Stochastic super-resolution for Gaussian textures

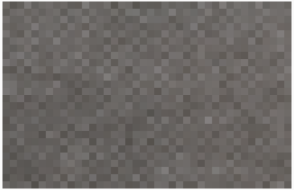
Émile Pierret[a], supervised by Bruno Galerne[a,b]

Journées Textures, Vannes, 10 octobre

[a] Institut Denis Poisson – Université d'Orléans, Université de Tours, CNRS
[b] Institut universitaire de France (IUF)

Introduction to stochastic super-resolution (SR)

The Single Image Super-Resolution (SISR)

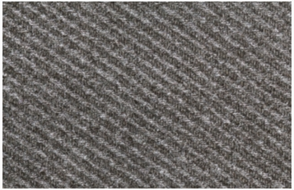


LR image: $\mathbf{u}_{\mathrm{LR}}$

$= \mathbf{S}$

convoluted image

$= \underbrace{\mathbf{SC}}_{=\mathbf{A}}$



HR image: $\mathbf{u}_{\mathrm{HR}}$

- SISR setting litterature: [Bruna et al., 2016][1] [Ledig et al., 2017][2], [Wang et al., 2019][3], [Johnson et al., 2016][4], [Hertrich, Houdard, et al., 2022][5], [Hertrich, Nguyen, et al., 2022][6], [Chatillon et al., 2022][7]

---

[1] Bruna, J., Sprechmann, P., & LeCun, Y. (2016). Super-Resolution with Deep Convolutional Sufficient Statistics. *ICLR 2016*

[2] Ledig, C., Theis, L., & Huszár, F. e. a. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *CVPR 2017*

[3] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., & Loy, C. C. (2019). ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. *ECCV 2018*

[4] Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. *ECCV 2016*

[5] Hertrich, J., Houdard, A., & Redenbach, C. (2022). Wasserstein Patch Prior for Image Superresolution. *IEEE Transactions on Computational Imaging*

[6] Hertrich, J., Nguyen, L. D. P., Aujol, J.-F., Bernard, D., Berthoumieu, Y., Saadaldin, A., & Steidl, G. (2022). PCA Reduced Gaussian Mixture Models with Applications in Superresolution. *Inverse Problems and Imaging*

[7] Chatillon, P., Gousseau, Y., & Lefebvre, S. (2022). A statistically constrained internal method for single image super-resolution. *ICPR 2022*

Stochastic super-resolution: an example

HR image

LR image (by a factor $r = 16$)

Image of Robert Hossein extracted from the dataset CelebA.

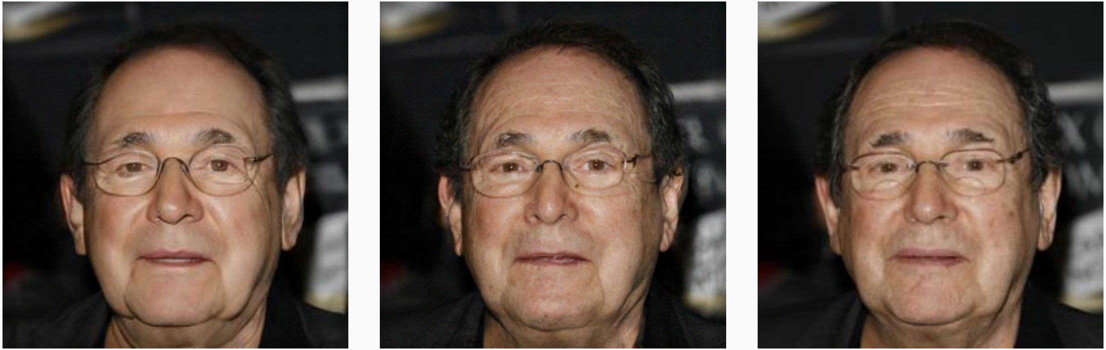


Image of Robert Hossein extracted from the dataset CelebA.

All these images have the same LR version !

The stochastic super-resolution

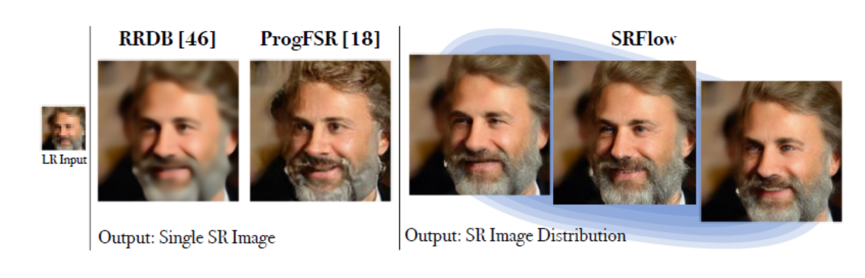


Image extracted from [Lugmayr et al., 2020]

Stochastic super-resolution litterature: SRFlow [Lugmayr et al., 2020][8], CEM [Bahat and Michaeli, 2020][9], SR3 [Saharia et al., 2022][10], DDRM [Kawar et al., 2022][11], DPS [Chung et al., 2023][12]

[8] Lugmayr, A., Danelljan, M., Van Gool, L., & Timofte, R. (2020). SRFlow: Learning the Super-Resolution Space with Normalizing Flow. *ECCV 2020*

[9] Bahat, Y., & Michaeli, T. (2020). Explorable super resolution. *CVPR*

[10] Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., & Norouzi, M. (2022). Image Super-Resolution Via Iterative Refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*

[11] Kawar, B., Elad, M., Ermon, S., & Song, J. (2022). Denoising diffusion restoration models. *ICLR Workshop on Deep Generative Models for Highly Structured Data*

[12] Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., & Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. *The Eleventh International Conference on Learning Representations*

Introduction to Gaussian microtextures

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [14]

Let $\mathbf{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{u} - m)$ its associated texton. Let $\mathbf{W}$ be a white Gaussian noise,

$$\mathbf{X} = \mathbf{t} \star \mathbf{W} \sim \mathrm{ADSN}(\mathbf{u}) = \mathscr{N}(\mathbf{0}, \mathbf{\Gamma}) \quad \textbf{which is a stationary law}$$

$\mathbf{u}$ Samples of $\mathrm{ADSN}(\mathbf{u})$

Image extracted from [Galerne et al., 2011a][13]

[13] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm_rpn

[14] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [14]

Let $\mathbf{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{u} - m)$ its associated texton. Let $\mathbf{W}$ be a white Gaussian noise,

$$\mathbf{X} = \mathbf{t} \star \mathbf{W} \sim \mathrm{ADSN}(\mathbf{u}) = \mathscr{N}(\mathbf{0}, \mathbf{\Gamma})$$ **which is a stationary law**

$\mathbf{\Gamma}$ represents the convolution by the kernel $\boldsymbol{\gamma} = \mathbf{t} \star \check{\mathbf{t}}$: $\mathbf{\Gamma}$ can be stored easily.

$\mathbf{u}$

Samples of $\mathrm{ADSN}(\mathbf{u})$

Image extracted from [Galerne et al., 2011a][13]

[13] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm_rpn

[14] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [14]

Let $\mathbf{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{u} - m)$ its associated texton. Let $\mathbf{W}$ be a white Gaussian noise,

$$\mathbf{X} = \mathbf{t} \star \mathbf{W} \sim \mathrm{ADSN}(\mathbf{u}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}) \quad \textbf{which is a stationary law}$$

$\boldsymbol{\Gamma}$ represents the convolution by the kernel $\boldsymbol{\gamma} = \mathbf{t} \star \check{\mathbf{t}}$: $\boldsymbol{\Gamma}$ can be stored easily.

$\mathbf{u}$

Samples of $\mathrm{ADSN}(\mathbf{u})$

Image extracted from [Galerne et al., 2011a][13]

[13] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm_rpn

[14] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [14]

Let $\mathbf{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{u} - m)$ its associated texton. Let $\mathbf{W}$ be a white Gaussian noise,

$$\mathbf{X} = \mathbf{t} \star \mathbf{W} \sim \mathrm{ADSN}(\mathbf{u}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}) \quad \textbf{which is a stationary law}$$

$\boldsymbol{\Gamma}$ represents the convolution by the kernel $\gamma = \mathbf{t} \star \check{\mathbf{t}}$: $\boldsymbol{\Gamma}$ can be stored easily.

$\mathbf{u}$

Samples of $\mathrm{ADSN}(\mathbf{u})$

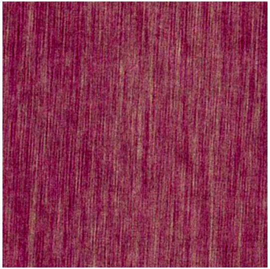


Image extracted from [Galerne et al., 2011a][13]

[13] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm_rpn

[14] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [14]

Let $\mathbf{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{u} - m)$ its associated texton. Let $\mathbf{W}$ be a white Gaussian noise,

$$\mathbf{X} = \mathbf{t} \star \mathbf{W} \sim \mathrm{ADSN}(\mathbf{u}) = \mathscr{N}(\mathbf{0}, \mathbf{\Gamma}) \quad \textbf{which is a stationary law}$$

$\mathbf{\Gamma}$ represents the convolution by the kernel $\gamma = \mathbf{t} \star \check{\mathbf{t}}$: $\mathbf{\Gamma}$ can be stored easily.

$\mathbf{u}$

Samples of $\mathrm{ADSN}(\mathbf{u})$

Image extracted from [Galerne et al., 2011a][13]

[13] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm_rpn

[14] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [14]

Let $\mathbf{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{u} - m)$ its associated texton. Let $\mathbf{W}$ be a white Gaussian noise,

$$\mathbf{X} = \mathbf{t} \star \mathbf{W} \sim \mathrm{ADSN}(\mathbf{u}) = \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}) \quad \textbf{which is a stationary law}$$

$\mathbf{\Gamma}$ represents the convolution by the kernel $\gamma = \mathbf{t} \star \check{\mathbf{t}}$: $\mathbf{\Gamma}$ can be stored easily.

$\mathbf{u}$

Samples of $\mathrm{ADSN}(\mathbf{u})$

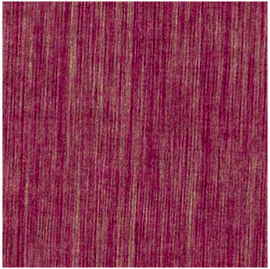


Image extracted from [Galerne et al., 2011a][13]

[13] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm_rpn

[14] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [14]

Let $\mathbf{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\mathbf{t} = \frac{1}{\sqrt{MN}}(\mathbf{u} - m)$ its associated texton. Let $\mathbf{W}$ be a white Gaussian noise,

$$\mathbf{X} = \mathbf{t} \star \mathbf{W} \sim \mathrm{ADSN}(\mathbf{u}) = \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}) \quad \textbf{which is a stationary law}$$

$\mathbf{\Gamma}$ represents the convolution by the kernel $\gamma = \mathbf{t} \star \check{\mathbf{t}}$: $\mathbf{\Gamma}$ can be stored easily.

$\mathbf{u}$

Samples of $\mathrm{ADSN}(\mathbf{u})$

Image extracted from [Galerne et al., 2011a][13]

[13] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line, 1.* https://doi.org/10.5201/ipol.2011.ggm_rpn

[14] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*

Stochastic SR for Gaussian microtextures

The conditional Gaussian simulation

We aim at sampling $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$, conditioned on $\mathbf{AX}$.

[15] Galerne, B., & Leclaire, A. (2017). Texture Inpainting Using Efficient Gaussian Conditional Simulation. *SIAM Journal on Imaging Sciences*

The conditional Gaussian simulation

We aim at sampling $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$, conditioned on $\mathbf{AX}$.

Following ideas from [Galerne and Leclaire, 2017][15], if $\tilde{\mathbf{X}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ is independent of $\mathbf{X}$, then:

$$\mathbb{E}(\mathbf{X}|\mathbf{AX}) + [\tilde{\mathbf{X}} - \mathbb{E}(\tilde{\mathbf{X}}|\mathbf{A}\tilde{\mathbf{X}})] \sim \mathbf{X}|\mathbf{AX}$$

[15] Galerne, B., & Leclaire, A. (2017). Texture Inpainting Using Efficient Gaussian Conditional Simulation. *SIAM Journal on Imaging Sciences*

The conditional Gaussian simulation

We aim at sampling $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$, conditioned on $\mathbf{AX}$.

Following ideas from [Galerne and Leclaire, 2017][15], if $\tilde{\mathbf{X}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ is independent of $\mathbf{X}$, then:

$$\mathbb{E}(\mathbf{X}|\mathbf{AX}) + [\tilde{\mathbf{X}} - \mathbb{E}(\tilde{\mathbf{X}}|\mathbf{A}\tilde{\mathbf{X}})] \sim \mathbf{X}|\mathbf{AX}$$

Furthermore, if $\mathbf{X}$ is zero-mean, there exists $\mathbf{\Lambda} \in \mathbb{R}^{\Omega_{M/r,N/r} \times \Omega_{M,N}}$ such that $\mathbb{E}(\mathbf{X}|\mathbf{AX}) = \mathbf{\Lambda}^T \mathbf{AX}$ and:

$$\mathbb{E}(\mathbf{X}|\mathbf{AX}) = \mathbf{\Lambda}^T \mathbf{AX} \iff \mathbf{A\Gamma A}^T \mathbf{\Lambda} = \mathbf{A\Gamma}.$$

[15] Galerne, B., & Leclaire, A. (2017). Texture Inpainting Using Efficient Gaussian Conditional Simulation. *SIAM Journal on Imaging Sciences*

The conditional Gaussian simulation

We aim at sampling $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$, conditioned on $\mathbf{AX}$.

