However, it fits well the targeted problem. Indeed, the problem is similar to a segmentation where signal has to be separated from noise. As a consequence, when properly feeding a trained Mask R-CNN network with noisy samples containing characters, one obtains

2. https://github.com/matterport/Mask_RCNN

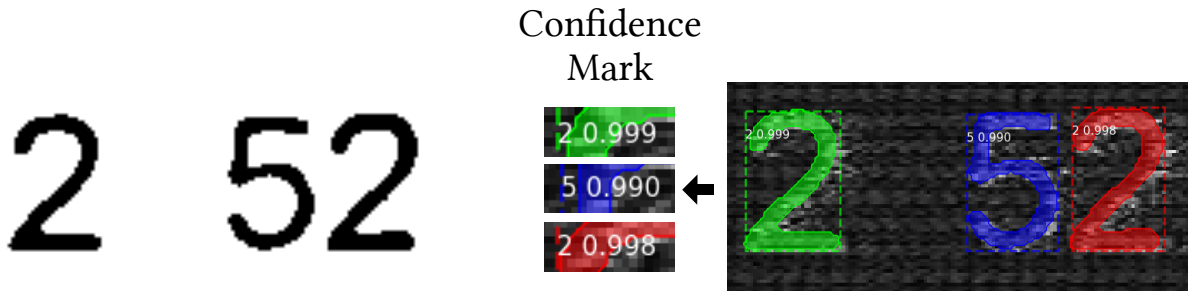


Figure 6.3 – The output of Mask R-CNN may be used in two ways. The segmentation can be drawn (left) and further processed by an Optical Character Recognition (OCR), or the Mask R-CNN classifier can directly infer the sample content (right) and propose some display and confidence information.

lists of labels (i.e. characters recognition), as well as their bounding boxes (characters localization) and binary masks representing the content of the original *clean* sample. The setup of the classification branch allows to be language-independent and to add classes other than characters.

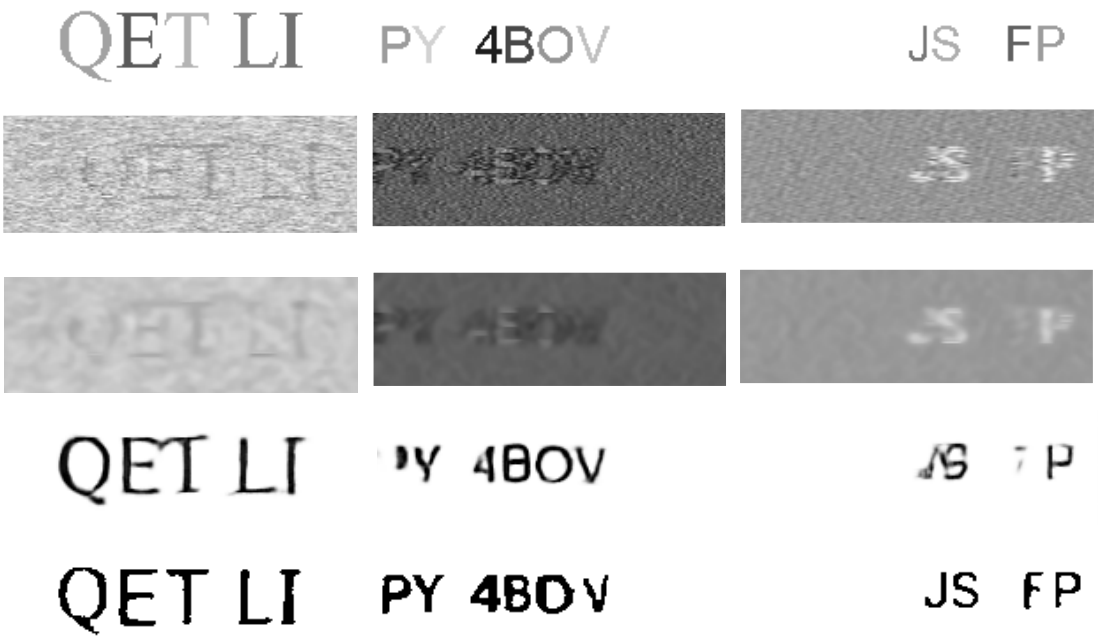
Two strategies can be employed to exploit Mask R-CNN components for the problem. The first idea is to draw the output masks of Mask R-CNN segmentation (Figure 6.3 left-hand side) and request an OCR to retrieve characters from the masks. A second possibility is to make use of the classification faculty of Mask R-CNN (Figure 6.3 right-hand side) and obtain a list of labels without using an OCR engine. The second method using the classifier of Mask R-CNN proves to be better in practice, as shown in Section 6.3.2.

The training strategy is to initialize the training process using pre-trained weights [Mah+18] for the MS COCO [Lin+14] dataset, made available by the authors of Mask R-CNN. First, the weights of the *backbone* are frozen and the *head* is trained to adapt to the application. Then, the weights of the *backbone* are relaxed and both *backbone* and *head* are trained together until convergence. This process is done to ensure the convergence and speed up training.

6.3 Experimental Results

6.3.1 Experimental Setup

The experimental setup is defined as follows: the eavesdropped display is 10 meters away from the interception antenna. A RF attenuator is inserted after the antenna. It ranges from 0 dB to 24 dB to simulate higher interception radius and generate a wide range of noise values.



(a) (b) (c)

Figure 6.4 – Three samples (left, middle, right) displayed at different stages of the interception/denoising pipeline. From top to bottom: the reference patch displayed on the screen; the patch after rasterization (raw patch); the patches denoised with Block-Matching 3D (BM3D), autoencoder and Mask R-CNN.

Compromising emanations are issued either by a VGA display, a DP-to-DVI cable or an HDMI connector. The interception system is depicted in Figure 6.1: the antenna is bilog, the SDR device automatically recovering parameters [De +18] is an Ettus X310 receiving with a 100 MHz bandwidth to recover the compromised information with a fine granularity [Kuh13]. The host computer running post-processing has a linux operating system, an Intel®Xeon®W-2125 Central Processing Unit (CPU) and an Nvidia GTX 1080 Ti Graphics Processing Unit (GPU). The host computer rasters the compromising data using the CPU while the proposed learning-based denoiser/classifier runs also on the GPU.

6.3.2 Performance Comparison Between Data Catchers

The purpose of the exposed method is to analyze compromising emanations. Once a signal is detected and rasterized, intercepted emanations should be classified into compromising or not. Figure 6.4 illustrates the outputs of different implemented denoisers. More examples are available at [3]. It is proposed to assess the data leak according to the ability of a model to retrieve original information. A ratio between the number of characters that a method correctly classifies from an intercepted sample, and the true number of characters in the corresponding *clean* reference is used as a metric.

The quality assessment method is the following. First, a sample containing a large number of characters is pseudo-randomly generated (similar to dataset construction). The sample is displayed on the eavesdropped screen and EM emanations are intercepted. The proposed denoising/retrieval is applied and the obtained results are compared to the reference sample. The method using Mask R-CNN produces directly a list of retrieved characters. Other methods, implemented to compare the efficiency of the proposal, use denoising in combination with the Tesseract [Smi07] OCR. Tesseract is a well performing OCR engine, retrieving characters from images. It produces a list of characters retrieved from a denoised sample. As the output of Tesseract is of the same type as the output of Mask R-CNN classification, metrics can be extracted to fairly compare methods.

An end-to-end evaluation is used measuring the quality of characters classification. A *F-score* (see Section 3.5.2) is computed. For simplification and not use an alignment process, a true positive is chosen here to be the recognition of a character truly existing in the reference sample.

3. https://github.com/opendenoising/extension

Denoiser	OCR	F-Score	precision	recall
Raw	Tesseract	0.04	0.20	0.02
BM3D		0.13	0.22	0.09
Noise2Noise		0.17	0.25	0.12
AutoEncoder		0.24	0.55	0.15
RaGAN		0.24	0.42	0.18
UNet		0.35	0.62	0.25
Mask R-CNN		0.55	**0.82**	0.42
Mask R-CNN	Mask R-CNN	**0.68**	**0.81**	**0.57**

Table 6.1 – Character recognition performance for several data catchers using either denoising and Tesseract, or Mask R-CNN (Mask R-CNN) classification. Mask R-CNN classifier outperforms others methods with a 0.68 *F-score* on the test set.

Table 6.1 presents the results of different data catchers on a test set of 12563 patches. All denoising methods are tested using Tesseract, and compared to Mask R-CNN classification used as OCR. Tesseract is first applied to raw (non-denoised) samples as a point of reference. BM3D is the only expert-based denoising solution tested. *Noise2Noise*, *AutoEncoder*, *RaGAN* and *UNet* are different deep learning networks configured as denoisers. As shown in Table 6.1, Mask R-CNN classification outperforms all other methods. The version of Mask R-CNN using its own classifier is better than the Tesseract OCR engine applied on Mask R-CNN segmentation mask output. It is also interesting to look at precision and recall scores that compose the *F-score*. Both Mask R-CNN methods perform better than other methods for the two indices. Precision is almost the same for both methods, meaning that they both present the same ratio of good decision. The difference lies in the recall score. The 0.42 recall score of the version using Tesseract is lower than the 0.57 score of the method using its own classifier, indicating that the latter version miss less characters. The main advantage of the Mask R-CNN is that the processing tasks to solve the final aim of textual information recovery are jointly optimized.

Another key performance indicator of learning-based algorithms is inference time (Table 6.2). The proposed implementation using Mask R-CNN infers results from an input sample of resolution 1200×1900 in 4.04s in average. This inference time, although lower than BM3D latency, is admittedly higher than other neural networks and hardly real-time. Nevertheless, the inference time of Mask R-CNN includes all the denoising/OCR process and provides a largely better retrieval score. In the context of a continuous listening of EM emanations, it provides an acceptable trade-off between processing time and interception performance. The optimization of the inference time could be considered as a future work with the recent advances in accelerating neural network inference [Zha+16; He+18].

Denoiser	OCR	Inference Timing (s)
Raw	Tesseract	0.19
BM3D		21.8
Autoencoder		1.15
Mask R-CNN		4.22
Mask R-CNN	Mask R-CNN	4.04

Table 6.2 – Inference time for several data catchers using Tesseract or Mask R-CNN classification as OCR. Input resolution is 1200×1900 and it is processed using a split in 28 patches. Mask R-CNN classifier is slower than the autoencoder but still faster than BM3D.

6.4 An Opening to Eavesdropped Natural Images

In this manuscript, we focused on textual images, when evoking eavesdropped images. As evoked in Chapter 3, the properties of textual images are different from those of natural images. To go further on the restoration and interpretation of eavesdropped images, we propose in this section a dataset made of natural eavesdropped samples and their clean references. Dubbed Natural Interception Dataset (NID), the dataset is publicly available on GitHub [4].

First, the acquisition process of the Natural Interception Dataset (NID) dataset is detailed. Discussions on the noise corrupting the dataset follow.

6.4.1 Dataset Construction

The NID is a dataset of natural eavesdropped images. The underlying data is made of Berkeley Segmentation Dataset (BSD) [Mar+01] samples. Using the protocol described in the following, we obtained the eavesdropped counterpart of all BSD68 samples as well as this of 424 BSD432 samples out of the original 432.

