Towards Eavesdropped Image Denoising

Florian LEMARCHAND [†] ,
Erwan NOGUES [†‡] and Maxime PELCAT [†]

[†] **Univ. Rennes, INSA Rennes, IETR - UMR CNRS 6164**

[‡] DGA-MI, Bruz, France

Who are we?

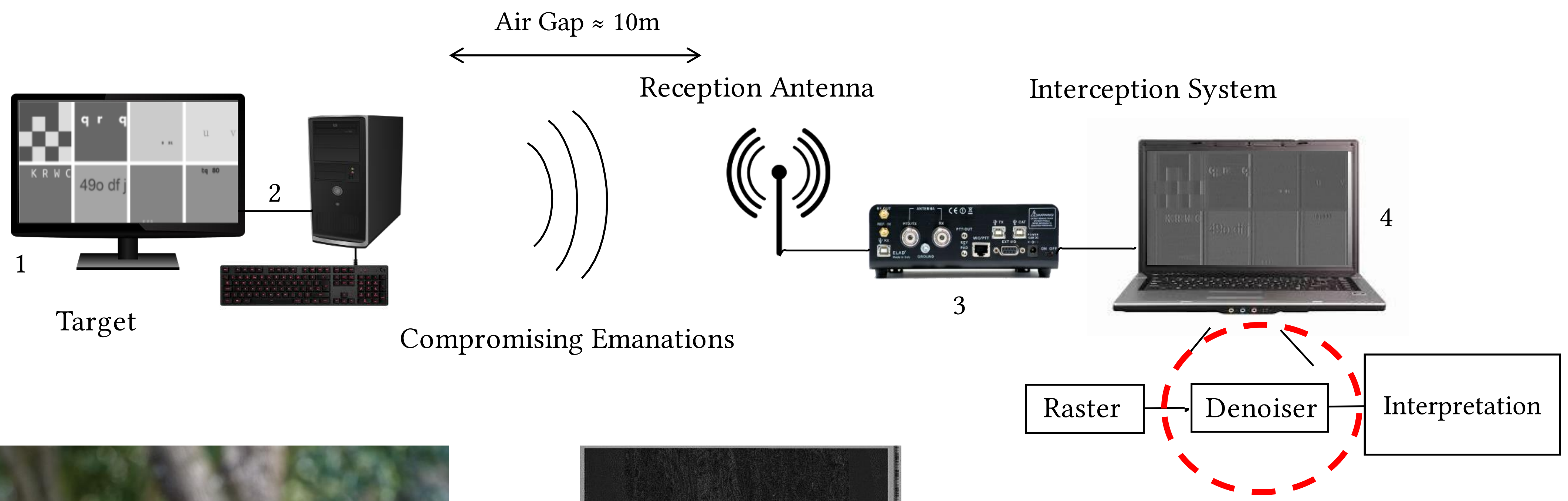
Florian Lemarchand

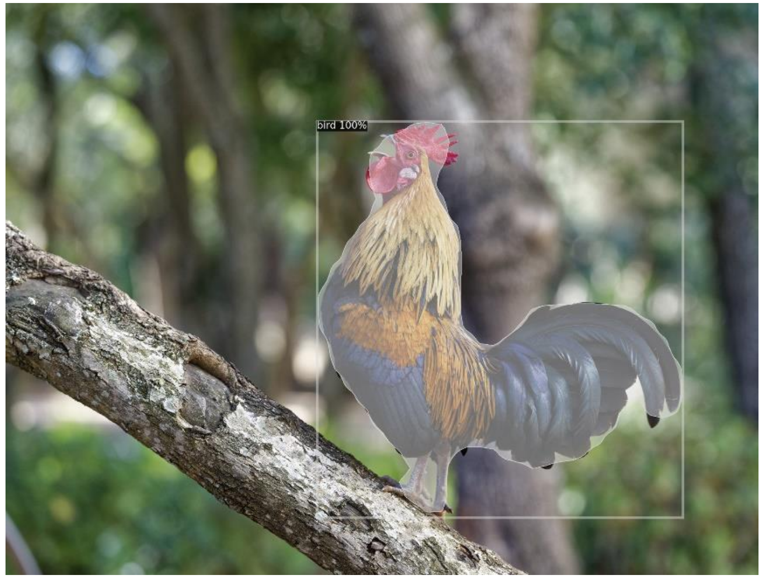
- Engineer INSA Rennes « Electronique et Informatique Industrielle (EII) », 2018
- PhD Student since October 2018:
    - Lab: “Institut D’Electronique et des Technologies du numéRique de Rennes” (IETR)
    - Team: “ Video Analysis and Architecture Design for Embedded Resources ” (VAADER)
    - PhD Founded by “ Pole d’Excellence Cyber ” (PEC) → Bretagne council et French ministry of armed forces
    - Advisors : Erwan Nogues and Maxime Pelcat
    - PhD Subject:
        - “Recognition of Images and Intercepted Signal using Artificial Intelligence ”
    - Technical Domains :
        - Image Restoration
        - Machine (Deep) Learning
- More information on my research on my webpage!
- Contact: florian.lemarchand@insa-rennes.fr
- What about you?
    - Background: Image Processing? Machine Learning?

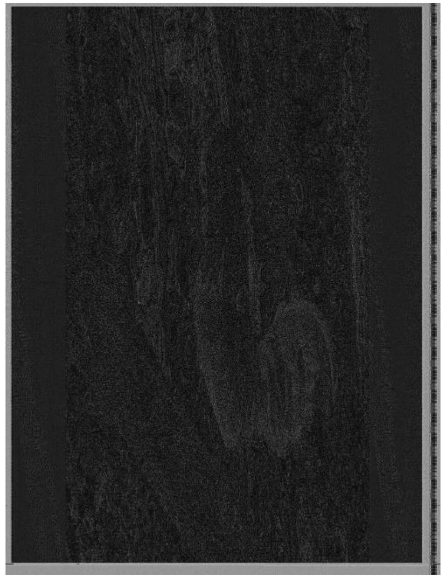
Plan

Plan

Context


Air Gap ≈ 10m
Reception Antenna
Interception System
1
2
3
4
Target
Compromising Emanations
Raster
Denoiser
Interpretation
How to automate?



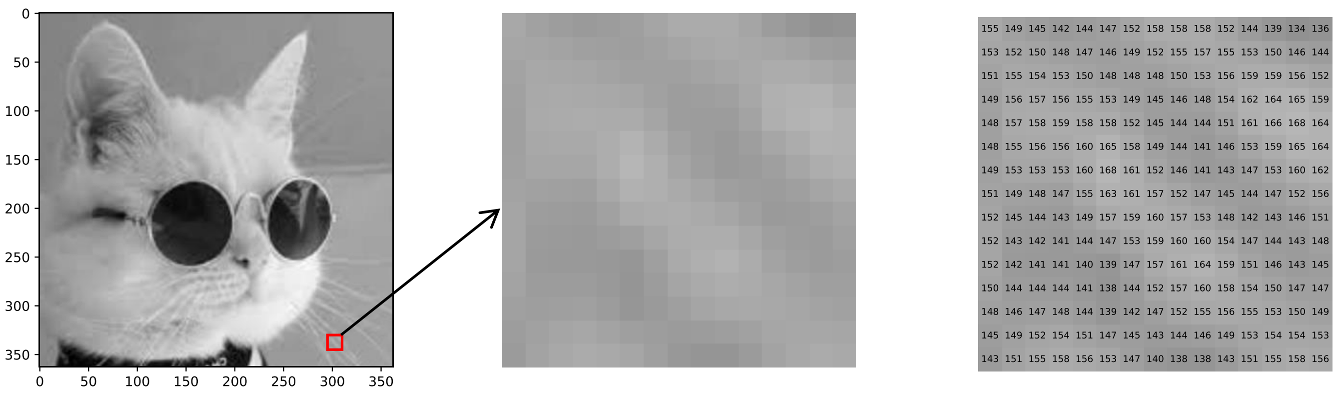


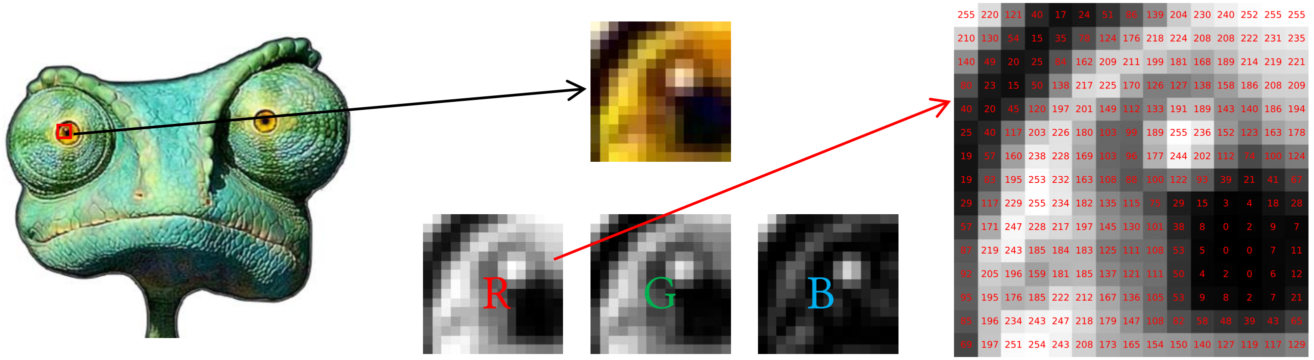
Plan

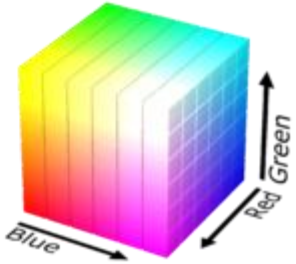
Problem Definition

Digital Image

- Pixel (Picture Element)
  → $p \in [0, 255]$ or $[0., 1.]$
- Image → HxWxC array of pixels
  - Height, Width, Channels
  - C = 1 for grayscale , C=3 for RGB (Red Green Blue), C>100 for hyperspectral
- Content:
  - Natural Images (pictures)
  - Synthetic Images (computer screen, video games, cartoon, …)







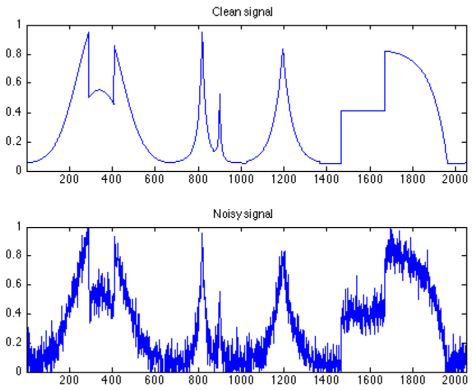
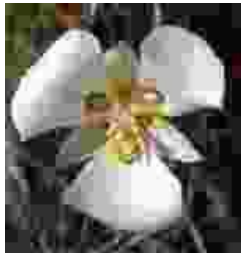


Image Noise

- Noise ≠ Signal
    - Signal is the information contained in an image
    - Noise is the undesired variation that disrupts the interpretation

- Noise Sources
    - Defects of sensing and transmission systems
        - Image sensors: Defects of hardware surfaces / Analogic to Digital conversion errors
        - Signal Loss (electro-magnetic interception)
        - Sensing content itself: when only few photons (space imaging)
        - Lossy Compression/Decompression (JPEG)
    - Poor acquisition conditions (light, rain, blur)
    - Falsification (incoherence in Bayer patterns)

- Noise Types:
    - Pixelwise
    - Spatially Correlated
    - Data Dependent

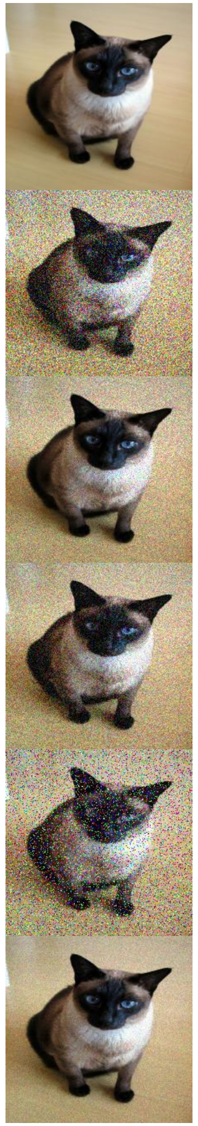


Problem Definition

Noise Models

Primary Noise:

- No Noise — $\emptyset$ — p_o
- Gaussian — $\mathcal{N}(\sigma_g)$ — $p_n = p_o + \mathcal{N}(\sigma_g)$
- Speckle — $\mathcal{S}(\sigma_s)$ — $p_n = p_o + \mathcal{N}(\sigma_g) \times p_o$
- Uniform — $\mathcal{U}(s)$ — $p_n = p_o + \mathcal{U}(s)$
- Bernoulli — $\mathcal{B}(p)$ — $p_n = \begin{cases} \text{choice(min, max), if rand()} \in [0, p[\\ p_n, \text{otherwise} \end{cases}$
- Poisson — $\mathcal{P}$ — $p_n = p_o + \mathcal{P}(p_o)$



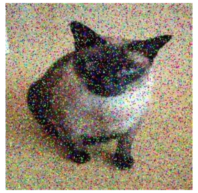
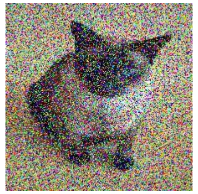
Sequential Mixture Noise:

$$x \rightarrow y = h_1(x) \rightarrow y' = h_2(h_1(x))$$

- Gaussian and Bernoulli

 $\xrightarrow{\mathcal{N}(\sigma_g)}$  $\xrightarrow{\mathcal{B}(p)}$ 

- Bernoulli and Speckle

 $\xrightarrow{\mathcal{B}(p)}$  $\xrightarrow{\mathcal{S}(\sigma_s)}$ 

Problem Definition

How to measure how noisy is an image?

