

Projet 7 :

Développez une preuve de concept



Stanford Dogs Dataset

Présentation :

Une preuve de concept (POC) est un petit projet ou test qui montre que quelque chose fonctionne comme prévu. C'est comme un prototype ou un essai qui prouve que l'idée est réalisable avant de la développer complètement.



Présentation :

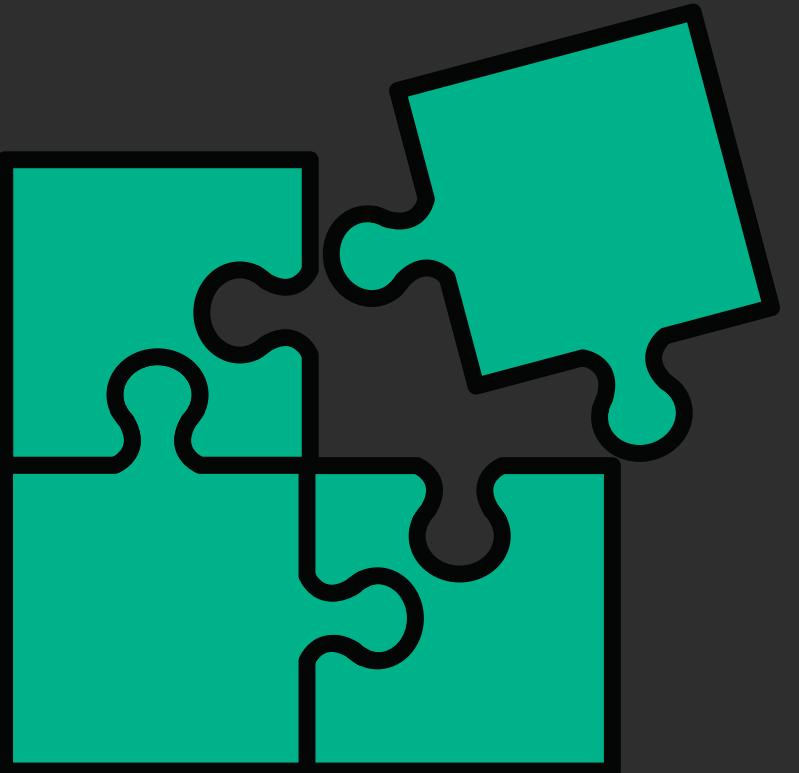
Une recherche approfondie dans le domaine du deep learning a révélé un sujet encore peu étudié. Des réseaux de neurones sont exploités dans le but d'augmenter la résolution spatiale d'une image, autrement dit, pour améliorer le niveau de détail d'une image donnée.



Stanford Dogs Dataset

Problème :

La résolution d'image unique, ou super-résolution d'image unique (SISR), est une technique qui permet d'améliorer la qualité et les détails d'une image à faible résolution en générant une version plus nette et détaillée de cette même image

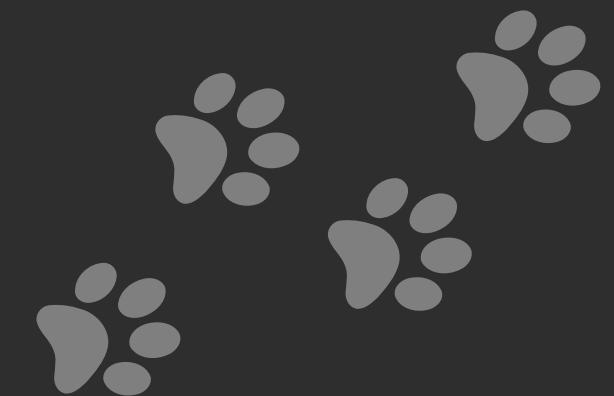


Stanford Dogs Dataset

Présentation :

Dans le projet précédent , une association de protection des animaux détient une BDD de photos de chiens qui continue de s'agrandir .