Building a supervised dataset of eavedropped samples is complicated due to the impairements evoked in Section 2.3. First, it is complicated to know the position of the desired image part in the eavesdropped reconstructed image. Second, the image is noisy by nature which makes difficult any correlation or corner detection technique. Identification markings like QR codes are also difficult to detect for noisy/reference pairing.

We use a method for building the dataset which is close but not equal to the one used in Section 6.2.2 to contruct the dataset of eavesdropped text images. The dataset is contructed in two phases. First, the images are displayed on a screen and eavesdropped with the same experimental system as the one of Figure 2.4. The images are jointly displayed with a QR code. This QR code enables pairing noisy and reference images since each image as its own. A cross sight is also drawn and used to retrieved the position of the data of interest in the eavesdropped image. Indeed, as can be seen on the top-right part of Figure 6.5, such sight still appears clearly in eavesdropped samples. The QR code cannot be used for this purpose since its content changes and no corner is always visible, depending on the encoded value. The image cannot be used either since the sharpness of its corner depends on its original content.

Second, the images are post-processed. The cross sight is first retrieved. The QR code being positioned directly on the right-bottom quarter, created by the sight, is cropped and interpreted. The top-left corner of the image is always positioned at the same position with respect

4. https://github.com/opendenoising/NID

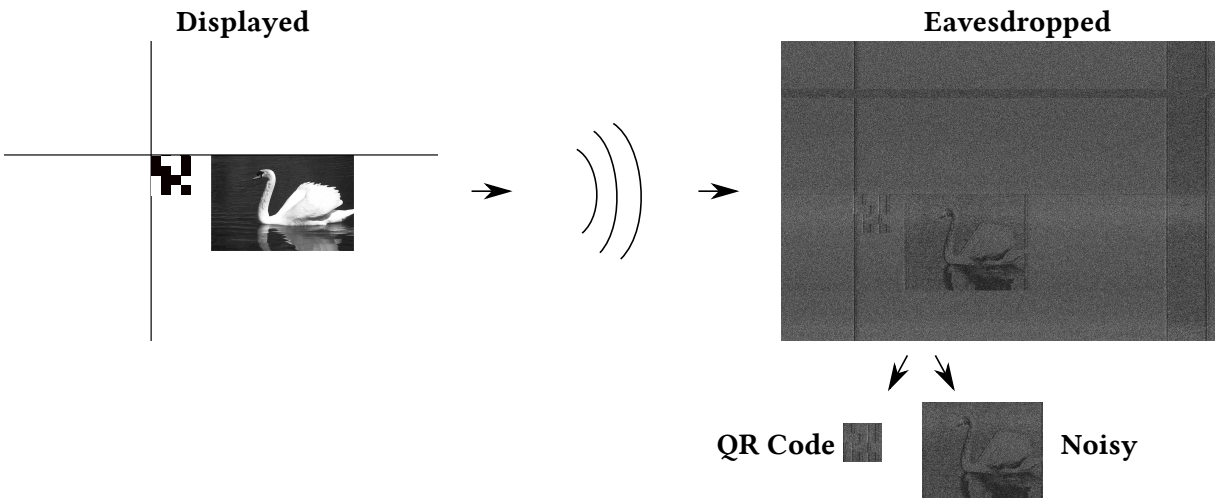


Figure 6.5 – Contruction of NID, a natual eavesdropped supervised dataset. A reference sample is displayed on a screen jointly with a QR code and a cross. The screen is eavesdropped and the image reconstrcuted. The cross is detected, the QR code cropped for reading and identification and the noisy image extracted.

to the sight. It can thus be cropped once the sight identified. The BSD dataset contains both portrait and landscape samples. A measure of standard deviation is then used to defined weither a landscape or portrait crop should be done. Indeed, the part of the eavesdropped sample that contains image data has a smaller standard deviation compared to the background, which is noise.

The noisy nature of the eavesdropped images complicates the identification of the cross sight as well as the reading of the QR Code. Even repeating the procedure several times per BSD image, it has been impossible to collect 8 reference/clean image pairs out of the 432 of the training set. For the potential users to be informed, these 8 images are identified and listed on the NID GitHub repository .

6.4.2 Does a Gaussian Denoiser Transfer to Eavesdropping?

In the previous section we proposed the NID. It is made of eavesdropped natural images. An interesting study would be to train ToxicAI, the direct interpretation proposed earlier in this chapter, on this dataset. We leave this study as a future work but we conduct in this section experiments to understand deeper the noise content of NID.

When looking at Figure 6.6, the appereance of the NID samples let us think that the noise corrupting the samples is something close to an Additive White Gaussian Noise (AWGN). An human would then accordingly select a Gaussian denoiser to address this corruption. We show

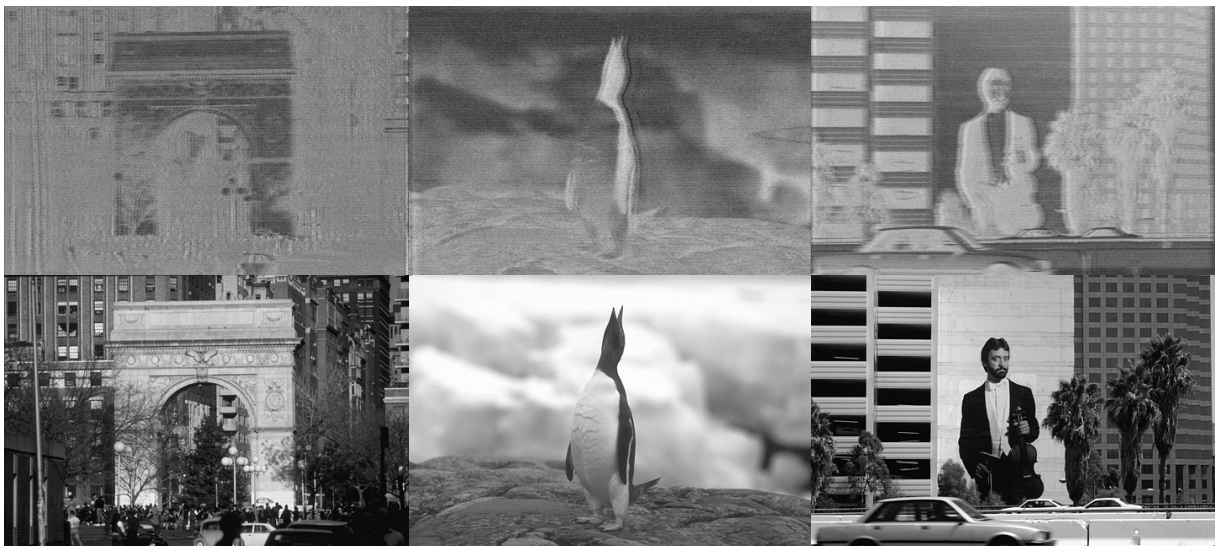


Figure 6.6 – Three NID samples and their references.

in the following that a Gaussian denoiser, applied as is, cannot remove the eavesdropping corruption. Also, we demonstrate that the poor performances of denoisers developed for Gaussian on eavesdropped text images in Chapter 4 do not come from the type of original content.

In Chapter 4, we compared different architectures on various types of noise. To serve this comparison, we trained from scratch and under equal conditions the different algorithms and evaluate them on the corruptions they were trained for. Instead, we choose here to train, using our OpenDenoising tool, two instances of an only architecture using two different datasets and observe how the obtained models transfers when evaluating on the other evaluation set. The two datasets are NID, noisy by nature, and BSD corrupted with AWGN.

For the experiments, we fix the denoising architecture to the one of Denoising Convolutional Neural Network (DnCNN) [Zha+17], with $L = 20$ layers. We choose this architecture since it is has proven efficient on AWGN removal. We do not apply data augmentation. The first dataset is made of BSD samples corrupted by an AWGN with $\sigma = 50$, its standard deviation. The second dataset is the above-presented NID. As we mentionned earlier, 8 BSD samples are missing in the training set of NID. To be consistent, we also removed the 8 missing samples from the Gaussian corrupted training dataset. For training, we use patching with size $p = 40$ for both datasets. The training last $N_{epochs} = 100$, the initial learning rate is $lr = 10^{-3}$ and is divided by 10 every 30 epochs. An Adam optimizer is applied following an Mean Square Error (MSE) loss function. The best model is selected during training based on the Peak Signal to Noise Ratio (PSNR) obtained on the validation set. For evaluation, we use the entire images

Training \ Evaluation	AWGN $\sigma = 50$	Eavesdropping
AWGN $\sigma = 50$	25.57/0.71	9.68/0.40
Eavesdropping	6.51/0.16	16.29/0.52
No Denoising	14.15/0.15	9.27/0.25

Table 6.3 – PSNR (dB) and SSIM measures for two DnCNNs. The models are trained on a dataset corrupted by AWGN and by eavesdropping. The evaluation is conducted for both models on the two corruptions. Metrics on the noisy sets are given as a reference.

of BSD68 accordingly corrupted and measure the performances using PSNR and Structural Similarity (SSIM).

Table 6.3 presents the results of the evaluation of the two trained models on the two corrupted evaluation sets. The metrics computed on the noisy sets are given as a reference. We observe from this tab that a denoiser trained for AWGN does not operate well when applied as is on eavesdropping samples. In fact, the model trained on AWGN obtains an average PSNR of 9.68dB on the eavesdropped evaluation set while its score was 25.57dB on its own evaluation set. regarding SSIM, the denoising results in a progressiion of 46%. However, the index value is still limited with a value of 0.4. The conclusion is the same on the opposite way, a denoiser trained on eavesdropped samples cannot restore AWGN-corrupted samples. The model trained on eavesdropped images obtains an average PSNR of 16.29dB on the eavesdropped evaluation set while its score drops to 6.51dB on the AWGN evaluation set. Applying a denoiser for eavesdropped samples on AWGN-corrupted samples even degrades the image quality.

Figure 6.7 displays visual results of applying the two models on the evaluation sets, for two images. The first column shows the noisy versions of the two images. The second column displays the images denoised using the model trained for AWGN removal. The third column presents the images denoised using the model trained for eavesdropping corruption removal. Finally, the clean images are given as reference. This figure shows that the eavesdropping corruption is poorly removed, even when using the model trained accordingly. It also confirms that it is not possible to transfer a model, learned on a denoising corruption, to another corruption. In fact, the images denoised by the model that was not trained for this corruption have a poor quality. As an example, on the second image, corrupted with AWGN and denoised with the model for eavesdropping corruption, the train is not visible after denoising.

