Real-Time Image Denoising with Embedded Deep Learning: Review, Perspectives and Application to Information System Security

Florian Lemarchand, Erwan Nogues, Maxime Pelcat

Real-Time Image Denoising with Embedded Deep Learning: Review, Perspectives and Application to Information System Security

Florian Lemarchand [1], Erwan Nogues [2] and Maxime Pelcat [3]
Univ. Rennes, INSA Rennes,CNRS, IETR - UMR CNRS 6164 [1,2,3]
DGA-MI [2]
Institut Pascal, Clermont-Ferrand, UMR CNRS 6602 [3]
firstname.lastname@insa-rennes.fr

***Abstract*—Any information system emits, by conduction or radiation, compromising signals likely to be intercepted by an attacker. These leakage signals usually have low signal-to-noise ratio and the security of information systems depends on the capacity of an attacker to denoise them. Denoising is a major topic in signal processing, currently revolutionized by deep learning methods. In particular, the scope of image denoising is large and ranges from classical and low footprint techniques to computationally intensive deep learning techniques. Deep learning algorithms typically run onto energy costly computers using Graphics Processing Units (GPUs) and are currently hardly available in an embedded context. This paper gives an overview of existing methods for embedded image denoising and proposes some perspectives. A case study is also presented that motivates our research on the domain.**

I. Introduction

All systems processing data introduce distortions and noise while processing. Indeed, noise is inherent to analog electronic devices and even digital processing introduces rounding and sampling noises. Although electronic circuits are ever more accurate, none of them can claim a noiseless processing. Different sources of noise appear along the stream of data from sensors to actuators. Denoising is then needed when noise jeopardizes the data interpretation. Embedded denoising close to sensors has several advantages when compared to denoising post-acquisition. However, it also introduces new challenges because state-of-the-art methods are computationally intensive and do not fit easily onto embedded platforms.

The paper is organized as follows. Section II overviews image denoising methods, image quality assessment metrics and embedded processing platforms. Section III proposes perspectives on embedding image denoising methods. A motivating case study requiring such advances, and relative to information systems protection, is depicted in Section IV before the conclusion of Section V.

II. Related Work

In this section, we focus on two areas, namely, image denoising and embedded processing platforms. We present an overview of both topics and point out both strengths and weakness of existing models and platforms.

A. Image Denoising

Image denoising by digital signal processing methods has been extensively studied [1] and is an essential step in many computer vision applications. It is based on assumptions of the nature of the target noise. As an example of classical denoising methods, a well known method for Gaussian white noise removal in images is Block-Matching 3D (BM3D) [2]. BM3D uses thresholding in the transform domain to remove data considered as noise.

Deep learning algorithms have recently stood out from the crowd for solving many signal processing problems. These trained models have an extreme ability to fit complex problems. Recent Graphics Processing Unit (GPU) architectures have been optimized to support deep learning workloads and have fostered ever deeper networks, extracting structured information from data and providing results where classical algorithms fail. The spread of deep learning has occurred in image denoising and several models initially developed for other purposes have been turned into denoisers [3]. Denoising Convolutionnal Neural Networks (DnCNNs) [4] are designed in this way. DnCNNs make use of Convolutional Neural Networks (CNNs) to blindly remove Gaussian noise, without prior knowledge on the noise level.

Supervised learning techniques such as denoising autoencoders [5], [6] are able to denoise images without restriction on the type of noise, as long as datasets are available with both noisy and reference images. Autoencoders algorithms learn to map their input image to a latent space (encoding) and project back the latent representation to the input space (decoding). Autoencoders learn a denoising model by minimizing a loss function which evaluates the difference between the autoencoder output and the reference. Recent methods, such as Noise2Noise [7], infer denoising strategies without any clean reference data. The Noise2Noise algorithm learns a representation of the noise by looking only at noisy samples.

B. Embedded Processing Platforms and their Limits

Most state of the art denoising methods currently run only onto powerful GPUs that consume hundreds of Watts. Porting these algorithms onto existing embedded platforms with power consumption typically under 20W is challenging for different

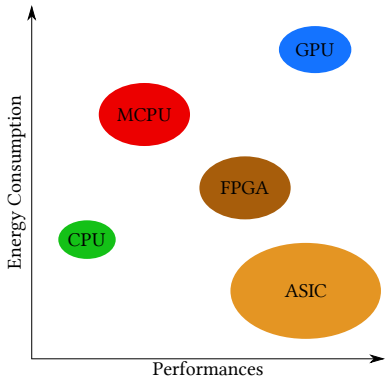

Fig. 1. Sketching performances versus energy consumption of several processing platforms when running DNNs. Scales are rough and are only for illustration purpose as well as the size of the ellipses representing the programming complexity of platforms.

reasons. These methods are either too large in terms of memory storage or too computationally intensive, meaning that system response time explodes.