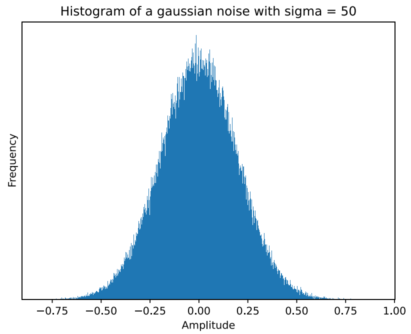
- Subjective/Qualitative rating
  - N subjects ask to rate image quality (A=x, B=y) or compare two versions (A > B)
    - Mean Opinion Score (MOS)
- Objective Metrics
  - Mean Squared Error (MSE) / Root MSE (RMSE) / Sum of Absolute Errors (SAE)

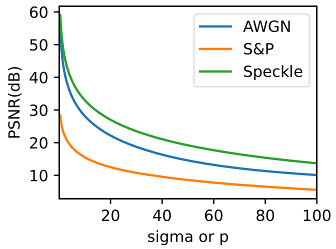
$$MSE(x,y) = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(y(i,j)-x(i,j))^2$$

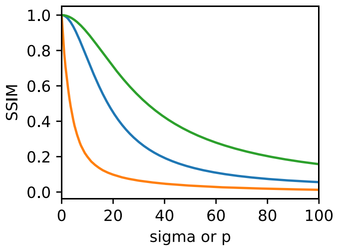
  - Peak Noise to Signal Ratio (PSNR)

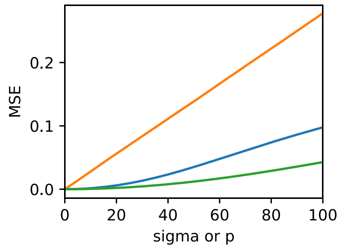
$$PSNR(x,y) = 10.log_{10}\left(\frac{MAX^2}{MSE}\right)$$

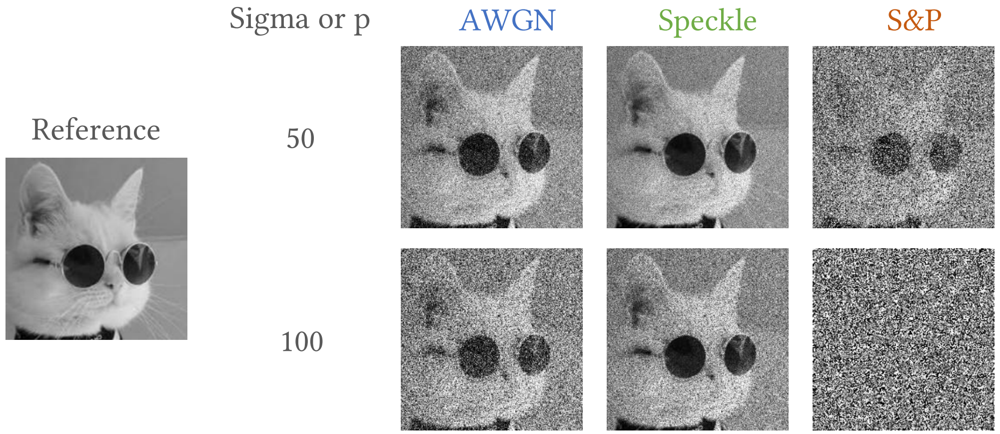
  - Structural SIMilarity (SSIM) [Wang04]→ measure spatial coherence
- Learned metrics:
  - Predict subjective rating using a Neural network [Talebi18]
- Histogram of pixel values







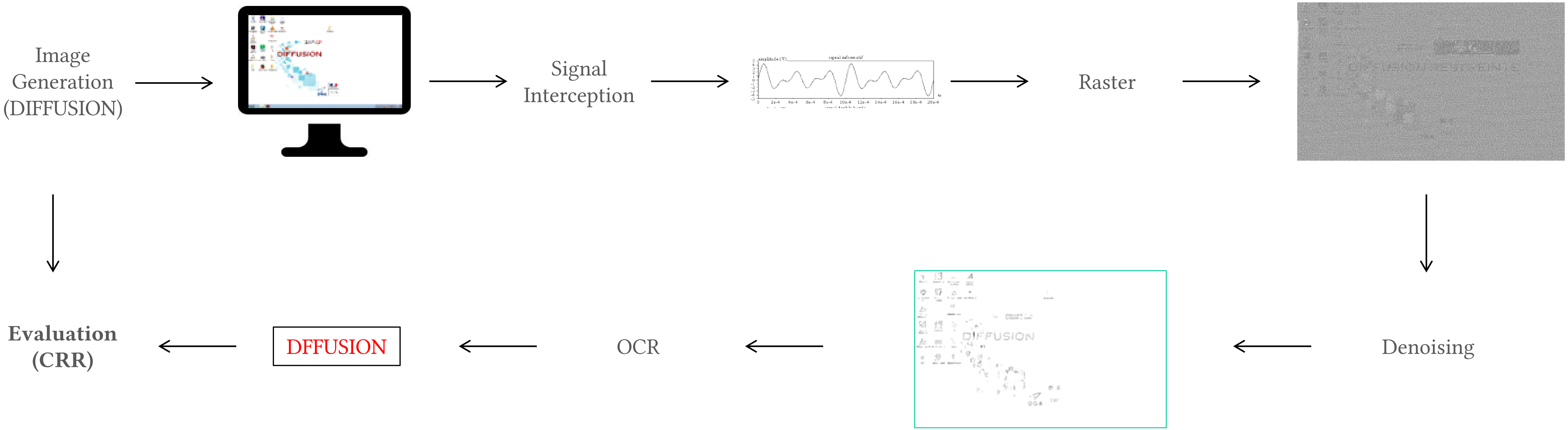




[Wang04] Wang, Zhou, et al. "Image quality assessment: from error visibility to structural similarity." IEEE transactions on image processing 13.4 (2004): 600-612.
[Talebi18] Talebi, Hossein, and Peyman Milanfar. "NIMA: Neural image assessment." IEEE Transactions on Image Processing 27.8 (2018): 3998-4011.

Problem Definition

- When usual metrics do not make sense: SSIM, PSNR, …
  - Use of application specific metrics, e.g.: character recognition a.k.a. Optical Caracter Recognition(OCR) [Lemarchand20]



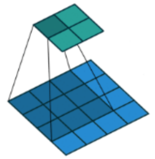
[Lemarchand20] F. Lemarchand, C. Marlin, F. Montreuil, E. Nogues, et M. Pelcat, « Electro-Magnetic Side-Channel Attack Through Learned Denoising and Classification », in ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, May 2020, p. 2882-2886, doi: 10.1109/ICASSP40776.2020.9053913.

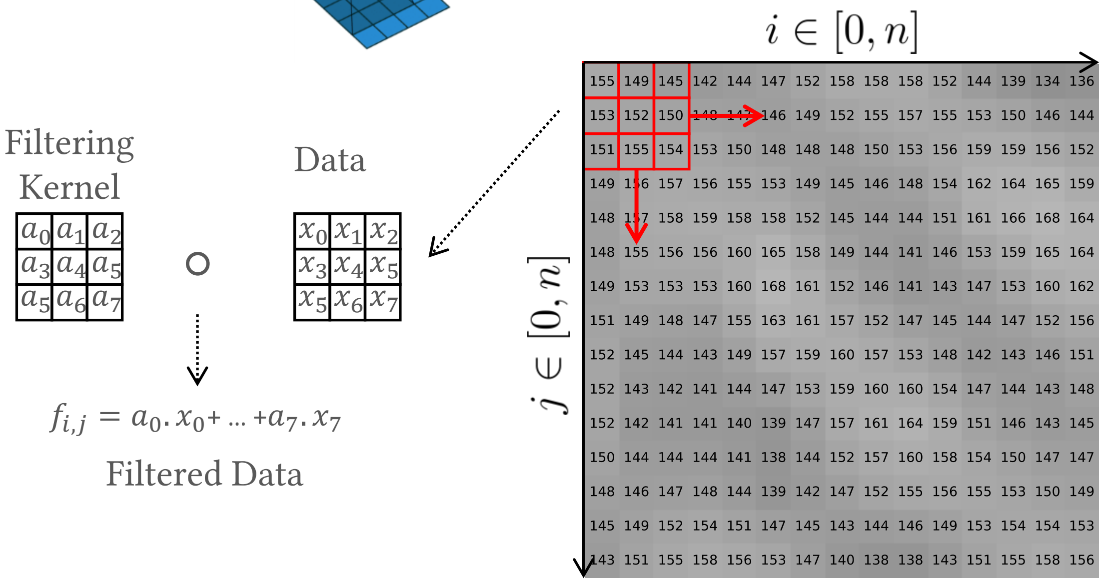
Plan

Expert-Based Denoising

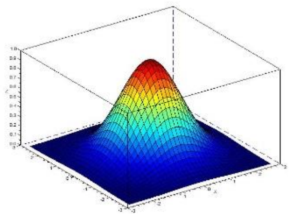
Filtering

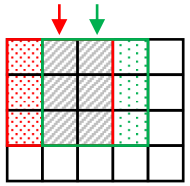
- Common kernels:
  - mean: $a_k = \frac{1}{kernel\_size}$
  - median / min / max: $out = operator(x_0, \ldots, x_k)$
  - Gaussian / approximate Gaussian
- Difficulties:
  - Padding: Add values around the image to enable kernel filtering
  - Computation optimizations:
    - Kernel Separability: Horizontal and Vertical slides computed separately
    - Previous results re-use
- Issue: Does not adapt to content



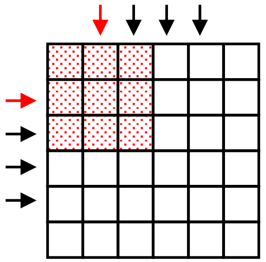


$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} = \frac{1}{3}\begin{bmatrix}1\\ 1\\ 1\end{bmatrix} * \frac{1}{3}\begin{bmatrix}1 & 1 & 1\end{bmatrix}$$

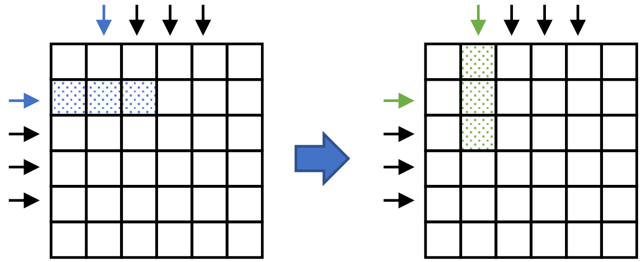




16 * 9 = 144 MACs



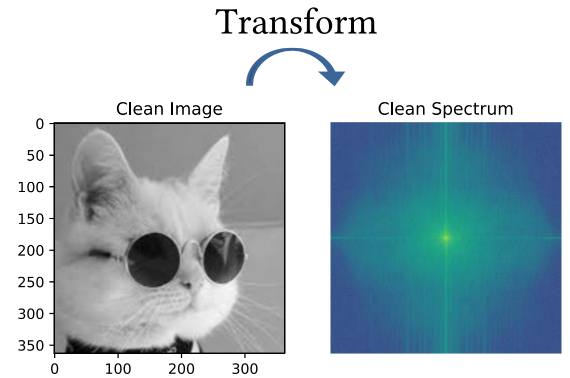
16 * 3 + 16 * 3 = 96 MACs



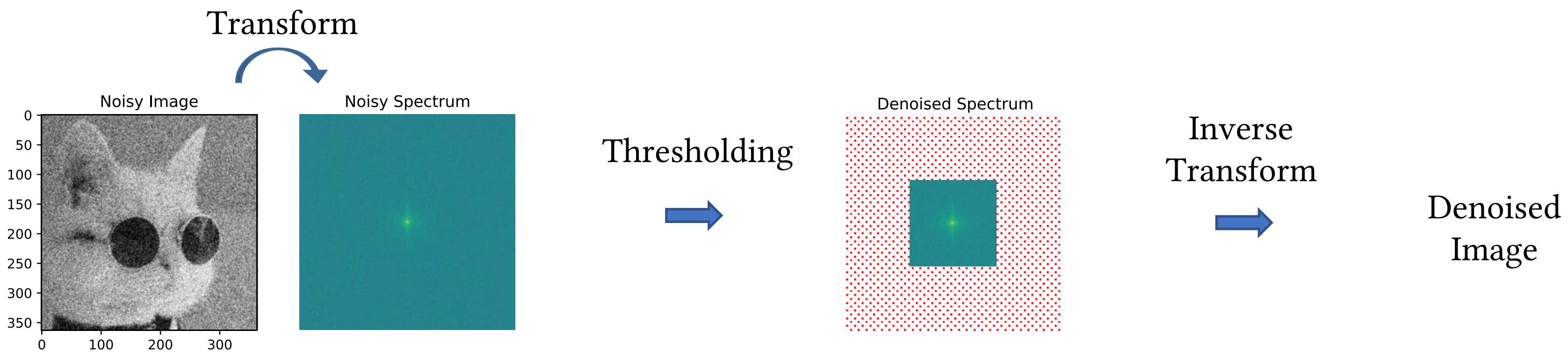
Expert-Based Denoising

Thresholding in the transform domain

- Transform the image in a sparse representation that concentrate the signal, small coefficients are considered as noise and threshold



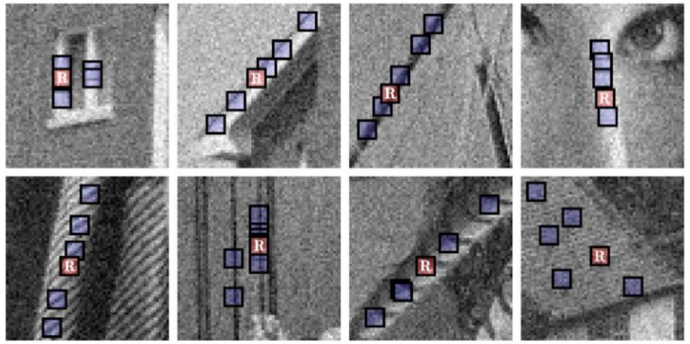
Transforms: FFT, DCT, Wavelets, …
Thresholds: Hard, Soft, Adaptive, Spatially Arbitrary, …



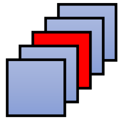
Expert-Based Denoising

BM3D: Block Matching 3D

Take a reference block and find “similar blocks”



Noisy Group

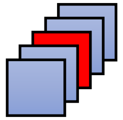


3D Transform

Thresholding

Inverse Transform

Denoised Group



Denoised Patch

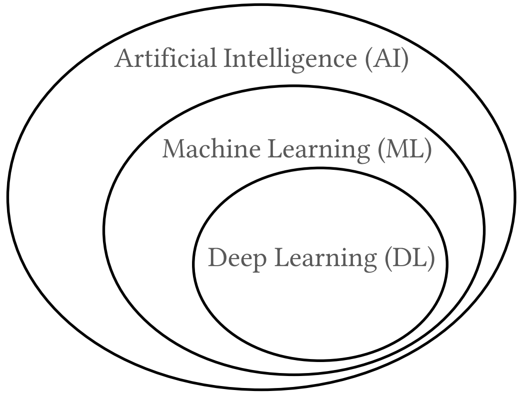
x2

Global Filtering to avoid Block Effect

[Dabov07] K. Dabov, A. Foi, V. Katkovnik, et K. Egiazarian, « Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering », IEEE Trans. on Image Process., vol. 16, n° 8, p. 2080-2095, août 2007, doi: 10.1109/TIP.2007.901238.

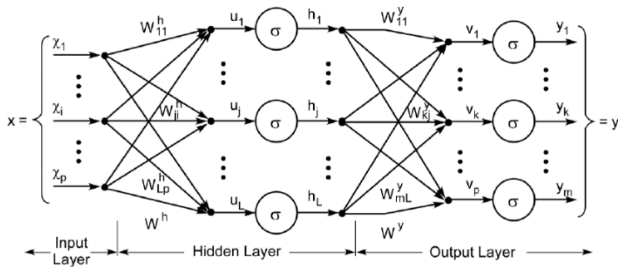
Plan

Deep Learning

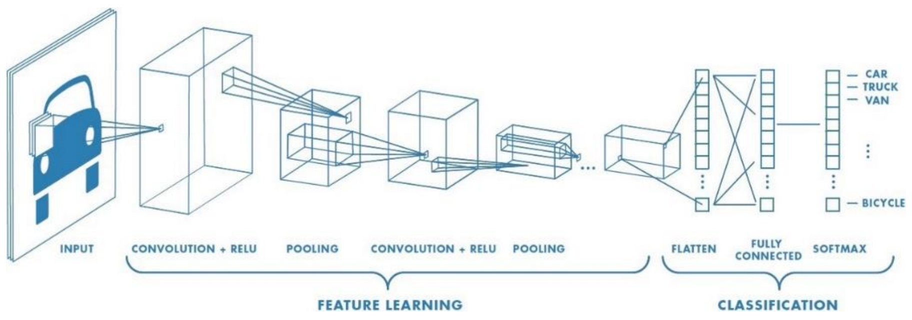


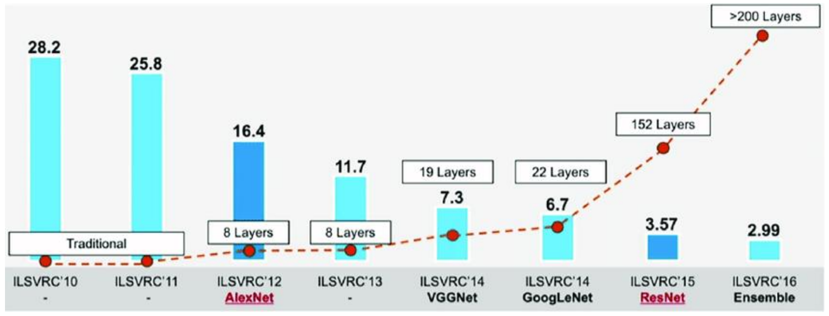
- AI → " *The effort to automate tasks normally performed by humans*"
- ML → The "program" defines itself the rules to solve a problem from data (examples)
- DL → ML that uses successive representations (layers), mostly abstract, to solve a problem
  - The number of representation layers is called depth

- Types of Deep Neural Networks:
  - Multi-Layer Perceptrons
    - All "neurons" are connected to each other and connections represented by learnable values (weights). The neuron itself is a non-linear activation function,



  - Convolutional Neural Networks [LeCun98]
    - The network is made of groups of filters (layers) convolved to the input or previous layer results resulting in feature maps,
    - The filters are learnable and outputs of layers are passed through activation function
    - First Success: Classification





[LeCun98] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et others, « Gradient-based learning applied to document recognition », Proceedings of the IEEE, vol. 86, n° 11, p. 2278–2324, 1998.