These experiments bring three majors conclusions. First, it is not possible to transfer directly a denoiser trained on a corruption A to the removal of a corruption B. Second, the

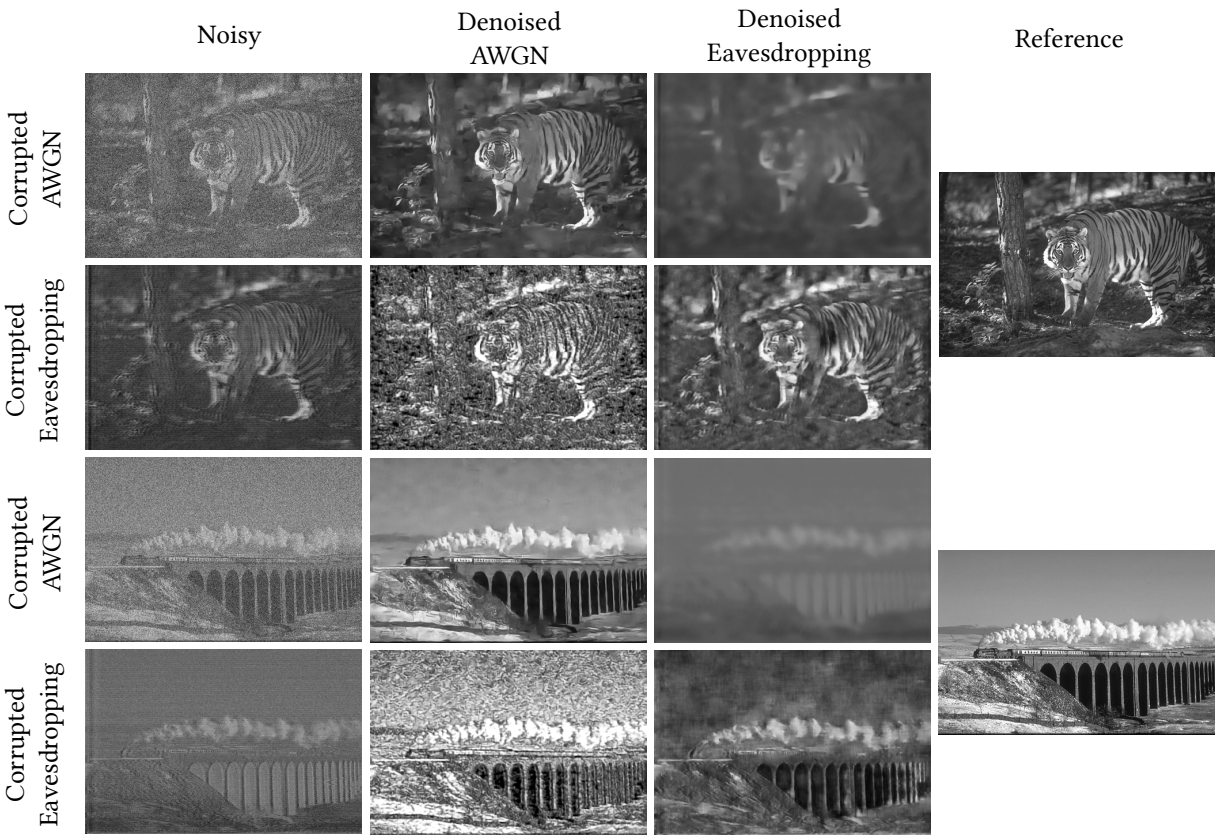


Figure 6.7 – Visual results of the experiments on NID. The denoising models are trained on a dataset corrupted by AWGN and by eavesdropping. The evaluation is conducted for both models on the two corruptions.

eavesdropping noise cannot be simulated using an AWGN distribution. Finally, the results of Table 6.3 suggest that the DnCNN architecture designed for AWGN does not perform well on eavesdropped image removal, even with according training. In fact, we observe that the Δ_{PSNR} (denoised PSNR - noisy PSNR) is 4.4dB lower for eavesdropping, compared to AWGN. It must be noted that the initial eavesdropped samples have a worst quality with an average PSNR of 9.27dB compared to the 14.15dB of the AWGN dataset.

Finally, a second case study is conducted. It aims at evaluating how a gaussian denoiser extend to the removal of eavedropping corruption. Two conclusions are drawn from the experiments. First, a model trained for AWGN cannot be used as is on eavesdropped samples. Second, when properly trained, the DnCNN architecture brings the same denoising ratio on AWGN and eavesdropping corruption.

6.5 Conclusions

This chapter has presented how an EM side-channel attack can be automated, from data retrieval to data interpretation, by employing deep learning methods. The employed experimental setup for demonstrating fully automated information extraction is based on the Mask R-CNN network and on the eavesdropping of textual information. The final setup is capable of recovering 57% of a leaked textual information from a standard screen.

This chapter shows that taking some assumptions on image content, the information interception process can be fully automated until semantic extraction. Such an automation opens for future work on automated audit and countermeasures.

CHAPTER 7

Conclusion

As shown throughout this document, Electro Magnetic (EM) compromising emanations are a threat to the confidentiality of Information Processing Equipments (IPEs) that handle sensitive information. Analyzing a side-channel, an attacker may have access to confidential information. Due to voltage changes, video signals transmitted through cables and connectors are subject to EM side-channel emanations. The transmitted images may be recovered from this side-channel but reconstructed images are strongly corrupted. At the same time as the research on enhancing EM side-channel attacks progresses, Deep Learning (DL) has revolutionnized image restoration and interpretation. In this context, this document has proposed studies to evaluate the interest of image processing and DL to reinforce the restoration and interpretation of eavesdropped images.

These studies demonstrate several elements. State of the art restoration and interpretation methods do not directly transfer to eavesdropped images. In fact, the corruption contained in these images is hybrid, because of the origins of the corruption (noise, interferences, information overlap), and does not follow simple parametric distributions. However, it is difficult to separate the different contributions of that hybrid corruption. While using a gradual removal strategy has proven its efficiency on mixtures of simple parametric noises, eavesdropping noise cannot be removed based on an usual parametric noise using such method. Training learning based image restoration with custom data, it is nevertheless possible to remove part of this complicated corruption. Finally, it has been proven that DL can automate the interpretation of eavesdropped images containing textual information. With an automated synchronization of

the interception process, this is a step further towards fully automated EM emanations exloitation.

The contributions presented in this thesis are among the first published on the use of image processing to enhance and automate the interpretation of images reconstructed from EM side-channel emanations. In addition to providing promising results, several implementations are provided open source on GitHub.

7.1 Research Contributions

To progress on the interpretation of eavesdropped images, Chapter 2 has given an overview of the state of the art rastering techniques as well as on the origins of the hybrid noise generated by the eavesdropping process. Given these findings, Chapter 3 has introduced image processing features likely to be used for eavesdropped image restoration and interpretation. The three main contributions proposed in this manuscript rely on these preliminary chapters and are summarized in this section.

7.1.1 Benchmarking of Image Restoration Algorithms

Chapter 3 has highlighted the complexity of fairly comparing methods evaluated in different conditions. To solve this issue, the OpenDenoising tool has been proposed in Chapter 4. OpenDenoising benchmarks image denoisers and aims at comparing methods on a common ground in terms of datasets, training parameters and evaluation metrics. Supporting several languages and learning frameworks, OpenDenoising is also extensible and open-source.

The second contribution of the chapter is a comparative study of image restoration in the case of a complex noise source, including eavesdropped images. Three major conclusions arise from the comparative study. First, the difference in terms of performance between expert-based and learning-based methods rises as the complexity of the noise grows. Second, the ranking of methods is strongly impacted by the nature of the noises. Finally, Multi-level Wavelet Convolutional Neural Network (MWCNN) proves to be the best method for the considered real-world interception restoration task. It slightly outperforms Denoising Convolutional Neural Network (DnCNN) and RED30 while being substantially faster, in inference mode.

These results show that restoring an image from a complex noise is not universally solved by a single method and that choosing a denoiser requires automated testing.

This chapter has led to the public release of the OpenDenoising benchmark tool [1].

7.1.2 Mixture Noise Denoising Using a Gradual Strategy

Chapter 2 and Chapter 3 have suggested that the corruption generated by the eavesdropping process is a sequential mixture of several primary corruptions.

Chapter 5 introduces a gradual image denoising strategy called NoiseBreaker. NoiseBreaker iteratively detects the image dominating noise using a trained classifier with an accuracy of 93% and 91% for grayscale and RGB samples, respectively, when primary noises are known and parametrized. Under the assumption of grayscale sequential noise mixtures, NoiseBreaker performs only 0.95dB under the supervised MWCNN denoiser without being trained on any mixture noise. Neither the classifier nor the denoisers are exposed to mixture noise during training. NoiseBreaker operates 2dB over the gradual denoising of [LSJ20] and 5dB over the state of the art self-supervised denoiser *Noise2Void*. When using RGB samples, NoiseBreaker operates 5dB over [LSJ20] while *Noise2Void* underperforms. Moreover, this paper demonstrates that making noise analysis guide the denoising is not only efficient on noise type, but also on noise intensity.

This chapter has demonstrated the practicality of NoiseBreaker on six different synthetic noise mixtures. Nevertheless, the NoiseBreaker version proposed in the chapter has not permitted to conclude on the efficiency of the method to restore eavesdropped images. Consequently, it is not possible to validate the hypothesis of the sequential composition of the eavesdropping corruption.

7.1.3 Direct Interpretation of Eavesdropped Images

Handling data while ensuring trust and privacy is challenging for information system designers. Chapter 6 presents how the attack surface can be enlarged with the introduction of deep learning in an EM side-channel attack. The proposed method, called ToxicAI, uses Mask R-CNN as denoiser and it automatically recovers more than 57% of characters, present in the test set. In comparison, the best denoising/Optical Character Recognition (OCR) pair retrieves 42% of characters. The proposal is software-based, and runs on the host computer of an off-the-shelf Software-Defined Radio (SDR) platform.

This chapter has led to the public release of two datasets of eavesdropped samples:

1. https://github.com/opendenoising/opendenoising-benchmark

- a dataset of eavesdropped images made of text characters and their references [2],
- a dataset of eavesdropped natural images, based on Berkeley Segmentation Dataset (BSD), dubbed Natural Interception Dataset (NID) [3].

7.2 Prospects – Future Works

The work presented in this document opens opportunities for future research on eavesdropped image restoration and interpretation. This section proposes research directions to go deeper using our proposed methods, but also more general research directions.

7.2.1 Signal Detection in Eavesdropping Noise

In Chapter 2 we explained why we study the application of image processing techniques to eavesdropped images. We then hypothetised that leveraging such methods, it could be possible to relax the required precision when setting up the raster process that transform the 1D signal into an image.

To continue in this direction it would be interesting to design a signal detection method in eavesdropped images. In fact, we hypothetize that when the interception system reconstructs an image from a signal that contains no information (no screen emanations or nothing displayed on the screen), the reconstruction should be composed of pure noise. Once a signal mixes to this pure noise (content on the screen), the recontructed image ditribution should deviate. Identifing these two cases would enable predicting whether a screen with potential compromising content is present in the area. Such problem may be solved creating a custom dataset and using a binary classifier predicting whether there is signal or not in an image.

7.2.2 Fine-Grain Modeling of the Eavesdropping Corruption

We proposed in Chapter 2 a list of the contributing elements to the strong hybrid corruption generated by the eavesdropping process. While questionning the content of this hybrid corruption, we did not give a fine-grain modeling.

Following the previous point on signal detection in noise, the same data could be used to model the noise generated by the side-channel. In fact, when no screen data is contained in the reconstructed image, an illustration of the noise distribution is accessible. The modeling of the

2. https://github.com/opendenoising/interception_dataset
3. https://github.com/opendenoising/NID

other part of the corruption seems more complicated since it depends on the characteristics of the data to be eavesdropped.