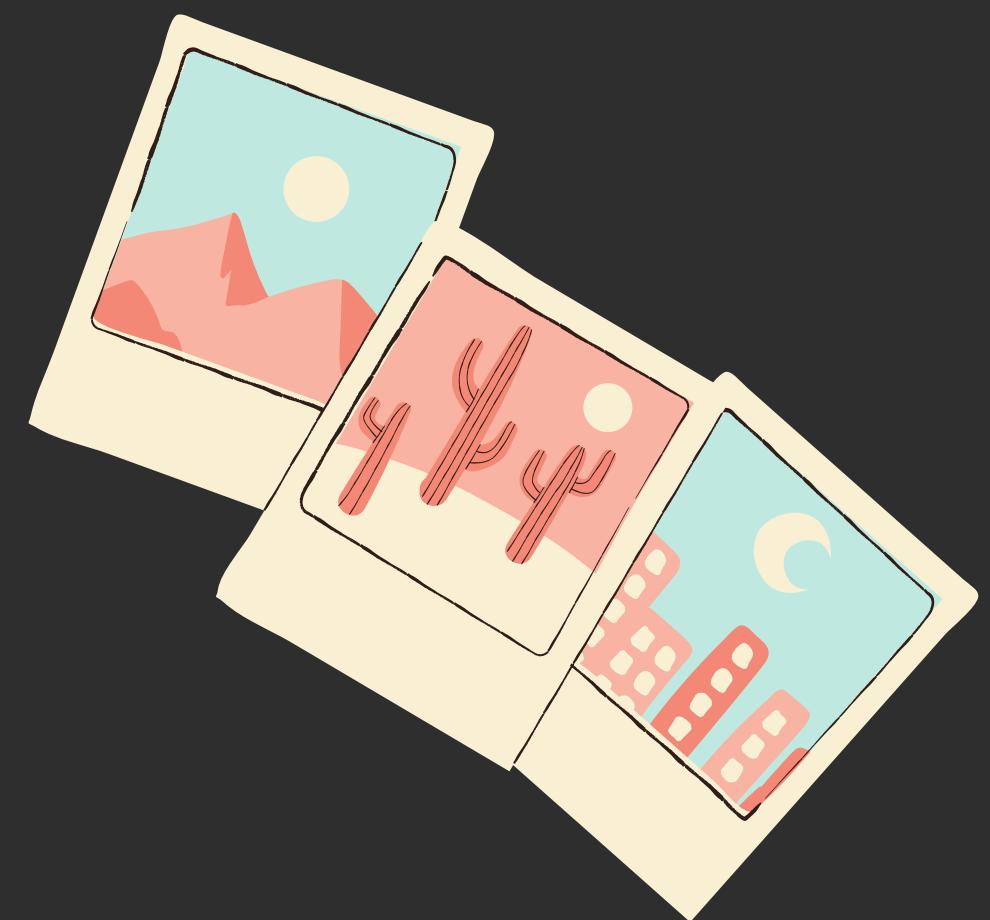
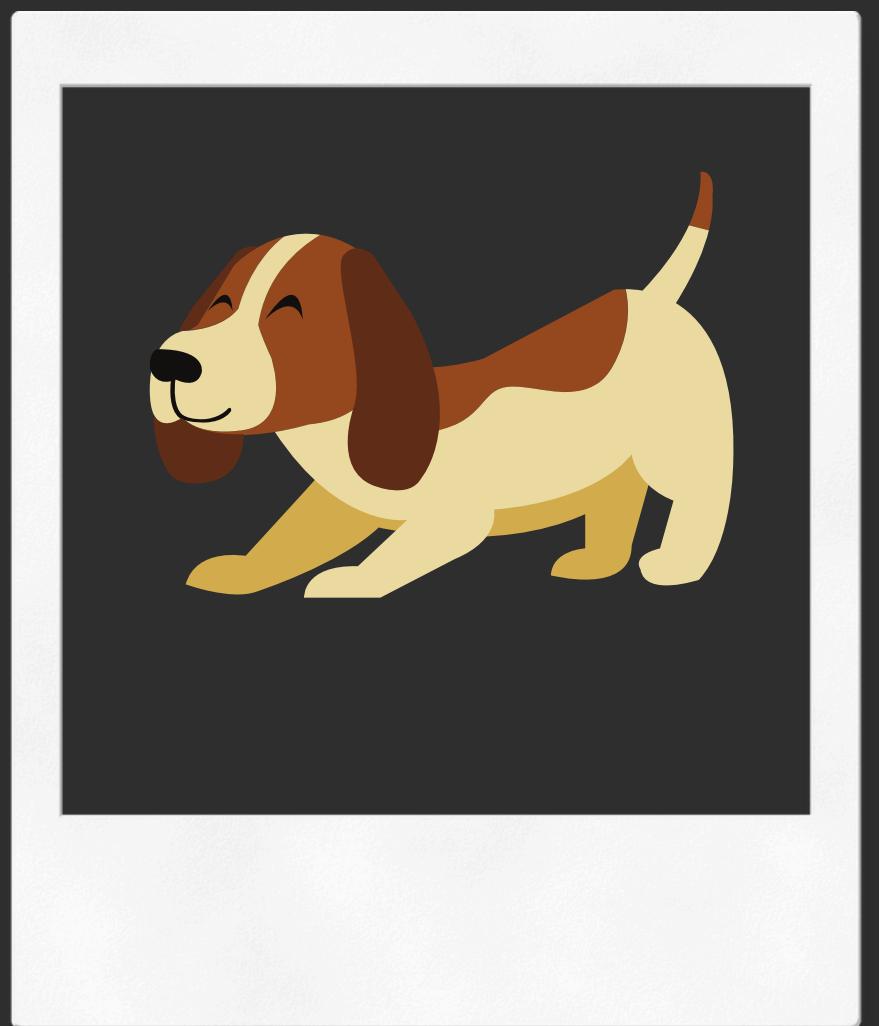
Mais certaines sont de mauvaises qualité