VAADER IETR research team

Learned Denoising

IETR

Feed-Forward Neural Network, Back-Propagation and Weights Optimisation

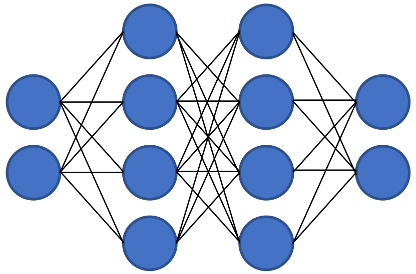
≠

Recurrent

$x, \hat{y}, y$: input, output and reference images
W: weights of the neural network
L: loss function
$Grad$: gradients of L according to W
lr: learning rate

W

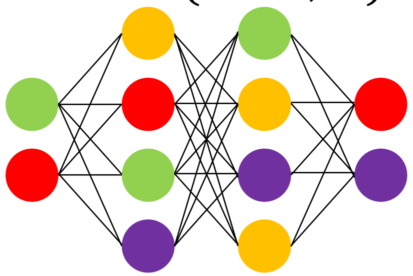
$x \;\longrightarrow$



$\longrightarrow \;\hat{y} = f(x, W)$

$\nearrow \qquad \searrow$

$W' = h(Grad, lr)$

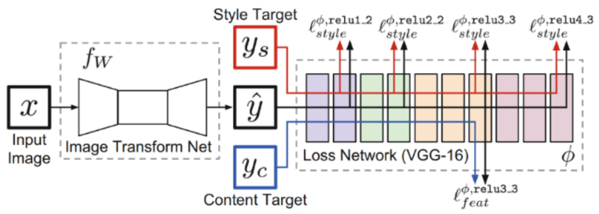


$\longleftarrow$

$Grad = \dfrac{\partial L}{\partial W} \;\longleftarrow\; L = g\,(\hat{y}, y)$

SGD[Sutskever13], RMSProp[Graves14], Adam[Kingma17], …

MSE, Sobel Loss, Perceptual Loss [Johnson16], …



[Kingma17] D. P. Kingma et J. Ba, « Adam: A Method for Stochastic Optimization », arXiv:1412.6980 [cs], janv. 2017
[Graves14] A. Graves, « Generating Sequences With Recurrent Neural Networks », arXiv:1308.0850 [cs], juin 2014
[Sutskever13] I. Sutskever, J. Martens, G. Dahl, et G. Hinton, « On the importance of initialization and momentum in deep learning », p. 14
[Johnson16] J. Johnson, A. Alahi, et L. Fei-Fei, « Perceptual Losses for Real-Time Style Transfer and Super-Resolution », in Computer Vision – ECCV 2016, vol. 9906, B. Leibe, J. Matas, N. Sebe, et M. Welling, Éd. Cham: Springer International Publishing, 2016, p. 694-7

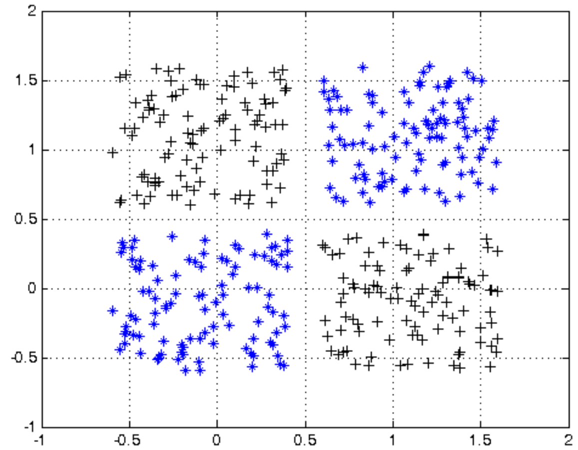
Convolutional Neural Networks

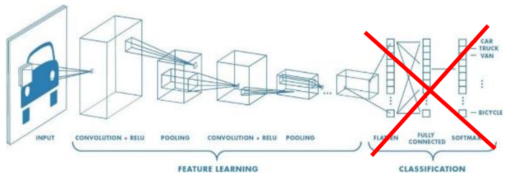
- Why convolutions?
  - Scientists used to filter with kernel!
  - Using filters requires less parameters than fully connecting layers

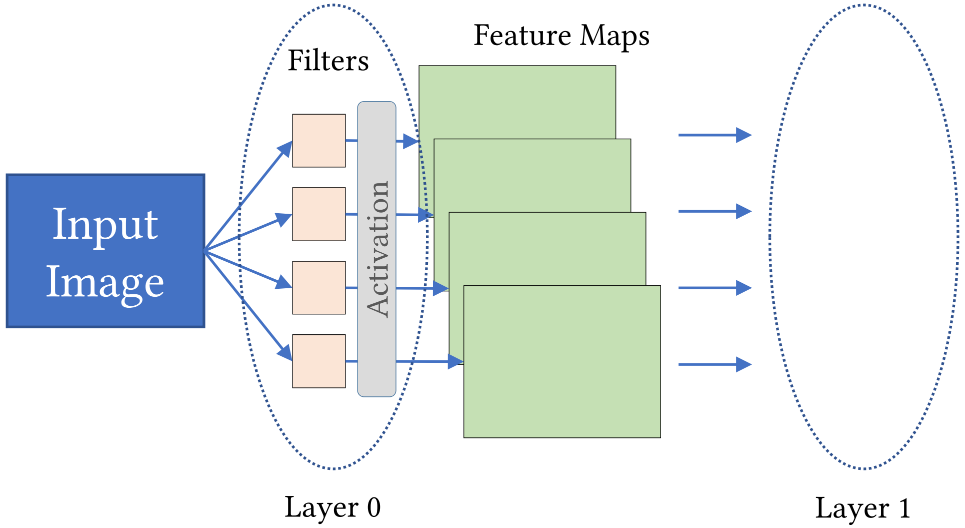
- Why activations?
  - An activation is a non-linear function. Non-Linearity is required for complex modelling
  - Without activations, all layers would collapse in one, being a linear combination of them,
  - It enables layers to be learned independently from others.

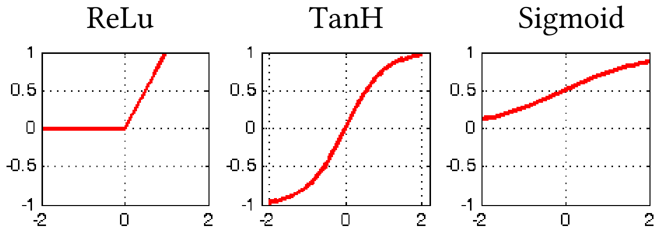
- For Denoising, three groups: GANs, Autoencoders, Others

- [Jain09] → First to use image to image network instead of image to class

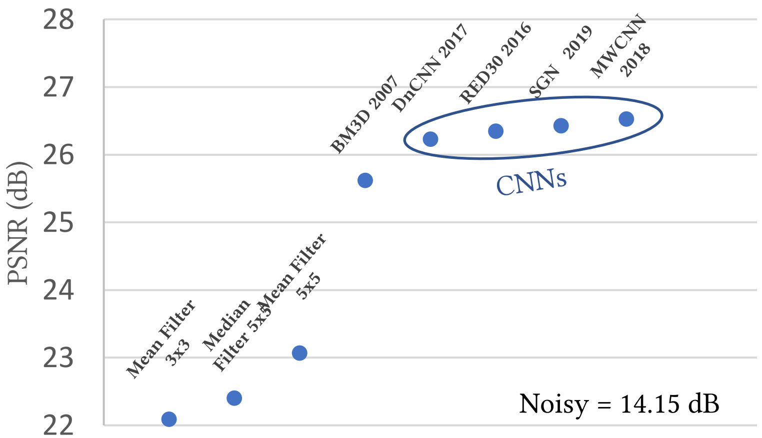








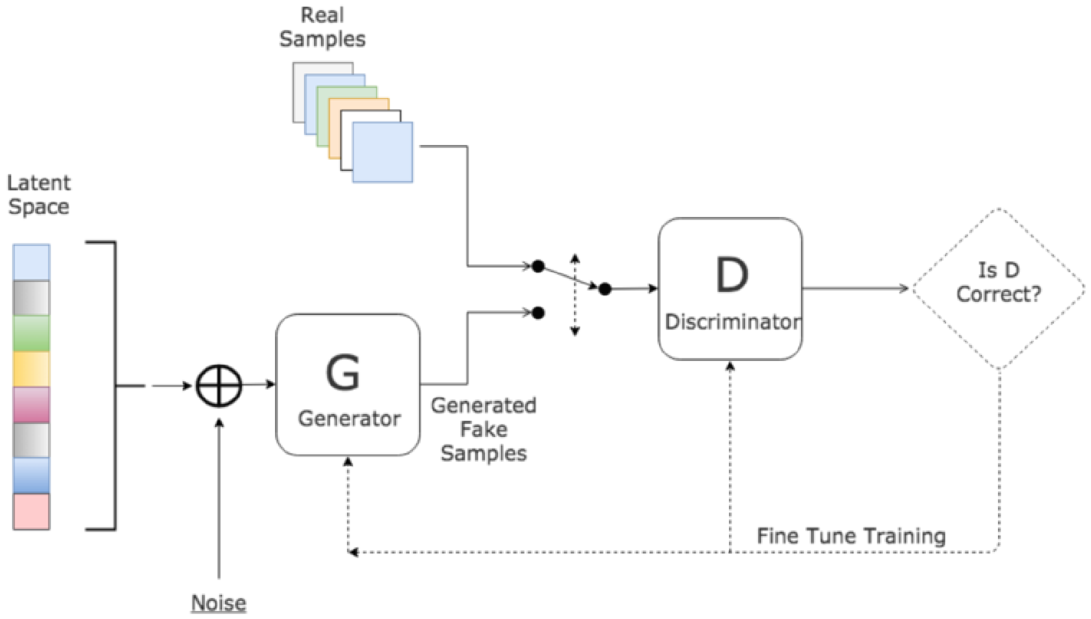
Evolution of Denoising Performances on BSD68 Grayscale AWGN 50

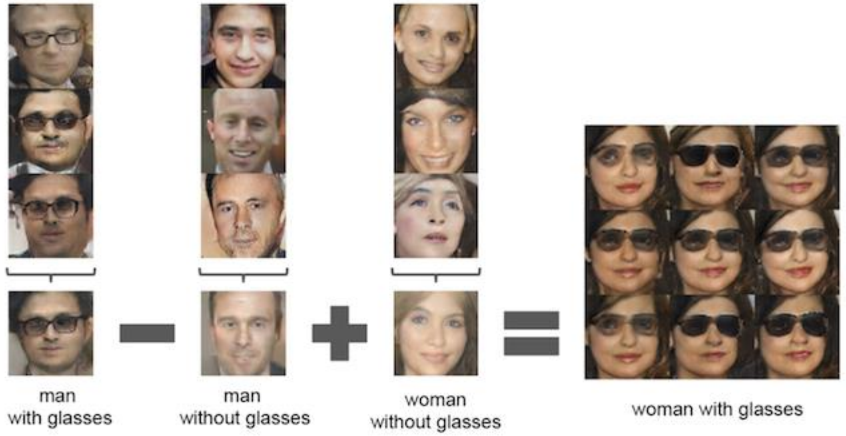


[Jain09] V. Jain et S. Seung, « Natural image denoising with convolutional networks », in Advances in neural information processing systems, 2009, p. 769–776.

CNNs for Denoising: GANs

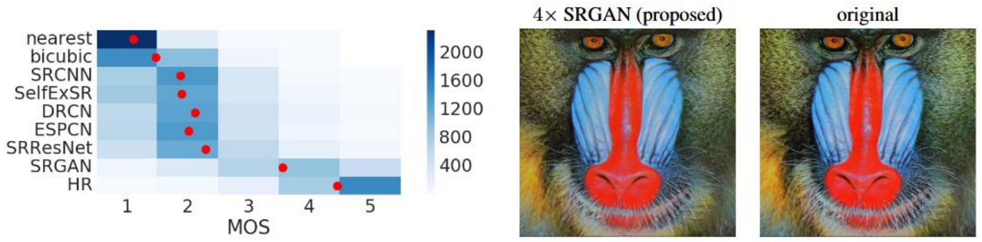
- Generative Adversarial Networks [Goodfellow14]
  - Principle
    - Two networks: a Generator G and a Discriminator D
    - G tries to generate an image close enough to real samples
    - D tries to determine if G samples are real of fake
    - G and D trained to fool each other
  - Interest?
    - Generate new samples from a distribution
    - Input an image instead of a noise vector to make G denoise





[Radford15]

[Karras17]



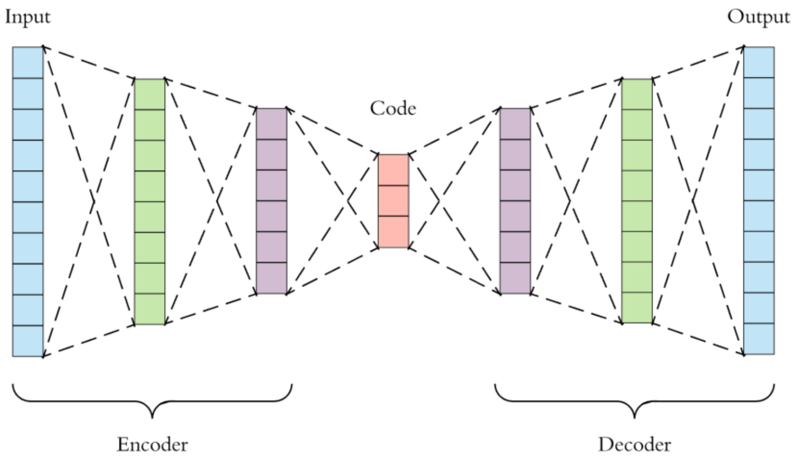
[Ledig17]

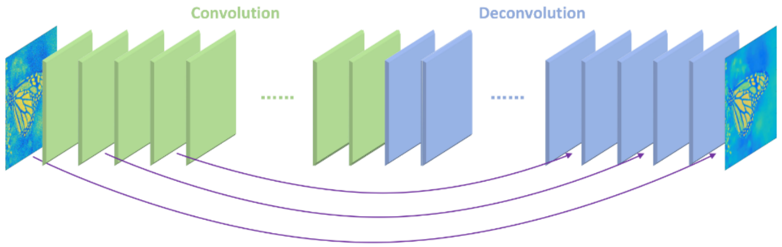
More Applications!

[Goodfellow14] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.
[Radford15] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).
[Karras17] T. Karras, T. Aila, S. Laine, et J. Lehtinen, « Progressive growing of gans for improved quality, stability, and variation », arXiv preprint arXiv:1710.10196, 2017
[Ledig17] C. Ledig et al., « Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network », in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, juill. 2017, p. 105-114, doi: 10.1109/CVPR.2017.19.