Additionally, leveraging learning algorithms, modeling the eavesdropping corruption using Generative Adversarial Networks (GAN) could also be a promising direction. Once a generative model is trained, it would be possible to generate training datasets for further restoration and interpretation methods.

7.2.3 Interpretability of Eavesdropped Images

In Chapter 6, we presented ToxicAI as proof of concept on the efficiency of learning algorithms on interpretation of eavesdropped images. ToxicAI is designed to retrieve characters in eavesdropped images. This contribution introduces a custom metric that consists in measuring the number of character retrieved in an image instead of using classical metrics like Peak Signal to Noise Ratio (PSNR) or Structural Similarity (SSIM). This metric is a first step in measuring the interpretability of eavesdropped images. However, this metric is limited to textual content.

A future work could consist in developing new methods to assess further the interpretability of eavesdropped images. As an example, a progressive tree testing could be proposed. From the root to the leafs, each node represents a test on the information extracted from the input image. The deeper the process goes, the deeper the knowledge extractable from the image and the higher the interpretability. The interpretability of eavesdropped images is a crucial parameter since it enables assessing the potential compromise of detected emanations. Knowing such measure would enable finer countermeasures to avoid sensitive information diffusion.

7.2.4 Extension to Other Noisy Data

Throughout this manuscript, we have worked with noisy data. In particular, the eavesdropped images that motivated our studies are corrupted by complex noises. There are other domains that deal with noisy images because of their acquisition conditions or sensing technologies. Among these domains medical images and spatial imaging seems related to our work because of their high corruption levels. Adapting our proposed methods to these applications may be a promising direction. Also, we hypothesize that the diversity of the corruption, contained in the two datasets we propose (NID in particular), may make our methods applicable directly to other complex problems. As an example, we have experienced good results at de-

tecting text on poor quality security camera images, using ToxicAI. This transfer faculty may be interesting for domains where it is complicated to gather large supervised datasets.

7.2.5 Embedding of Proposed Methods

The eavesdropping restoration and interpretation experiments of that manuscript have been conducted offline, i.e. the eavesdropped images are acquired and interpreted later. When auditing sytems, an operator looking for risk of compromising emanations has to be mobile. In that context, the embedding of our proposed solutions is an important future work.

NoiseBreaker: Chapter 5 proposed NoiseBreaker, a gradual denoising method for sequential mixture noise removal. NoiseBreaker is made of two parts. First, a classifier determines what is the dominant primary noise in an image to be restored. Then, the denoiser trained to remove the according primary noise is applied to the image. The process is iterative and operates until no noise is detected anymore. The memory and computation footprint are not studied in the chapter. However, the use of several primary denoiser implies keeping them in memory. Furthermore, the principle of an iterative algorithm means running several denoisers sequentially. These points open opportunities for the optimisation of NoiseBreaker.

First, primary denoisers have state of the art architectures for simplicity. Conducting a design study, e.g. using OpenDenoising benchmark, would enable reducing the number of parameters of the architectures, and then the memory they require. We introduced class refinement to target more precise corruptions, i.e. several denoisers address the same noise but with different parameter ranges. Increasing the number of primary denoisers reduces the noise parameters ranges that each denoiser is responsible for. A relief of the number of parameters in the architecture of each denoiser could be considered under such context which would lighten NoiseBreaker.

ToxicAI: Chapter 6 introduced ToxicAI, an information retrieval method for eavesdropped images. ToxicAI uses the default architecture of Mask-RCNN which contains a ResNet101 of more that 44 millions parameters. This architecture while giving good results is heavy in terms of memory and computations, at inference time. A study may be conducted on the impact of replacing the back-end of ToxicAI by a lighter architecture, like e.g. a MobileNet.

APPENDIX A

French Summary

A.1 Contexte

La tendance récente consiste à rendre les données numériques disponibles à tout moment et en tout lieu, ce qui crée de nouvelles menaces pour la confidentialité. En particulier, si l'on considère les données hautement confidentielles, là où les informations imprimées étaient protégées physiquement et n'étaient accessibles qu'aux personnes autorisées, les données sont aujourd'hui numériques. Les données sont échangées et consultées en utilisant des systèmes d'information (SI) et leurs afficheurs vidéo correspondants. Si les principaux efforts de sécurité se concentrent aujourd'hui sur le côté réseau des systèmes, il existe d'autres menaces de sécurité.

Un *canal auxiliaire* est un chemin de données non intentionnel en opposition avec le canal traditionnel. En particulier, les canaux auxiliaires électro-magnétiques (EM) sont dus aux champs émis par les câbles et connecteurs vidéo lorsque leur tension interne change. Ces canaux auxiliaires sont dangereux car ils propagent des données non chiffrées en dehors du système physique. Ces émissions peuvent être corrélées à une information confidentielle. Un attaquant recevant le signal et connaissant les protocoles de communication peut accéder illégalement aux informations originales traitées par le SI. Dans ces conditions, l'attaquant peut reconstruire l'image affichée sur l'écran attaqué connecté au SI. Il a été démontré que le contenu d'un écran peut être recontruit à des dizaines de mètres [DSV20a]. Depuis les exploits des pionniers [Van85], de nombreux travaux ont été publiés sur la reconstruction d'images à partir d'émanations EM par canal auxiliaire, et ce domaine de recherche est toujours dy-

namique [Lav+21]. Les travaux de l'état de l'art menés sur ce sujet ont principalement porté sur l'amélioration de la reconstruction d'un point de vue du traitement du signal.

Récemment, le domaine du traitement d'images a été révolutionné par l'apparentissage machine et surtout l'apprentissage profond. Ces algorithmes, qui apprennent des tâches à partir de données, ont dépassé les performances des algorithmes experts de l'état de l'art sur plusieurs tâches de vision par ordinateur. En particulier, l'une des tâches qui a bénéficié des algorithmes d'apprentissage est la classification sémantique du contenu des images. Dans cette tâche, les algorithmes de l'état de l'art sont aujourd'hui capables d'automatiser l'interprétation des images. Cependant, ces méthodes d'interprétation sont conçues pour des images naturelles non corrompues. La restauration d'images est la tâche qui consiste à supprimer les altérations des images. La restauration d'images a également beaucoup bénéficié des algorithmes d'apprentissage. En effet, les algorithmes récents surpassent les anciens algorithmes de l'état de l'art, tant sur les performances objectives que subjectives. Cependant, les algorithmes de restauration d'images de l'état de l'art se concentrent sur des corruptions bien définies, qui suivent une distribution paramétrique, régie par quelques paramètres.

Les images reconstruites à partir des émanations EM sont fortement corrompues pour plusieurs raisons. Il y a d'abord une perte de données et des interférences inhérentes au processus d'émission/réception EM. Il existe également des défauts dans la synchronisation de la reconstruction, lors du passage du signal 1D à une image. Enfin, les défauts du matériel du système d'interception introduisent des erreurs. Il en découle trois questions que nous étudions dans ce manuscrit : **Quel est le type de corruption généré par la reconstruction d'émanations EM ? Peut-elle être réduite à une composition de bruits aux distribution paramétriques ? Comment les méthodes actuelles de restauration d'images se comportent-elles sur une image interceptée ?**

L'audit des systèmes d'information traitant des données confidentielles est actuellement réalisé par des experts. Une fois le système d'interception en place, l'expert évalue la compromission de l'équipement audité, en s'appuyant sur son expérience. Ce protocole d'audit prend du temps et est sujet à la perception humaine. Vient alors une autre question que nous étudions dans ce manuscrit : **Peut-on utiliser l'apprentissage profond pour automatiser l'interprétation sémantique d'images interceptées ?**

A.2 Objectifs et contributions de cette thèse

L'objectif principal de cette thèse est d'analyser comment les techniques d'apprentissage profond peuvent être appliquées aux images interceptées et si elles peuvent automatiser l'interprétation de ces images. Bien que la reconstruction d'émanations EM et le traitement d'images par apprentissage profond soient deux domaines très étudiés, leur utilisation concomitante est une avancée récente.

Après avoir passé en revue les travaux fondamentaux sur l'interception et l'interprétation d'images bruitées, nous proposons un ensemble d'expériences et de contributions pour étudier la faisabilité de l'exploitation automatique des images d'interception.

Trois contributions principales sont proposées dans ce document. Elles sont parmi les premières études des émanations EM d'un point de vue traitement d'image. En conséquence, cette thèse est l'une des premières tentatives d'application de l'apprentissage profond pour l'automatisation de l'exploitation d'images d'interception. Les trois principales contributions de cette thèse sont brièvement présentées ci-dessous.

A.2.1 Comparaison d'algorithmes de restauration d'images

Comparer équitablement les débruiteurs est devenu compliqué avec l'utilisation d'algorithmes d'apprentissage. En effet, les algorithmes peuvent être entraînés et évalués sur différents ensembles de données, ce qui rend la comparaison injuste sans ré-entraînement. C'est un problème lorsqu'on cherche des méthodes pour un nouveau système. L'outil proposé, nommé OpenDenoising, évalue les débruiteurs d'images et vise à comparer les méthodes sur un terrain commun en termes de jeux de données, de paramètres d'apprentissage et de métriques d'évaluation. Supportant plusieurs langages et outils d'apprentissage, OpenDenoising est également extensible et open-source.

La deuxième contribution du chapitre est une étude comparative de la restauration d'images dans le cas d'une source de bruit complexe. Les expériences de cette étude comparative sont utilisées comme étude de cas pour l'outil proposé. Plusieurs conclusions sont tirées de l'étude comparative. Premièrement, il existe une différence en termes de performance entre les méthodes basées sur l'expertise et celles basées sur l'apprentissage. Cette différence augmente avec la complexité du bruit. Deuxièmement, le classement des méthodes est fortement influencé par la nature des bruits. Ces résultats montrent que la restauration d'une image corrompue par un bruit complexe n'est pas universellement résolue par une seule méthode et que le choix d'un débruiteur nécessite des tests automatisés.

Ce chapitre a conduit à la publication de l'outil OpenDenoising [1]. Ces travaux ont été présentés lors de la conférence internationale *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* en 2020 [Lem+20c].

A.2.2 Débruitage graduel de mélanges de bruit

Les chapitres préliminaires suggèrent que la corruption générée par le processus d'interception EM est un mélange séquentiel de plusieurs corruptions primaires. En conséquence, le chapitre 5 introduit une stratégie graduelle de débruitage d'image appelée NoiseBreaker. NoiseBreaker détecte itérativement le bruit dominant de l'image à l'aide d'un classificateur entraîné avec une précision de 93 % et 91 % pour les échantillons en niveaux de gris et rouge-vert-bleu (RVB), respectivement. Dans l'hypothèse de mélanges de bruits séquentiels en niveaux de gris, NoiseBreaker obtient une performance de $0,95$dB en dessous du débruitage supervisé utilisant MWCNN sans avoir été entraîné sur un quelconque mélange de bruits. Ni le classificateur, ni les débruiteurs ne sont exposés au mélanges de bruit pendant l'entraînement. NoiseBreaker opère 2dB au dessus du débruitage graduel de [LSJ20] et 5dB au dessus de l'état de l'art du débruitage auto-supervisé *Noise2Void*. Lorsqu'il utilise des échantillons RVB, NoiseBreaker a une performance supérieure de 5dB à celle de [LSJ20] alors que *Noise2Void* est moins performant. De plus, cet article démontre que l'utilisation de l'analyse du bruit pour guider le débruitage est efficace non seulement sur le type de bruit, mais aussi sur son intensité.

