ANR Prise 3D

Thesis : Noise detection in stereoscopic synthesis images using machine learning

Jérôme BUISINE

LISIC - Imap

Table of contents

Noise in synthesis images

Output overview

(a) After 50 samples **(b)** After 300 samples **(c)** After 1200 samples

Figure 1: Preview of the images obtained by the Maxwell rendering engine of the Cuisine01 (D) scene at different generation times

Sub image overview

(a) After 50 samples **(b)** After 300 samples **(c)** After 1200 samples

Figure 2: Preview of the sub images obtained by the Maxwell render engine of the Cuisine01 (D) scene at different generation times

Noise overview

As we can see after 50 minutes of generation, a perceptual noise is generated due to the Monte-Carlo (stochastic) process.

Problematic

- How to detect this **perceptual** noise ?

Problematic

- How to detect this **perceptual** noise ?

- How to **quantify** it ?

Image quality

Image quality metrics

Image quality assessment (IQA) metrics can be divided in three categories :

- Full reference (FR) metrics

Image quality metrics

Image quality assessment (IQA) metrics can be divided in three categories :

- Full reference (FR) metrics
- Reduced reference (RR) metrics

Image quality metrics

Image quality assessment (IQA) metrics can be divided in three categories :

- Full reference (FR) metrics
- Reduced reference (RR) metrics
- No-reference (NR) metrics

Image quality metrics

Image quality assessment (IQA) metrics can be divided in three categories :

- Full reference (FR) metrics
- Reduced reference (RR) metrics
- No reference (NR) metrics

Image quality metrics

- Full reference :
  - Peak Signal to Noise Ratio (PSNR)
  - Structural Similarity Index Metrix (SSIM) [Wang et al., 2004]
  - Multi-Scale SSIM (MS-SSIM) [Wang et al., 2003]
  - ...
- No-reference :
  - Blind Image Quality Index (BIQI) [Moorthy and Bovik, 2010]
  - Blind Referenceless Image Spatial Quality Evaluator (BRISQUE) [Mittal et al., 2012]
  - Perception-based Image QUality Evaluator (PIQUE) [N. et al., 2015]
  - ...

Image quality metrics Databases

Database of natural images are available with distortions applied on these images.

A **subjective** score is then associated to these images.

- TID2008
- LIVE
- CSIQ
- ...

Image quality metrics Databases

Database of natural images are available with distortions applied on these images.

A **subjective** score is then associated to these images.

- TID2008
- LIVE
- CSIQ
- ...

Model objective

Correlate as well as possible with the subjective scores.

Image quality metrics Databases

Database of natural images are available with distortions applied on these images.

A **subjective** score is then associated to these images.

- TID2008
- LIVE
- CSIQ
- ...

Synthesis images database

Currently in the literature, there is no database that identifies the noise present into synthesis images.

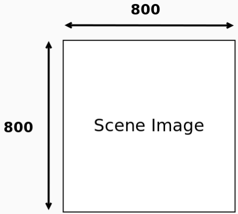
Database

Scenes

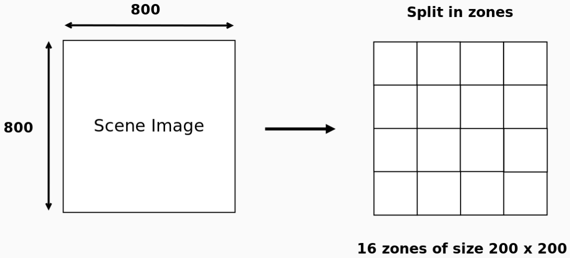
ID	Scene	Indices (samples)	Step	Images
A	Appart02	$20 \rightarrow 900$	10	89
D	Cuisine01	$20 \rightarrow 1200$	10	119
G	SdbCentre	$20 \rightarrow 950$	10	94
H	SdbDroite	$20 \rightarrow 950$	10	94