CNNs for Denoising: Auto-encoders

- Autoencoders [Vincent10]
  - Principle
    - Bottleneck network that learns dimension reduction without supervision
    - Input is corrupted (noise, sparsity), the network learns to reconstruct original input ignoring the noise
    - Resulting encoding keeping the most important information for reconstruction
  - Interest
    - Input a noisy image and learn to reconstruct its clean version (supervised)



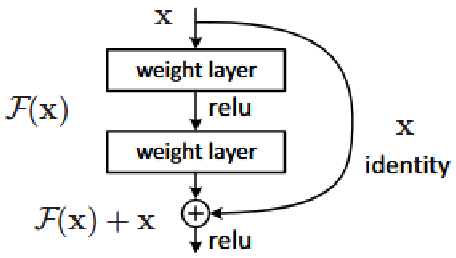


[Mao16]

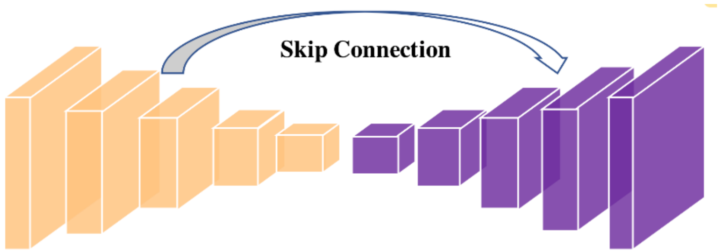
[Vincent10] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, et P. A. Manzagol, « Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion », Journal of Machine Learning Research, vol. 11, p. 3371--3408, 2010.
[Mao16] X. Mao, C. Shen, et Y.-B. Yang, « Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections », Advances in Neural Information Processing Systems 29 (NIPS 2016), p. 9, 2016.

CNNs for Denoising: Others

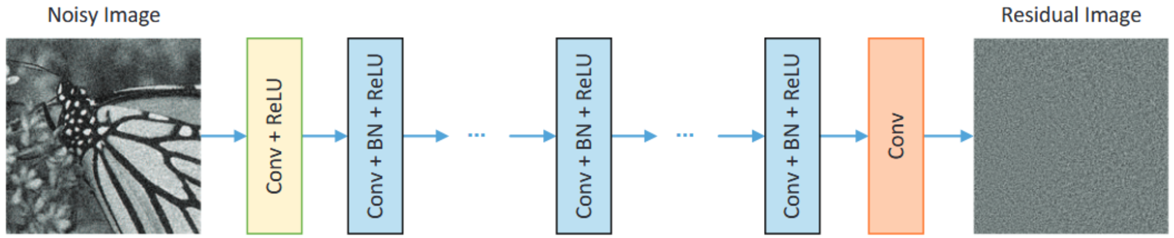
- Residual Learning [He16]
  - Learn to predict the residual signal instead of the signal itself
  - Gives a reference of what is to be reconstructed
  - Enables learning deeper networks
  - RedNet [Mao16] is an autoencoder with skip-connections between layers of same size
  - DnCNN [Zhang17] uses a global residual
    - It learns the noise instead of the denoised signal



[He16]



[Mao16]

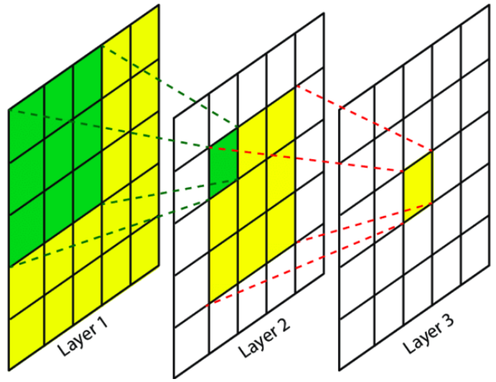


[Zhang17]

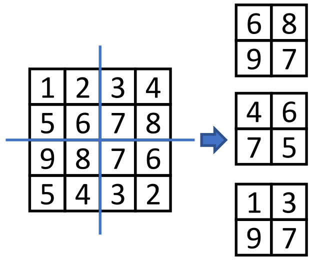
[He16] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[Zhang17] Zhang, Kai, et al. "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising." IEEE Transactions on Image Processing 26.7 (2017): 3142-3155.
[Mao16] X. Mao, C. Shen, et Y.-B. Yang, « Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections », Advances in Neural Information Processing Systems 29 (NIPS 2016), p. 9, 2016.

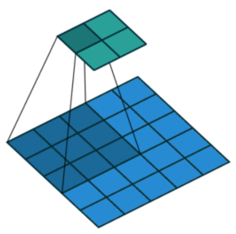
CNNs for Denoising: Others

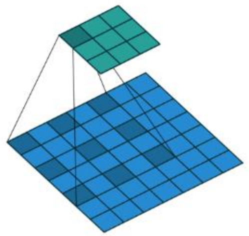
- Multi-Scale Learning
  - Use feature maps at different scales into the network
  - Different justifications:
    - Reduce the computations in the branches of lower scales
    - Enables the network to use information at different resolution
      - An homogeneous block is learned easily at low scale
      - An high frequency block is learned better at high resolution
    - Enlarge the receptive field
  - Types of up/down-samplings:
    - Down: Pooling, Strided Convolution, Dilated Convolution, Pixel Shuffle [Shi16]
    - Up: Bicubic, Nearest Neighbor, Transpose-Convolution, Pixel Unshuffle

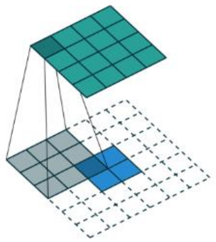


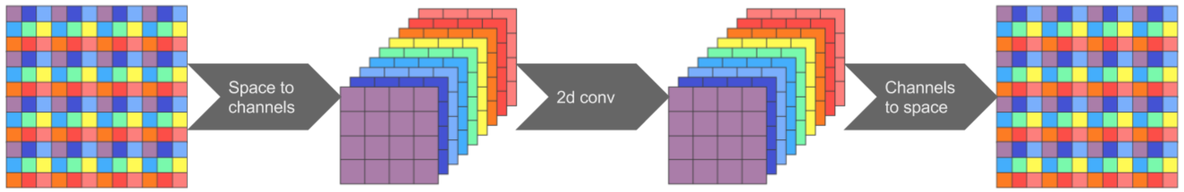
Receptive Field

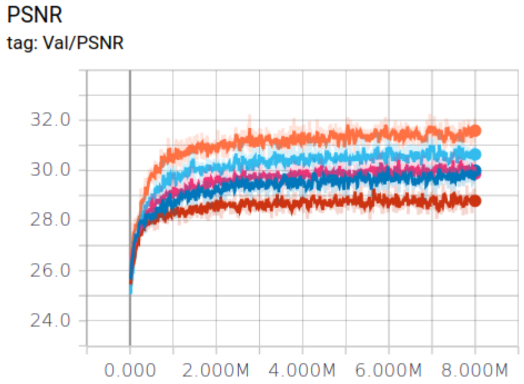



Strided Conv


Dilated Conv


Transpose Conv



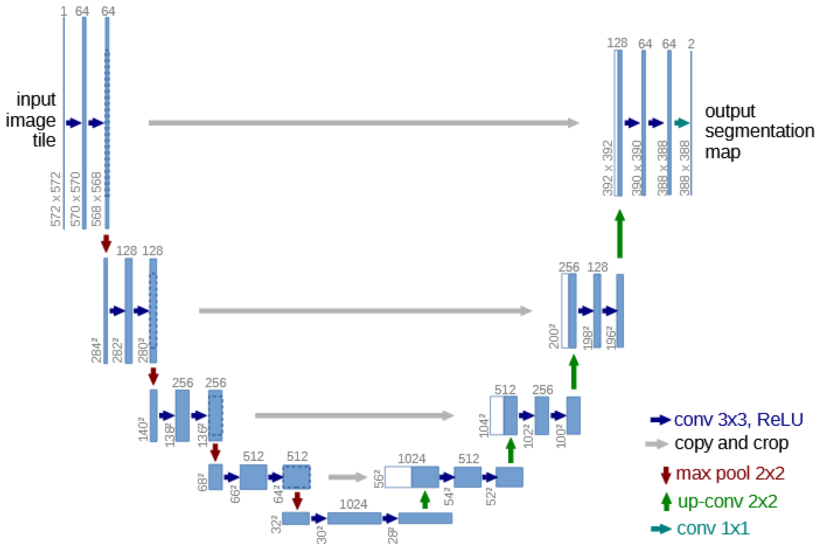


Name	Smoothed	Value
AvgPooling	30.00	30.23
Bicubic	29.87	29.68
MaxPooling	28.78	28.72
NearestNeighbor	30.64	30.83
PixelShuffle	31.59	31.37

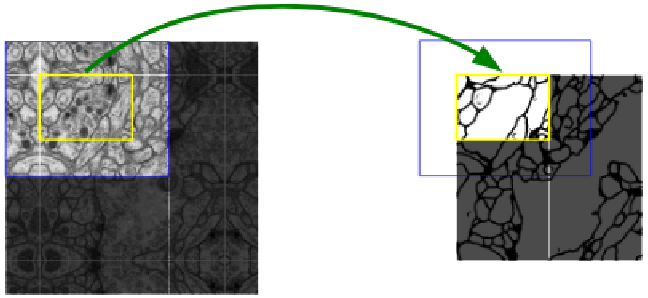
[Shi16] Shi, Wenzhe, et al. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

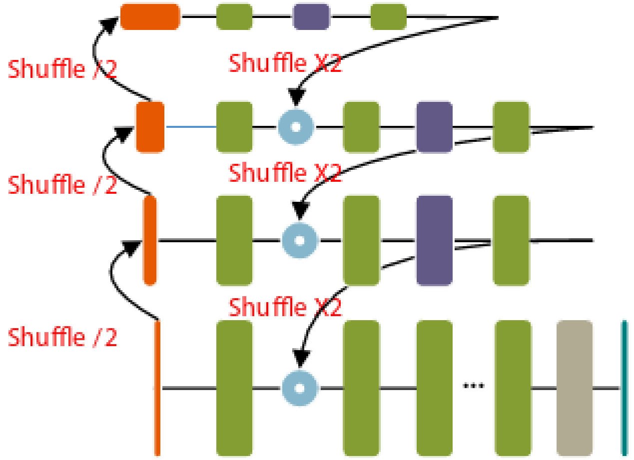
CNNs for Denoising: Others

- Multi-Scale Learning
  - U-Net [Ronneberger15]
    - First to use U formed network
  - Self-Guided Network (SGN) [Gu19]
    - Self-guidance of features by lower-level (scale) features
    - Faster to train, better convergence, lighter network
      - 4x times smaller/faster than RedNet [Mao16]



[Ronneberger15]

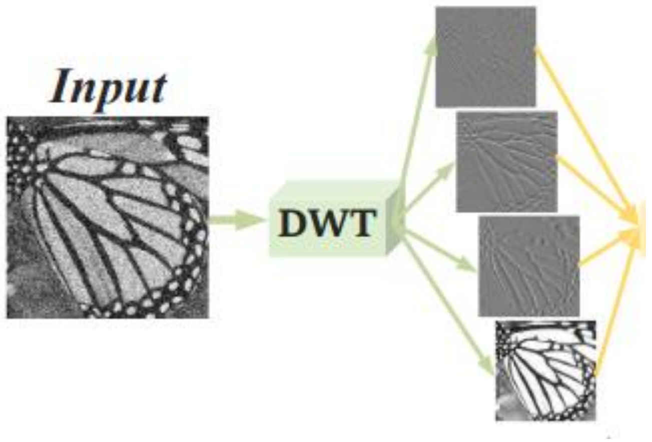




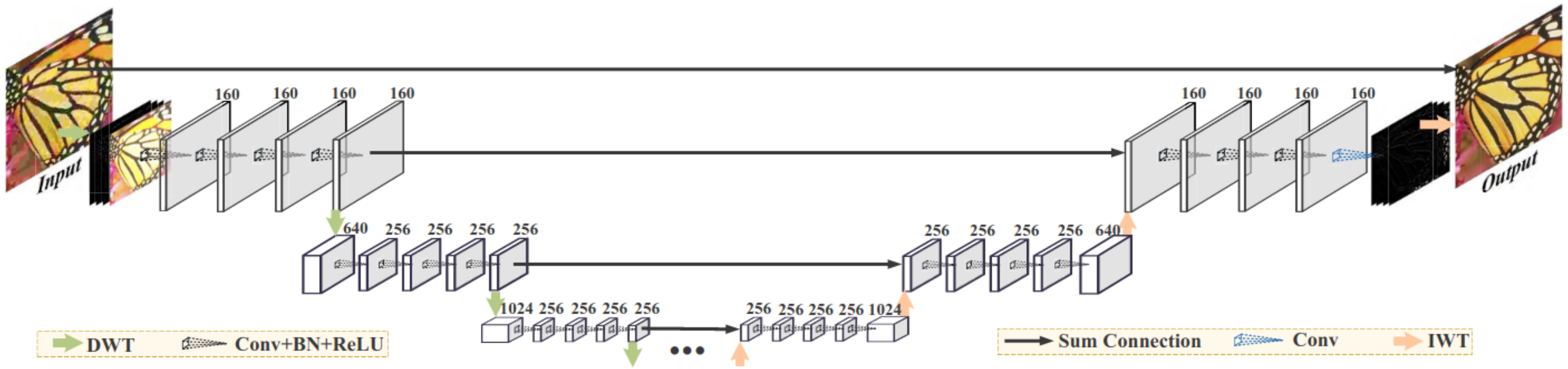
[Gu19]

[Ronneberger15] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.
[Gu19] Gu, Shuhang, et al. "Self-guided network for fast image denoising." Proceedings of the IEEE International Conference on Computer Vision. 2019.

- Multi-Scale Learning
  - Multi-level Wavelet CNN (MWCNN) [Liu18]
    - Use Wavelet decomposition as down/up sampling operator
      - No Information loss
    - Introduction of expert based knowledge into the network



$$\mathbf{f}_{LL} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{f}_{LH} = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}, \mathbf{f}_{HL} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, \mathbf{f}_{HH} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

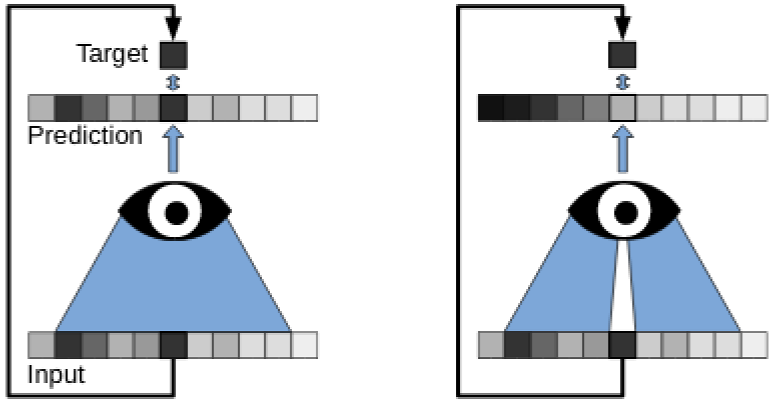


[Liu18] Liu, Pengju, et al. "Multi-level wavelet-CNN for image restoration." Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2018.

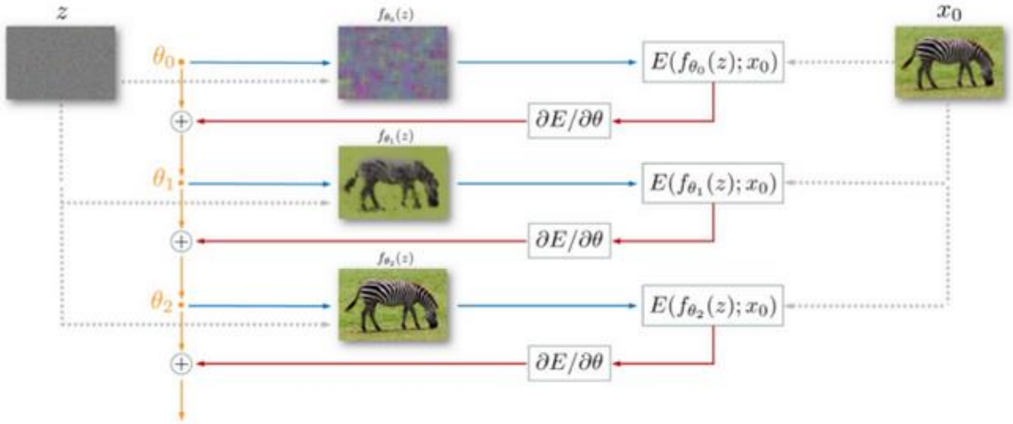
Towards Less Supervision

- GANs

- Noise2Void (N2V) [Krull19]
  - Self-Supervised Learning → Learned to reconstruct an image using only itself with some pixels removed
  - Assumption of pixel-independent noise



- Deep Image Prior [Ulyanov18]
  - Counter intuitive strategy!
  - Learns a randomly initialized neural network Θ that maps a vector z to the noisy image.
  - The network "resists" to learn the target itself because of its inner prior on natural image, coming from its handcrafted architecture.
  - Eventually, once an optimal point reached, forward z and obtained the denoised image!