Following ideas from [Galerne and Leclaire, 2017][15], if $\tilde{\mathbf{X}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ is independent of $\mathbf{X}$, then:

$$\mathbb{E}(\mathbf{X}|\mathbf{AX}) + [\tilde{\mathbf{X}} - \mathbb{E}(\tilde{\mathbf{X}}|\mathbf{A}\tilde{\mathbf{X}})] \sim \mathbf{X}|\mathbf{AX}$$

Furthermore, if $\mathbf{X}$ is zero-mean, there exists $\mathbf{\Lambda} \in \mathbb{R}^{\Omega_{M/r,N/r} \times \Omega_{M,N}}$ such that $\mathbb{E}(\mathbf{X}|\mathbf{AX}) = \mathbf{\Lambda}^T \mathbf{AX}$ and:

$$\mathbb{E}(\mathbf{X}|\mathbf{AX}) = \mathbf{\Lambda}^T \mathbf{AX} \iff \mathbf{A\Gamma A}^T \mathbf{\Lambda} = \mathbf{A\Gamma}.$$

Consequence: To sample $\mathbf{u}_{\mathrm{SR}} \sim \mathrm{ADSN}(\mathbf{u}) = \mathscr{N}(\mathbf{0}, \mathbf{\Gamma})$, conditioned on $\mathbf{Au}_{\mathrm{SR}} = \mathbf{u}_{\mathrm{LR}}$, we aim:

$$\mathbf{\Lambda}^T \mathbf{u}_{\mathrm{LR}} + (\tilde{\mathbf{u}} - \mathbf{\Lambda}^T \mathbf{A}\tilde{\mathbf{u}}) \quad \text{with } \tilde{\mathbf{u}} \sim \mathrm{ADSN}(\mathbf{u}) = \mathscr{N}(\mathbf{0}, \mathbf{\Gamma})$$

[15] Galerne, B., & Leclaire, A. (2017). Texture Inpainting Using Efficient Gaussian Conditional Simulation. *SIAM Journal on Imaging Sciences*

The Gaussian SR method

$$\mathbf{u}_{\mathrm{SR}} = \underbrace{\mathbf{\Lambda}^T \mathbf{u}_{\mathrm{LR}}}_{\text{Kriging component}} + \underbrace{\tilde{\mathbf{u}} - \mathbf{\Lambda}^T \mathbf{A} \tilde{\mathbf{u}}}_{\text{Innovation component}} \quad \text{with } \tilde{\mathbf{u}} \sim \mathrm{ADSN}(\mathbf{u}) = \mathscr{N}(\mathbf{0}, \mathbf{\Gamma})$$

with $\mathbf{\Lambda}$ verifying the kriging equation:

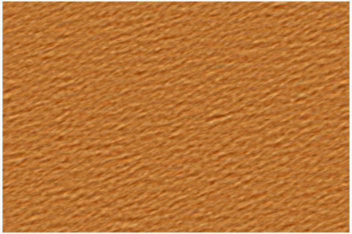
$$\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda} = \mathbf{A}\mathbf{\Gamma}. \qquad (1)$$

SR image: $\mathbf{u}_{\mathrm{SR}}$

=

Kriging component: $\mathbf{\Lambda}^T \mathbf{u}_{\mathrm{LR}}$

+



Innovation component: $\tilde{\mathbf{u}} - \mathbf{\Lambda}^T \mathbf{A} \tilde{\mathbf{u}}$

Image extracted from [Galerne et al., 2011a]

The Gaussian SR method

$$\mathbf{u}_{\mathrm{SR}} = \underbrace{\mathbf{\Lambda}^T \mathbf{u}_{\mathrm{LR}}}_{\text{Kriging component}} + \underbrace{\tilde{\mathbf{u}} - \mathbf{\Lambda}^T \mathbf{A} \tilde{\mathbf{u}}}_{\text{Innovation component}} \quad \text{with } \tilde{\mathbf{u}} \sim \mathrm{ADSN}(\mathbf{u}) = \mathscr{N}(\mathbf{0}, \mathbf{\Gamma})$$

with $\mathbf{\Lambda}$ verifying the kriging equation:

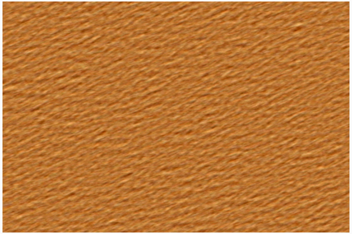
$$\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda} = \mathbf{A}\mathbf{\Gamma}. \tag{1}$$

SR image: $\mathbf{u}_{\mathrm{SR}}$

=

Kriging component: $\mathbf{\Lambda}^T \mathbf{u}_{\mathrm{LR}}$

+



Innovation component: $\tilde{\mathbf{u}} - \mathbf{\Lambda}^T \mathbf{A} \tilde{\mathbf{u}}$

Image extracted from [Galerne et al., 2011a]

The Gaussian SR method

$$\mathbf{u}_{\mathrm{SR}} = \underbrace{\boldsymbol{\Lambda}^T \mathbf{u}_{\mathrm{LR}}}_{\text{Kriging component}} + \underbrace{\tilde{\mathbf{u}} - \boldsymbol{\Lambda}^T \mathbf{A} \tilde{\mathbf{u}}}_{\text{Innovation component}} \quad \text{with } \tilde{\mathbf{u}} \sim \mathrm{ADSN}(\mathbf{u}) = \mathscr{N}(\mathbf{0}, \boldsymbol{\Gamma})$$

with $\boldsymbol{\Lambda}$ verifying the kriging equation:

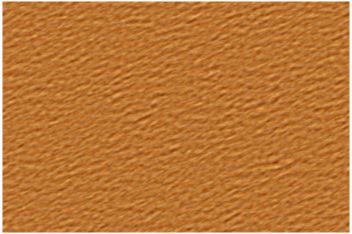
$$\mathbf{A} \boldsymbol{\Gamma} \mathbf{A}^T \boldsymbol{\Lambda} = \mathbf{A} \boldsymbol{\Gamma}. \tag{1}$$

SR image: $\mathbf{u}_{\mathrm{SR}}$

=

Kriging component: $\boldsymbol{\Lambda}^T \mathbf{u}_{\mathrm{LR}}$

+



Innovation component: $\tilde{\mathbf{u}} - \boldsymbol{\Lambda}^T \mathbf{A} \tilde{\mathbf{u}}$

Image extracted from [Galerne et al., 2011a]

The Gaussian SR method

$$\mathbf{u}_{\mathrm{SR}} = \underbrace{\boldsymbol{\Lambda}^T \mathbf{u}_{\mathrm{LR}}}_{\text{Kriging component}} + \underbrace{\tilde{\mathbf{u}} - \boldsymbol{\Lambda}^T \mathbf{A}\tilde{\mathbf{u}}}_{\text{Innovation component}} \quad \text{with } \tilde{\mathbf{u}} \sim \mathrm{ADSN}(\mathbf{u}) = \mathscr{N}(\mathbf{0}, \boldsymbol{\Gamma})$$

with $\boldsymbol{\Lambda}$ verifying the kriging equation:

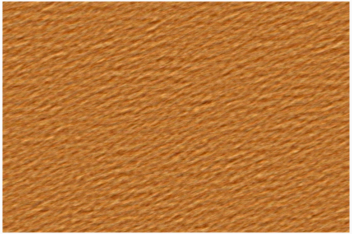
$$\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^T\boldsymbol{\Lambda} = \mathbf{A}\boldsymbol{\Gamma}. \tag{1}$$

SR image: $\mathbf{u}_{\mathrm{SR}}$

=

Kriging component: $\boldsymbol{\Lambda}^T \mathbf{u}_{\mathrm{LR}}$

+



Innovation component: $\tilde{\mathbf{u}} - \boldsymbol{\Lambda}^T \mathbf{A}\tilde{\mathbf{u}}$

Image extracted from [Galerne et al., 2011a]

$$\mathbf{u}_{\mathrm{SR}} = \underbrace{\mathbf{\Lambda}^T \mathbf{u}_{\mathrm{LR}}}_{\text{Kriging component}} + \underbrace{\tilde{\mathbf{u}} - \mathbf{\Lambda}^T \mathbf{A}\tilde{\mathbf{u}}}_{\text{Innovation component}} \quad \text{with } \tilde{\mathbf{u}} \sim \mathrm{ADSN}(\mathbf{u}) = \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$$

with $\mathbf{\Lambda}$ verifying the kriging equation:

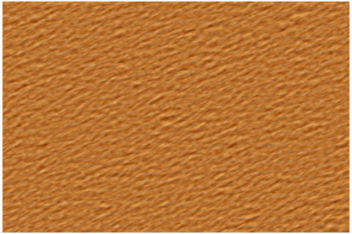
$$\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda} = \mathbf{A}\mathbf{\Gamma}. \tag{1}$$

SR image: $\mathbf{u}_{\mathrm{SR}}$

=

Kriging component: $\mathbf{\Lambda}^T \mathbf{u}_{\mathrm{LR}}$

+



Innovation component: $\tilde{\mathbf{u}} - \mathbf{\Lambda}^T \mathbf{A}\tilde{\mathbf{u}}$

Image extracted from [Galerne et al., 2011a]

The Gaussian SR method

$$\mathbf{u}_{\mathrm{SR}} = \underbrace{\mathbf{\Lambda}^T \mathbf{u}_{\mathrm{LR}}}_{\text{Kriging component}} + \underbrace{\tilde{\mathbf{u}} - \mathbf{\Lambda}^T \mathbf{A} \tilde{\mathbf{u}}}_{\text{Innovation component}} \quad \text{with } \tilde{\mathbf{u}} \sim \mathrm{ADSN}(\mathbf{u}) = \mathscr{N}(\mathbf{0}, \mathbf{\Gamma})$$

with $\mathbf{\Lambda}$ verifying the kriging equation:

$$\mathbf{A\Gamma A}^T \mathbf{\Lambda} = \mathbf{A\Gamma}. \tag{1}$$


SR image: $\mathbf{u}_{\mathrm{SR}}$

=


Kriging component: $\mathbf{\Lambda}^T \mathbf{u}_{\mathrm{LR}}$

+

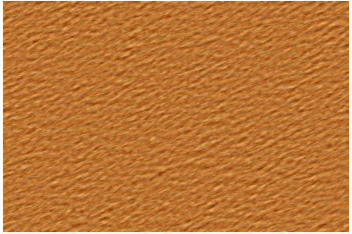

Innovation component: $\tilde{\mathbf{u}} - \mathbf{\Lambda}^T \mathbf{A} \tilde{\mathbf{u}}$

Image extracted from [Galerne et al., 2011a]

The Gaussian SR method

$$\mathbf{u}_{\mathrm{SR}} = \underbrace{\boldsymbol{\Lambda}^T \mathbf{u}_{\mathrm{LR}}}_{\text{Kriging component}} + \underbrace{\tilde{\mathbf{u}} - \boldsymbol{\Lambda}^T \mathbf{A} \tilde{\mathbf{u}}}_{\text{Innovation component}} \quad \text{with } \tilde{\mathbf{u}} \sim \mathrm{ADSN}(\mathbf{u}) = \mathscr{N}(\mathbf{0}, \boldsymbol{\Gamma})$$

with $\boldsymbol{\Lambda}$ verifying the kriging equation:

$$\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^T\boldsymbol{\Lambda} = \mathbf{A}\boldsymbol{\Gamma}. \tag{1}$$

Two problems:

1. Solving the kriging equation.
2. Storing $\boldsymbol{\Lambda} \in \mathbb{R}^{\Omega_{M/r,N/r} \times \Omega_{M,N}}$.

Solving the kriging equation

$$\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^T\boldsymbol{\Lambda} = \mathbf{A}\boldsymbol{\Gamma}. \tag{1}$$

Proposition 1: Kriging as a convolution

$\boldsymbol{\Lambda} = \left(\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^T\right)^\dagger \mathbf{A}\boldsymbol{\Gamma}$ is an exact solution of kriging Equation 1 and for all $\boldsymbol{v} \in \mathbb{R}^{\Omega_{M/r,N/r}}$,

$$\boldsymbol{\Lambda}^T\boldsymbol{v} = \boldsymbol{\lambda} \star (\mathbf{S}^T\boldsymbol{v}) \tag{2}$$

where $\boldsymbol{\lambda} = \mathbf{t} \star \check{\mathbf{t}} \star \check{\mathbf{c}} \star (\mathbf{S}^T\boldsymbol{\kappa}^\dagger)$ with $\boldsymbol{\kappa} = \mathbf{S}(\mathbf{t} \star \check{\mathbf{t}} \star \mathbf{c} \star \check{\mathbf{c}}) \in \mathbb{R}^{\Omega_{M/r,N/r}}$ and $\boldsymbol{\kappa}^\dagger$ the convolution kernel defined in Fourier domain by

$$\widehat{\boldsymbol{\kappa}}^\dagger(\omega) = \begin{cases} \frac{1}{\widehat{\boldsymbol{\kappa}}(\omega)} & \text{if } \widehat{\boldsymbol{\kappa}}(\omega) \neq 0, \\ 0 & \text{otherwise,} \end{cases} \qquad \omega \in \mathbb{R}^{\Omega_{M/r,N/r}}.$$

Summary

- We know $\mathbf{\Gamma}$ such that $\mathbf{u}_{\mathrm{HR}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and $\mathbf{u}_{\mathrm{LR}} = \mathbf{A}\mathbf{u}_{\mathrm{HR}}$

Summary

- We know $\mathbf{\Gamma}$ such that $\mathbf{u}_{\mathrm{HR}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and $\mathbf{u}_{\mathrm{LR}} = \mathbf{A}\mathbf{u}_{\mathrm{HR}}$
- We can easily compute and store $\mathbf{\Lambda}$ such that $\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda} = \mathbf{A}\mathbf{\Gamma}$

Summary

- We know $\mathbf{\Gamma}$ such that $\mathbf{u}_{\mathrm{HR}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and $\mathbf{u}_{\mathrm{LR}} = \mathbf{A}\mathbf{u}_{\mathrm{HR}}$
- We can easily compute and store $\mathbf{\Lambda}$ such that $\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda} = \mathbf{A}\mathbf{\Gamma}$
- We can simulate SR samples $\mathbf{u}_{\mathrm{SR}}$ compatible with $\mathbf{u}_{\mathrm{LR}}$ computing $\tilde{\mathbf{u}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and

$$\mathbf{u}_{\mathrm{SR}} = \mathbf{\Lambda}^T\mathbf{u}_{\mathrm{LR}} + \tilde{\mathbf{u}} - \mathbf{\Lambda}^T\mathbf{A}\tilde{\mathbf{u}}$$

Summary

- We know $\boldsymbol{\Gamma}$ such that $\mathbf{u}_{\mathrm{HR}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$ and $\mathbf{u}_{\mathrm{LR}} = \mathbf{A}\mathbf{u}_{\mathrm{HR}}$
- We can easily compute and store $\boldsymbol{\Lambda}$ such that $\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^T\boldsymbol{\Lambda} = \mathbf{A}\boldsymbol{\Gamma}$
- We can simulate SR samples $\mathbf{u}_{\mathrm{SR}}$ compatible with $\mathbf{u}_{\mathrm{LR}}$ computing $\tilde{\mathbf{u}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$ and

$$\mathbf{u}_{\mathrm{SR}} = \boldsymbol{\Lambda}^T\mathbf{u}_{\mathrm{LR}} + \tilde{\mathbf{u}} - \boldsymbol{\Lambda}^T\mathbf{A}\tilde{\mathbf{u}}$$

Problem: $\boldsymbol{\Gamma}$ is extracted from $\mathbf{u}_{\mathrm{HR}}$

Summary

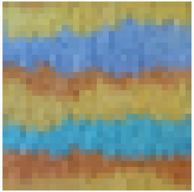
- We know $\mathbf{\Gamma}$ such that $\mathbf{u}_{\mathrm{HR}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and $\mathbf{u}_{\mathrm{LR}} = \mathbf{A}\mathbf{u}_{\mathrm{HR}}$
- We can easily compute and store $\mathbf{\Lambda}$ such that $\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda} = \mathbf{A}\mathbf{\Gamma}$
- We can simulate SR samples $\mathbf{u}_{\mathrm{SR}}$ compatible with $\mathbf{u}_{\mathrm{LR}}$ computing $\tilde{\mathbf{u}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and

$$\mathbf{u}_{\mathrm{SR}} = \mathbf{\Lambda}^T\mathbf{u}_{\mathrm{LR}} + \tilde{\mathbf{u}} - \mathbf{\Lambda}^T\mathbf{A}\tilde{\mathbf{u}}$$

Problem: $\mathbf{\Gamma}$ is extracted from $\mathbf{u}_{\mathrm{HR}}$

Solution : Use a reference image with the same correlation information.