Table 1: Reduced database information

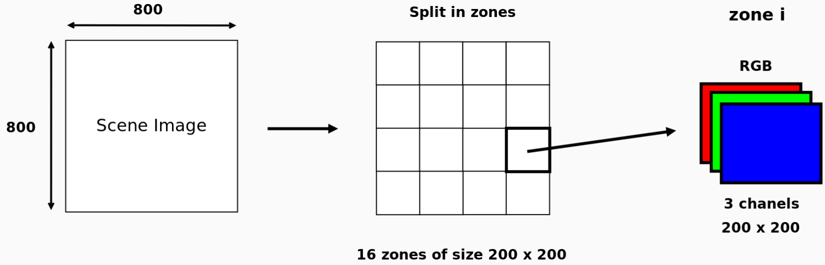
Database explanations



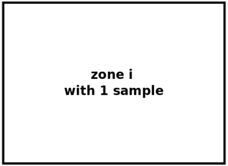
Database explanations



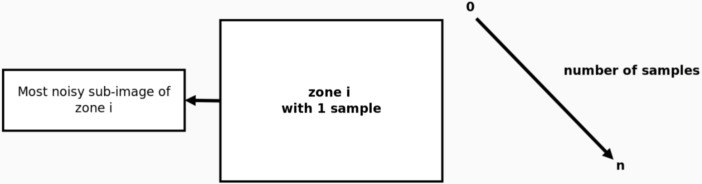
Database explanations



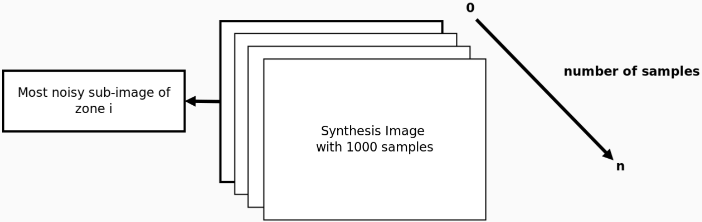
Subjective perceptual threshold



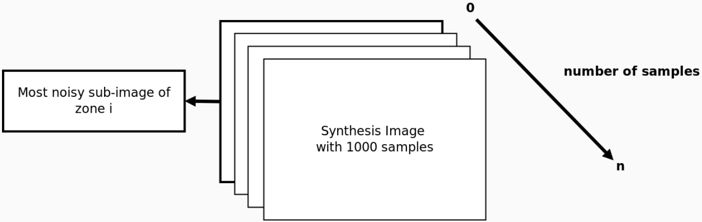
Subjective perceptual threshold



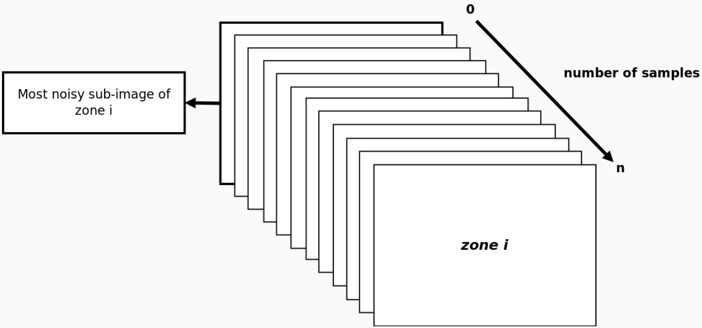
Subjective perceptual threshold



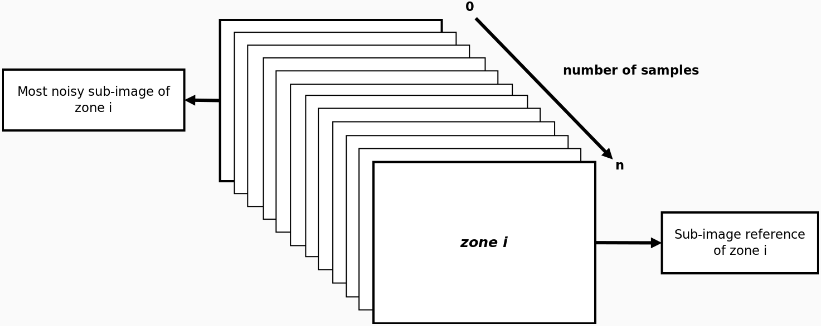
Subjective perceptual threshold



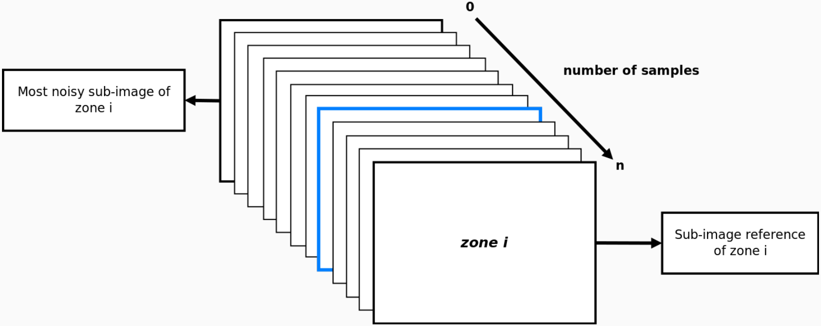
Subjective perceptual threshold



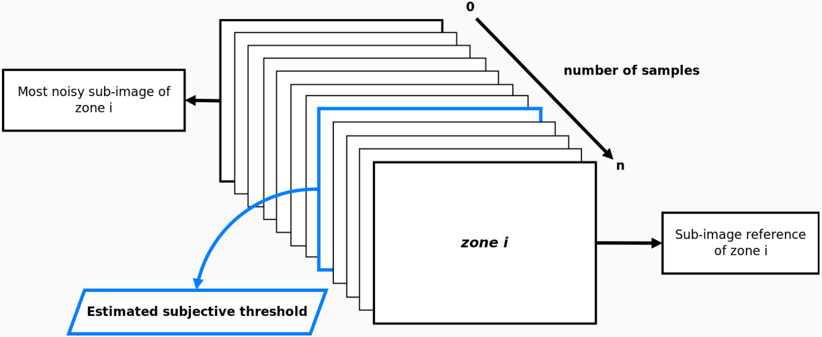
Subjective perceptual threshold



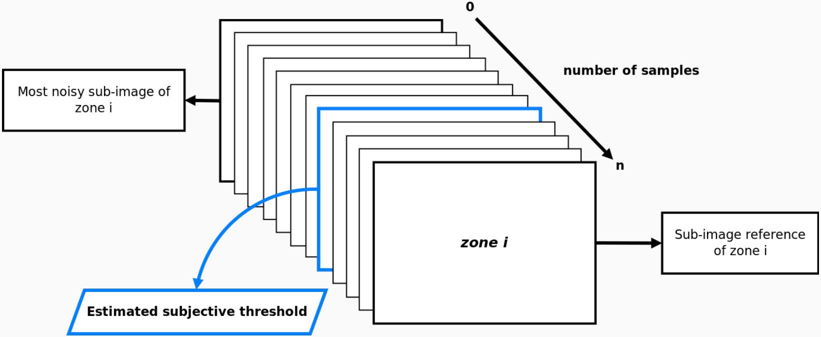
Subjective perceptual threshold



Subjective perceptual threshold



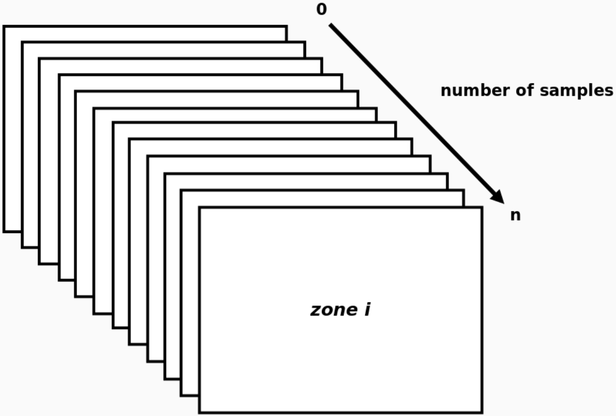
Subjective perceptual threshold



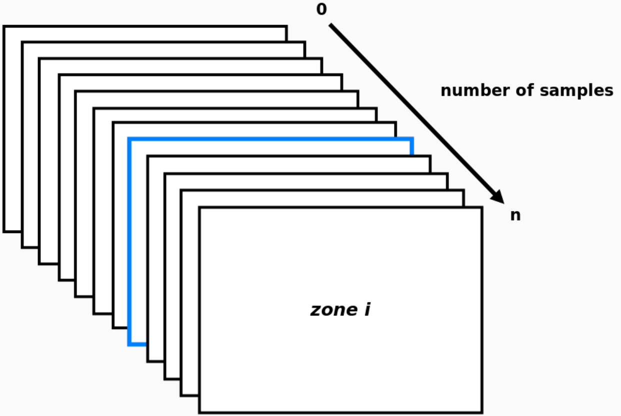
Information

The final threshold for a zone is the **mean** of all subjective thresholds obtained.