Ce chapitre démontre l'aspect pratique de NoiseBreaker sur six différents mélanges de bruits synthétiques. Néanmoins, la version de NoiseBreaker proposée dans le chapitre n'a pas permis de conclure quand à l'efficacité de la méthode pour restaurer des images interceptées. Par conséquent, l'hypothèse de la composition séquentielle de la corruption d'interception n'est pas validée.

Ce travail a donné lieu à une présentation lors du workshop *IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)* en 2020 [Lem+20a].

A.2.3 Interprétation directe d'images interceptées

Ce travail est présenté dans le dernier chapitre de contribution du manuscrit. Le début du manuscrit étudie l'applicabilité de l'apprentissage profond pour restaurer des images interceptées. Ce dernier travail se concentre sur l'interprétation et étudie son automatisation sur des images textuelles. L'introduction de l'apprentissage profond dans une attaque de type

1. https://github.com/opendenoising/opendenoising-benchmark

canal auxiliaire EM est étudiée. La méthode proposée utilise Mask R-CNN comme débruiteur et récupère automatiquement plus de 57% des caractères présents dans le jeu de test et ce pour une large gamme de distances d'interception. La proposition est logicielle et s'exécute sur l'ordinateur hôte d'une plateforme radio-logicielle prête à l'emploi.

Ce chapitre a conduit à la diffusion publique de deux ensembles de données d'images interceptées :

- un jeu de données d'images synthétiques d'interception composées de caractères de texte et de leurs références [2],
- un jeu de données d'images naturelles interceptées, basé sur BSD, nommé NID [3].

Ce travail a été présenté lors de la conférence *Conference on Artifical Intelligence for Defense (CAID)*, en 2019 [Lem+19] et lors de la conférence *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* en 2020 [Lem+20b].

A.3 Plan du Manuscrit

Le chapitre 2 présente ce qu'est une interception électro-magnétique et en quoi elle constitue une menace pour la confidentialité des système d'information utilisant des écrans. Les caractéristiques de l'interception sont étudiées. En particulier, le lien est fait entre les corruptions trouvées dans les images et leur origine physique. Enfin, des arguments sont donnés qui motivent l'étude du traitement d'images pour améliorer l'interprétation des images interceptées.

Le chapitre 3 donne une définition du bruit dans une image. Les principales distributions de bruit dans les images sont détaillées, ce qui ouvre la voie à l'introduction de compositions plus complexes de ces distributions. Le chapitre passe ensuite en revue l'état de l'art des méthodes de restauration et d'interprétation d'images. Une distinction est faite entre les algorithmes experts et ceux basés sur l'apprentissage. L'avancée en termes de performance permise par ces derniers est discutée. Des métriques d'évaluation et d'optimisation ainsi que des jeux de données sont présentés pour l'évaluation de la qualité des images et de la classification. Enfin, la terminologie des algorithmes d'apprentissage ainsi que des discussions sur leurs forces et les questions ouvertes pour notre étude de cas sont proposées.

Le chapitre 4 propose un outil extensible et open-source pour évaluer les algorithmes de débruitage équitablement. Ensuite, une étude comparative de l'état de l'art des débruiteurs est

2. https://github.com/opendenoising/interception_dataset
3. https://github.com/opendenoising/NID

présentée. Cette étude comparative apporte également des premières réponses sur la suppression du bruit d'interception dans les images.

Le chapitre 5 présente NoiseBreaker, une méthode progressive de débruitage d'images qui s'attaque à la suppression du bruit de mélange séquentiel. Les travaux connexes de l'état de l'art sont exposés avant de détailler la méthode proposée qui s'appuie sur une stratégie itérative. Le bruit dominant est détecté avant d'être supprimé. La méthode est comparée à l'état de l'art avant d'être discutée dans une étude d'ablation.

Le chapitre 6 aborde l'interprétation directe d'images interceptées en proposant ToxicAI. Les travaux connexes sont présentés avant de définir l'architecture de ToxicAI. La construction de l'ensemble de données personnalisées, open-source, d'écrans interceptées contenant des caractères, utilisé pour former ToxicAI, est détaillée. Ensuite, des expériences sont menées sur la proposition et les résultats sont comparés à l'état de l'art. Enfin, un jeu de données open-source d'images naturelles interceptées est proposé pour étendre ToxicAI.

Le chapitre 7 conclut le manuscrit. Tout d'abord, les questions adressées dans le document sont rappelées et les contributions sont résumées. En s'appuyant sur les principes proposés dans ce document, des directions de recherche pour le futur de l'interprétation des images interceptées sont proposées.

La figure A.1 illustre l'organisation de ce document. Cette figure met en évidence les liens entre les chapitres présentés ci-dessus.

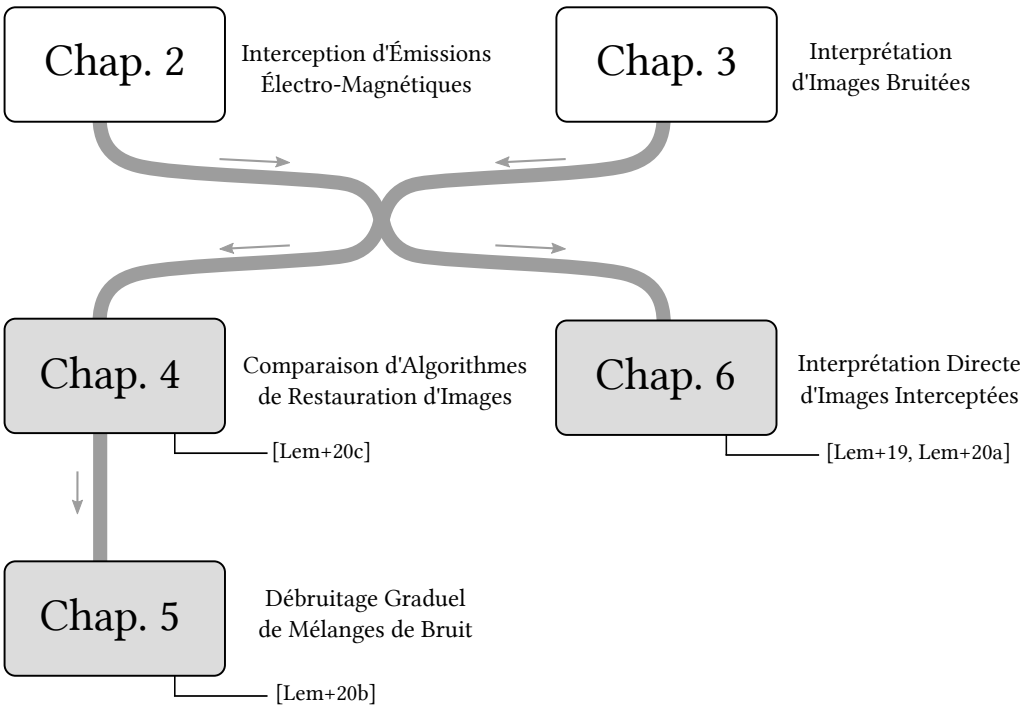


Figure A.1 – Structure générale du document. Les chapitres sur l'état de l'art sont affichés en blanc, tandis que les chapitres de contribution sont en gris.

List of Figures

List of Tables

Acronyms

AM Amplitude Modulated. 22

AWGN Additive White Gaussian Noise. 33, 40, 50, 52, 55, 57, 58, 60, 61, 63, 64, 71, 79, 98–102

BM3D Block-Matching 3D. 37, 38, 40, 50, 57, 60, 61, 63, 64, 72, 76, 79, 81, 82, 93, 95

BSD Berkeley Segmentation Dataset. 13, 46, 50, 97–99, 106, 113

CIA Confidentiality Integrity Accessibility. 17, 20

CNN Convolutional Neural Network. 37, 38, 41, 48, 50, 57, 58, 70

CPU Central Processing Unit. 21, 60, 94

CRT Cathode Ray Tube. 20, 23

CV Computer Vision. 10, 39, 41, 50

DIP Deep Image Prior. 39, 40

DIV2K DIVerse 2K. 46, 50, 56

DL Deep Learning. 10–12, 41, 50, 51, 61, 86, 103

DnCNN Denoising Convolutional Neural Network. 38, 40, 50, 57, 60, 61, 64, 65, 68, 71, 76, 99, 100, 102, 104

DP Display Port. 90, 94

DVI Digital Visual Interface. 21, 90, 94

DWT Discrete Wavelet Transform. 28, 29, 41, 49

EM Electro Magnetic. 9–12, 18–21, 25, 27, 28, 31, 52, 61, 87, 88, 90, 94, 95, 103–105, 116, 117

FC Fully Connected. 41, 50, 73

FCN Fully Convolutional Network. 91

FM Frequency Modulated. 22

G2G Generated-Artificial-Noise to Generated-Artificial-Noise. 35, 39, 40, 71, 79

GAN Generative Adversarial Networks. 38, 107

GCBD GAN-CNN based Blind Denoiser. 35, 39, 40

GPU Graphics Processing Unit. 21, 50, 60, 94

HDMI High-Definition Multimedia Interface. 21, 24, 25, 90, 94

HOG Histograms of Oriented Gradients. 41

IDCN Implicit Dual-domain Convolutional Network. 37, 40

ILSVRC ImageNet Large Scale Visual Recognition Competition. 41, 46, 50, 71, 83

IPE Information Processing Equipment. 9, 13, 17, 20, 21, 28, 39, 103

IPT Image Processing Transformer. 39, 40

ISS Information System Security. 17

kNN k-Nearest Neighbors. 41

LCD Liquid Crystal Display. 20, 34

LED Light-Emitting Diode. 34

ML Machine Learning. 10, 21, 31, 32, 37, 38, 42, 50

MLP Muli-Layer Perceptron. 41

MOS Mean Opinion Score. 43

MSE Mean Square Error. 42, 43, 45, 99

MWCNN Multi-level Wavelet Convolutional Neural Network. 12, 37, 40, 58, 60, 61, 64, 65, 75, 76, 79, 81, 82, 85, 86, 104, 105, 112

N2N Noise2Noise. 39, 40, 58, 70

N2S Noise2Self. 39, 40

N2V Noise2Void. 39, 40, 58, 70, 76, 79, 81, 82

NID Natural Interception Dataset. 13, 65, 97–99, 106, 107, 113, 117

NLP Natural Language Processing. 39

NTIRE New Trends in Image Restoration. 46

OCR Optical Character Recognition. 12, 92, 94–96, 105

PSNR Peak Signal to Noise Ratio. 42, 43, 45, 62, 78, 80–82, 99, 100, 102, 107

RED Residual Encoder-Decoder Network. 38, 40

RF Radio Frequency. 89, 90, 92

RGB Red Green Blue. 25, 32, 42, 86

RNN Recurrent Neural Network. 47

RoI Region of Interest. 91

RPN Region Proposal Network. 91

SDR Software-Defined Radio. 12, 20–23, 87–89, 94, 105

SGN Self-Guided Network. 38, 40, 75, 76, 79

SIFT Scale Invariant Feature Transform. 39

SNR Signal to Noise Ratio. 25, 27

SRResNet Super-Resolution Residual Network. 75, 76, 79

SSIM Structural Similarity. 42, 43, 45, 78, 80, 81, 100, 107

SURF Speed-Up Robust Features. 41

SVM Support Vector machine. 41

VDU Video Display Unit. 9, 13, 17, 20, 21

VGA Video Graphics Array. 21, 90, 94

XAI eXplainable Artificial Intelligence. 51

Personal Publications

[Lem+19] Florian Lemarchand, Cyril Marlin, Florent Montreuil, Erwan Nogues, and Maxime Pelcat, « ToxicIA: Apprentissage Profond Appliquéà l'Analyse des Signaux Parasites Compromettants », *in*: *C&ESAR 2019 IA & Défense*, 2019.