[Krull19] Krull, Alexander, Tim-Oliver Buchholz, and Florian Jug. "Noise2void-learning denoising from single noisy images." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019.
[Ulyanov18] Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky. "Deep image prior." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

Learning Frameworks

- Theano first DL framework (FW) 2007, no longer maintained since 2017
- Caffe (2013), Berkley Artificial Intelligence Research (BAIR), Caffe2 (2017), Facebook
- Tensorflow (2015), Google → First to be massively used, lot of open-source code
- Keras: Interface over Tensorflow (2015), Francois Cholet , now Google
- Pytorch: Native Python interface with Torch backend (2017), Facebook → Used in Practical Work
- MatConvNet (Matlab), CNTK (Microsoft), ….
- N2D2: Only French FW? CEA List, industrials and academic partners (2017)
- ONNX common interface between FWs, Facebook and Microsoft
    - Enables alternating between FWs
- Perceptilabs: graphs to Tensorflow via a GUI

PyTorch

TensorFlow

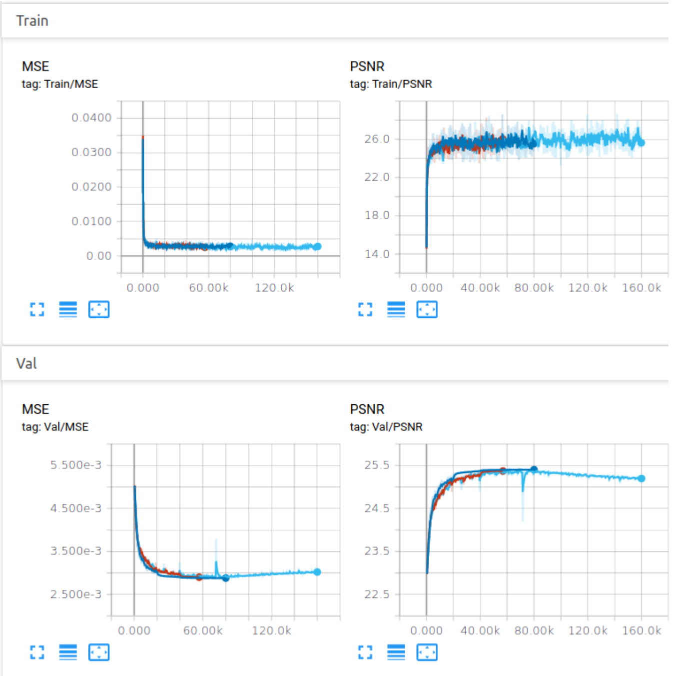
N2D2

Keras

ONNX

Prototyping Process

- Prepare the dataset
  - Select the data according to the problem to solve
  - Data Augmentation: rotation, flips, noising→ Bring diversity/ Make the learning more robust
- Design the network architecture
  - Still empirical for now, Some attempt to automate: reinforcement learning driven denoising toolbox [Yu18], genetic algo for architecture [Suganuma18]
- Choose the optimization scheme
  - Optimizer: Type of gradient-based optimization strategy , LR Decay (Step, Exponential, Adaptive, ...)
  - Loss type, Number of Iteration, Evaluation Strategy
- Train
  - Optimal on Graphics Processing Units (GPUs) [for now]...
  - Monitoring → Tensorboard
- Post-training Optimization:
  - Weight quantization/pruning ([TensorRT], self-ensemble inference)
- Test and integration



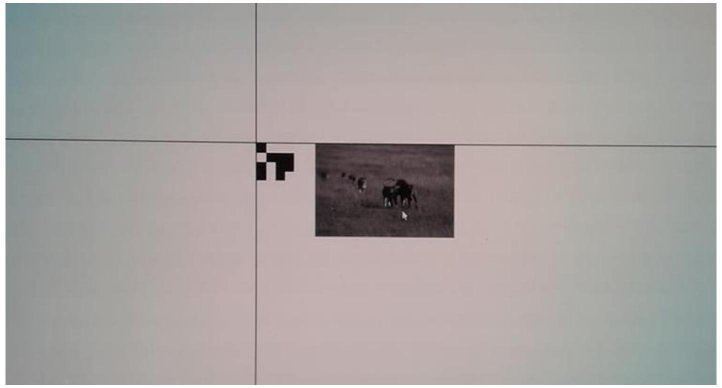
[Yu18] K. Yu, C. Dong, L. Lin, et C. C. Loy, « Crafting a Toolchain for Image Restoration by Deep Reinforcement Learning », in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, juin 2018, p. 2443-2452, doi: 10.1109/CVPR.2018.00259.[Suganuma18] M. Suganuma, M. Ozay, et 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, Stockholmsmässan, Stockholm Sweden, juill. 2018, vol. 80, p. 4771–4780.

Plan

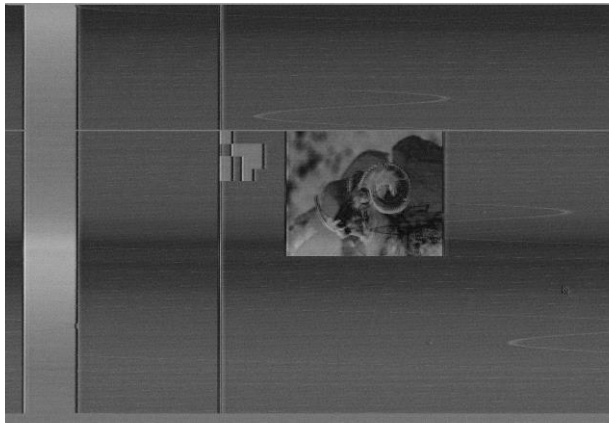
Why is it complicated?

Dataset Construction

Reference display on screen
with sight and QR code



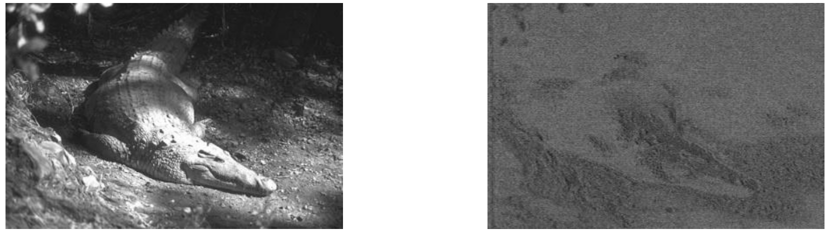
Intercepted image:
The size is different and the position unknown

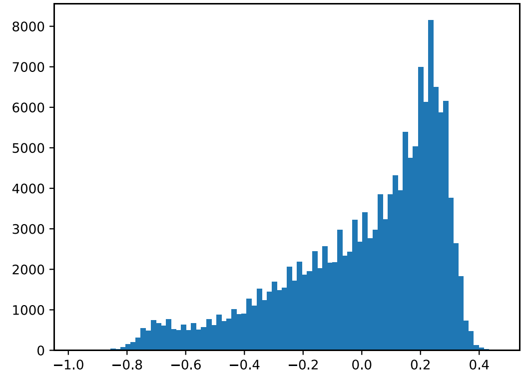


Noise Distribution

Reference

Intercepted

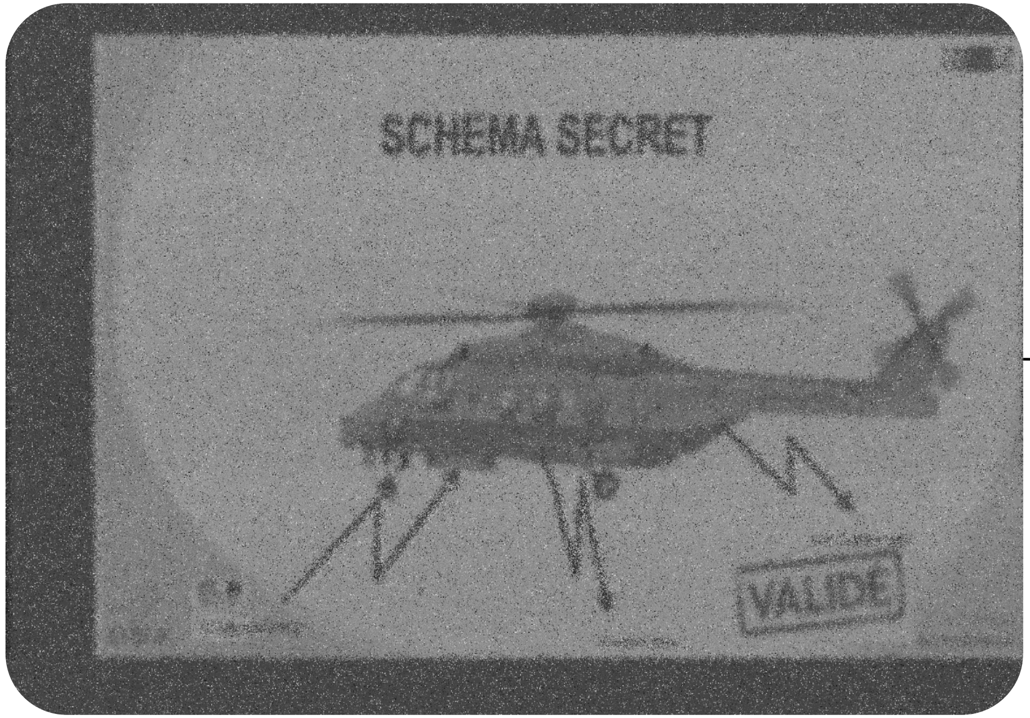




Hist (intercept-ref)

Eavesdropped Image Denoising

Interception Noise



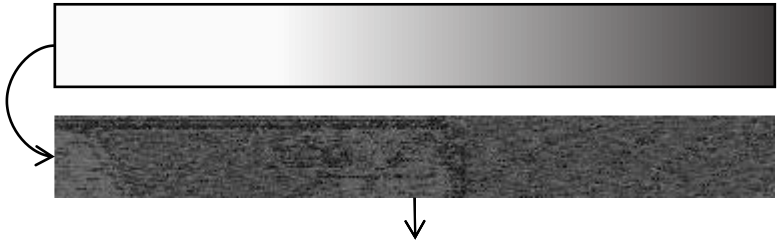
Example of an eavesdropped image with "good" interception conditions

Known Noises:

- **Gaussian**
- **Speckle**
- **Bernoulli**
- …

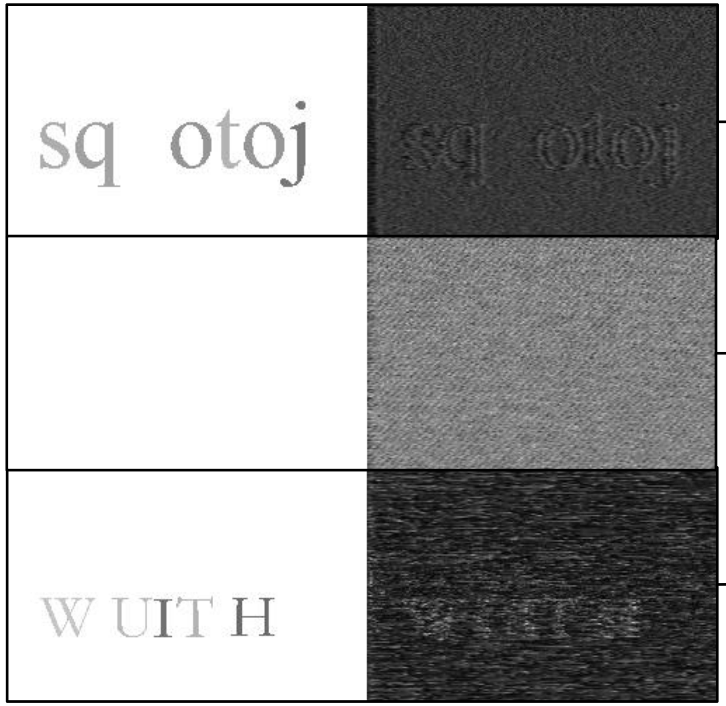
Displayed

Intercepted



Gradient Destruction

Displayed | Intercepted



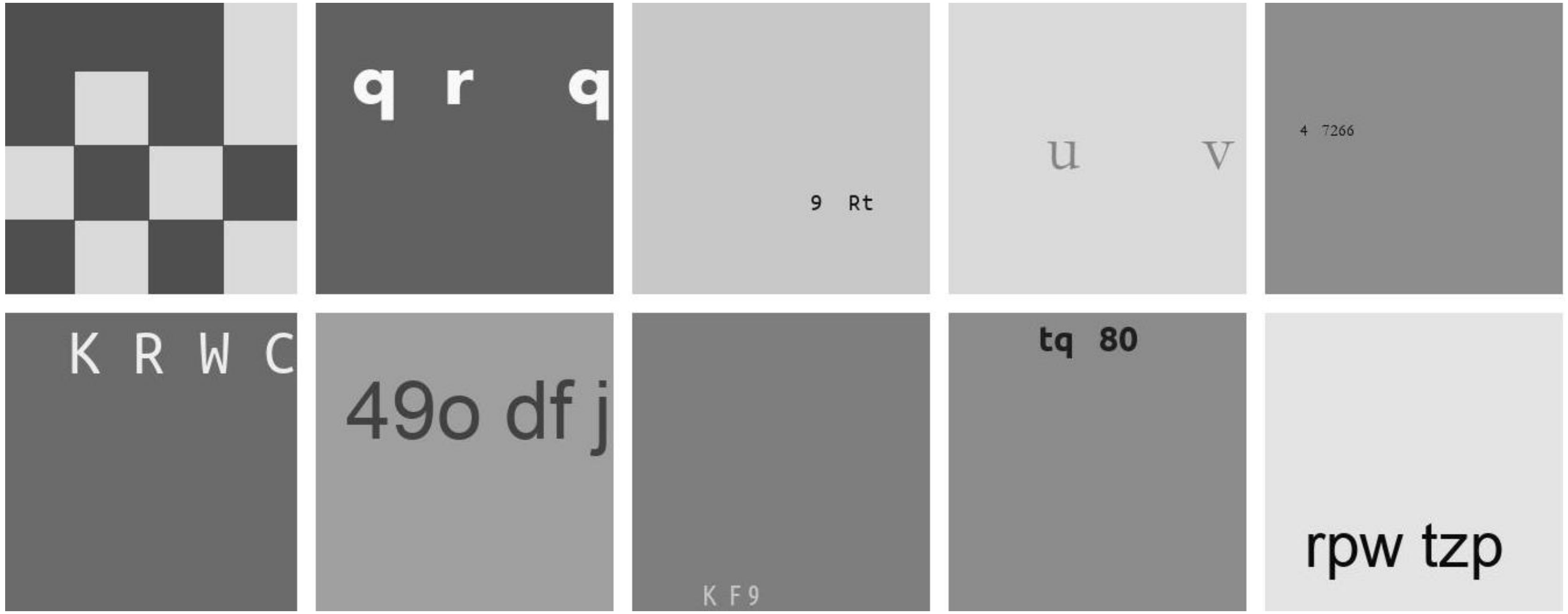
Echo

Pattern Noise

Spatially Correlated Noise

Specific Dataset Building

Automated Creation:
- Caracters
- Size
- Font
- Positionning
- Labels saving
- QR Code Writing



[Lemarchand20] F. Lemarchand, C. Marlin, F. Montreuil, E. Nogues, et M. 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), Barcelona, Spain, May 2020, p. 2882-2886, doi: 10.1109/ICASSP40776.2020.9053913.