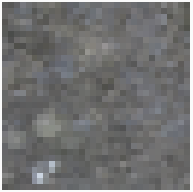
LR image



HR image

Reference image

LR image



HR image

Reference image

Algorithm

- **Input:** An image $\mathbf{u}_{\mathrm{LR}} \in \mathbb{R}^{\Omega_{M/r,N/r}}$, r the zoom factor, $\mathbf{t}$ the convolution kernel of the ADSN model, $\mathbf{c}$ the kernel of the convolution of the zoom-out operator $\mathbf{A} = \mathbf{S}\mathbf{C}_{\mathbf{c}}$.
- **Preprocessing:**
- Compute the grayscale mean m from $\mathbf{u}_{\mathrm{LR}}$ and set $\mathbf{u}_{\mathrm{LR}} := \mathbf{u}_{\mathrm{LR}} - m\mathbf{1}_{\Omega_{M,N}}$
- **Step 1: Computation of the kriging kernel**
- Store the DFT transform of the kernel $\boldsymbol{\lambda} = \mathbf{t} \star \check{\mathbf{t}} \star \check{\mathbf{c}} \star \mathbf{S}^T(\boldsymbol{\kappa}^\dagger)$
- **Step 2: Simulation of $\mathbf{u}_{\mathrm{SR}}$**
- Sample $\tilde{\mathbf{u}} = \mathbf{t} \star \boldsymbol{w}$ where $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{\Omega_{M,N}})$
- Compute $\mathbf{u}_{\mathrm{SR}} = \boldsymbol{\lambda} \star \mathbf{S}^T(\mathbf{u}_{\mathrm{LR}} - \mathbf{A}\tilde{\mathbf{u}}) + \tilde{\mathbf{u}}$
- **Postprocessing:**
- **Output:** $m\mathbf{1}_{\Omega_{M,N}} + \mathbf{u}_{\mathrm{SR}}$

Examples

LR image | HR image | Reference image | Sample | Kriging | Innovation

HR size is 208×208 and $r = 8$.

Too small ? (27)

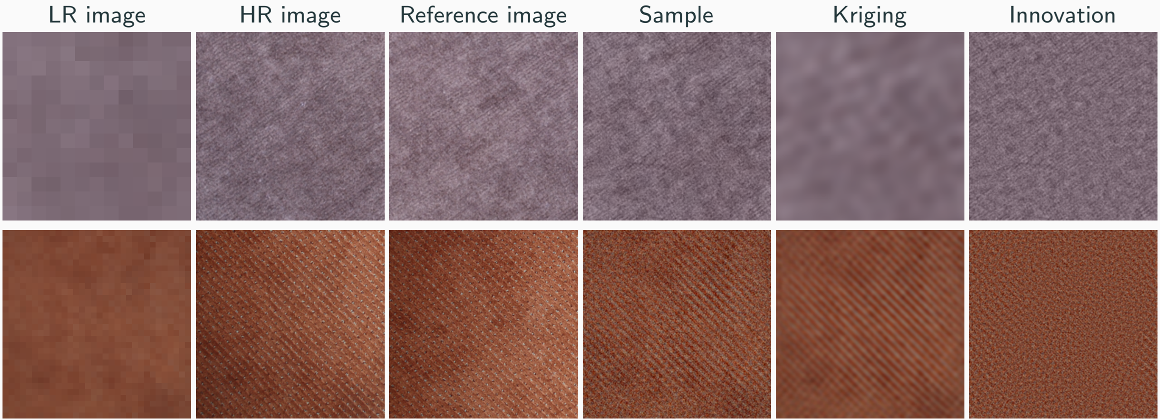
Examples

LR image | HR image | Reference image | Sample | Kriging | Innovation

HR size is 208×208 and $r = 8$.

Too small ? (27)

Examples



HR size is 208×208 and $r = 8$.

Too small ? (27)

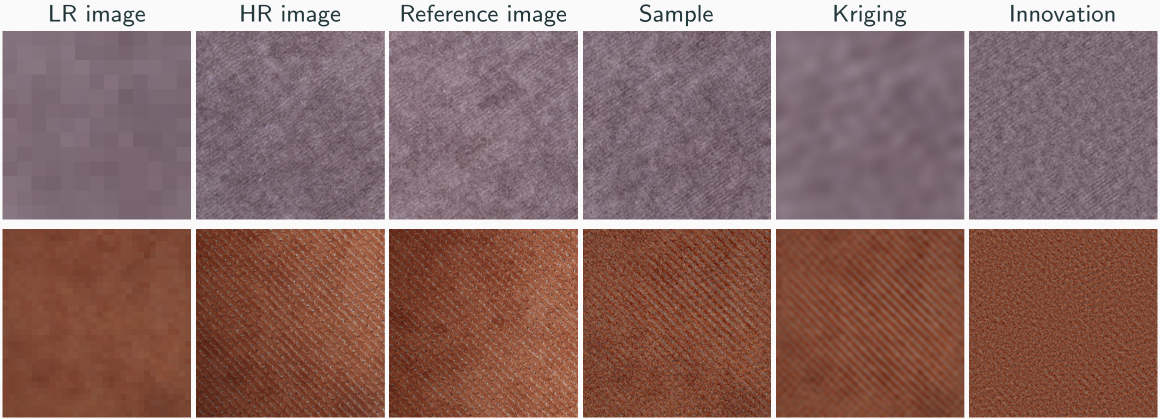
Examples

LR image | HR image | Reference image | Sample | Kriging | Innovation

HR size is 208×208 and $r = 8$.

Too small ? (27)

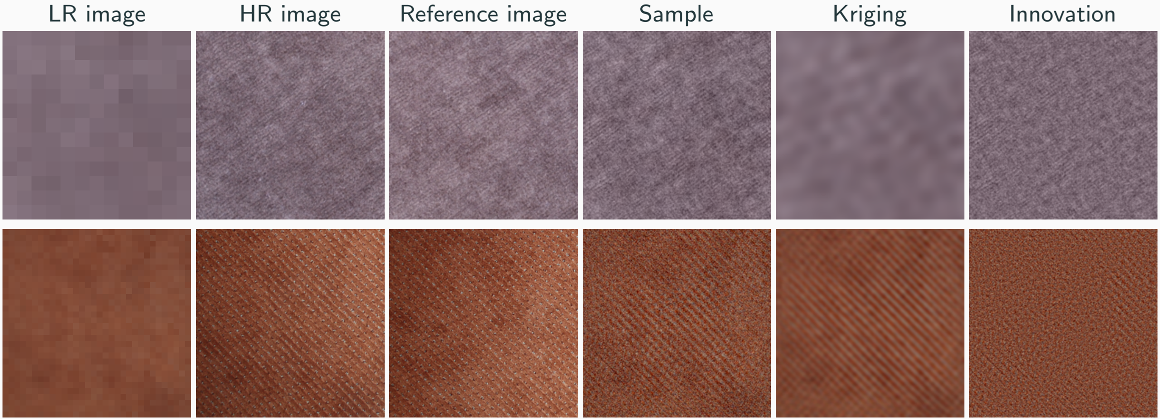
Examples



HR size is 208×208 and $r = 8$.

Too small ? (27)

Examples

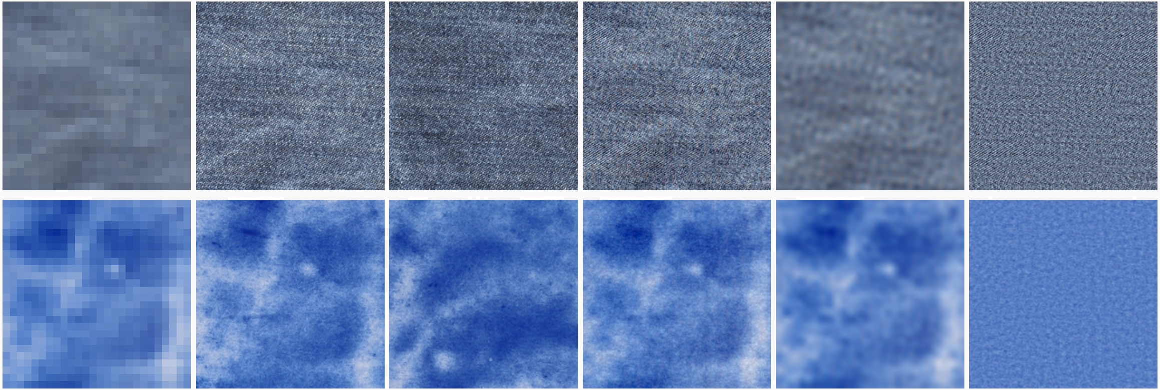


HR size is 208×208 and $r = 8$.

Too small ? (27)

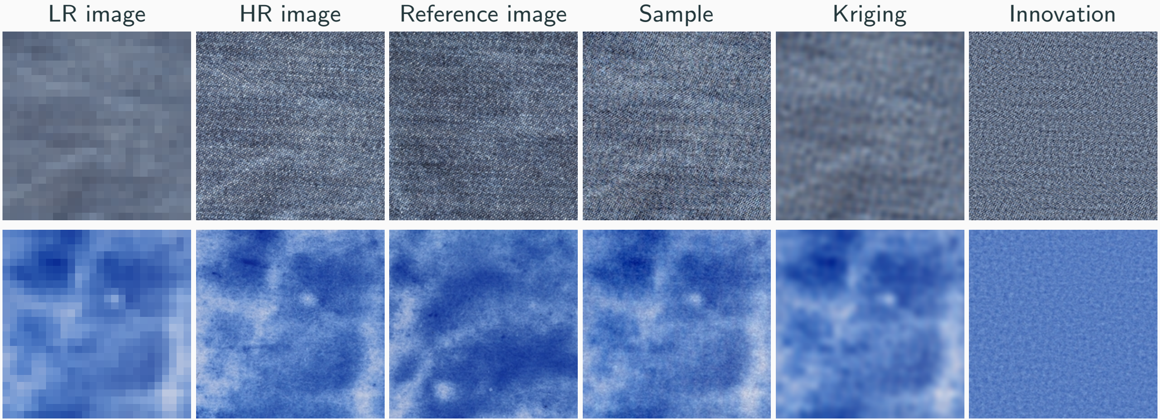
Examples

LR image | HR image | Reference image | Sample | Kriging | Innovation



HR size is 208×208 and $r = 8$.

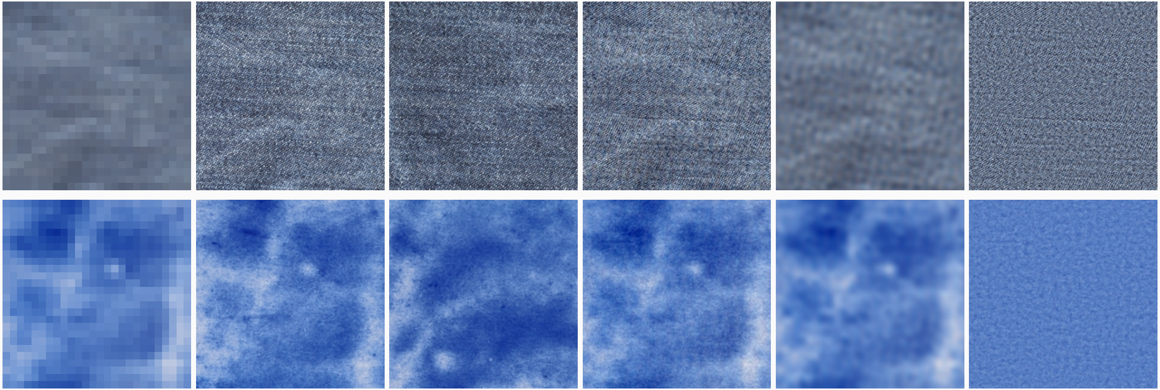
Examples



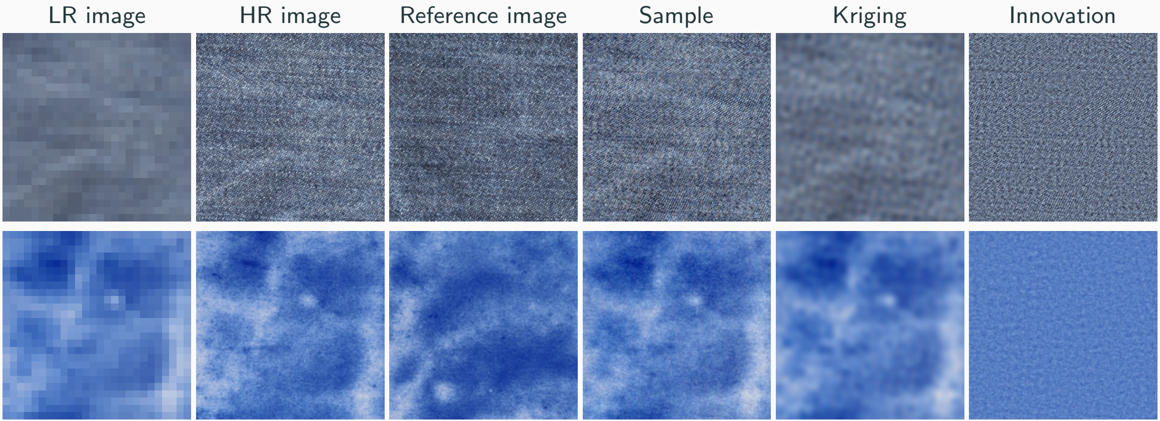
HR size is 208×208 and $r = 8$.

Examples

LR image | HR image | Reference image | Sample | Kriging | Innovation

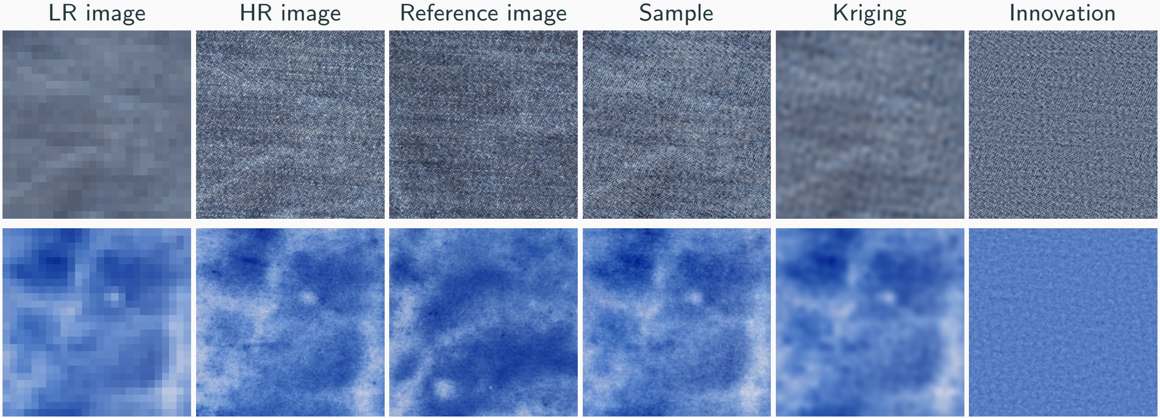


HR size is 208×208 and $r = 8$.