[LNP19] Florian Lemarchand, Erwan Nogues, and Maxime Pelcat, « Real-Time Image Denoising with Embedded Deep Learning: Review, Perspectives and Application to Information System Security », *in*: *RESSI 2019 Rendez-Vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information*, 2019.

[Lem+20a] Florian Lemarchand, Thomas Findeli, Erwan Nogues, and Maxime Pelcat, « Noisebreaker: Gradual image denoising guided by noise analysis », *in*: *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2020, pp. 1–6.

[Lem+20b] Florian Lemarchand, Cyril Marlin, Florent Montreuil, Erwan Nogues, and Maxime Pelcat, « Electro-Magnetic Side-Channel Attack Through Learned Denoising and Classification », *in*: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2882–2886.

[Lem+20c] Florian Lemarchand, Eduardo Fernandes Montesuma, Maxime Pelcat, and Erwan Nogues, « OpenDenoising: an Extensible Benchmark for Building Comparative Studies of Image Denoisers », *in*: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 2648–2652.

Bibliography

[AB18] Josue Anaya and Adrian Barbu, « RENOIR-A benchmark dataset for real noise reduction evaluation », *in*: *Journal of Visual Communication and Image Representation* (2018), pp. 144–154 (cit. on p. 46).

[Abd+20] A. Abdelhamed, M. Afifi, R. Timofte, and M. S. Brown, « Ntire 2020 challenge on real image denoising: Dataset, methods and results », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 496–497 (cit. on pp. 35, 69).

[ALB18] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown, « A High-Quality Denoising Dataset for Smartphone Cameras », *in*: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018 (cit. on pp. 34, 46).

[AT17] E. Agustsson and R. Timofte, « NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study », en, *in*: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA: IEEE, July 2017, pp. 1122–1131 (cit. on pp. 46, 47, 56, 76, 79).

[Bay+08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, « Speeded-Up Robust Features (SURF) », en, *in*: *Computer Vision and Image Understanding* 110.*3* (June 2008), pp. 346–359 (cit. on p. 41).

[BCM05] A. Buades, B. Coll, and J. M. Morel, « A Review of Image Denoising Algorithms, with a New One », en, *in*: *Multiscale Modeling & Simulation* 4.*2* (2005), pp. 490–530 (cit. on pp. 36, 37, 40).

[BJ15] A. K. Boyat and B. K. Joshi, « A Review Paper : Noise Models in Digital Image Processing », en, *in*: *Signal & Image Processing : An International Journal* 6.*2* (Apr. 2015), pp. 63–75 (cit. on p. 36).

[BR19] J. Batson and L. Royer, « Noise2Self: Blind Denoising by Self-Supervision », *in*: *International Conference on Machine Learning*, 2019, pp. 524–533 (cit. on pp. 35, 39, 40, 67, 71).

[BSH12] Harold C Burger, Christian J Schuler, and Stefan Harmeling, « Image denoising: Can plain neural networks compete with BM3D? », *in*: *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, pp. 2392–2399 (cit. on p. 50).

[BT07] ITU-R Recommendation BT, « Methodology for the subjective assessment of video quality in multimedia applications », *in*: *Proceedings of the International Telecommunication Union* (2007) (cit. on p. 43).

[Bur98] Christopher JC Burges, « A tutorial on support vector machines for pattern recognition », *in*: *Data mining and knowledge discovery* 2.*2* (1998), pp. 121–167 (cit. on p. 41).

[Che+18] J. Chen, J. Chen, H. Chao, and M. Yang, « Image Blind Denoising with Generative Adversarial Network Based Noise Modeling », en, *in*: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, June 2018, pp. 3155–3164 (cit. on pp. 35, 39, 40).

[Che+20] Hanting Chen et al., « Pre-Trained Image Processing Transformer », en, *in*: *arXiv:2012.00364 [cs]* (Dec. 2020), arXiv: 2012.00364 (cit. on pp. 39, 40).

[CM10] P. Chatterjee and P. Milanfar, « Is Denoising Dead? », *in*: *IEEE Transactions on Image Processing* 19.*4* (Apr. 2010), pp. 895–911 (cit. on pp. 36, 55).

[CPM19] S. Cha, T. Park, and T. Moon, « GAN2GAN: Generative Noise Learning for Blind Image Denoising with Single Noisy Images », en, *in*: *arXiv:1905.10488 [cs, eess]* (May 2019), arXiv: 1905.10488 (cit. on pp. 35, 39, 40, 67, 71, 79).

[CTH20] Yucheng Chen, Yingli Tian, and Mingyi He, « Monocular Human Pose Estimation: A Survey of Deep Learning-based Methods », en, *in*: *Computer Vision and Image Understanding* 192 (Mar. 2020), arXiv: 2006.01423, p. 102897 (cit. on p. 39).

[Dab+07] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, « Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering », en, *in*: *IEEE Transactions on Image Processing* 16.*8* (Aug. 2007), pp. 2080–2095 (cit. on pp. 36–38, 40, 50, 55, 57, 72, 77, 79).

[Dau88] Ingrid Daubechies, « Orthonormal bases of compactly supported wavelets », *in*: *Communications on pure and applied mathematics* 41.*7* (1988), pp. 909–996 (cit. on p. 28).

[De +18] P. De Meulemeester, L. Bontemps, B. Scheers, and G. A. E. Vandenbosch, « Synchronization retrieval and image reconstruction of a video display unit exploiting its compromising emanations », en, *in*: *2018 International Conference on Military Communications and Information Systems (ICMCIS)*, Warsaw: IEEE, 2018, pp. 1–7 (cit. on pp. 20, 94).

[De 21] Pieterjan De Meulemeester, *Compromising Electromagnetic Radiation of Information Displays*, eng, 2021 (cit. on pp. 21, 22).

[Den+09] J. Deng, W. Dong, R. Socher, L. Li, Kai L., and Li F-F., « ImageNet: A large-scale hierarchical image database », en, *in*: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL: IEEE, June 2009, pp. 248–255 (cit. on pp. 41, 46, 47, 61, 71, 76).

[Dos+20] Alexey Dosovitskiy et al., « An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale », en, *in*: *arXiv:2010.11929 [cs]* (Oct. 2020), arXiv: 2010.11929 (cit. on p. 39).

[DSV20a] P. De Meulemeester, B. Scheers, and G.A.E. Vandenbosch, « Eavesdropping a (Ultra-)High-Definition Video Display from an 80 Meter Distance Under Realistic Circumstances », en, *in*: *2020 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI)*, Reno, NV, USA: IEEE, July 2020, pp. 517–522 (cit. on pp. 10, 21, 22, 27, 109).

[DSV20b] Pieterjan De Meulemeester, Bart Scheers, and Guy A.E. Vandenbosch, « Differential Signaling Compromises Video Information Security Through AM and FM Leakage Emissions », en, *in*: *IEEE Transactions on Electromagnetic Compatibility* 62.*6* (Dec. 2020), pp. 2376–2385 (cit. on p. 22).

[DSV20c] Pieterjan De Meulemeester, Bart Scheers, and Guy A.E. Vandenbosch, « Reconstructing Video Images in Color Exploiting Compromising Video Emanations », en, *in*: *2020 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, Rome, Italy: IEEE, Sept. 2020, pp. 1–6 (cit. on p. 22).

[DT05] N. Dalal and B. Triggs, « Histograms of Oriented Gradients for Human Detection », en, *in*: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, San Diego, CA, USA: IEEE, 2005, pp. 886–893 (cit. on p. 41).

[DV18] Vincent Dumoulin and Francesco Visin, « A guide to convolution arithmetic for deep learning », en, *in*: *arXiv:1603.07285 [cs, stat]* (Jan. 2018), arXiv: 1603.07285 (cit. on p. 49).

[EFJ09] F. Estrada, D. Fleet, and A. Jepson, « Stochastic Image Denoising », en, *in*: *Procedings of the British Machine Vision Conference 2009*, London: British Machine Vision Association, 2009, pp. 117.1–117.11 (cit. on p. 57).

[FKE07] Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, « Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images », en, *in*: *IEEE Transactions on Image Processing* 16.5 (May 2007), pp. 1395–1411 (cit. on pp. 37, 40).

[Ge+17] Qian Ge, Yuval Yarom, Frank Li, and Gernot Heiser, « Your Processor Leaks Information - and There's Nothing You Can Do About It », en, *in*: *arXiv:1612.04474 [cs]* (Sept. 2017), arXiv: 1612.04474 (cit. on p. 18).

[Gen+18] D. Genkin, M. Pattani, R. Schuster, and E. Tromer, « Synesthesia: Detecting Screen Content via Remote Acoustic Side Channels », en, *in*: *arXiv:1809.02629* (2018) (cit. on pp. 20, 87).

[Gow+07] R.D. Gow et al., « A comprehensive tool for modeling CMOS image-sensor-noise performance », *in*: *IEEE Transactions on Electron Devices* 54.6 (2007), pp. 1321–1329 (cit. on pp. 35, 67).

[Gu+19] Shuhang Gu, Yawei Li, Luc Van Gool, and Radu Timofte, « Self-Guided Network for Fast Image Denoising », *in*: *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2511–2520 (cit. on pp. 38, 40, 49, 50, 70, 75, 76, 79).

[Gun+19] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang, « XAI—Explainable artificial intelligence », *in*: *Science Robotics* 4.*37* (2019) (cit. on p. 51).

[Guo+03] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer, « KNN model-based approach in classification », *in*: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, Springer, 2003, pp. 986–996 (cit. on p. 41).

[Guo+17] Tiantong Guo, Hojjat Seyed Mousavi, Tiep Huu Vu, and Vishal Monga, « Deep wavelet prediction for image super-resolution », *in*: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 104–113 (cit. on p. 28).

[Ha+19] Viet Khanh Ha et al., « Deep Learning Based Single Image Super-resolution: A Survey », en, *in*: *International Journal of Automation and Computing* 16.*4* (Aug. 2019), pp. 413–426 (cit. on p. 35).