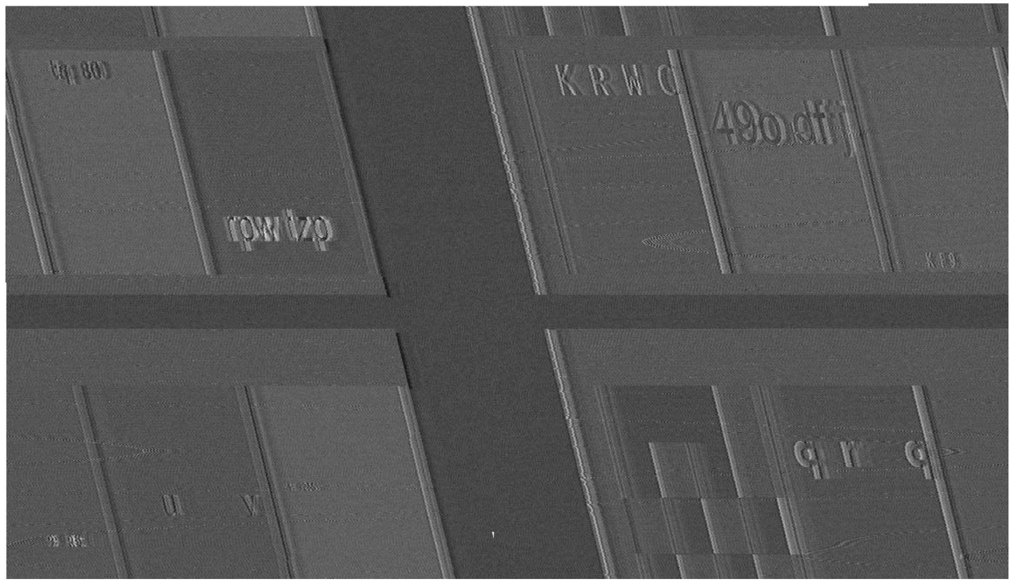
Specific Dataset Building

Automated Creation:
- Caracters
- Size
- Font
- Positionning
- Labels saving
- QR Code Writing

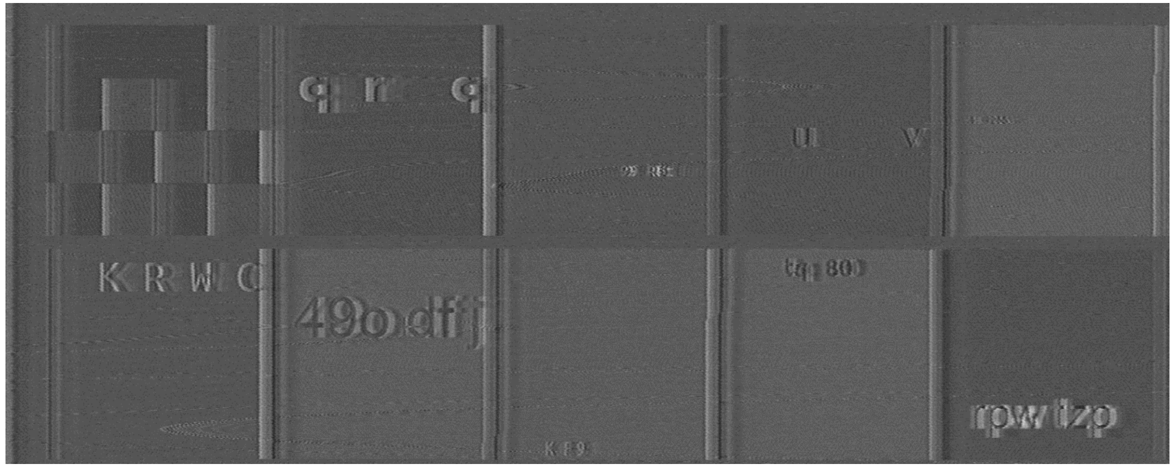
→ **Interception** →

Interception and Spatial Rearrangement:
- Corner Detect.
- Shearing Correct.
- Global image Reorganizing

Intercepted



Rearranged



Specific Dataset Building

Automated Creation:
- Caracters
- Size
- Font
- Positionning
- Labels saving
- QR Code Writing

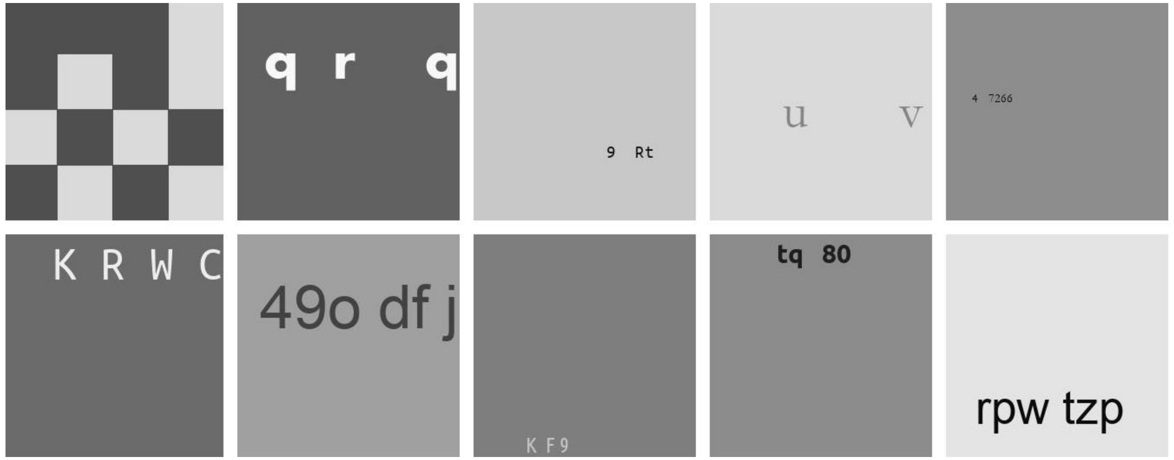
→ **Interception** →

Interception and Spatial Rearrangement:
- Corner Detect.
- Shearing Correct.
- Global image Reorganizing

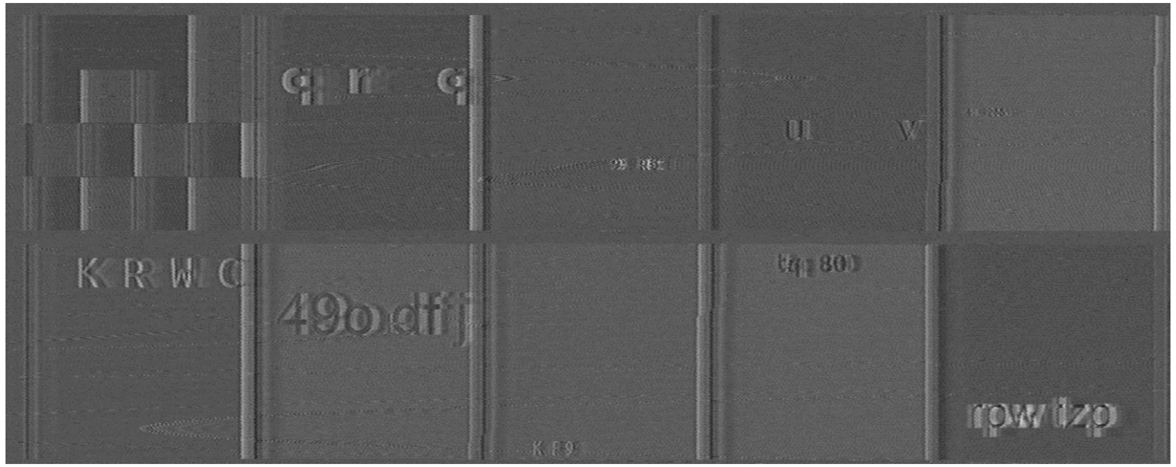
→

Time Alignement With Reference:
- Timestamp
- QR Code Valid

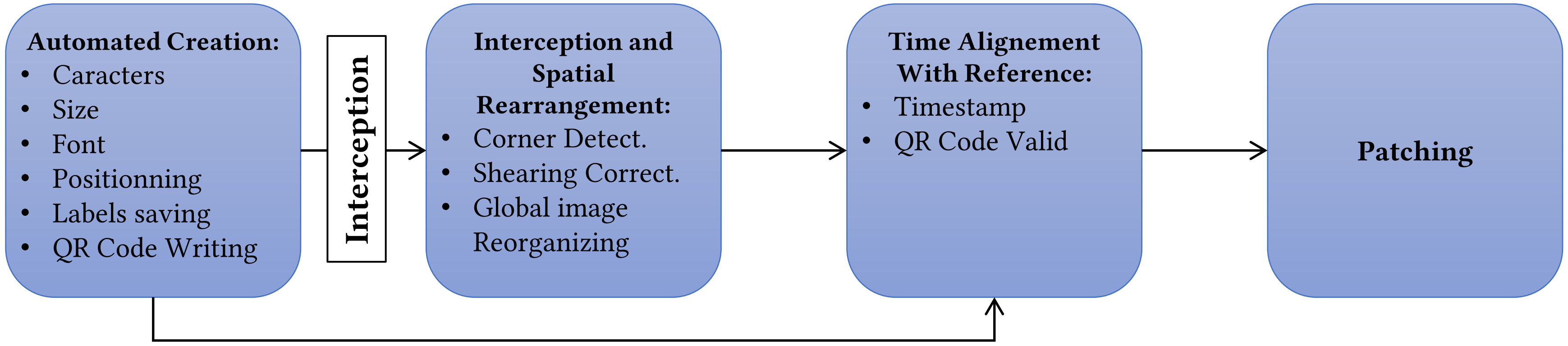
Reference

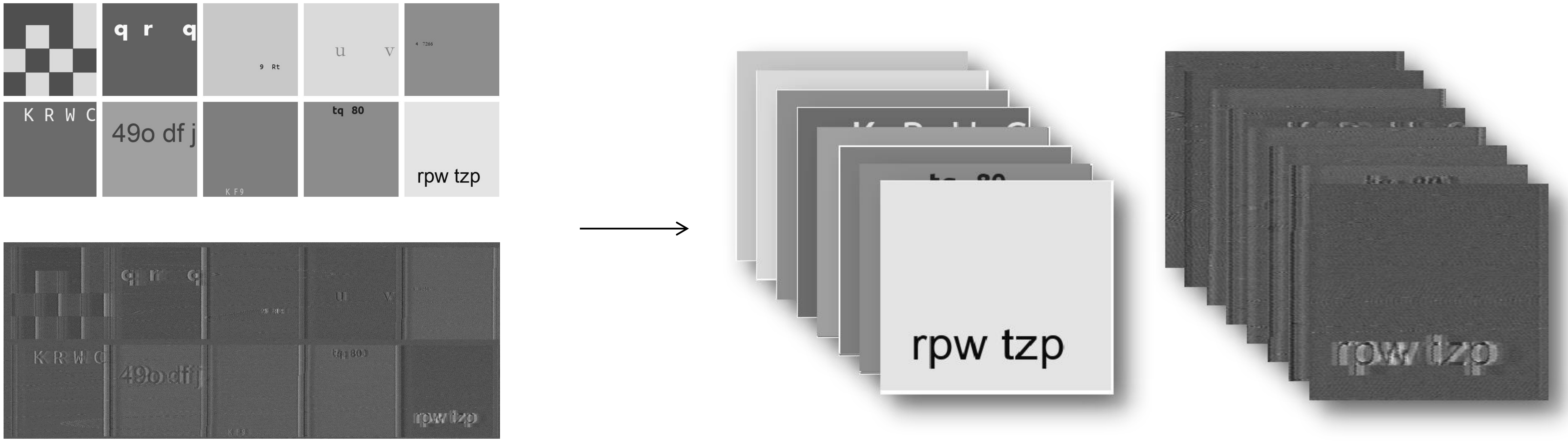


Intercepted



Specific Dataset Building





Specific Dataset Building

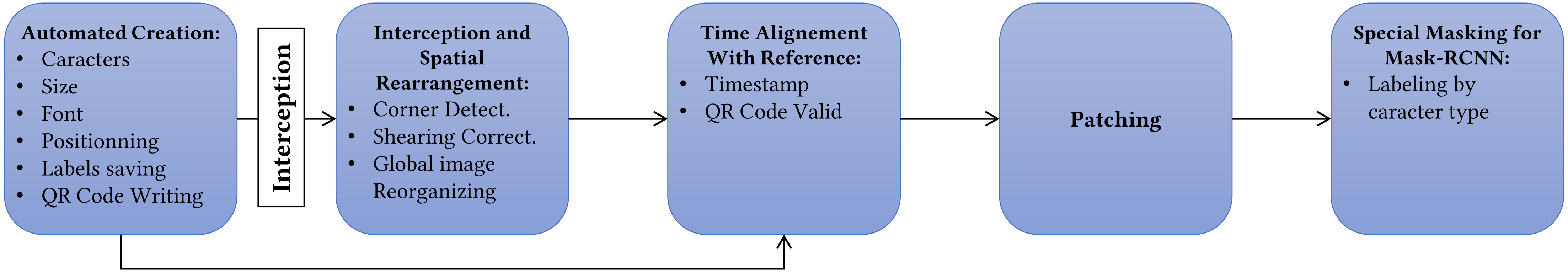
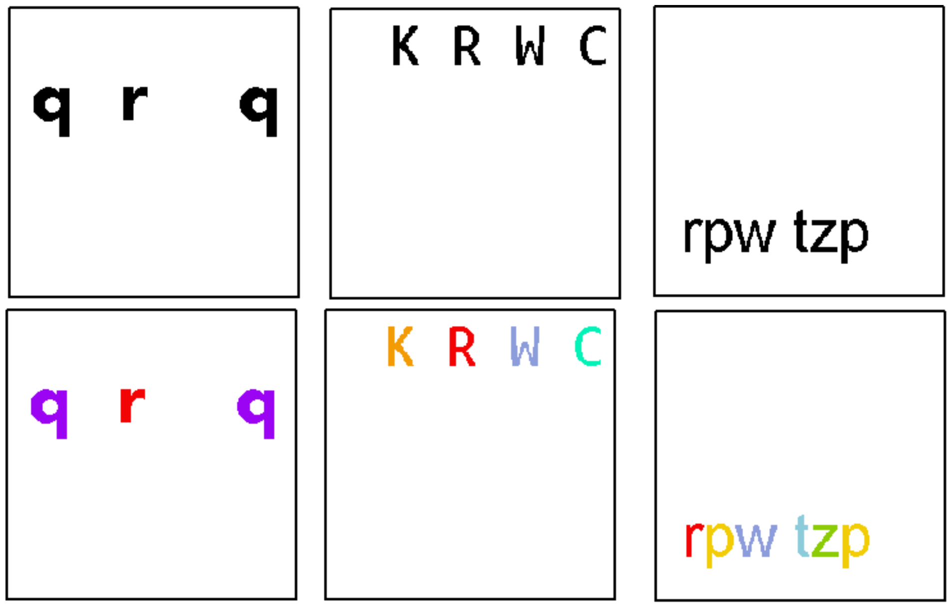


Image Label

Caracter Label



Specific Dataset Building

Automated Creation:
- Caracters
- Size
- Font
- Positionning
- Labels saving
- QR Code Writing