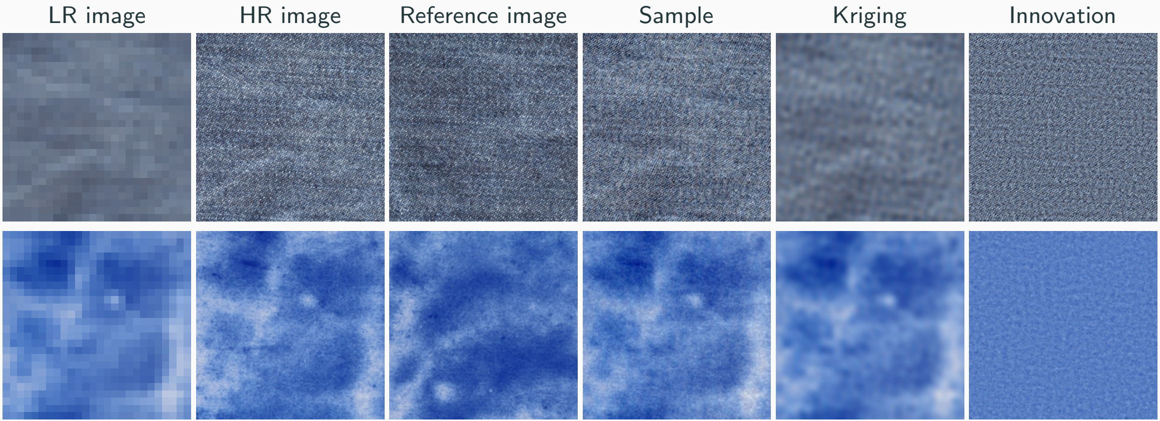
HR size is 208×208 and $r = 8$.

Examples



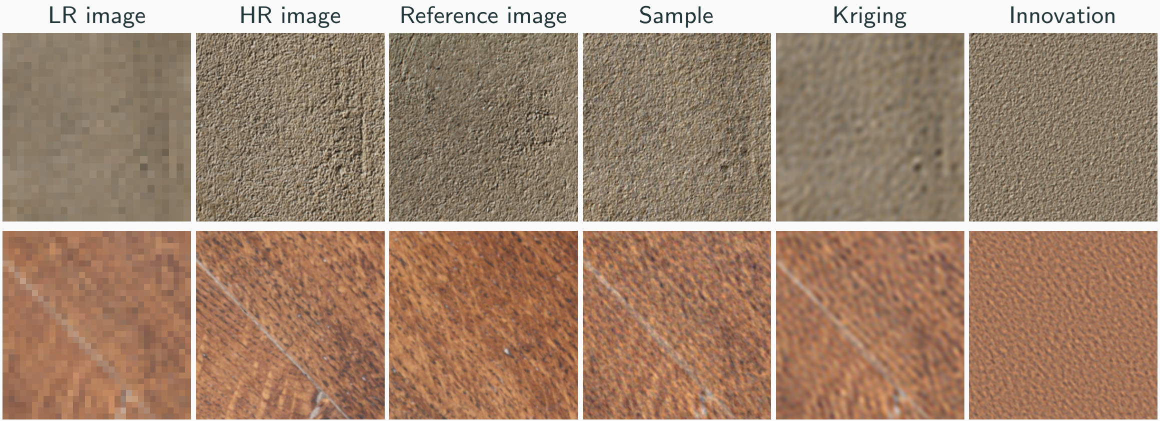
HR size is 208×208 and $r = 8$.

Examples



HR size is 208×208 and $r = 8$.

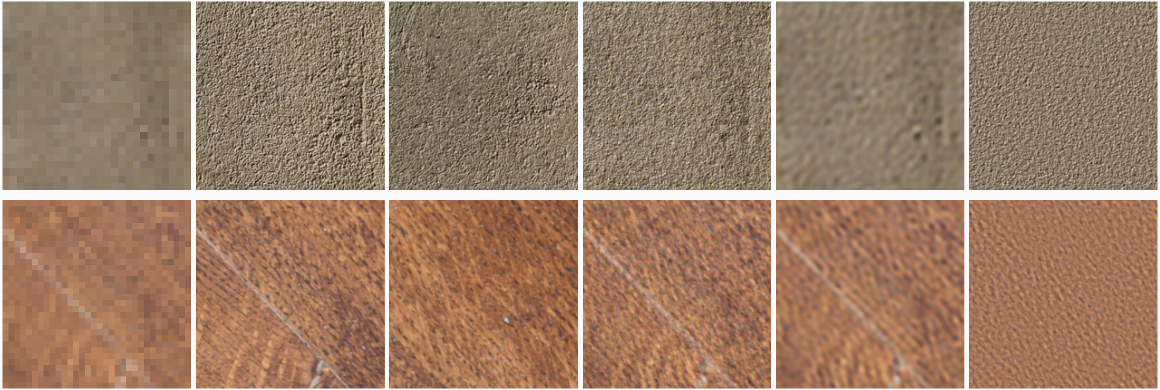
Examples



HR size is 208×208 and $r = 8$.

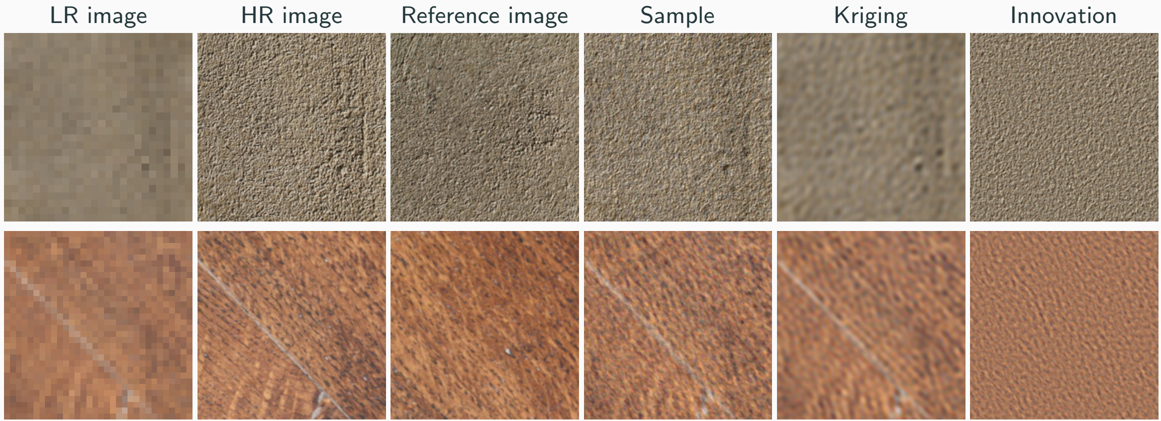
Examples

LR image | HR image | Reference image | Sample | Kriging | Innovation



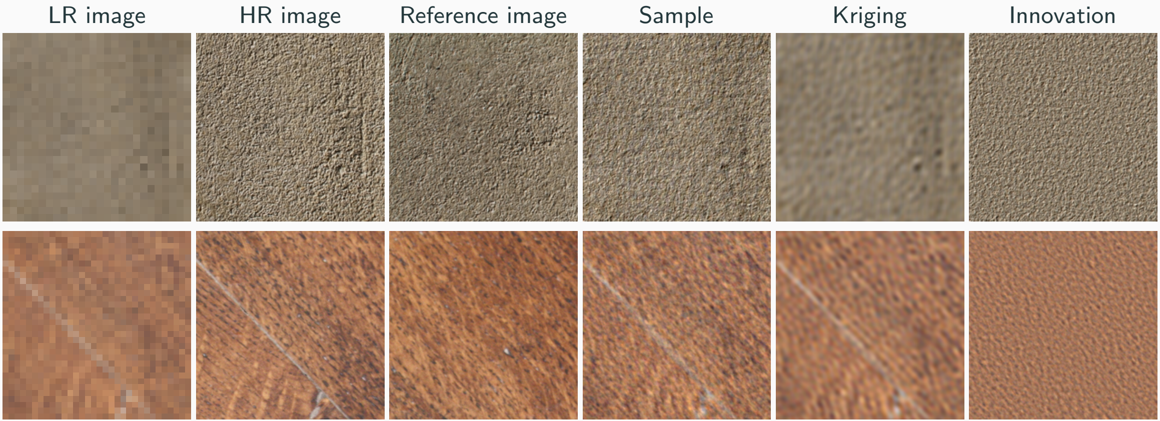
HR size is 208×208 and $r = 8$.

Examples



HR size is 208×208 and $r = 8$.

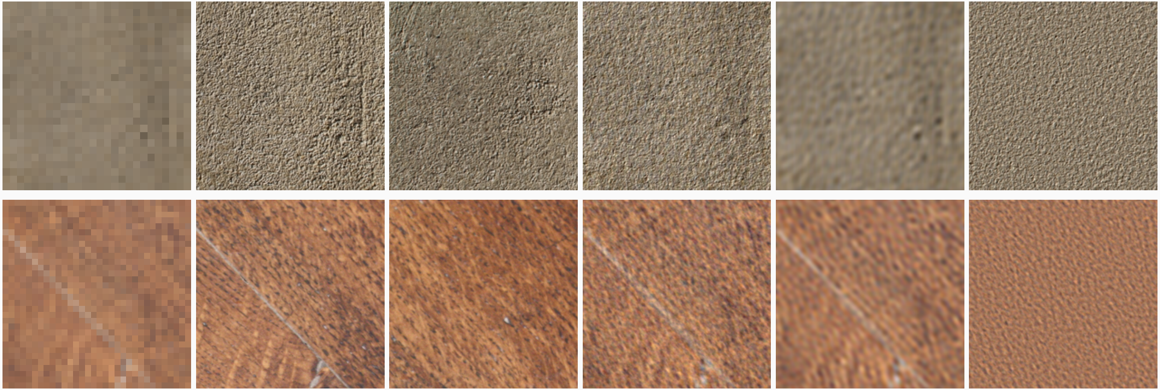
Examples



HR size is 208×208 and $r = 8$.

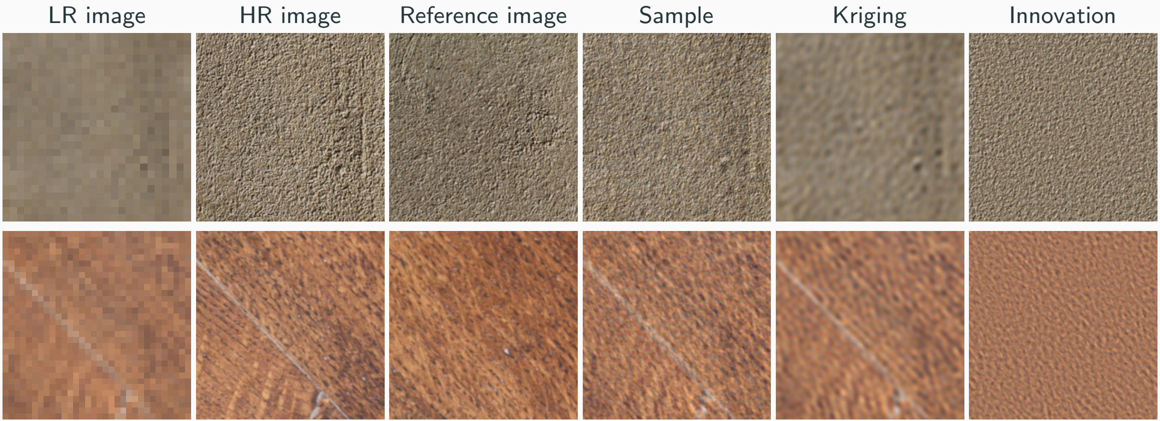
Examples

LR image | HR image | Reference image | Sample | Kriging | Innovation



HR size is 208×208 and $r = 8$.

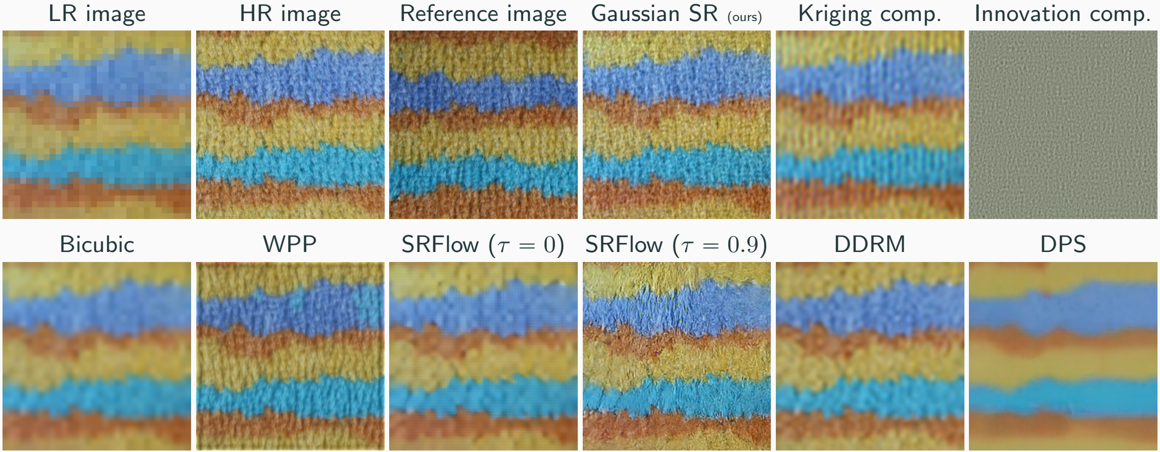
Examples



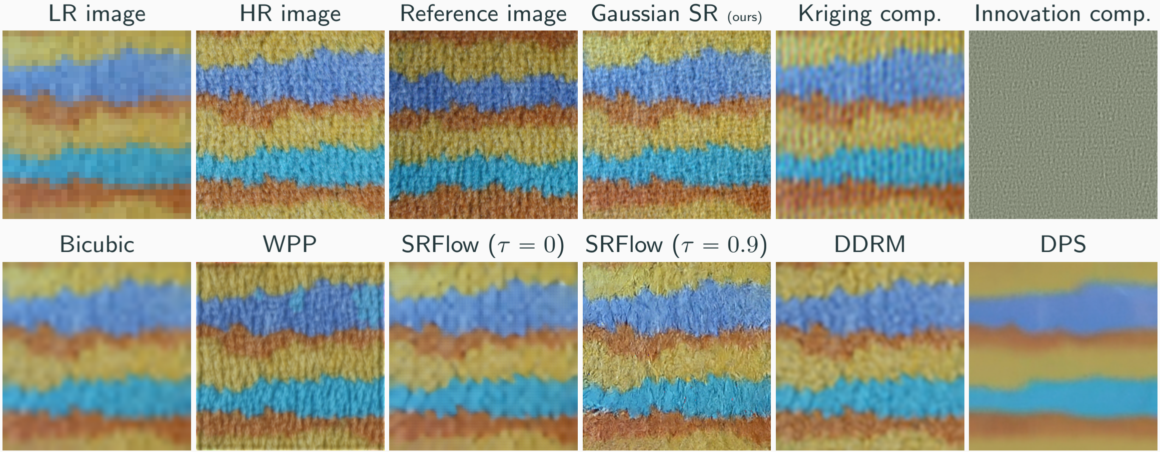
HR size is 208×208 and $r = 8$.