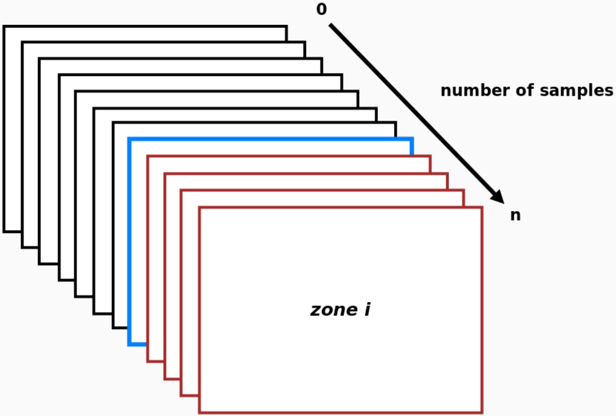
How sub-images are classified ?



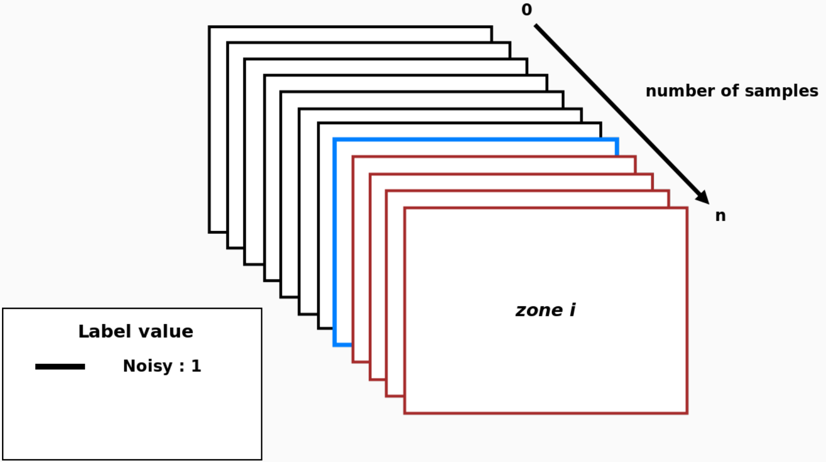
How sub-images are classified ?



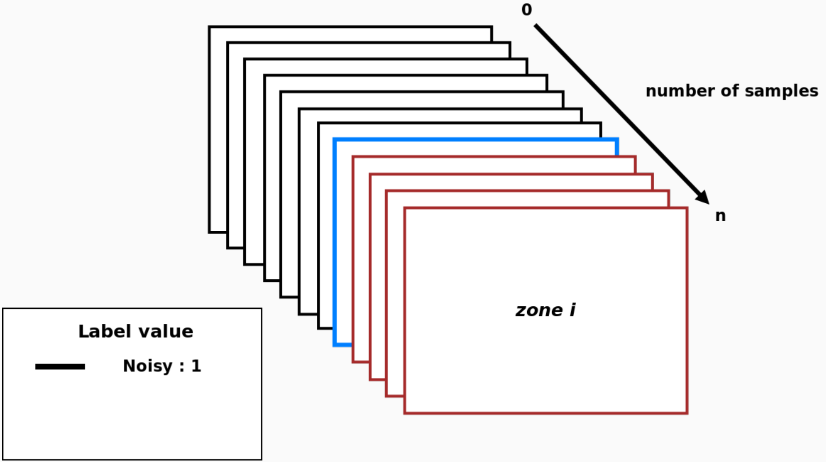
How sub-images are classified ?



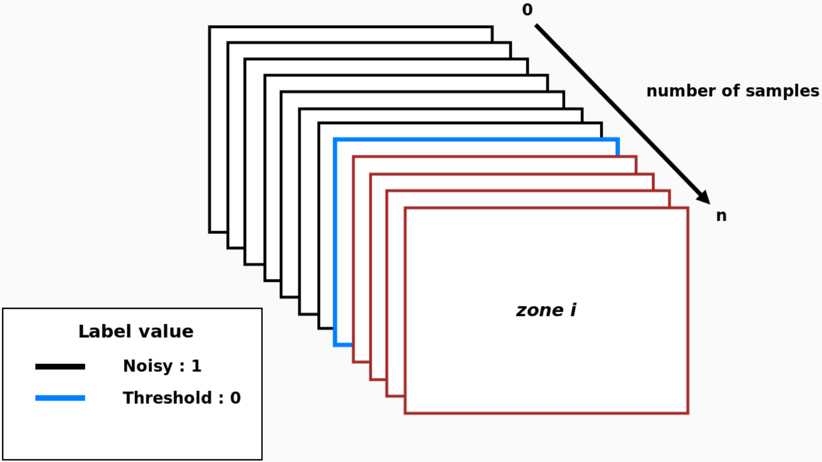
How sub-images are classified ?