Figure 1 sketches embedded platforms efficiency based on their processing performance and energy consumption, when executing DNNs. The size of the ellipses roughly represents the programming effort to obtain a functional system. The general purpose Central Processing Unit (CPU) is easily programmable but does not really suit DNNs because of a lack of parallelism. Parallelism can be improved by increasing the number of cores, leading to multi or many-core CPUs (MCPUs). The drawback of these platforms is their programming complexity that skyrockets with the number of cores.

GPUs are currently the most used platforms to run DNNs. Indeed, they present an important degree of parallelism and thus achieve good performances. Furthermore, most deep learning frameworks propose abstraction layers that allow developers to design deep learning applications without even writing a code of GPU-specific programming language. The main drawback of high-end GPUs is their energy consumption. These systems require up to several hundred Watts. Moderns platforms referred as embedded GPUs propose less energy consuming processing elements while still showing good results and good programmability.

Recently, authors have proposed to use Field Programmable Gate Arrays (FPGAs) to perform all or part of DNN computation. As an example, in [8], Abdelouahab *et al.* propose some tactics to map CNNs onto FPGA. They also work on strategies where only several layers of a networks are mapped on FPGA as a pre-processing and rest is done using other processing elements. This can be viewed as a combination of software and hardware acceleration.

Application-Specific Integrated Circuits (ASICs) are hardware defined circuits tailored for a specific application. Since they are application-specific, their circuitry is optimized for a given problem. ASICs can thus achieve orders of magnitudes better efficiency, both in terms of energy consumption and performances. On the other hand, an important effort is required to design an ASIC. It is expensive and no change can be made once the circuit produced. Because of that, ASICs are used only for large markets.

III. Research Perspectives

In this section, we present some perspectives for embedded deep learning denoisers that we intend to explore in the next years. Different strategies are evoked and discussed, assuming that denoising is performed by a trained neural network.

A. Networks Reductions

A possible way to reduce the needed memory storage and the computational burden is to reduce the size of the networks. In [9], Han *et al.* propose a method to compress neural networks without significant loss of accuracy. For instance, they achieve a weight storage reduction of 35, 39 and 49x on AlexNet, LeNet and VGG-16, respectively. Their method operates by pruning low impact connections, quantizing the network by sharing parameters, and finally applying Huffman Coding. Network quantization is further studied in [10] where all inputs and weights are binary quantized. Authors main interest is to run inference of DNNs onto mobile CPUs.

B. New Networks and Strategies

Instead of *compressing* existing networks, an avenue of research is to develop new networks. Since DNNs appeared and started to draw the attention of the image processing community, the major way of gaining algorithmic performance has been to make networks larger and choose their size at the limit of what can reasonably be trained with backpropagation on GPUs. Additionally to growing networks, new methods have significantly improved DNN performances. In 2016, [11] presented some tricks to raise network sizes up to a hundred layers like ResNet101. The following year, [12] proposed a network using new strategies claiming to obtain the same accuracy as AlexNet with 50x fewer parameters. The same year, MobileNets [13] proposed a series of DNNs designed for mobile vision applications.

When different steps of a deep learning algorithm are time multiplexed, high complexity translates into large response times. The algorithm may fit on the platform but the computational strength is too low to provide a response time necessary for the application requirements. In that case, solutions exist to accelerate the execution using approximate computing and hardware optimization. As an example, in [14], Marty *et al.* use overclocking and fault tolerance to accelerate a CNN implemented on a FPGA.

IV. Motivating Case Study

In this section, we present a system that benefits from advances in the field of embedded image denoising. All electronic devices produce Electro Magnetic (EM) emanations that not only interfere with radio devices but also compromise the data managed by the information system. A third party may perform a side-channel analysis and recover the original information, hence compromising the privacy of the system. While pioneering work of the domain focused