Comparison of Gaussian SR with other methods

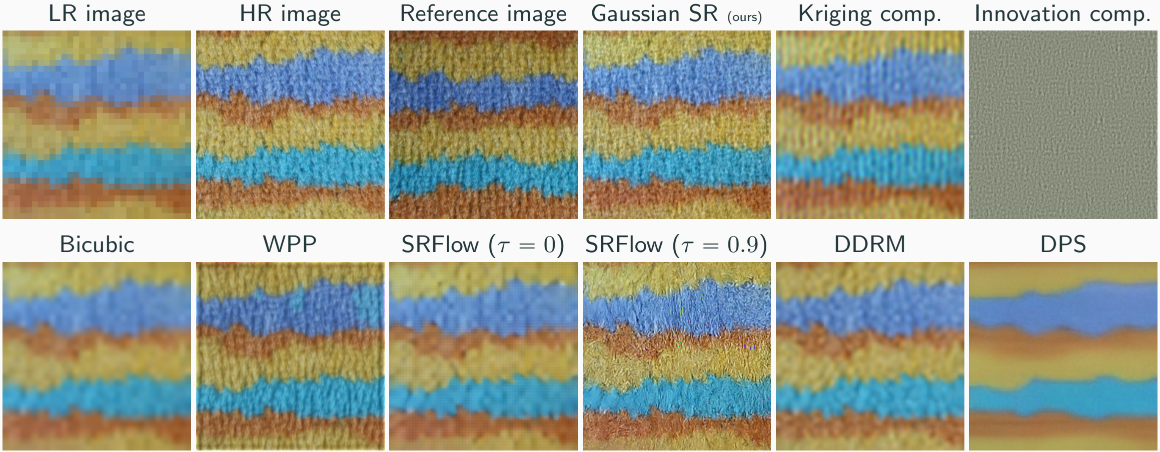
Comparison with other methods



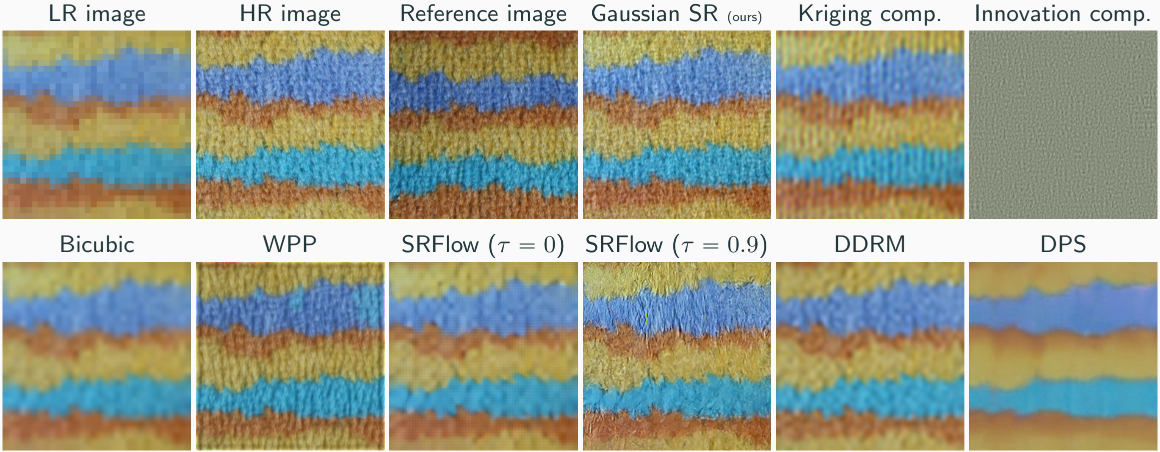
Comparison with other methods



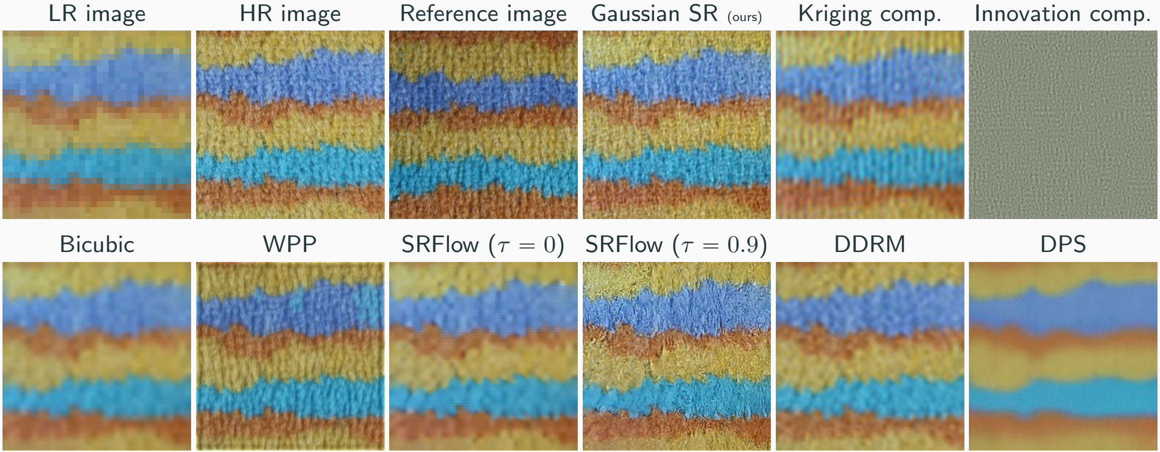
Comparison with other methods



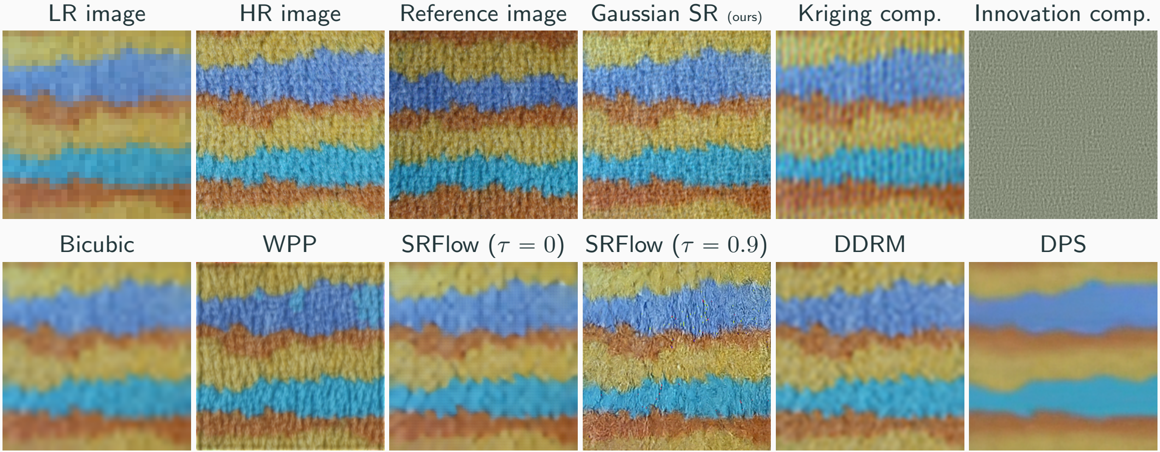
Comparison with other methods



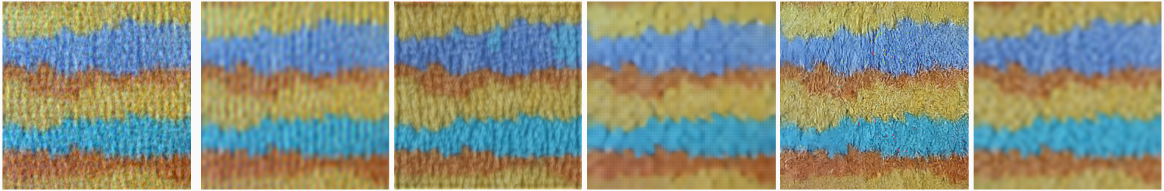
Comparison with other methods



Comparison with other methods

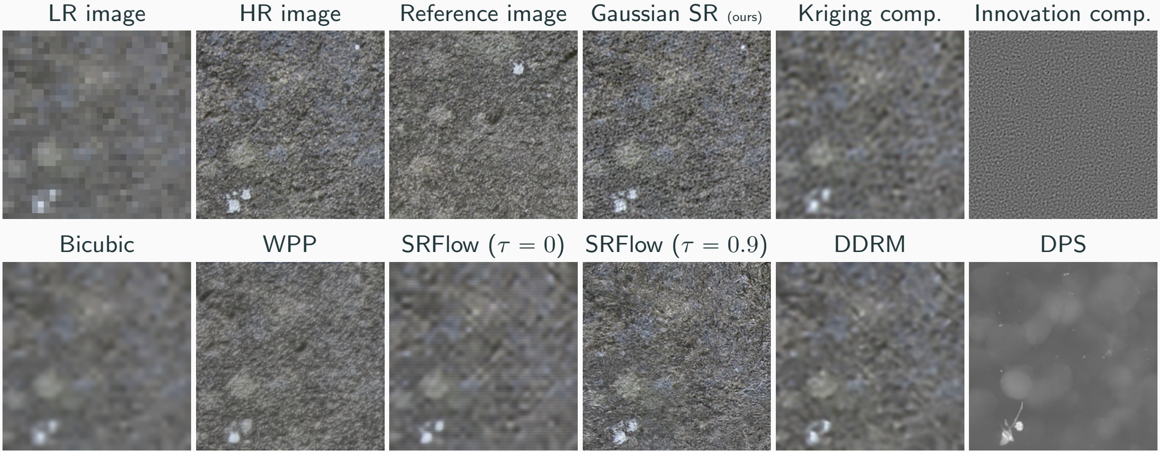


Gaussian SR (ours) | Kriging comp. | WPP | SRFlow $(\tau = 0)$ | SRFlow $(\tau = 0.9)$ | DDRM

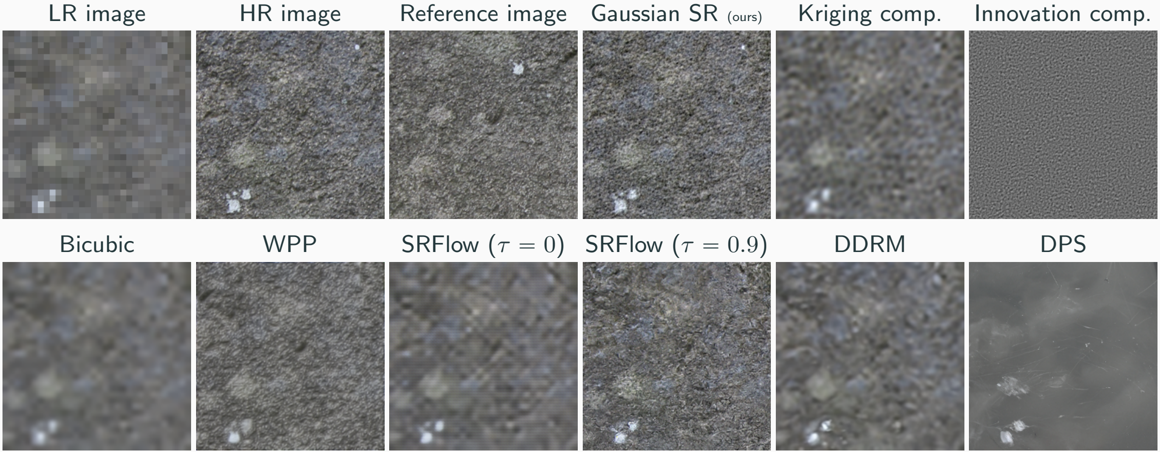


	PSNR ↑	LR-PSNR ↑	SSIM ↑	LPIPS ↓	Time
Gaussian SR (ours)	17.05±0.04	159.24±0.04	0.08±0.01	**0.22±0.01**	**0.01** (CPU)
Kriging comp.	18.76	**159.30**	0.11	0.75	-
WPP	20.04	29.29	0.14	0.36	44.0 (GPU)
SRFlow $(\tau = 0)$	21.44	54.61	**0.20**	0.70	0.22 (GPU)
SRFlow $(\tau = 0.9)$	18.29±0.36	55.13±0.15	0.12±0.01	0.30±0.03	0.19 (GPU)
DDRM	**21.57±0.05**	56.59±0.17	**0.20±0.00**	0.70±0.02	1.66 (GPU)