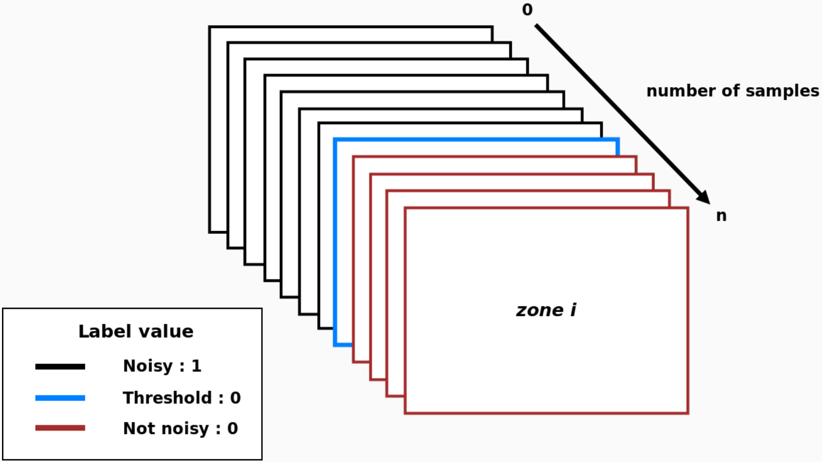
How sub-images are classified ?



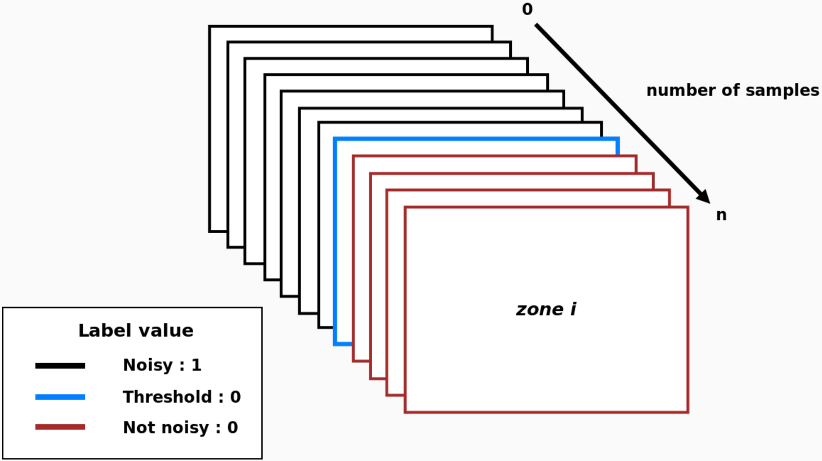
How sub-images are classified ?



How sub-images are classified ?



How sub-images are classified ?

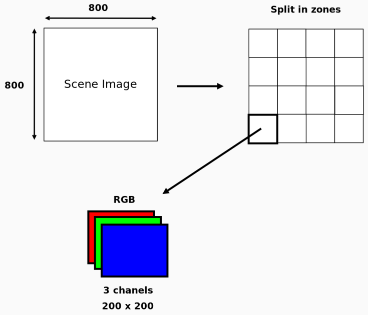


Label attribution

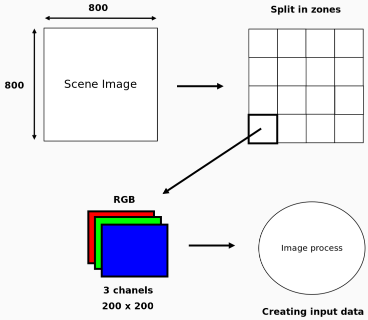
For each zone, a **label** value (0 or 1) is associated to the sub-image obtained at *n* samples during the generation.

Current works

What we need ?

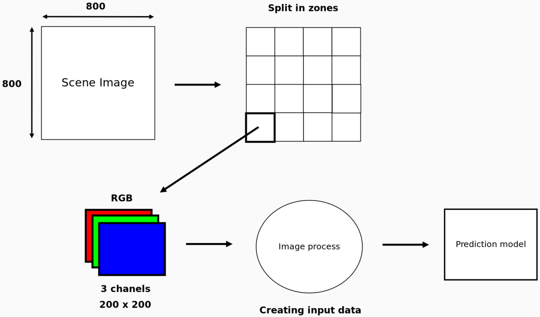


What we need ?

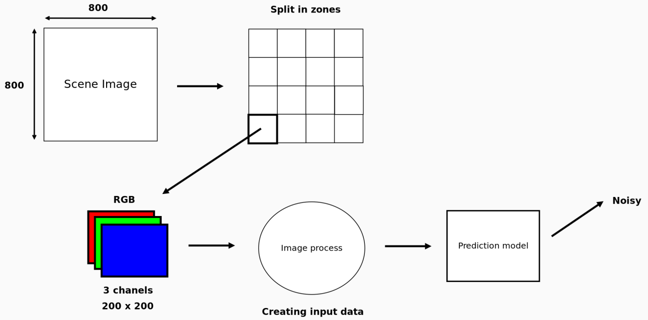


800
800
Scene Image
Split in zones
RGB
3 chanels
200 x 200
Image process
Creating input data

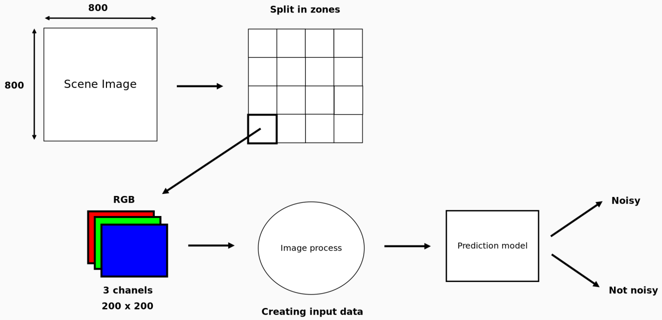
What we need ?


800
800
Scene Image
Split in zones
RGB
3 chanels
200 x 200
Image process
Creating input data
Prediction model

What we need ?