Interception

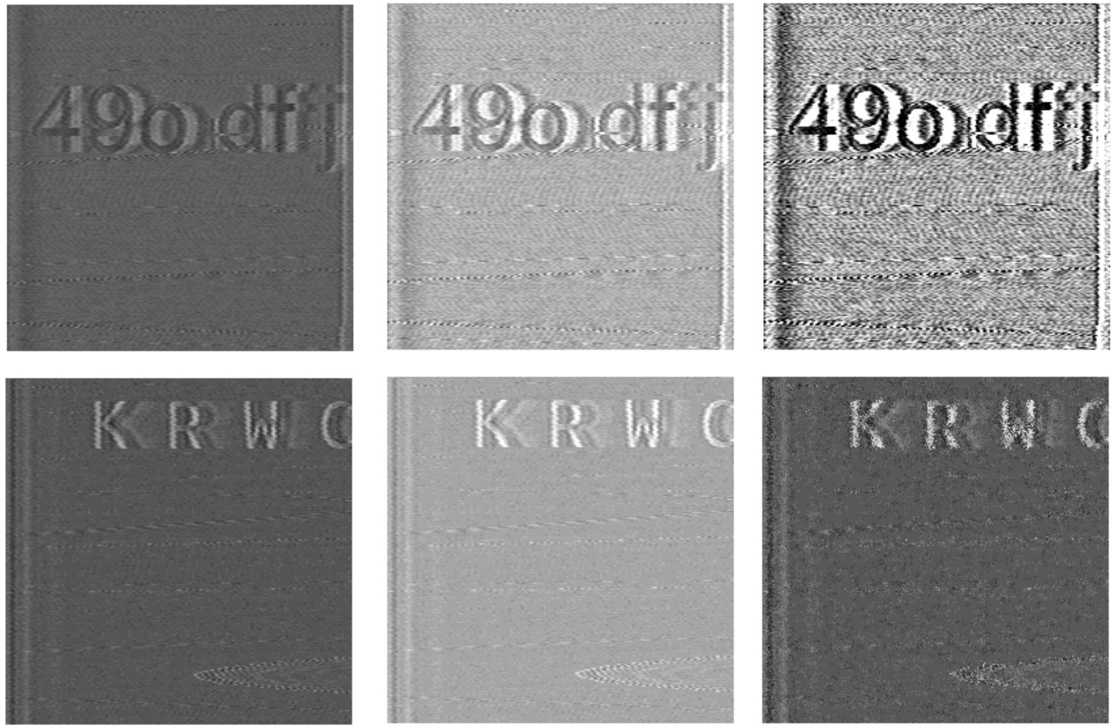
Interception and Spatial Rearrangement:
- Corner Detect.
- Shearing Correct.
- Global image Reorganizing

Time Alignement With Reference:
- Timestamp
- QR Code Valid

Patching

Special Masking for Mask-RCNN:
- Labeling by caracter type

Data-Augmentation



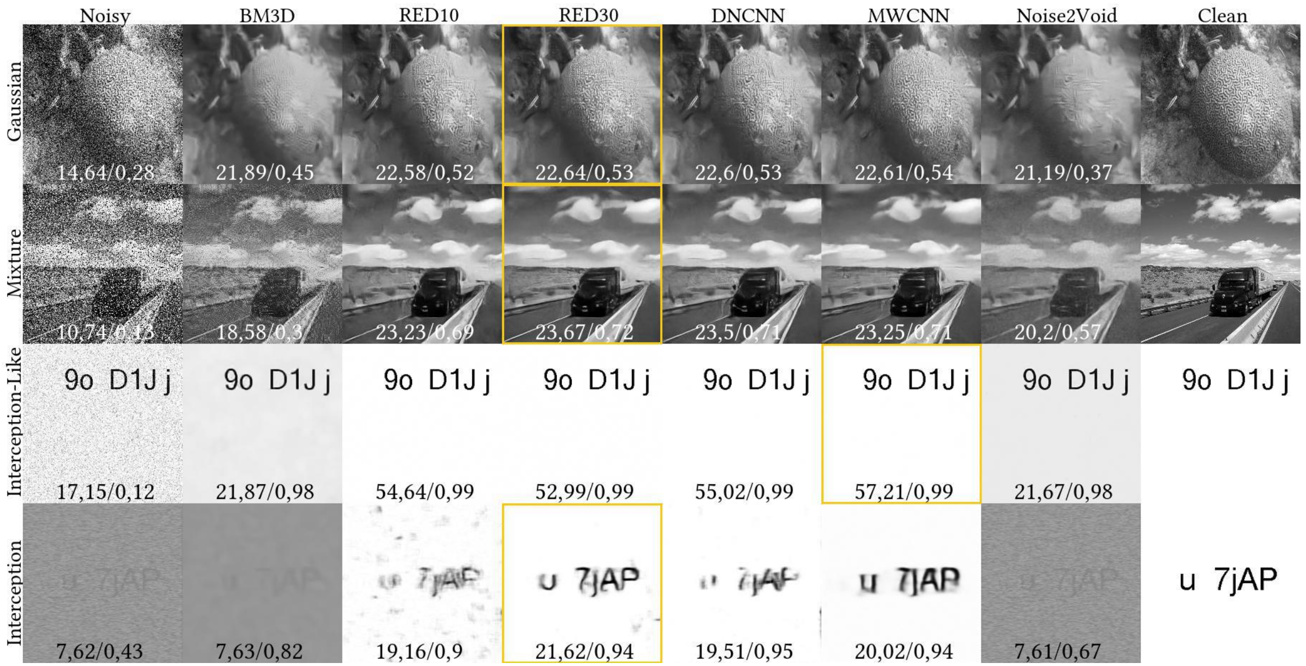
Specific Dataset Building

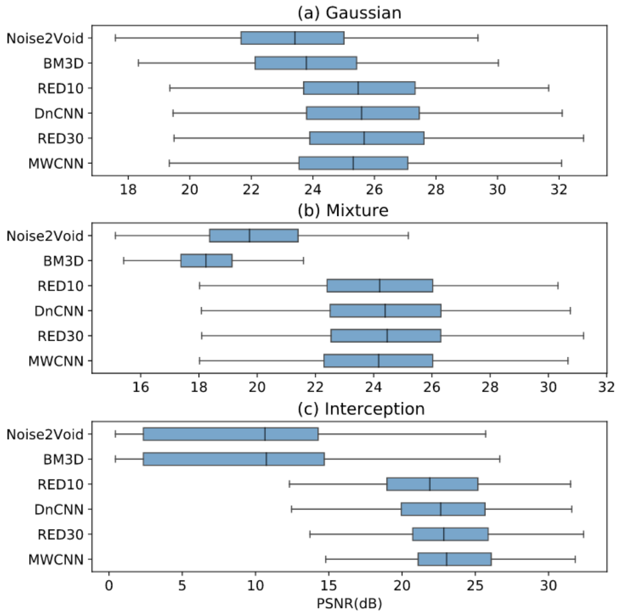
Obtained Corpus

- Available: https://github.com/opendenoising/interception_dataset
- Size :
  - Samples size: 256x256x1
  - Database size: 98.725 training samples/ 12.563 test and validation samples
- Acquiring parameters:
  - Connectors: DVI, VGA, DP, HDMI
  - 3 antennas
  - Different distances
  - 3 screens with different resolutions
  - Zoom 100% to maintain font scales

Eavesdropped Image Denoising

Interception Noise and Existing Algorithms



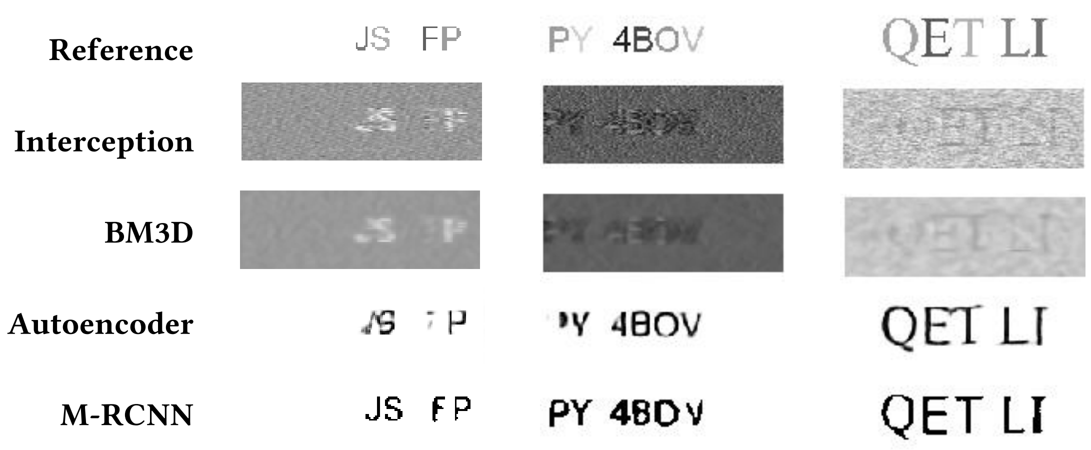


[Lemarchand20] Lemarchand, Florian, et al. "OpenDenoising: an Extensible Benchmark for Building Comparative Studies of Image Denoisers." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.

- Compared Methods:

  Denoising: BM3D [3], Autoencoder [4], Noise2Noise [5], DnCNN [6], Mask-RCNN [7]

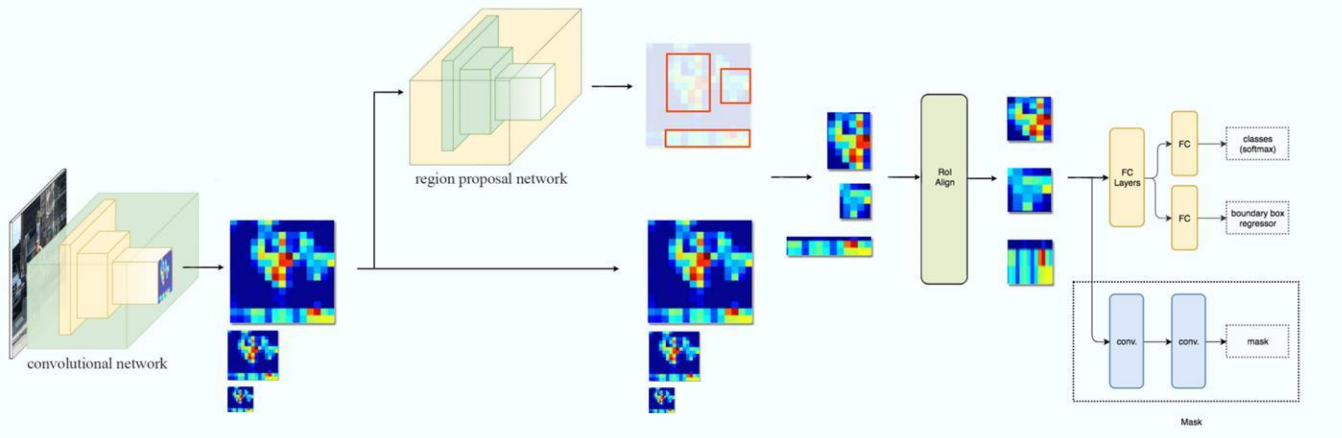
  + OCR: Tesseract [8]

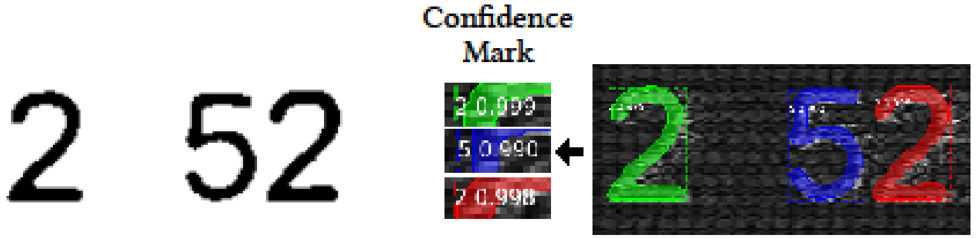


- Our Proposal:

  Join Denoising and Classification

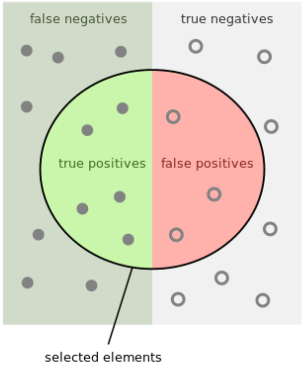
  → Mask-RCNN



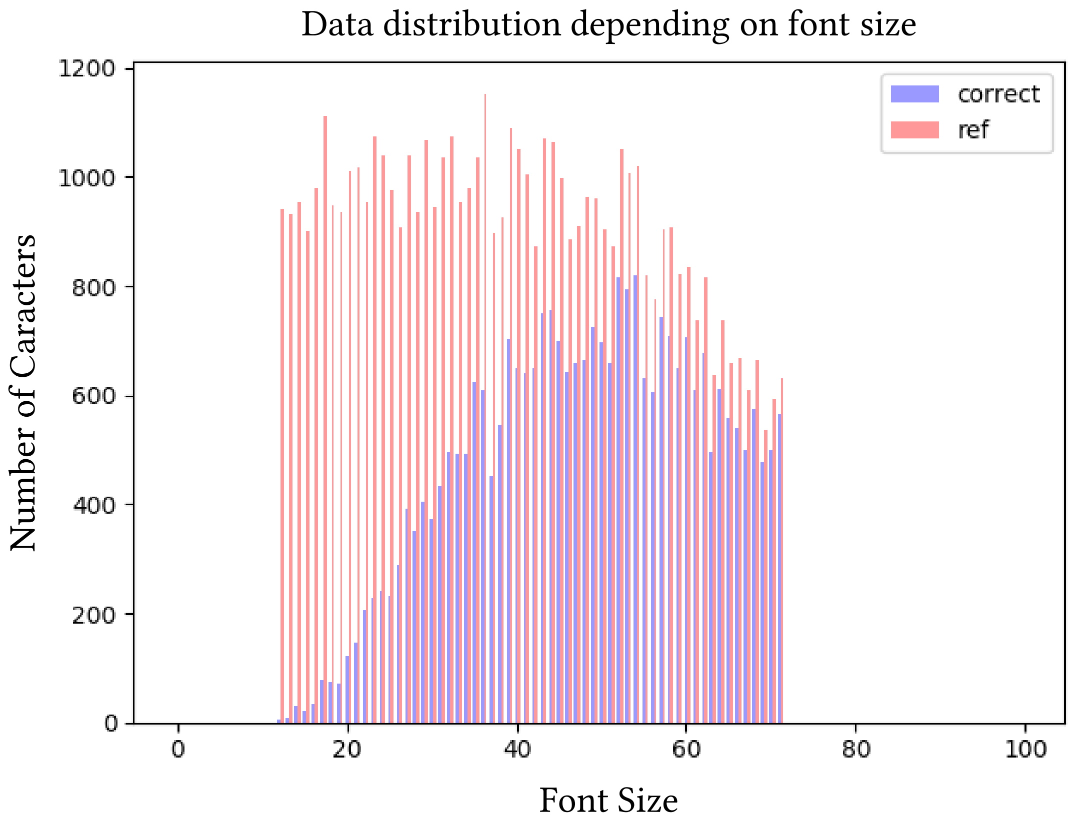


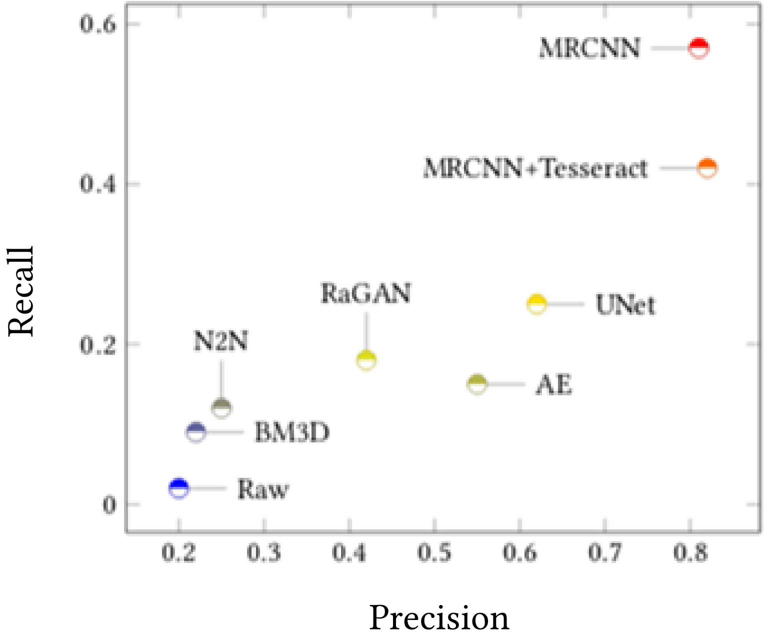
Architecture	OCR	Processing Type	F-Score (Caracter-wise)
Raw	Tesseract	∅	0,02
BM3D		Denoising	0,18
Auto-Encoder		Denoising	0,21
SegNet [9]		Semantic Segmentation	0,23
RaGAN [10]		Denoising	0,24
DnCNN		Denoising	0,30
U-Net [11]		Denoising	0,31
Mask-RCNN		Instance Segmentation	0,55
	∅		0,68