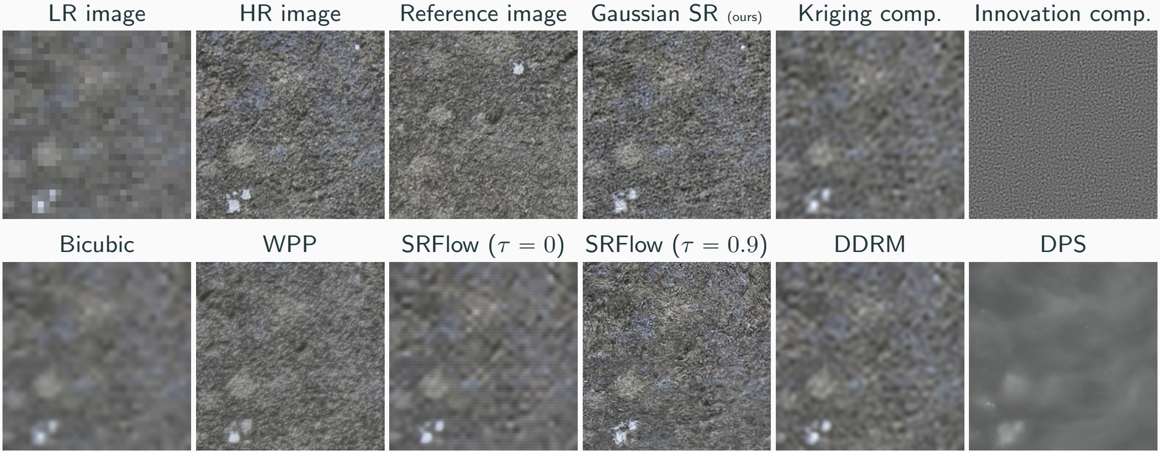
Comparison with other methods



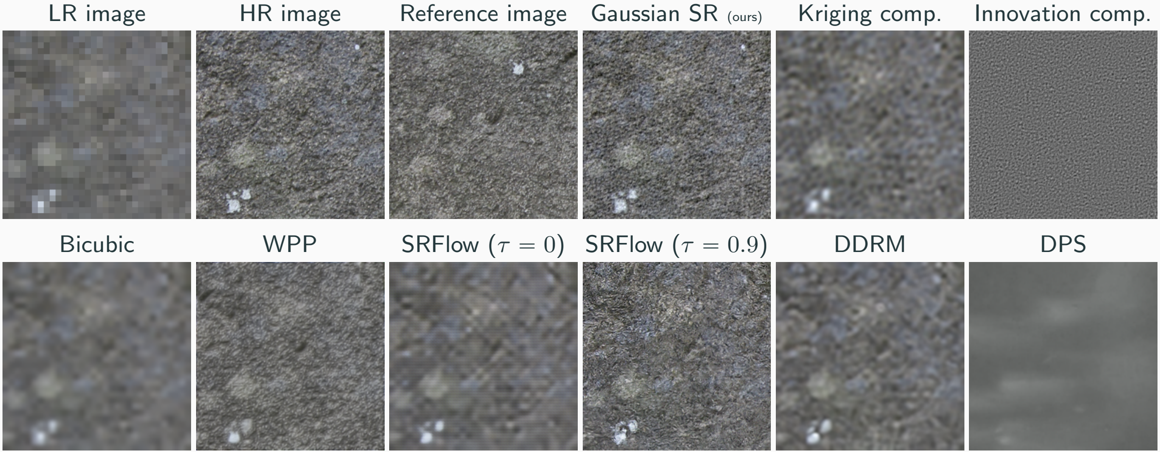
Comparison with other methods



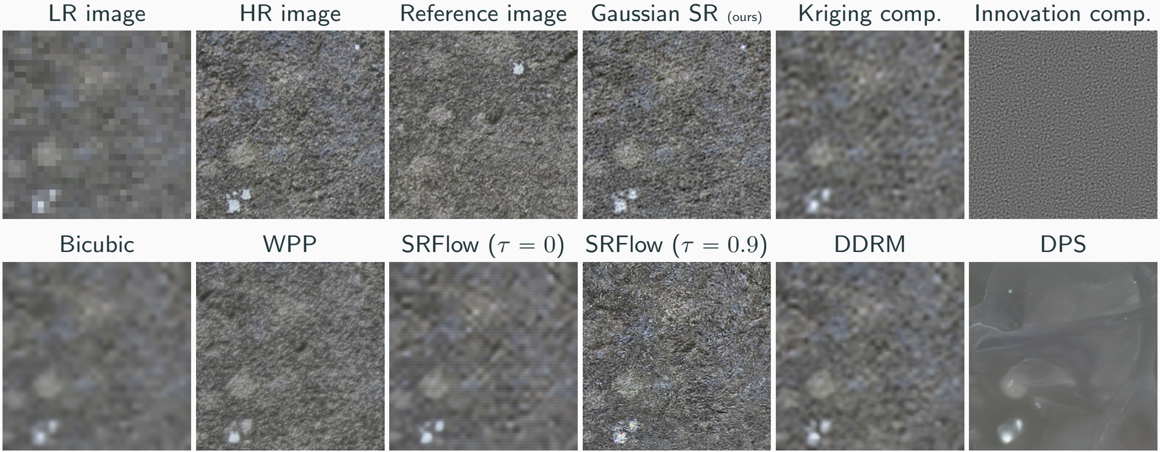
Comparison with other methods

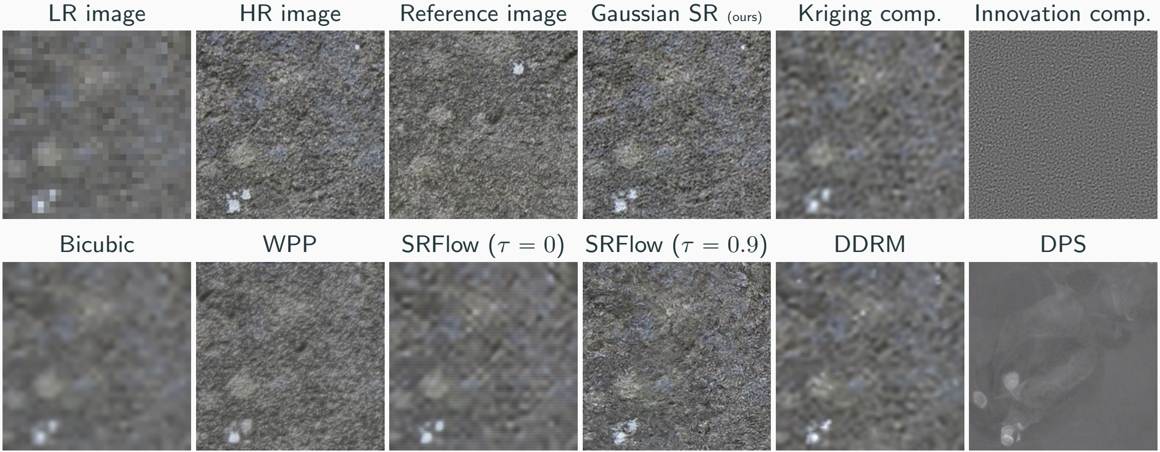


Comparison with other methods

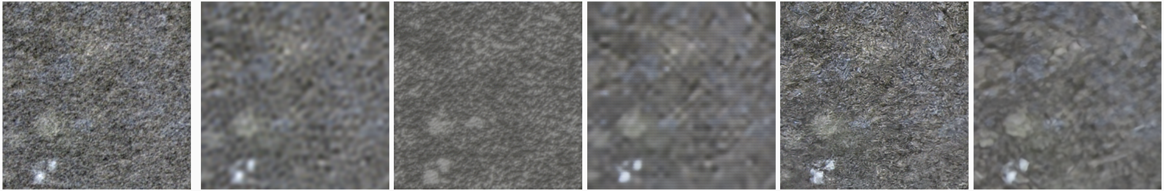


Comparison with other methods




LR image
HR image
Reference image
Gaussian SR (ours)
Kriging comp.
Innovation comp.
Bicubic
WPP
SRFlow ($\tau = 0$)
SRFlow ($\tau = 0.9$)
DDRM
DPS

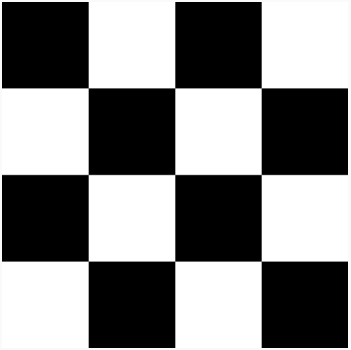
Comparison with other methods



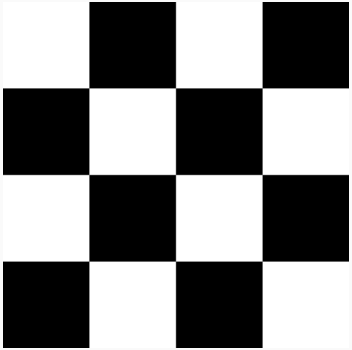
	PSNR ↑	LR-PSNR ↑	SSIM ↑	LPIPS ↓	Time
Gaussian SR (ours)	19.14±0.09	**154.52±0.36**	0.20±0.01	**0.23±0.01**	**0.01** (CPU)
Kriging comp.	21.42	154.47	**0.30**	0.52	-
WPP	17.68	18.84	0.19	0.30	64.0 (GPU)
SRFlow ($\tau = 0$)	21.64	51.63	0.29	0.54	0.23 (GPU)
SRFlow ($\tau = 0.9$)	18.21±0.53	54.02±0.23	0.16±0.02	0.39±0.06	0.20 (GPU)
DDRM	**22.44±0.04**	56.02±0.20	**0.30±0.00**	0.55±0.01	1.68 (GPU)

Metrics: a toy example

Checkerboard 1



Checkerboard 2

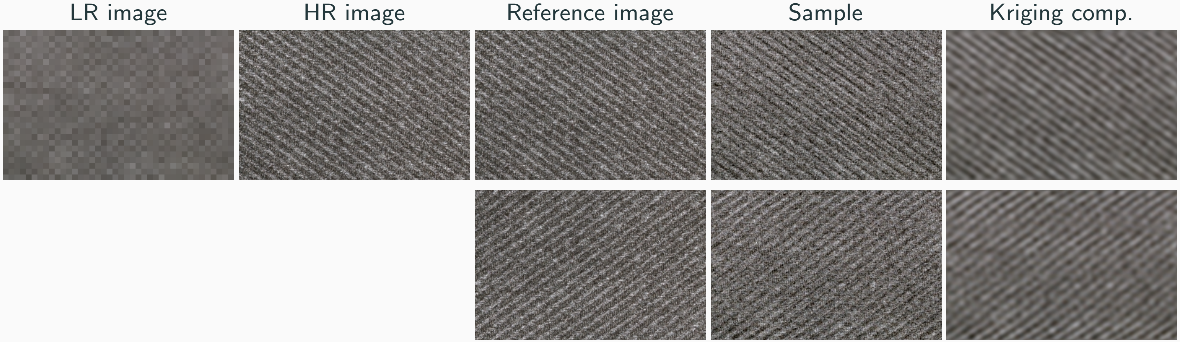


Mean image

Metrics w.r.t Checkerboard 1		
PSNR (dB) (↑)	0	$\mathbf{6.02}$
SSIM (↑)	-0.59	$\mathbf{0.04}$
LPIPS (↓)	$\mathbf{0.46}$	0.91

Limitations of the method

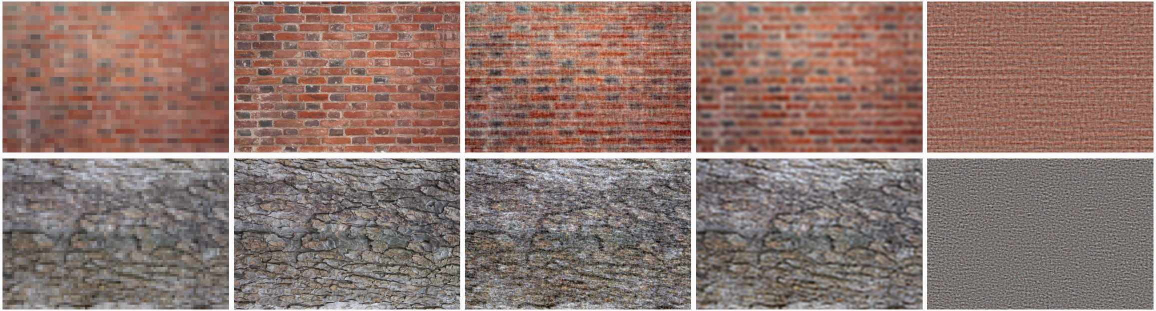
Reference choice



- The kriging component is computed using the covariance extracted from the reference image.

Structured textures

LR image $\mathbf{u}_{\mathrm{LR}}$ | HR image $\mathbf{u}_{\mathrm{HR}}$ | SR Gaussian sample $\mathbf{u}_{\mathrm{SR}}$ | Kriging comp. $\mathbf{\Lambda}^T\mathbf{u}_{\mathrm{LR}}$ | Innovation comp. $\tilde{\mathbf{u}} - \mathbf{\Lambda}^T\mathbf{A}\tilde{\mathbf{u}}$



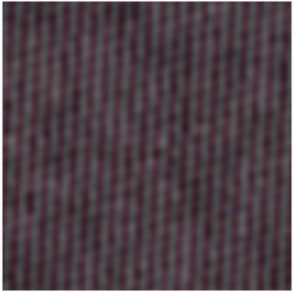
→ Not stationary textures

Instability case

HR image
$\mathbf{u}_{\mathrm{HR}}$

Reference
$\mathbf{u}_{\mathrm{ref}}$

Kriging comp.
$\mathbf{\Lambda}^T \mathbf{u}_{\mathrm{LR}}$



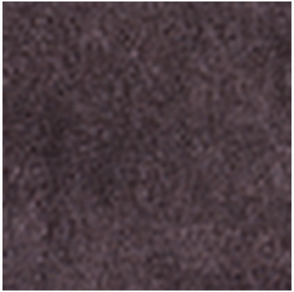
Sample
$\mathbf{u}_{\mathrm{SR}}$

Instability case

HR image $\mathbf{u}_{\mathrm{HR}}$

Reference $\mathbf{u}_{\mathrm{ref}}$

Kriging comp. $\boldsymbol{\Lambda}^T \mathbf{u}_{\mathrm{LR}}$



Sample $\mathbf{u}_{\mathrm{SR}}$

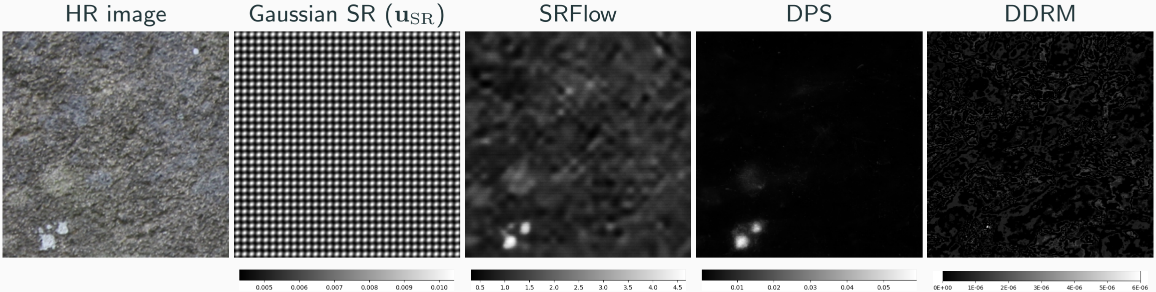
Instability case

Proposition 2: Stability of the kriging operator on the subspace of the LR ADSN samples

Let $\mathbf{\Lambda}^T = \mathbf{\Gamma}\mathbf{A}^T \left(\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\right)^\dagger \in \mathbb{R}^{\Omega_{M,N} \times \Omega_{M/r,N/r}}$. Then,

$$\forall \mathbf{W} \in \mathbb{R}^{\Omega_{M,N}}, \left\| \mathbf{\Lambda}^T \mathbf{A}(t \star \mathbf{W}) \right\|_2 \leqslant \|\mathbf{C}_\mathbf{t}\|_2 \|\mathbf{W}\|_2 \leqslant \|\mathbf{t}\|_1 \|\mathbf{W}\|_2. \quad (3)$$

- This proposition ensures the stability when the covariance matrix is extracted from the HR image.
- In practice, cases of instability are rare.



→ No adaptative variance.

In fact, $\mathbf{u}_{\mathrm{SR}}$ follows a cyclostationary texture law [Lutz et al., 2021][16].

[16] Lutz, N., Sauvage, B., & Dischler, J.-M. (2021). Cyclostationary gaussian noise: Theory and synthesis. *Computer Graphics Forum, 40*, xx–yy

Extension

Extension to blur operators

The previous method only uses

$$\mathbf{A} = \mathbf{S}\mathbf{C} \tag{4}$$

$\rightarrow$ $\mathbf{C}$ can be any convolution operator.

Motion blur followed by a subsampling operator with stride $r = 4$

Kernel | Degraded image | Ground truth | Reference | Kriging comp. | Innovation comp. | Sample

Conclusion

Our method has a limited scope but it is well-posed mathematically and provides an efficient sampler.

Lessons from this particular case:

- Discussion about the metrics.
- Inability of deep learning models to achieve SR of textures.

Limitations:

- The SR is achieved in the noiseless case.
- The stationarity assumption is very strong: details are affected in the SR samples.

Prepprint: Pierret, E., & Galerne, B. (2024). Stochastic super-resolution for gaussian microtextures.
https://arxiv.org/abs/2405.15399

Conclusion

Our method has a limited scope but it is well-posed mathematically and provides an efficient sampler.

Lessons from this particular case:

- Discussion about the metrics.
- Inability of deep learning models to achieve SR of textures.

Limitations:

- The SR is achieved in the noiseless case.
- The stationarity assumption is very strong: details are affected in the SR samples.

Prepprint: Pierret, E., & Galerne, B. (2024). Stochastic super-resolution for gaussian microtextures. https://arxiv.org/abs/2405.15399

Thank you for your attention !

References

Bahat, Y., & Michaeli, T. (2020). Explorable super resolution. *CVPR*.

Bruna, J., Sprechmann, P., & LeCun, Y. (2016). Super-Resolution with Deep Convolutional Sufficient Statistics. *ICLR 2016*.

Chatillon, P., Gousseau, Y., & Lefebvre, S. (2022). A statistically constrained internal method for single image super-resolution. *ICPR 2022*.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., & Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. *The Eleventh International Conference on Learning Representations*.

Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*, *1*. https://doi.org/10.5201/ipol.2011.ggm_rpn

Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*.

Galerne, B., & Leclaire, A. (2017). Texture Inpainting Using Efficient Gaussian Conditional Simulation. *SIAM Journal on Imaging Sciences*.

Hertrich, J., Houdard, A., & Redenbach, C. (2022). Wasserstein Patch Prior for Image Superresolution. *IEEE Transactions on Computational Imaging*.

Hertrich, J., Nguyen, L. D. P., Aujol, J.-F., Bernard, D., Berthoumieu, Y., Saadaldin, A., & Steidl, G. (2022). PCA Reduced Gaussian Mixture Models with Applications in Superresolution. *Inverse Problems and Imaging*.

Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. *ECCV 2016*.