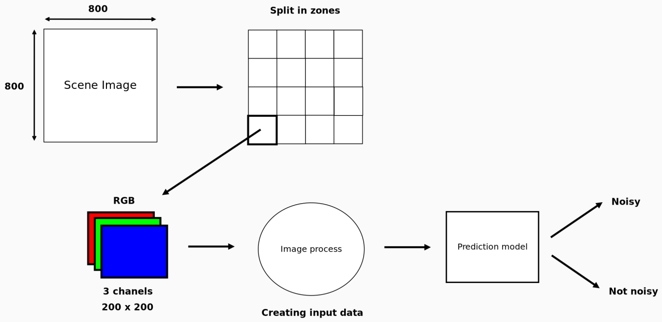

800
Split in zones
800
Scene Image
RGB
Noisy
Image process
Prediction model
3 chanels
200 x 200
Creating input data

What we need ?



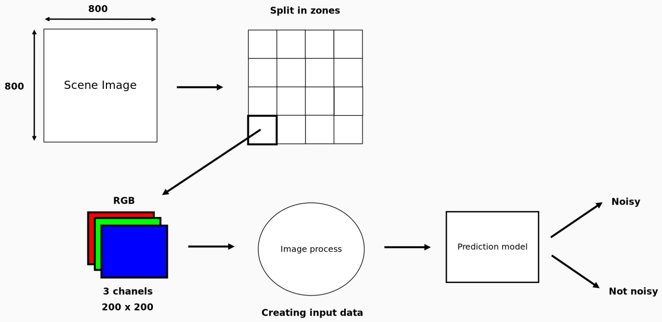
800
800
Scene Image
Split in zones
RGB
3 chanels
200 x 200
Image process
Creating input data
Prediction model
Noisy
Not noisy

What we need ?



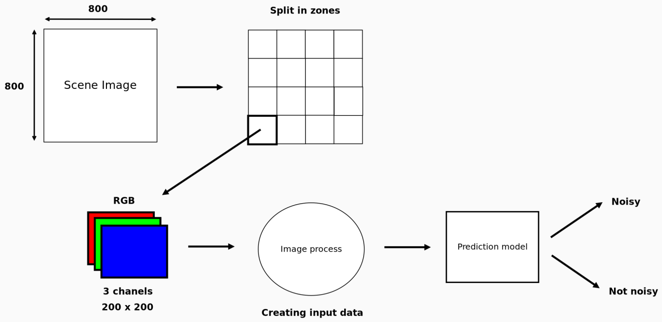
- What kind of data model wants in order to classify as well as possible sub-images

What we need ?



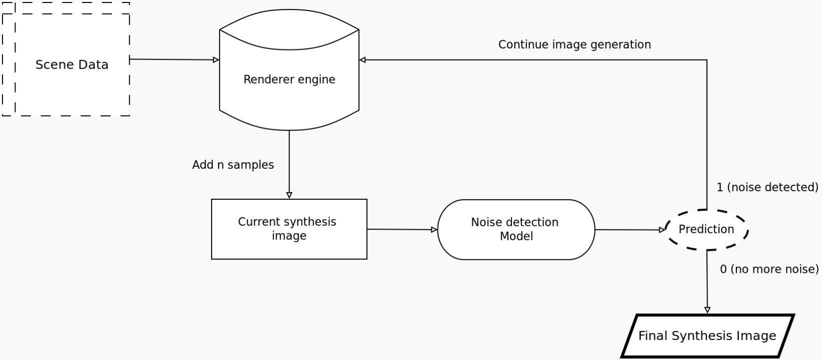
- What kind of data model wants in order to classify as well as possible sub-images
- The whole sub-image or reduced information ?

What we need ?

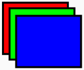


- What kind of data model wants in order to classify as well as possible sub-images
- The whole sub-image or reduced information ?
- What kind of data well described the perceived noise in image ?

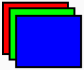
Expected model interactions



Reduction of canals

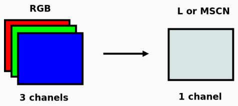


Reduction of canals

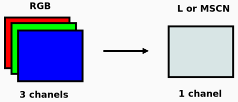


- Using **L** (luminance) canal from L*a*b transformation.
- Using the Mean Substracted Contrast Normalized (MSCN, see Eq. 3) matrix.

Pool of final features



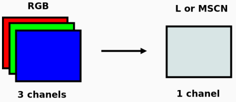
Pool of final features



Hypothesis

Low bits values from images perhaps keep information about noise

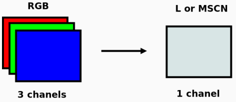
Pool of final features



Hypothesis

Low bits values from images perhaps keep information about noise

Pool of final features



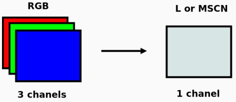
Hypothesis

Low bits values from images perhaps keep information about noise

low_bits_3

1	1	1	1	1	1	1	1	$\Longrightarrow$	1	1	1	0	0	0	0	0

Pool of final features



Hypothesis

Low bits values from images perhaps keep information about noise

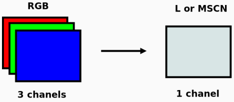
low_bits_3

1	1	1	1	1	1	1	1	$\Rightarrow$	1	1	1	0	0	0	0	0

low_bits_4_shifted_2

1	1	1	1	1	1	1	1	$\Rightarrow$	0	0	1	1	1	1	0	0

Pool of final features



Hypothesis

Low bits values from images perhaps keep information about noise

low_bits_3

1	1	1	1	1	1	1	1	$\Longrightarrow$	1	1	1	0	0	0	0	0

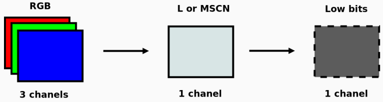
low_bits_4_shifted_2

1	1	1	1	1	1	1	1	$\Longrightarrow$	0	0	1	1	1	1	0	0

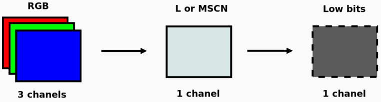
$$F = \{lab, mscn, low\_bits\_4\_shifted\_2\} + [low\_bits\_i]$$

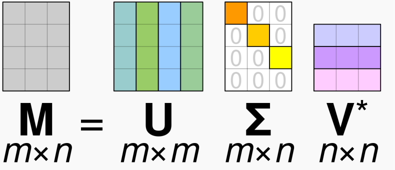
with $i \in [2, 6]$ and $|F| = 8$

Dimension reduction using SVD

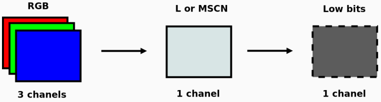


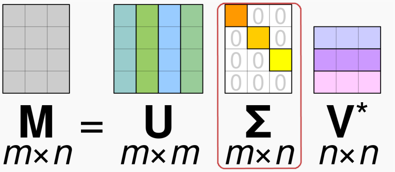
Dimension reduction using SVD





Dimension reduction using SVD





Why the use of SV vector ?