[Hay+14] Y. Hayashi, N. Homma, M. Miura, T. Aoki, and H. Sone, « A Threat for Tablet PCs in Public Space: Remote Visualization of Screen Images Using EM Emanation », en, *in*: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*, Scottsdale, Arizona, USA: ACM Press, 2014, pp. 954–965 (cit. on p. 20).

[Hay+17] Y. Hayashi, N. Homma, Y. Toriumi, K. Takaya, and T. Aoki, « Remote Visualization of Screen Images Using a Pseudo-Antenna That Blends Into the Mobile Environment », en, *in*: *IEEE Transactions on Electromagnetic Compatibility* 59.*1* (2017), pp. 24–33 (cit. on p. 20).

[HD19] Dan Hendrycks and Thomas Dietterich, « Benchmarking Neural Network Robustness to Common Corruptions and Perturbations », en, *in*: (2019), p. 16 (cit. on p. 41).

[He+16a] K. He, X. Zhang, S. Ren, and J. Sun, « Deep Residual Learning for Image Recognition », en, *in*: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, 2016, pp. 770–778 (cit. on pp. 41, 71, 91).

[He+16b] K. He, X. Zhang, S. Ren, and J. Sun, « Identity Mappings in Deep Residual Networks », en, *in*: *Computer Vision – ECCV 2016*, vol. 9908, Cham: Springer International Publishing, 2016, pp. 630–645 (cit. on pp. 41, 71, 73, 84).

[He+17] K. He, G. Gkioxari, P. Dollar, and R. Girshick, « Mask R-CNN », en, *in*: *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice: IEEE, 2017, pp. 2980–2988 (cit. on pp. 27, 88).

[He+18] Y. He, J. Lin, Z. Liu, H. Wang, L-J. Li, and S. Han, « AMC: AutoML for Model Compression and Acceleration on Mobile Devices », en, *in*: *Computer Vision – ECCV 2018*, ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, vol. 11211, Cham: Springer International Publishing, 2018, pp. 815–832 (cit. on p. 95).

[How+17] A.G. Howard et al., « MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications », en, *in*: *arXiv:1704.04861* (2017) (cit. on pp. 71, 73, 84).

[Hua+17] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, « Densely Connected Convolutional Networks », en, *in*: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, July 2017, pp. 2261–2269 (cit. on pp. 41, 71, 73, 84).

[JAF16] J. Johnson, A. Alahi, and L. Fei-Fei, « Perceptual Losses for Real-Time Style Transfer and Super-Resolution », en, *in*: *Computer Vision – ECCV 2016*, ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling, vol. 9906, Cham: Springer International Publishing, 2016, pp. 694–711 (cit. on p. 64).

[JIP10] Haris Javaid, Aleksander Ignjatovic, and Sri Parameswaran, « Fidelity metrics for estimation models », en, *in*: *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, USA: IEEE, Nov. 2010, pp. 1–8 (cit. on p. 64).

[JS09] V. Jain and S. Seung, « Natural image denoising with convolutional networks », *in*: *Advances in neural information processing systems*, 2009, pp. 769–776 (cit. on pp. 38, 40, 70).

[KB14] D. P. Kingma and J. Ba, « Adam: A method for stochastic optimization », *in*: *arXiv preprint arXiv:1412.6980* (2014) (cit. on p. 77).

[KBJ19] A. Krull, T-O. Buchholz, and F. Jug, « Noise2Void - Learning Denoising From Single Noisy Images », en, *in*: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), p. 9 (cit. on pp. 39, 40, 58, 77, 79).

[Koc+18] Paul Kocher et al., « Spectre Attacks: Exploiting Speculative Execution », en, *in*: *arXiv:1801.01203 [cs]* (Jan. 2018), arXiv: 1801.01203 (cit. on p. 18).

[Kri09] Alex Krizhevsky, « Learning Multiple Layers of Features from Tiny Images », en, *in*: (2009), p. 60 (cit. on pp. 46, 47).

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, « ImageNet Classification with Deep Convolutional Neural Networks », *in*: *Advances in Neural Information Processing Systems*, ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, vol. 25, Curran Associates, Inc., 2012 (cit. on pp. 41, 50).

[Kuh02] Markus Guenther Kuhn, « Compromising emanations: eavesdropping risks of computer displays », PhD Thesis, Citeseer, 2002 (cit. on p. 26).

[Kuh13] M. G. Kuhn, « Compromising Emanations of LCD TV Sets », en, *in*: *IEEE Transactions on Electromagnetic Compatibility* 55.*3* (2013), pp. 564–570 (cit. on pp. 20, 61, 87, 89, 94).

[Lav+21] Corentin Lavaud, Robin Gerzaguet, Matthieu Gautier, Olivier Berder, Erwan Nogues, and Stephane Molton, « Whispering Devices: A Survey on How Side-channels Lead to Compromised Information », en, *in*: *Journal of Hardware and Systems Security* (Mar. 2021) (cit. on pp. 10, 18, 19, 110).

[LeC+98] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al., « Gradient-based learning applied to document recognition », *in*: *Proceedings of the IEEE* 86.*11* (1998), pp. 2278–2324 (cit. on pp. 41, 49, 71).

[Led+17] C. Ledig et al., « Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network », en, *in*: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, July 2017, pp. 105–114 (cit. on pp. 70, 75, 76, 79).

[Leh+18] J. Lehtinen et al., « Noise2Noise: Learning Image Restoration without Clean Data », en, *in*: *CoRR* (2018) (cit. on pp. 39, 40, 58, 70).

[Lem+19] Florian Lemarchand, Cyril Marlin, Florent Montreuil, Erwan Nogues, and Maxime Pelcat, « ToxicIA: Apprentissage Profond Appliquéà l'Analyse des Signaux Parasites Compromettants », *in*: *C&ESAR 2019 IA & Défense*, 2019 (cit. on pp. 13, 113).

[Lem+20a] Florian Lemarchand, Thomas Findeli, Erwan Nogues, and Maxime Pelcat, « Noisebreaker: Gradual image denoising guided by noise analysis », *in*: *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2020, pp. 1–6 (cit. on pp. 12, 112).

[Lem+20b] Florian Lemarchand, Cyril Marlin, Florent Montreuil, Erwan Nogues, and Maxime Pelcat, « Electro-Magnetic Side-Channel Attack Through Learned Denoising and Classification », *in*: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2882–2886 (cit. on pp. 13, 113).

[Lem+20c] Florian Lemarchand, Eduardo Fernandes Montesuma, Maxime Pelcat, and Erwan Nogues, « OpenDenoising: an Extensible Benchmark for Building Comparative Studies of Image Denoisers », *in*: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 2648–2652 (cit. on pp. 11, 112).

[LH18] Zhizhong Li and Derek Hoiem, « Learning without Forgetting », en, *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.*12* (Dec. 2018), pp. 2935–2947 (cit. on p. 51).

[Li+16] Zhi Li, Anne Aaron, Ioannis Katsavounidis, Anush Moorthy, and Megha Manohara, « Toward a practical perceptual video quality metric », *in*: *The Netflix Tech Blog* 6.*2* (2016) (cit. on p. 43).

[Li+20a] Qiufu Li, Linlin Shen, Sheng Guo, and Zhihui Lai, « Wavelet Integrated CNNs for Noise-Robust Image Classification », en, *in*: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, June 2020, pp. 7243–7252 (cit. on p. 41).

[Li+20b] Xin Li et al., « Learning Disentangled Feature Representation for Hybrid-distorted Image Restoration », en, *in*: *arXiv:2007.11430 [cs, eess]* (July 2020), arXiv: 2007.11430 (cit. on p. 28).

[Lin+14] T.-Y. Lin et al., « Microsoft COCO: Common Objects in Context », en, *in*: *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 740–755 (cit. on p. 92).

[Liu+18] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, « Multi-level Wavelet-CNN for Image Restoration », en, *in*: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Salt Lake City, UT: IEEE, June 2018, pp. 886–88609 (cit. on pp. 35–38, 40, 49, 50, 55, 58, 70, 75, 76, 79).

[Low04] David G. Lowe, « Distinctive Image Features from Scale-Invariant Keypoints », en, *in*: *International Journal of Computer Vision* 60.*2* (Nov. 2004), pp. 91–110 (cit. on p. 41).

[LSJ20] F. Liu, Q. Song, and G. Jin, « The classification and denoising of image noise based on deep neural networks », en, *in*: *Applied Intelligence* (Mar. 2020) (cit. on pp. 12, 72, 74, 76–79, 81, 82, 86, 105, 112).

[LTO13] X. Liu, M. Tanaka, and M. Okutomi, « Single-image noise level estimation for blind denoising », *in*: *IEEE transactions on image processing* 22.*12* (2013), pp. 5226–5237 (cit. on pp. 38, 40, 71).

[Mah+18] D. Mahajan et al., « Exploring the Limits of Weakly Supervised Pretraining », en, *in*: *Computer Vision – ECCV 2018*, ed. by Martial Hebert, Cristian Sminchisescu, and Yair Weiss, vol. 11206, Cham: Springer International Publishing, 2018, pp. 185–201 (cit. on p. 92).

[Mar+01] 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 », en, *in*: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, Vancouver, BC, Canada: IEEE Comput. Soc, 2001, pp. 416–423 (cit. on pp. 46, 47, 56, 77, 97).

[MG18] A. Mikolajczyk and M. Grochowski, « Data augmentation for improving deep learning in image classification problem », en, *in*: *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, Swinoujście: IEEE, 2018, pp. 117–122 (cit. on p. 91).

[Min+20] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos, « Image Segmentation Using Deep Learning: A Survey », en, *in*: *arXiv:2001.05566 [cs]* (Nov. 2020), arXiv: 2001.05566 (cit. on p. 39).

[Mit] Joe Mitola, « As communications technology continues its rapid transition from analog to digital, more functions of contemporary radio systems are implemented in software, leading toward the software radio. What distinguishes soft-

ware radio architectures? What new capabilities are more economically accessible in software radios than digital radios? What are the pitfalls? And the prognosis? », en, *in*: (), p. 13 (cit. on p. 20).

[MSY16] X. Mao, C. Shen, and Y-B. Yang, « Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections », en, *in*: *Advances in Neural Information Processing Systems 29 (NIPS 2016)* (2016), p. 9 (cit. on pp. 35, 38, 40, 57, 70).

[Mur+06] Kevin P Murphy et al., « Naive bayes classifiers », *in*: *University of British Columbia* 18.*60* (2006) (cit. on p. 41).

[Mye99] G. Myers, « A Fast Bit-vector Algorithm for Approximate String Matching Based on Dynamic Programming », *in*: *J. ACM* 46.*3* (1999), pp. 395–415 (cit. on p. 89).

[Nat82] National Security Agency, *NACSIM 5000 TEMPEST FUNDAMENTALS*, 1982 (cit. on p. 19).

[PR17] T. Plotz and S. Roth, « Benchmarking Denoising Algorithms with Real Photographs », en, *in*: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, July 2017, pp. 2750–2759 (cit. on pp. 35, 56, 67, 69).

[RD18] P.-M. Ricordel and E. Duponchelle, « Risques associés aux signaux parasites compromettants : le cas des câbles DVI et HDMI », *in*: *Symposium sur la Sécurité des Technologies de l'Information et des Communications (SSTIC)*, 2018 (cit. on p. 22).