Kawar, B., Elad, M., Ermon, S., & Song, J. (2022). Denoising diffusion restoration models. *ICLR Workshop on Deep Generative Models for Highly Structured Data*.

Ledig, C., Theis, L., & Huszár, F. e. a. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *CVPR 2017*.

Lugmayr, A., Danelljan, M., Van Gool, L., & Timofte, R. (2020). SRFlow: Learning the Super-Resolution Space with Normalizing Flow. *ECCV 2020*.

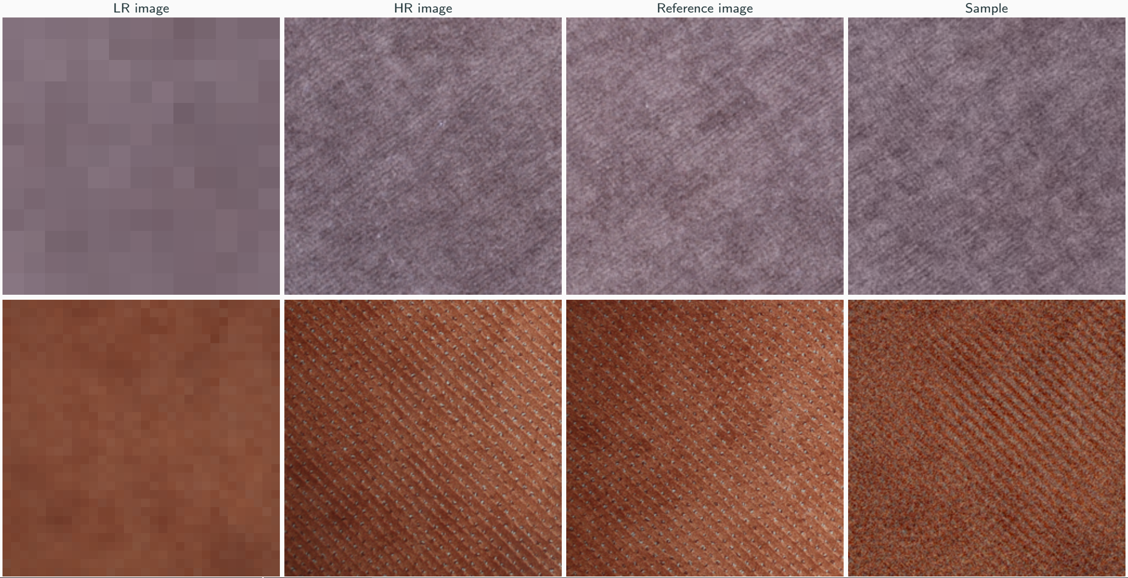
Lutz, N., Sauvage, B., & Dischler, J.-M. (2021). Cyclostationary gaussian noise: Theory and synthesis. *Computer Graphics Forum*, *40*, xx–yy.

Pierret, E., & Galerne, B. (2024). Stochastic super-resolution for gaussian microtextures. https://arxiv.org/abs/2405.15399

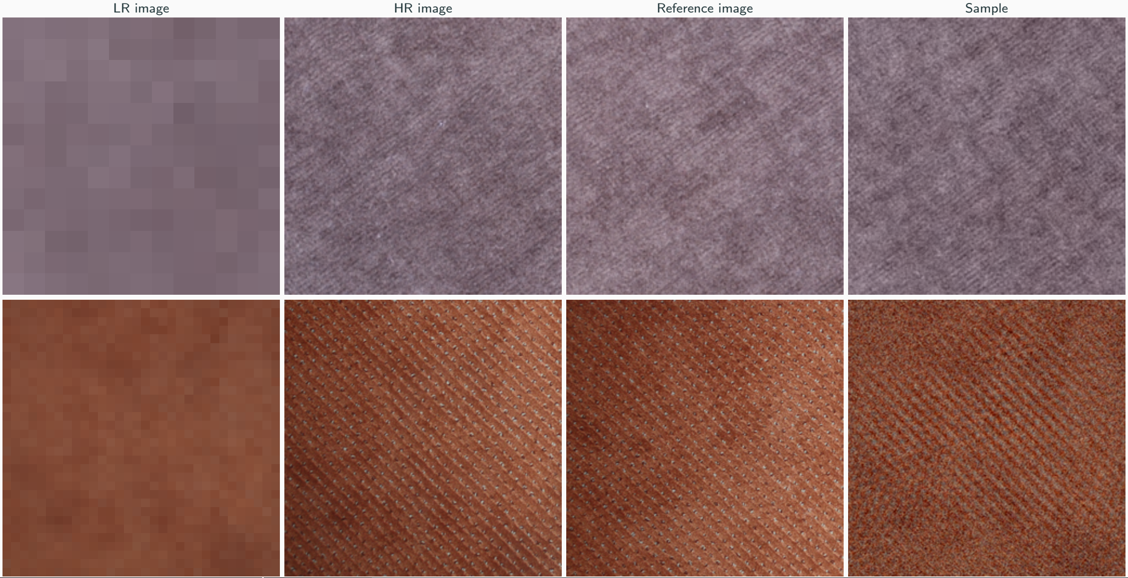
Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., & Norouzi, M. (2022). Image Super-Resolution Via Iterative Refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., & Loy, C. C. (2019). ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. *ECCV 2018.*