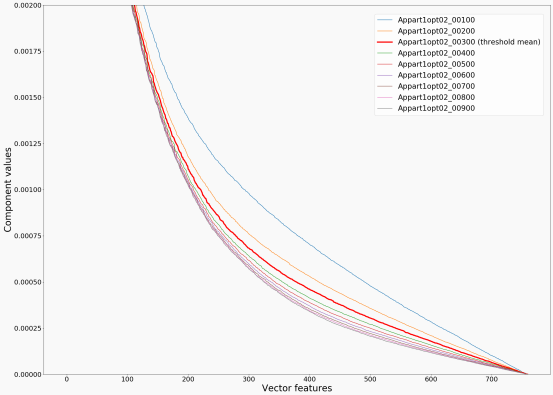
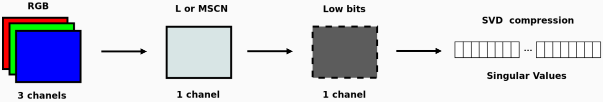
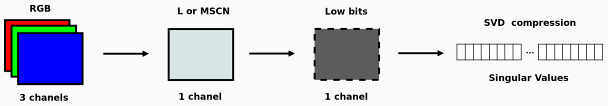


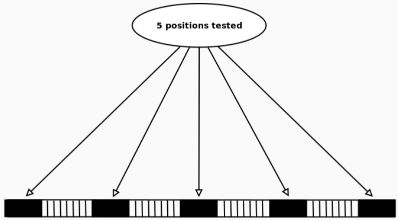
Figure 3: Singular values vector obtained from images of Appart02 (A) scene with **L** chanel

Another reduction



Another reduction

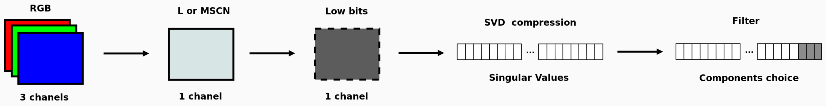




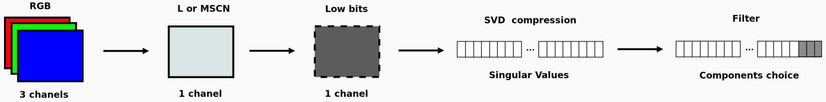
- Positions set, P with $|P| = 5$
- Potential sub-vectors size,

$$N = [4, 8, 16, 26, 32, 40]$$

Other parameters : normalization



Other parameters : normalization

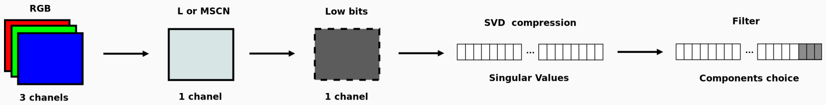


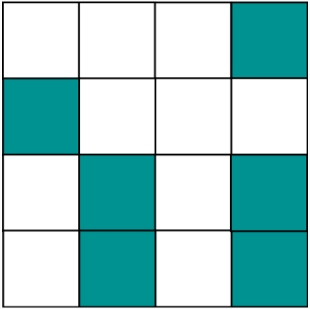
$$K = [svd, svdn, svdne]$$

where

- svd : no normalization
- svdn : sub-vector is normalized itself
- svdne : sub-vector is normalized depending the **min** and **max** sub-vectors interval values from the whole dataset

Other parameters : training data

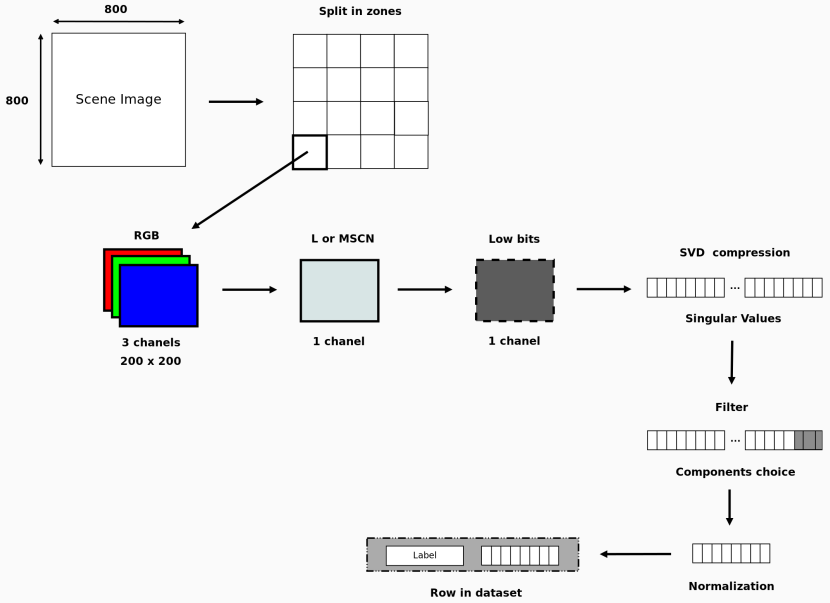




Zones are selected randomly with

$$Z = [4, 6, 8, 10, 12]$$

Final features as model input



Kind of models

We define 3 model architectures to fit as well as possible final features :

- M1 : **Support Vector Machine**
- M2 : **Ensemble_model** (3 sub-models)
- M3 : **Ensemble_model_v2** (5 sub-models)

Ensemble models configurations

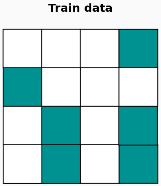
These ensemble models are in fact **voting classifier** with principle of fair voting and regulated on *soft*.