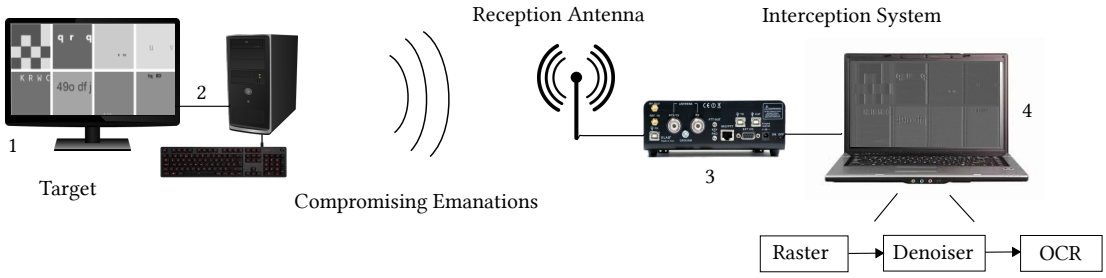

Fig. 2. Proposed System: it includes an eavesdropped screen (1) displaying sensitive information connected to an information system (2), and an interception chain including an SDR receiver (3) sending samples to a host computer (4) that implements signal processing including a deep learning denoiser and CR.

on analog signals [15], recent studies extend eavesdropping exploits using an EM side-channel attack to digital signals and embedded circuits [16]. 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 on advanced signal processing to further stretch the limits of the side-channel attack using EM emanations [17].

In that context, the method drawn in Figure 2 is proposed as an audit solution to assess the security of an information system and especially its display. The system first intercepts the emanations and reconstructs it using a raster. Image information comes with a very strong noise of complex nature. A computer is then used to denoise the reconstructed image and extract its content. It is assumed that the displayed and intercepted sample contains only characters. The content is extracted using transfer learning on a pre-trained implementation of Mask R-CNN [18]. Once content is found, it is possible to compare it to the content of the reference sample displayed on the audited screen. The more content is retrieved, the bigger the information leakage is.

To make the audit more applicable to real systems, the whole interception system, with SDR, raster, denoiser, and interpreter, should run onto a unique platform. Currently, this is not possible since the denoiser/interpreter is run on a computer and does not fit any embedded platform. This context motivates our studies on embedded deep learning-based image denoising.

V. Conclusion

In a context of constant growing attention on deep learning techniques, their embedding becomes a major topic of research. Denoising is required in many image processing applications to enhance the raw data acquired from a given source. We propose in this paper a brief review of conventional and deep learning denoising techniques, as well as of embedded platforms likely to host denoising algorithms. We notice that state-of-the-art denoising models are not designed for embedded systems. Embedded deep learning exists, but mainly image classification and segmentation have been studied in this context. From that observation, we propose perspectives to go forward on embedded deep learning for image denoising.

Acknowledgment

This study is supported by the "Pôle d'Excellence Cyber".

References

[1] A. Buades, B. Coll, and J. M. Morel. A Review of Image Denoising Algorithms, with a New One. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[3] Chunwei Tian, Yong Xu, Lunke Fei, and Ke Yan. Deep Learning for Image Denoising: A Survey. *arXiv:1810.05052*, 2018.

[4] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. arXiv: 1608.03981.

[5] 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. *Journal of Machine Learning Research*, 11:3371–3408, 2010.

[6] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pages 234–241. Springer, 2015.

[7] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila. Noise2noise: Learning Image Restoration without Clean Data. *CoRR*, 2018.

[8] K. Abdelouahab, M. Pelcat, J. Serot, C. Bourrasset, and F. Berry. Tactics to Directly Map CNN Graphs on Embedded FPGAs. *IEEE Embedded Systems Letters*, 9(4):113–116, 2017.

[9] S. Han, H. Mao, and W.J. Dally. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *arXiv:1510.00149*, 2015.

[10] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. In *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 525–542. Springer, 2016.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, 2016. IEEE.

[12] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer. SqueezeNet: AlexNet-level Accuracy with 50x Fewer Parameters and <1mb Model Size. *arXiv:1602.07360*, 2016.

[13] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861*, 2017.

[14] T. Marty, T. Yuki, and S. Derrien. Algorithm Level Timing Speculation for Convolutional Neural Network Accelerators. Technical Report, 2018.

[15] W. Van Eck. Electromagnetic radiation from video display units: An eavesdropping risk? *Computers & Security*, 4(4):269–286, 1985.

[16] M. G. Kuhn. Compromising Emanations of LCD TV Sets. *IEEE Transactions on Electromagnetic Compatibility*, 55(3):564–570, 2013.

[17] D. Genkin, M. Pattani, R. Schuster, and E. Tromer. Synesthesia: Detecting Screen Content via Remote Acoustic Side Channels. *arXiv:1809.02629*, 2018.

[18] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, Venice, 2017. IEEE.