Examples



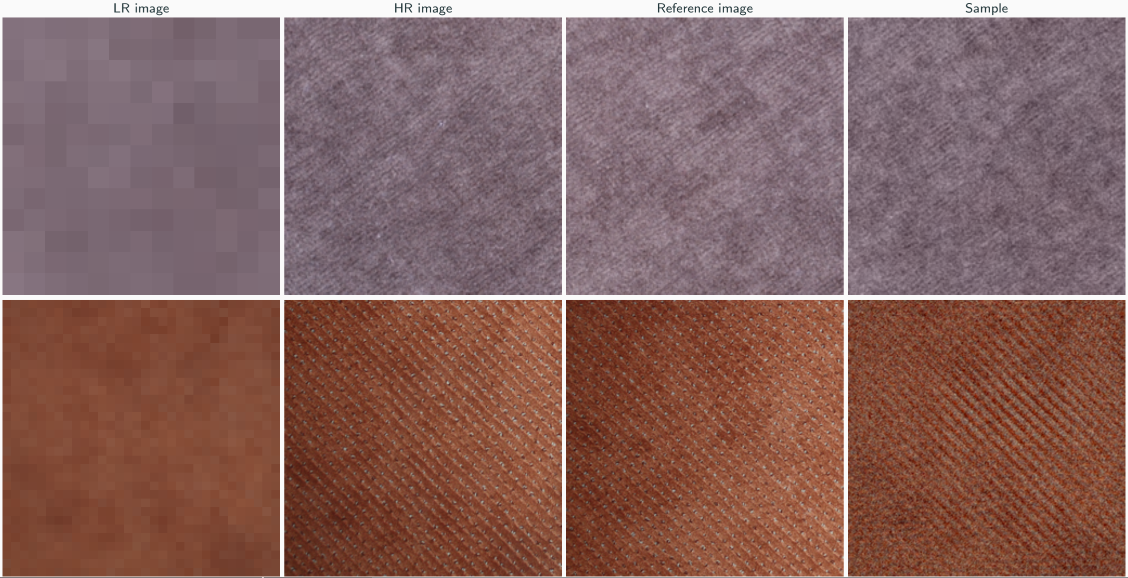
Examples



Examples

LR image | HR image | Reference image | Sample

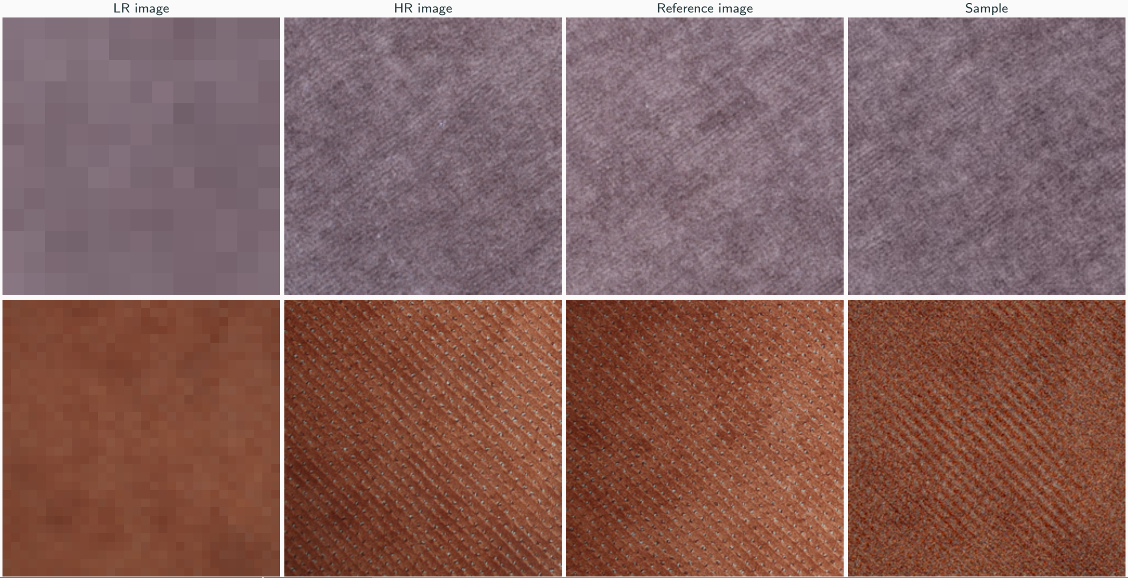
Examples



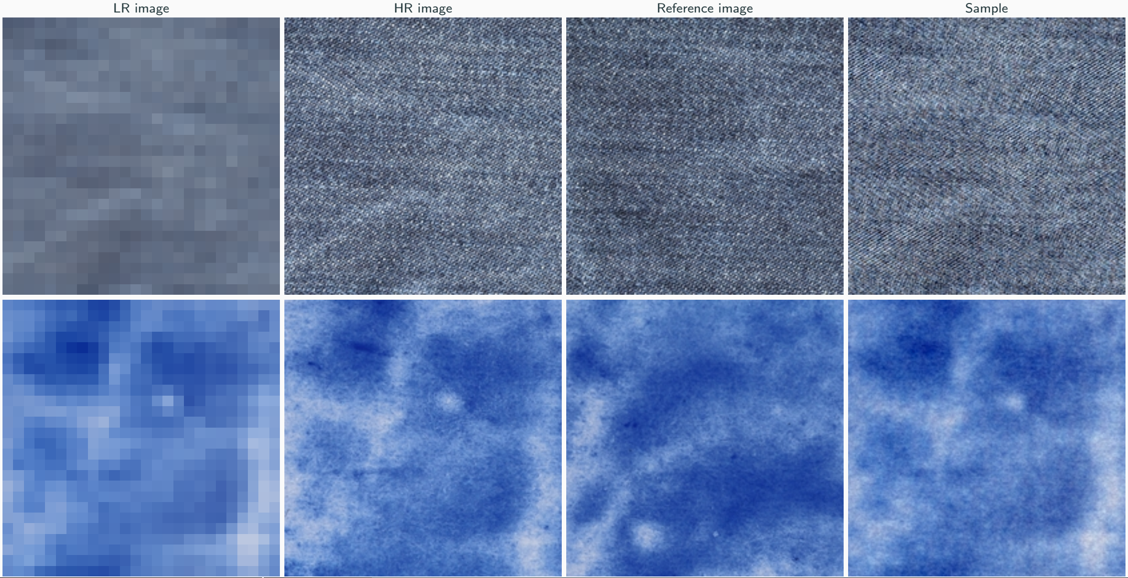
Examples

LR image | HR image | Reference image | Sample

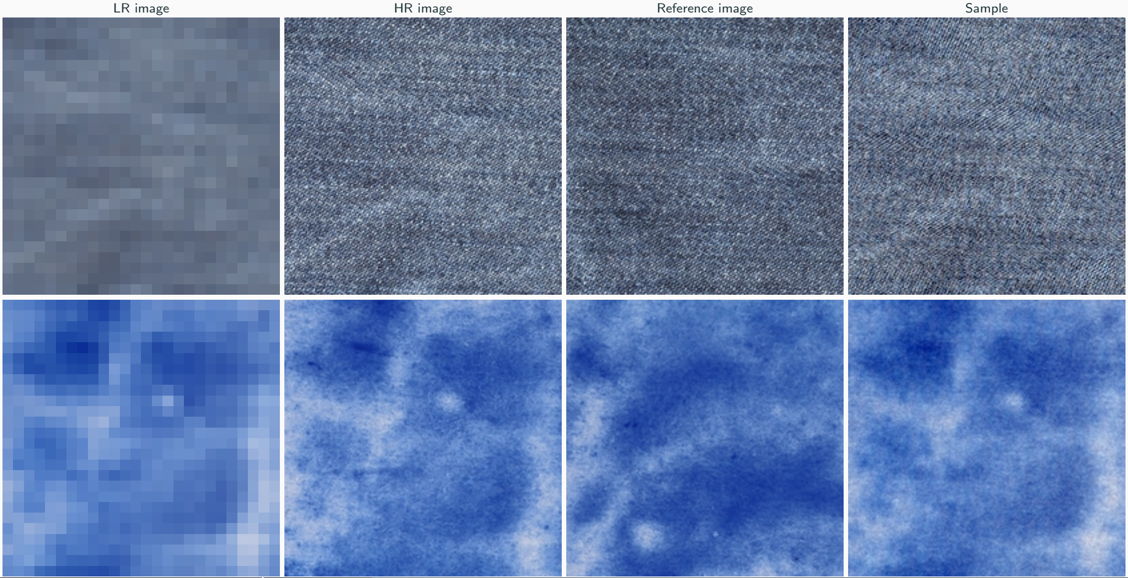
Examples



Examples

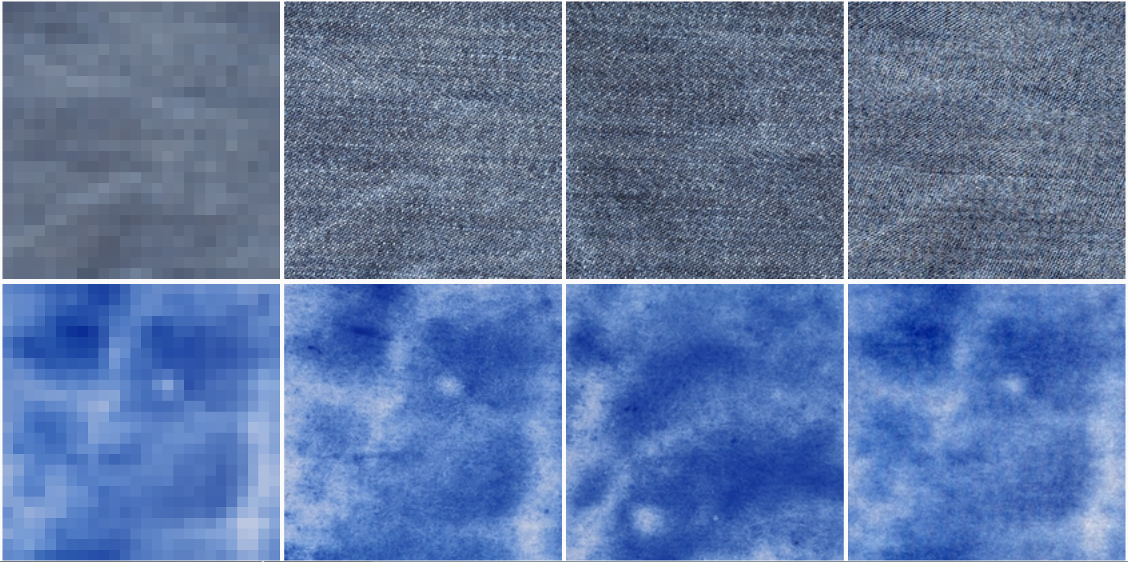


Examples

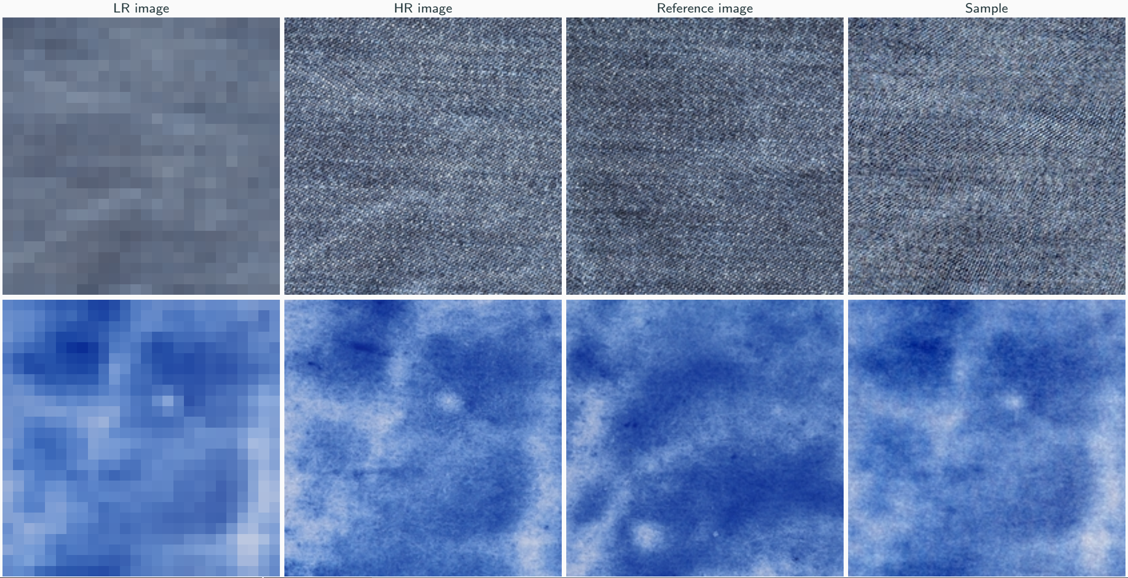


Examples

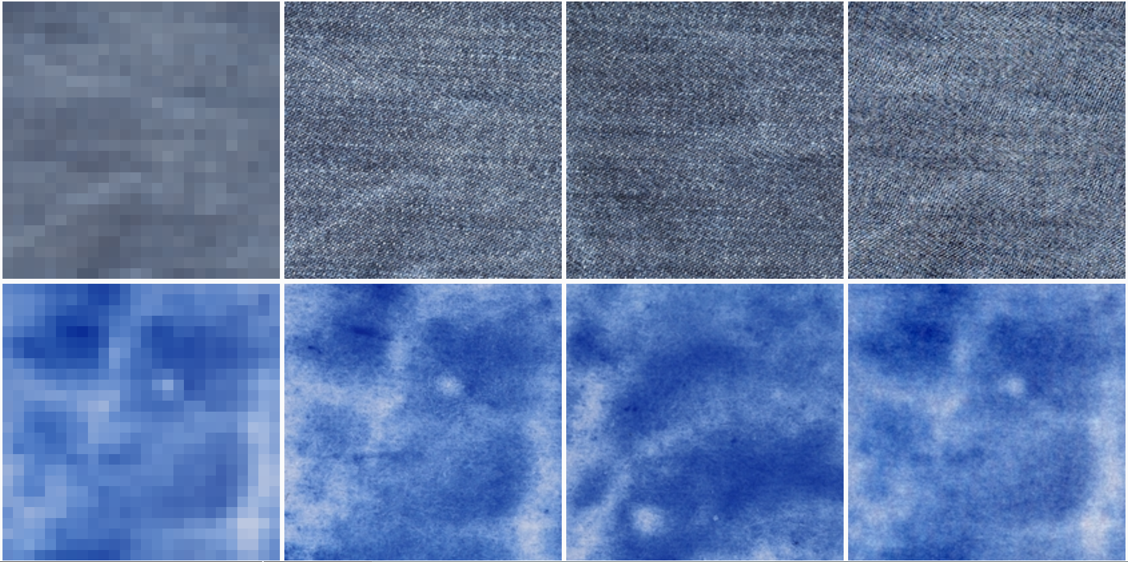
LR image | HR image | Reference image | Sample



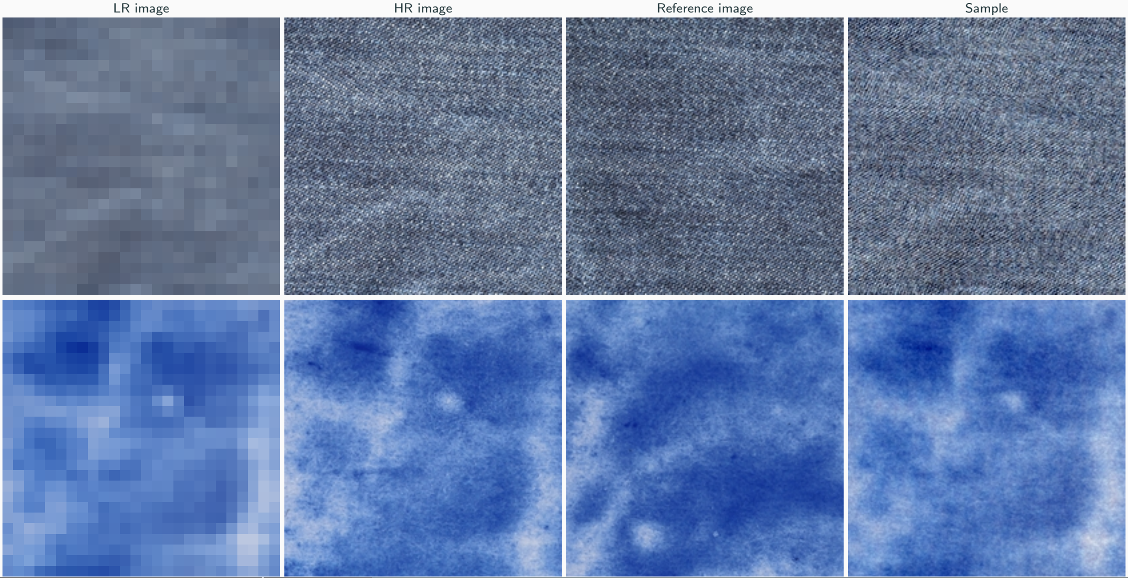
Examples



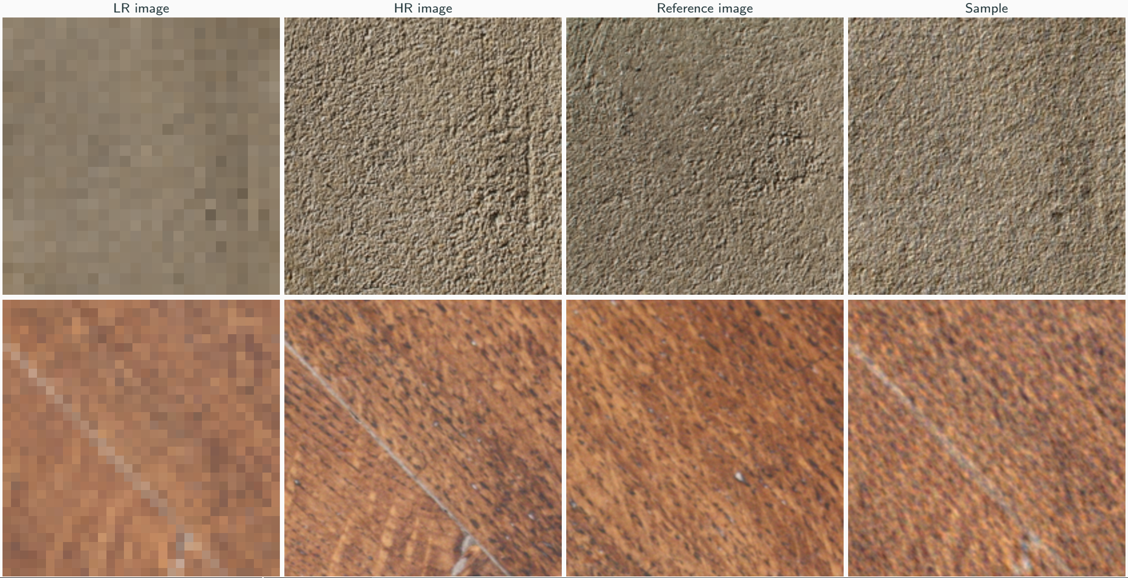
Examples



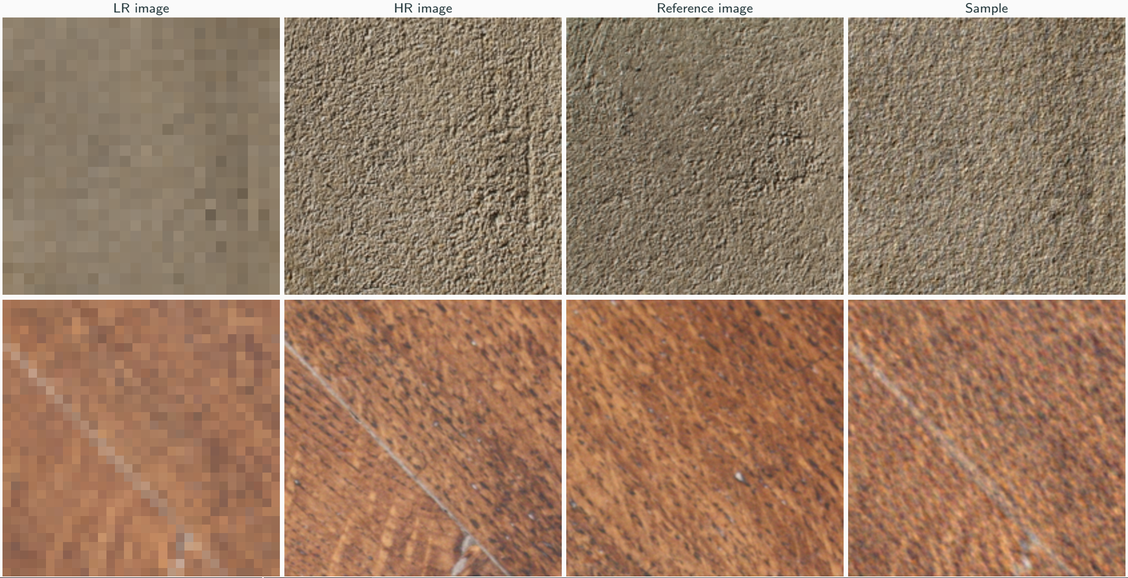
Examples



Examples



Examples



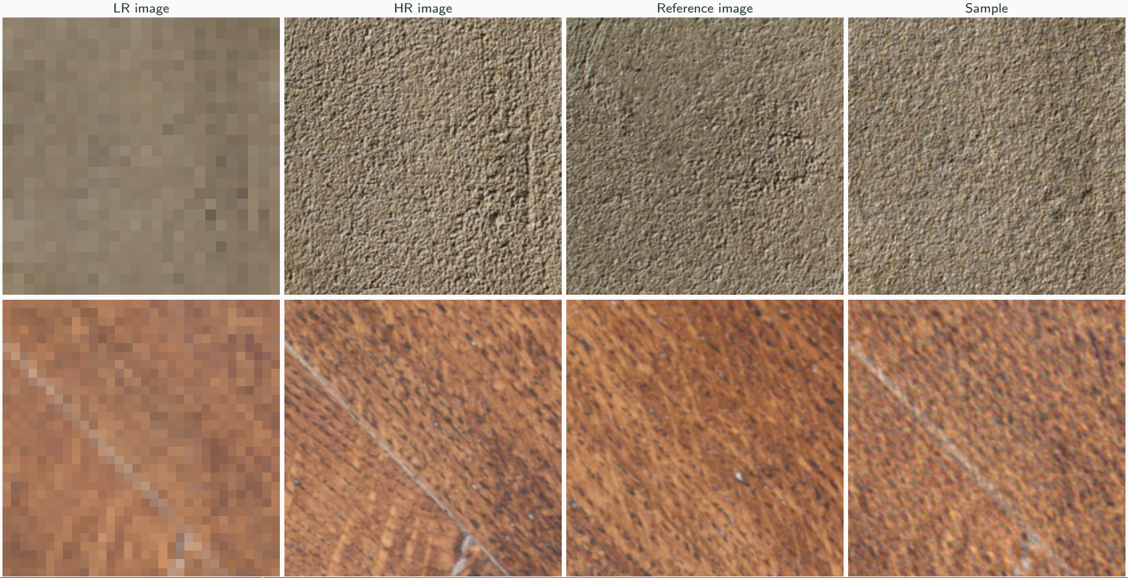
Examples

LR image | HR image | Reference image | Sample

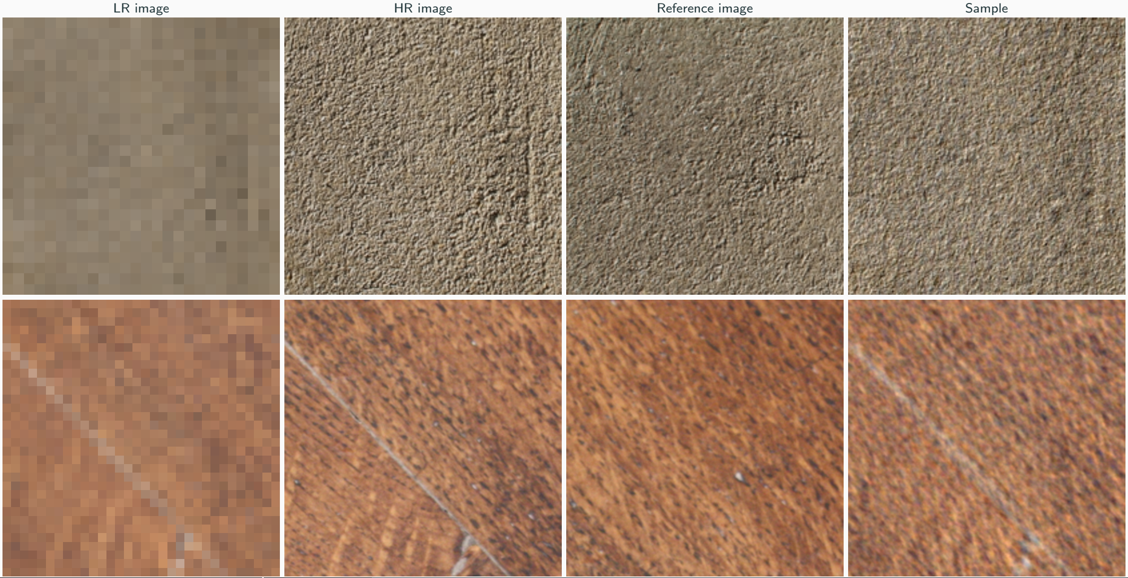
Examples

LR image | HR image | Reference image | Sample

Examples



Examples



Go back : 14