- F-Score $= 2.\frac{precision\,.\,recall}{precision + recall}$
- $precision = \frac{TP}{TP+FP}$
- $recall = \frac{TP}{TP+FN}$



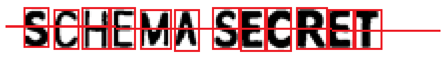
ToxicIA


Data distribution depending on font size
Number of Caracters
Font Size
correct
ref
0
200
400
600
800
1000
1200
20
40
60
80
100

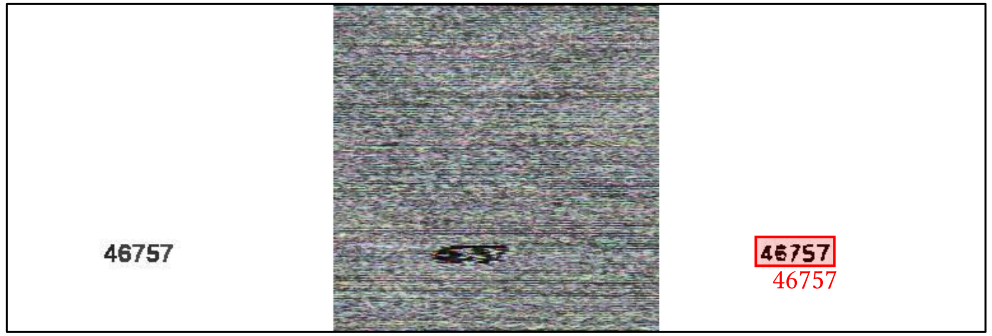

Recall
Precision
MRCNN
MRCNN+Tesseract
RaGAN
UNet
N2N
AE
BM3D
Raw
0
0.2
0.4
0.6
0.3
0.5
0.7
0.8

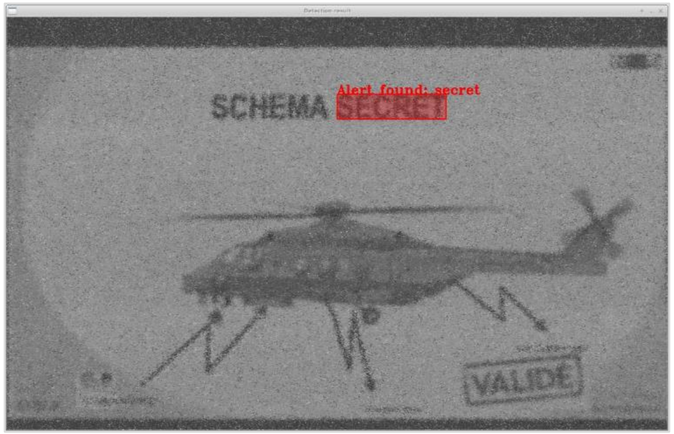
ToxicIA

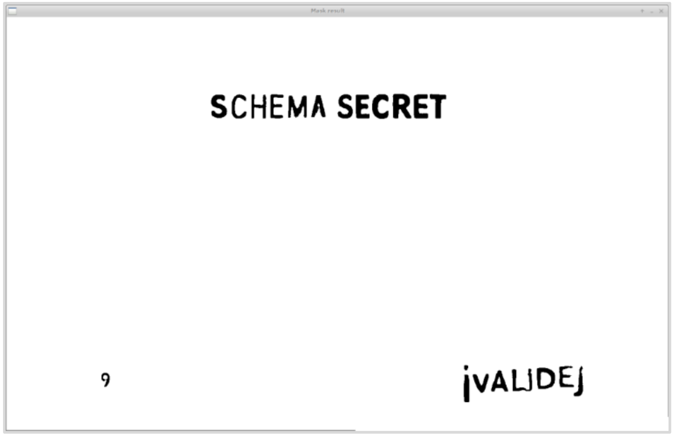
- Mask-RCNN + Post Processing:
  - Text Line detection: Hough Transform

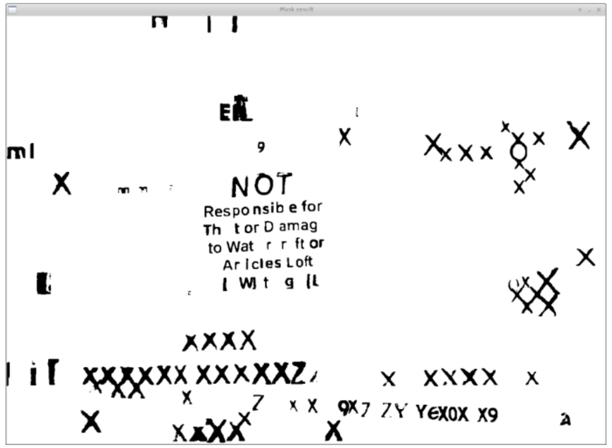


  - Approximate sub-string search: Bitap [12]
    - Found string: schemaseoret
    - Researched word: secret

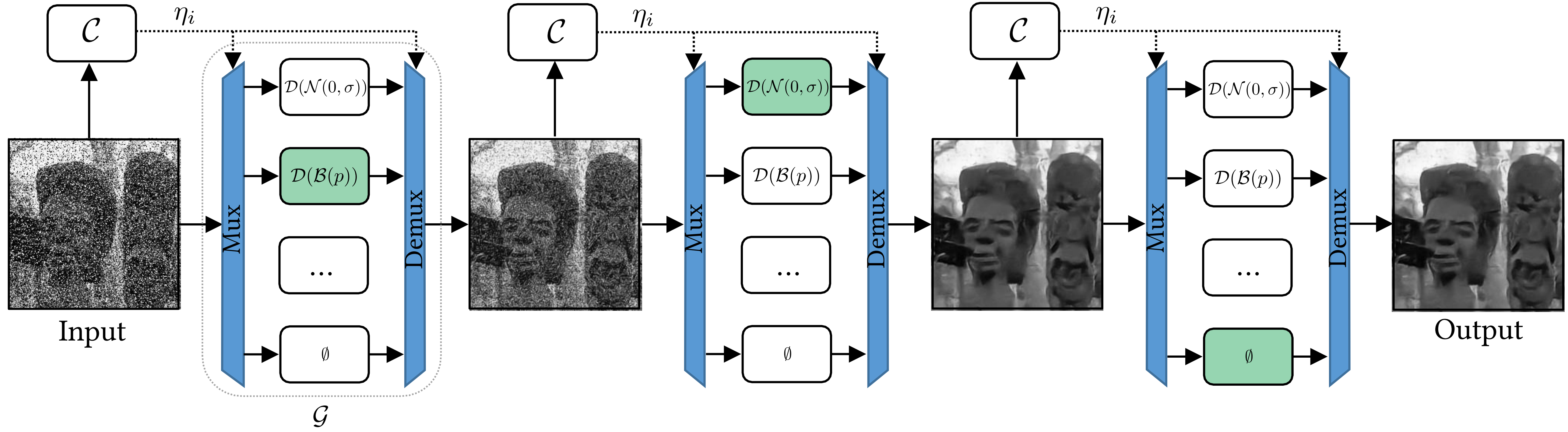








NoiseBreaker



[Lemarchand20] F. Lemarchand, E. Nogues, et M. Pelcat, « NoiseBreaker: Gradual Image Denoising Guided by Noise Analysis », in MMSP20.

Primary Noise Classes and Denoisers Architectures

Class Refinement

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

Dedicated Denoising Architecture

NoiseBreaker

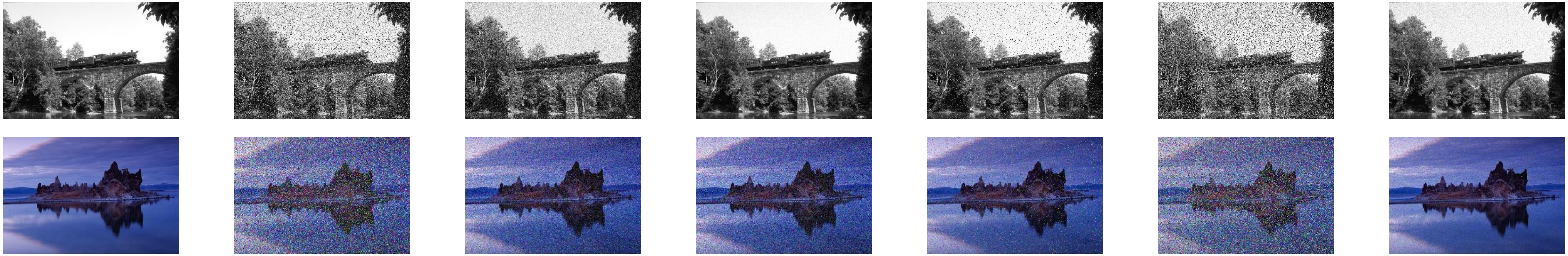
Evaluation Noise Mixtures

	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	$\mathcal{S}([0, 55])$	$\mathcal{P}$

- Same configuration as Liu et al.

Evaluation Images Examples

Ref C_0 C_1 C_2 C_3 C_4 C_5



NoiseBreaker

Evaluation metrics: PSNR, SSIM

Evaluation data: BSD68-Grayscale [10], BSD68-RGB [10]

- BM3D/CBM3D applied with $\sigma = 50$
- N2V retrained on each mixture
- Results of Liu et al. taken from paper

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	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	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.	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**

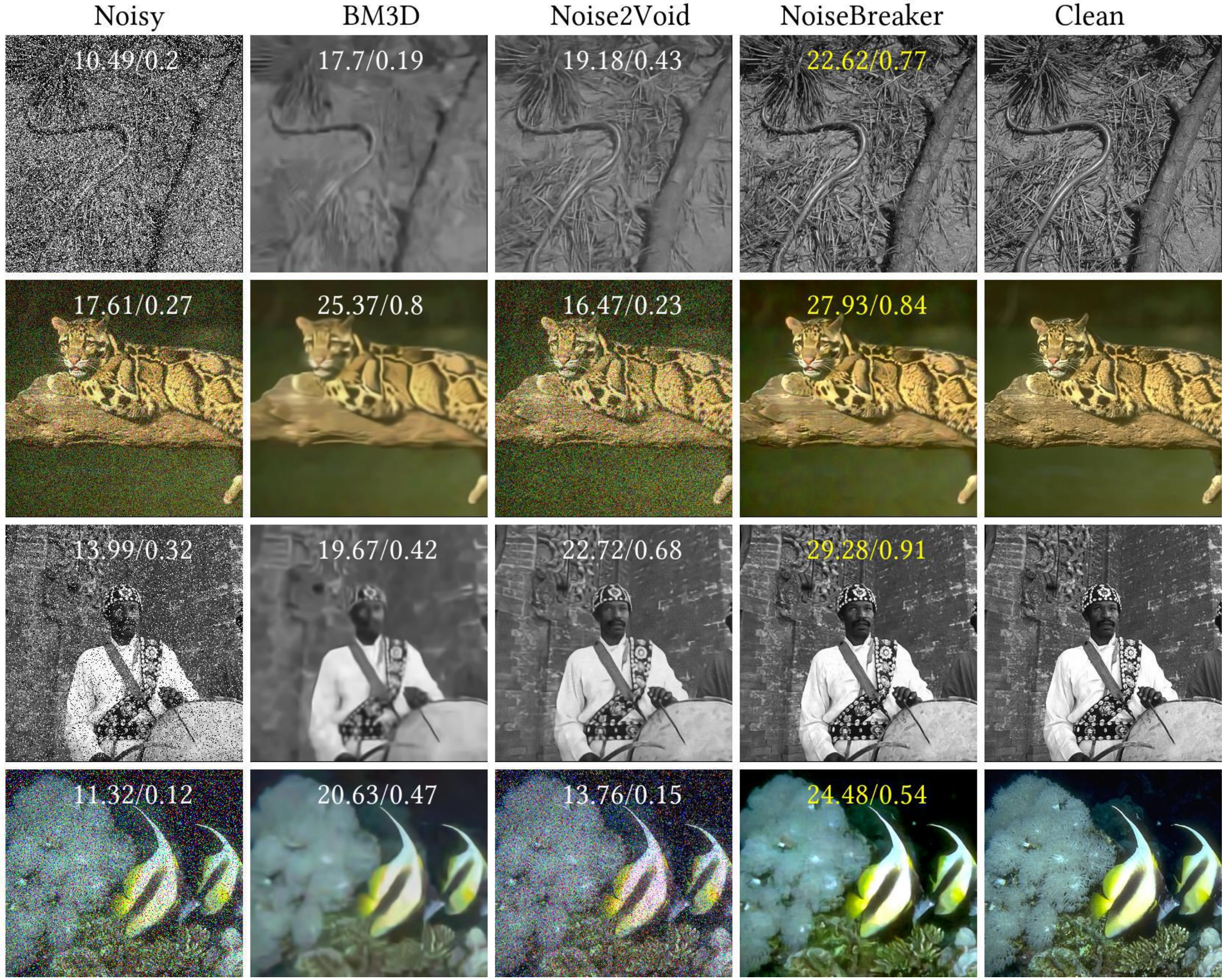
⇩ + 2dB PSNR, +13% SSIM

⇩ + 4,8dB PSNR, +38% SSIM

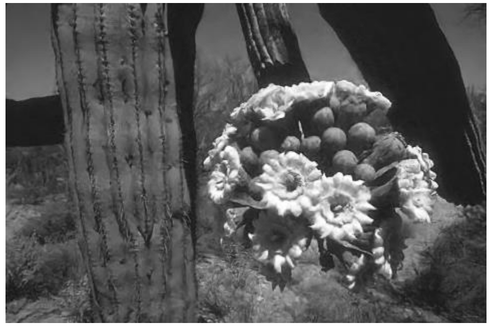
[10] D. Martin, C. Fowlkes, D. Tal, et J. Malik, « A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics », in Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, Vancouver, BC, Canada, 2001, vol. 2, p. 416-423, doi: 10.1109/ICCV.2001.937655.

NoiseBreaker

Subjective Results



Discussion



First denoising step may remove the second noise.

A wrong denoiser may be applied.

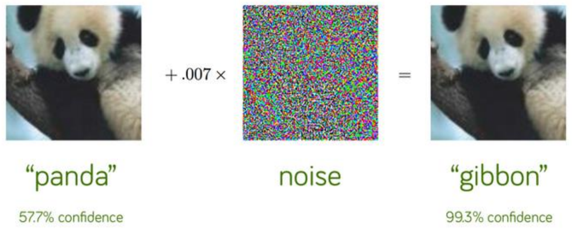
Noisy image may be classified as clean when low noise intensity.

Plan

Challenges and Perspectives

- Eavesdropped Image Denoising
  - Building of large and representative dataset:
    - Clean references expensive to obtain
    - Two interception campaigns can be very different
      - Type of antenna, distance, perturbations (phones, …), raster settings
  - Unknown and ‘Unstable’ Noise model
    - Video denoising to benefit from time integration

- Deep Learning (DL)
  - Requires large datasets and labelisation for supervised learning Fine-Tuning
    - Advances on few-shot learning→ Learning from only few examples [Koch15]
  - DL is resource-hungry: both computation and memory → Specific hardware and energy consumption
    - New training strategies? On CPU?
    - Fixed-Point Mixed-precision Networks [Micikevicius17]
  - Explainability
    - XAI: eXplainable Artificial Intelligence [Zhou16]
  - Security
    - How to test all responses to input?
    - Adversarial Networks



[Micikevicius17] Micikevicius, Paulius, et al. "Mixed precision training." arXiv preprint arXiv:1710.03740 (2017)
[Koch15] Koch, Gregory, Richard Zemel, and Ruslan Salakhutdinov. "Siamese neural networks for one-shot image recognition." ICML deep learning workshop. Vol. 2. 2015.
[Zhou16] Zhou, Bolei, et al. "Learning deep features for discriminative localization." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

Practical Work Summary

- Supervision: Maxime Pelcat and Florian Lemarchand
- PW1 : Basics of Image Processing and Denoising (1h45)
- PW2 : Toward Eavesdropping Denoising (1h45)