Parameters : total combinations

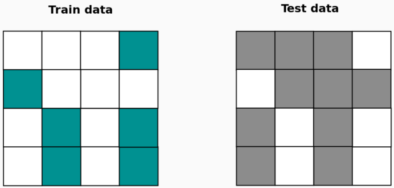
Finally using all these parameters, we have a lot of combinations :

$$
\begin{aligned}
r &= 3 \times |F| \times |P| \times |N| \times |K| \times |Z| \\
&= 3 \times 8 \times 5 \times 6 \times 3 \times 5 \\
&= 10800
\end{aligned}
$$

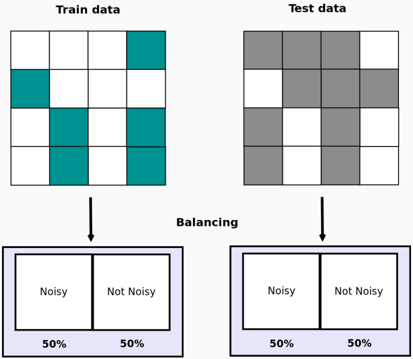
Build of specific dataset



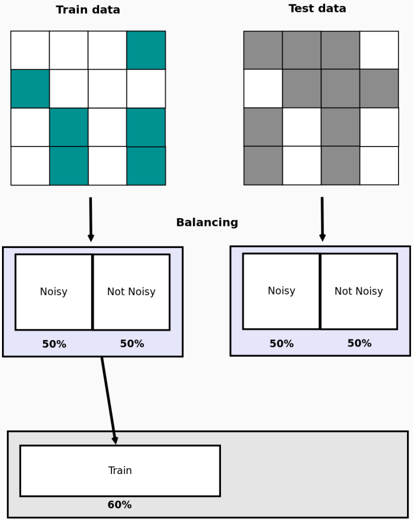
Build of specific dataset



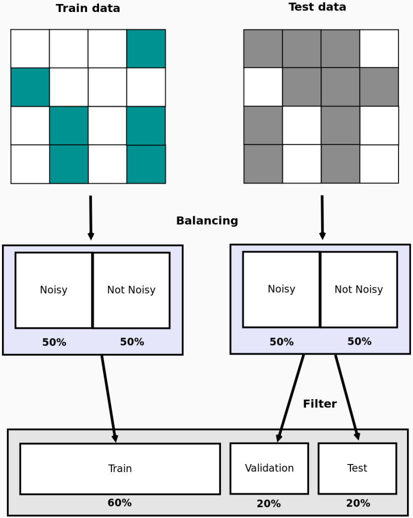
Build of specific dataset



Build of specific dataset



Build of specific dataset



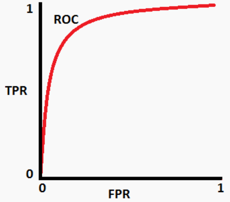
How to compare model ?

The Area Under The Curve Receiver Operating Characteristics score is used to compare these models based on **test** dataset.

How to compare model ?

The Area Under The Curve Receiver Operating Characteristics score is used to compare these models based on **test** dataset.

AUC - ROC score is a performance measurement for classification problem. It tells how much model is capable of distinguishing between classes.



First results

Model	feature	size	interval	zones	ROC Train	ROC Val	ROC Test
M3	lab (svd)	40	[80, 120[	12	0.9418	0.9023	0.9219
M2	lab (svd)	32	[84, 116[	4	0.9158	0.8724	0.9153
M2	lab (svd)	40	[80, 120[	12	0.9629	0.9049	0.9145
M2	lab (svdne)	26	[87, 113[	6	0.9337	0.8763	0.9089
M3	low_bits_2 (svd)	40	[0, 40[	12	0.9567	0.8417	0.9081

Table 2: The 5 best models found based on AUC ROC score

Simulation from best model

Figure 4: Simulation of each zone obtained on scene Appart02 (A)

Simulation from best model

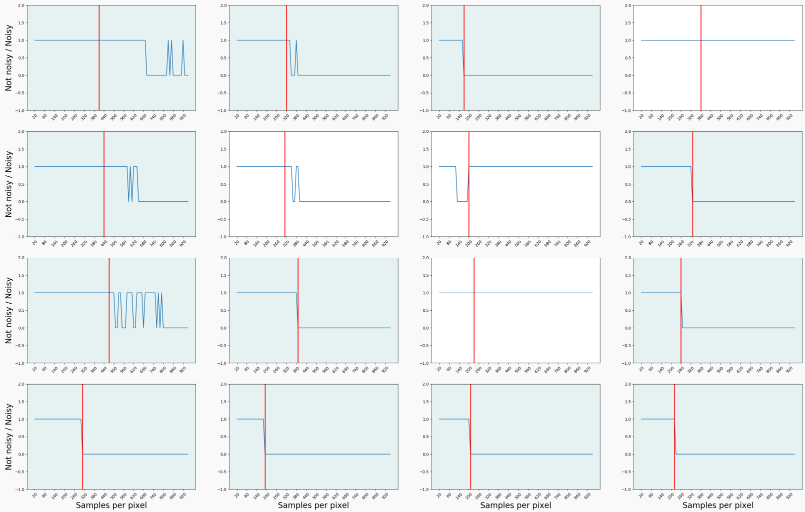


Figure 5: Simulation of each zone obtained on scene SdbDroite (H)

Other approaches

Features

- Statistics approach, use of statistics from sub-block
  - Mean, Median, Percentile at 25%, Percentile at 75%, Variance, Area under curvex

Other approaches

Features

- Statistics approach, use of statistics from sub-block
  - Mean, Median, Percentile at 25%, Percentile at 75%, Variance, Area under curvex
- Use of MSCN statistics

Other approaches

Features

- Statistics approach, use of statistics from sub-block
  - Mean, Median, Percentile at 25%, Percentile at 75%, Variance, Area under curvex
- Use of MSCN statistics
- Use of correlation matrix from SV