[Ren+17] S. Ren, K. He, R. Girshick, and J. Sun, « Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks », en, *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.*6* (2017), pp. 1137–1149 (cit. on p. 91).

[RFB15] O. Ronneberger, P. Fischer, and T. Brox, « U-Net: Convolutional Networks for Biomedical Image Segmentation », en, *in*: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, Springer, 2015, pp. 234–241 (cit. on pp. 39, 58).

[SA10] Y Suzuki and Y Akiyama, « Jamming technique to prevent information leakage caused by unintentional emissions of PC video signals », en, *in*: *2010 IEEE International Symposium on Electromagnetic Compatibility*, Fort Lauderdale, FL: IEEE, July 2010, pp. 132–137 (cit. on p. 19).

[SDC19] D. Sil, A. Dutta, and A. Chandra, « Convolutional Neural Networks for Noise Classification and Denoising of Images », en, *in*: *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, Kochi, India: IEEE, Oct. 2019, pp. 447–451 (cit. on p. 72).

[Shi+16] Wenzhe Shi et al., « Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network », en, *in*: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, June 2016, pp. 1874–1883 (cit. on p. 49).

[SLD17] E. Shelhamer, J. Long, and T. Darrell, « Fully Convolutional Networks for Semantic Segmentation », en, *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.*4* (Apr. 2017), pp. 640–651 (cit. on p. 91).

[Smi07] R. Smith, « An Overview of the Tesseract OCR Engine », en, *in*: *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, Curitiba, Parana, Brazil: IEEE, Sept. 2007, pp. 629–633 (cit. on p. 94).

[SOO18] M. Suganuma, M. Ozay, and T. Okatani, « Exploiting the Potential of Standard Convolutional Autoencoders for Image Restoration by Evolutionary Search », *in*: *Proceedings of the 35th International Conference on Machine Learning*, ed. by Jennifer Dy and Andreas Krause, vol. 80, Proceedings of Machine Learning Research, Stockholmsmässan, Stockholm Sweden: PMLR, July 2018, pp. 4771–4780 (cit. on p. 38).

[SP11] Jorge Sánchez and Florent Perronnin, « High-dimensional signature compression for large-scale image classification », *in*: *CVPR 2011*, IEEE, 2011, pp. 1665–1672 (cit. on p. 50).

[SSR11] U. Schmidt, K. Schelten, and S. Roth, « Bayesian deblurring with integrated noise estimation », *in*: *CVPR 2011*, IEEE, 2011, pp. 2625–2632 (cit. on pp. 38, 40, 71).

[SZ15] Karen Simonyan and Andrew Zisserman, « Very Deep Convolutional Networks for Large-Scale Image Recognition », en, *in*: *arXiv:1409.1556 [cs]* (Apr. 2015), arXiv: 1409.1556 (cit. on p. 57).

[Taj+16] N. Tajbakhsh et al., « Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? », en, *in*: *IEEE Transactions on Medical Imaging* 35.*5* (May 2016), arXiv: 1706.00712, pp. 1299–1312 (cit. on p. 71).

[Tia+20] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C-W. Lin, « Deep Learning on Image Denoising: An overview », en, *in*: *arXiv:1912.13171 [cs, eess]* (Jan. 2020), arXiv: 1912.13171 (cit. on p. 35).

[TO03] Antonio Torralba and Aude Oliva, « Statistics of natural image categories », *in*: *Network: computation in neural systems* 14.*3* (2003), pp. 391–412 (cit. on p. 33).

[Tou+21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou, « Training data-efficient image transformers & distillation through attention », en, *in*: *arXiv:2012.12877 [cs]* (Jan. 2021), arXiv: 2012.12877 (cit. on p. 39).

[UVL20] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, « Deep Image Prior », en, *in*: *International Journal of Computer Vision* 128.*7* (July 2020), pp. 1867–1888 (cit. on pp. 39, 40).

[Van85] W. Van Eck, « Electromagnetic radiation from video display units: An eavesdropping risk? », en, *in*: *Computers & Security* 4.*4* (1985), pp. 269–286 (cit. on pp. 10, 20, 61, 87, 109).

[Vas+17] Ashish Vaswani et al., « Attention Is All You Need », en, *in*: *arXiv:1706.03762 [cs]* (Dec. 2017), arXiv: 1706.03762 (cit. on p. 39).

[VES15] VESA, *About DisplayPort*, en-US, 2015 (cit. on p. 21).

[Vin+10] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.A. Manzagol, « Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion », en, *in*: *Journal of Machine Learning Research* 11 (2010), pp. 3371–3408 (cit. on p. 57).

[VP09] M. Vuagnoux and S. Pasini, « Compromising Electromagnetic Emanations of Wired and Wireless Keyboards », fr, *in*: *Proceedings of the 18th USENIX Security Symposium* (2009), pp. 1–16 (cit. on p. 20).

[Wan+04] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, « Image Quality Assessment: From Error Visibility to Structural Similarity », en, *in*: *IEEE Transactions on Image Processing* 13.*4* (Apr. 2004), pp. 600–612 (cit. on p. 42).

[WCH20] Zhihao Wang, Jian Chen, and Steven C.H. Hoi, « Deep Learning for Image Super-resolution: A Survey », *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), pp. 1–1 (cit. on p. 35).

[Xu+18] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang, « Real-world Noisy Image Denoising: A New Benchmark », en, *in*: *arXiv:1804.02603 [cs]* (Apr. 2018), arXiv: 1804.02603 (cit. on p. 56).

[YK16] Fisher Yu and Vladlen Koltun, « MULTI-SCALE CONTEXT AGGREGATION BY DILATED CONVOLUTIONS », en, *in*: (2016), p. 13 (cit. on p. 49).

[YS18] Dong Yang and Jian Sun, « BM3D-Net: A Convolutional Neural Network for Transform-Domain Collaborative Filtering », en, *in*: *IEEE Signal Processing Letters* 25.*1* (Jan. 2018), pp. 55–59 (cit. on pp. 38, 40).

[Yua+20] S. Yuan, R. Timofte, A. Leonardis, and G. Slabaugh, « Ntire 2020 challenge on image demoireing: Methods and results », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 460–461 (cit. on pp. 34, 69).

[Zha+14] Qian Zhao, Deyu Meng, Zongben Xu, Wangmeng Zuo, and Lei Zhang, « Robust Principal Component Analysis with Complex Noise », en, *in*: *International Conference on Machine Learning (ICML)* (2014), p. 9 (cit. on pp. 34, 71).

[Zha+16] C. Zhang, Z. Fang, P. Zhou, P. Pan, and J. Cong, « Caffeine: towards uniformed representation and acceleration for deep convolutional neural networks », en, *in*: *Proceedings of the 35th International Conference on Computer-Aided Design - ICCAD '16*, Austin, Texas: ACM Press, 2016, pp. 1–8 (cit. on p. 95).

[Zha+17] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, « Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising », en, *in*: *IEEE Transactions on Image Processing* 26.*7* (July 2017), pp. 3142–3155 (cit. on pp. 36, 38, 40, 50, 55, 57, 70, 76, 99).

[Zhe+19] B. Zheng, Y. Chen, X. Tian, F. Zhou, and X. Liu, « Implicit Dual-domain Convolutional Network for Robust Color Image Compression Artifact Reduction », en, *in*: *IEEE Transactions on Circuits and Systems for Video Technology* (2019), pp. 1–1 (cit. on pp. 37, 38, 40).

INSA de RENNES
Service des Formations

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse:
Deep-Learning Based Exploitation of Eavesdropped Images

Nom Prénom de l'auteur : LEMARCHAND FLORIAN

Membres du jury :
- Madame YANG Fan
- Monsieur STRAUSS Olivier
- Monsieur PELCAT Maxime
- Monsieur GOOSSENS Bart
- Monsieur PUECH William
- Monsieur BERRY François
- Monsieur COTTAIS Emmanuel
- Monsieur NOGUES Erwan

Président du jury : William PUECH

Date de la soutenance : 29 Septembre 2021

Reproduction de la these soutenue

☒ Thèse pouvant être reproduite en l'état
☐ Thèse pouvant être reproduite après corrections suggérées

Fait à Rennes, le 29 Septembre 2021

Signature du président de jury

Le Directeur,

A. MIRAOUI

DOCTORAT BRETAGNE LOIRE MATHSTIC

INSA RENNES

Titre : Exploitation d'Images Interceptées Basée Apprentissage Profond

Mot clés : Interception, Emanations Electro-Magnétiques, Apprentissage Profond, Débruitage

Résumé : La tendance récente est de rendre les données numériques disponibles à tout moment et en tout lieu, ce qui crée de nouvelles menaces de confidentialité. Les données sont échangées et consultées à l'aide de systèmse d'information (SI) et de leurs écrans. Un canal auxiliaire correspond à un chemin de données non intentionnel en opposition avec le canal traditionnel. En particulier, les canaux auxiliaires électromagnétiques (EM) sont dus aux champs émis par les câbles et les connecteurs vidéo lorsque leur tension interne change. Il a été démontré que le contenu d'un écran peut être recontruit à des dizaines de mètres à partir des émanations par canal auxiliaire. Jusqu'à aujourd'hui, les travaux menés de l'état de l'art sur la reconstruction d'images, à partir d'émanations EM, se sont principalement concentrés sur un point de vue traitement du signal.

Récemment, le domaine du traitement d'image a été révolutionné par l'apprentissage profond. Ces algorithmes ont dépassé les performances des algorithmes experts de l'état de l'art. Dans ce manuscrit, il est montré que les méthodes par apprentissage profond pour la restauration et l'interprétation d'images peuvent être appliquées aux images reconstruites depuis l'interception d'émanations EM. Le manuscrit étudie la corruption impliquée par l'interception et démontre que cette corruption est complexe. Le manuscrit propose des expériences et des contributions sur l'application des techniques par apprentissage profond à la restauration et à l'automatisation de l'interprétation des images interceptées.

Title: Deep-Learning Based Exploitation of Eavesdropped Images

Keywords: Eavesdropping, Electro-Magnetic Emanations, Deep Learning, Denoising

Abstract: The recent trend of processing is to make digital data available anytime anywhere, creating new confidentiality threats. Data is exchanged and consulted using Information Processing Equipments(IPEs) and their according Video Display Units (VDUs). A side-channel corresponds to an unintended data path in opposition to the legacy channel. In particular, Electro Magnetic (EM) side-channels are due to fields emitted by video cables and connectors when their inner voltage changes. It has been shown that the content of screen can be reconstructed from tens of meters using side-channel emanations. Until today, the work conducted on state of the art of image reconstruction, from EM emanations has focused on a signal processing point of view. Image processing has recently been revolutionnized by Deep Learning (DL). These algorithms have overpassed the performances of state of the art expert algorithms. In this thesis, we show that DL methods for image restoration and interpretation can successfully be applied to images reconstructed from EM emanations. The manuscript studies the image corruptions implied by eavesdropping and demonstrates that these corruptions are complex. The manuscript proposes experiments and contributions on the application of DL techniques to eavesdropped image restoration and interpretation automation.