Other approaches

Features

- Statistics approach, use of statistics from sub-block
  - Mean, Median, Percentile at 25%, Percentile at 75%, Variance, Area under curvex
- Use of MSCN statistics
- Use of correlation matrix from SV
- Use of correlation between SV and labels

Other approaches

Features

- Statistics approach, use of statistics from sub-block
  - Mean, Median, Percentile at 25%, Percentile at 75%, Variance, Area under curvex
- Use of MSCN statistics
- Use of correlation matrix from SV
- Use of correlation between SV and labels

Model

- Use of Deep Learning approach as model

Other approaches

Features

- Statistics approach, use of statistics from sub-block
    - Mean, Median, Percentile at 25%, Percentile at 75%, Variance, Area under curvex
- Use of MSCN statistics
- Use of correlation matrix from SV
- Use of correlation between SV and labels

Model

- Use of Deep Learning approach as model

Remark

All of these approaches seems to give same results as before.. Hence, **overfitting** or bad approximation.

Conclusion

Conclusion

Difficult to have a model which **generalizes** for each scene but why ?

Conclusion

Difficult to have a model which **generalizes** for each scene but why ?

- Scene seems to have each own components to describe the noise
- Difficult to find the best components from data

Conclusion

Difficult to have a model which **generalizes** for each scene but why ?

- Scene seems to have each own components to describe the noise
- Difficult to find the best components from data

Solution :

- Find a way to choose components depending of the scene ?
- Use of work of André and Rémi : SV entropy

Future works

Noise detection

- Study of Singular Values behaviors from few noises
- Use of Generative Adversarial Network model
- Use of Transfer learning (Alexnet, Resnet, ...)
- Work with University of Lille 3
- Tackle the **stereoscopic** aspect

Future works

Noise detection

- Study of Singular Values behaviors from few noises
- Use of Generative Adversarial Network model
- Use of Transfer learning (Alexnet, Resnet, ...)
- Work with University of Lille 3
- Tackle the **stereoscopic** aspect

Denoising

- Use of Deep Learning approaches (autoencoder and others...) to denoise images
- Create custom denoiser for synthesis images

Development

Developments are centralized into the *IPFML* python package (`https://github.com/jbuisine/IPFML`).

Questions?

References i

- Mittal, A., Moorthy, A. K., and Bovik, A. C. (2012).
  No-reference image quality assessment in the spatial domain.
  IEEE Trans. Image Processing, 21(12):4695–4708.
- Moorthy, A. K. and Bovik, A. C. (2010).
  A two-step framework for constructing blind image quality indices.
  IEEE Signal Process. Lett., 17(5):513–516.
- N., V., D., P., Bh., M. C., Channappayya, S. S., and Medasani, S. S. (2015).
  Blind image quality evaluation using perception based features.
  In *Twenty First National Conference on Communications, NCC 2015, Mumbai, India, February 27 - March 1, 2015*, pages 1–6.

References ii

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004).
Image quality assessment: from error visibility to structural similarity.
IEEE Trans. Image Processing, 13(4):600–612.

Wang, Z., Simoncelli, E. P., and Bovik, A. C. (2003).
Multiscale structural similarity for image quality assessment.
In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee.

Backup slides (PSNR)

$$PSNR = 10 \times log_{10}\left(\frac{d^2}{MSE}\right) \tag{1}$$

where d is the signal dynamics (the maximum possible value for a pixel). In the standard case of an image where the components of a pixel are encoded on 8 bits, $d = 255$ and MSE (see Eq. 2) is the mean square error between the 2 images.

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

where I_o is the distorted image and I_r the reference image, both of size $m \times n$

Backup slides (MSCN)

To calculate the MSCN matrix, we must first convert our RGB image to a grayscale image. The MSCN will extract Natural Structure Scene (NSS) information from this grayscale image. An operation is applied to the luminance image $I(i, j)$ to produce :

$$\hat{I}(i, j) = \frac{I(i, j) - \mu(i, j)}{\sigma(i, j) + C} \tag{3}$$

where $i \in 1, 2...M, j \in 1, 2...N$ are spatial indices, M, N are respectively the height and width of the image, C is constant, which is set to 1 to avoid instability

Backup slides (MSCN)

and where,

$$\mu(i,j) = \sum_{k=-K}^{K} \sum_{l=-L}^{L} w_{k,l} I_{k,l}(i,j) \tag{4}$$

$$\sigma(i,j) = \sqrt{\sum_{k=-K}^{K} \sum_{l=-L}^{L} w_{k,l} (I_{k,l}(i,j) - \mu(i,j))^2} \tag{5}$$

In (4) and (5) $w = \{w_{k,l} | k = -K, ..., K, l = -L, ..., L\}$ is a circularly symmetrical 2D Gaussian weighting function sampled at 3 standard deviations ($K = L = 3$) and recalculated at unit volume. Then, the transformed luminance values of (3) are called Mean Subtracted Constrast Normalized (MSCN) coefficients.

Kind of models

We define 3 models architecture to fit as well as possible final features :

- **Support Vector Machine** (with RBF kernel and cross validation process)

Kind of models

We define 3 models architecture to fit as well as possible final features :

- **Support Vector Machine** (with RBF kernel and cross validation process)
- **Ensemble_model**, composed of :
  - SVM with same configuration as previous model
  - Random Forest with 100 estimators
  - Logistic Regression with *liblinear* kernel

Kind of models

We define 3 models architecture to fit as well as possible final features :

- **Support Vector Machine** (with RBF kernel and cross validation process)
- **Ensemble_model**, composed of :
  - SVM with same configuration as previous model
  - Random Forest with 100 estimators
  - Logistic Regression with *liblinear* kernel
- **Ensemble_model_v2**, composed of :
  - SVM with same configuration as previous model
  - Random Forest with 100 estimators
  - Logistic Regression with *liblinear* kernel
  - KNeighbors Classifier
  - Gradient Boosting Classifier with 100 estimators and learn step set to 1.0.