Stanford Dogs Dataset

Présentation :

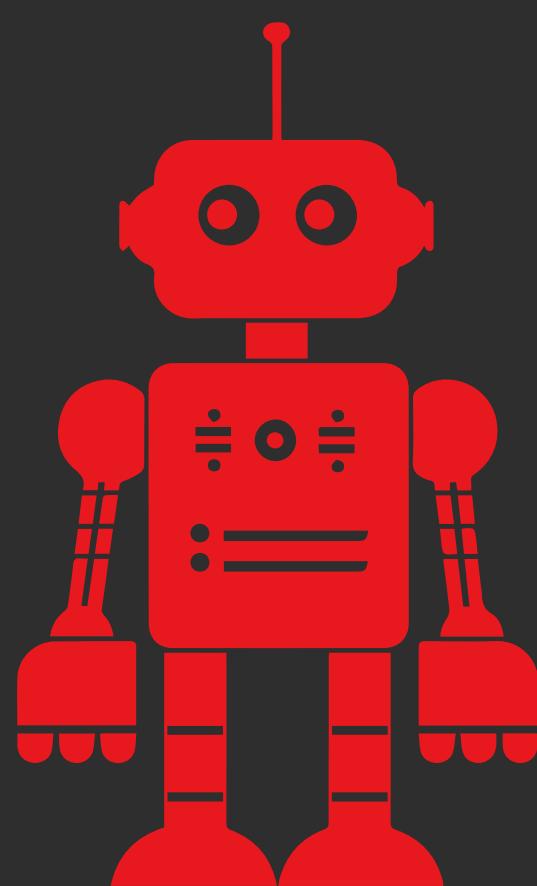
Pouvons-nous améliorer la qualité des photos de chiens ?



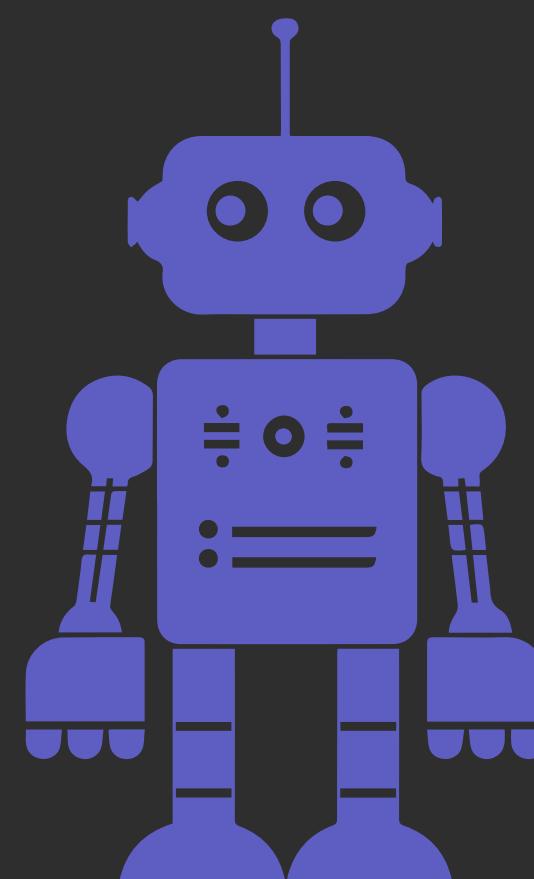
Stanford Dogs Dataset

Dans le domaine complexe de la résolution d'images, une structure particulière a émergé dans divers articles sur ce sujet .

Ce sont les réseaux adverses génératifs (GANs)



VS



Stanford Dogs Dataset

Les GANs (Generative Adversarial Networks)

- Deux intelligences artificielles travaillent ensemble :
 - Le générateur crée des images .
 - Le discriminateur essaie de deviner si ces images sont fausses ou réelles.



Stanford Dogs Dataset

Le but du générateur est de tromper le discriminateur en produisant des images de plus en plus réalistes.

Au fil du temps, les deux réseaux s'améliorent mutuellement, permettant de générer des images très convaincantes.



Stanford Dogs Dataset

Pour améliorer la résolution d'images de chiens avec des GANs :

1. Le générateur crée des images haute résolution à partir d'images floues.
2. Le discriminateur compare ces images avec de vraies images haute résolution pour détecter les fausses.
3. Entraînement : Le générateur améliore ses images pour tromper le discriminateur, tandis que ce dernier devient meilleur pour faire la distinction.

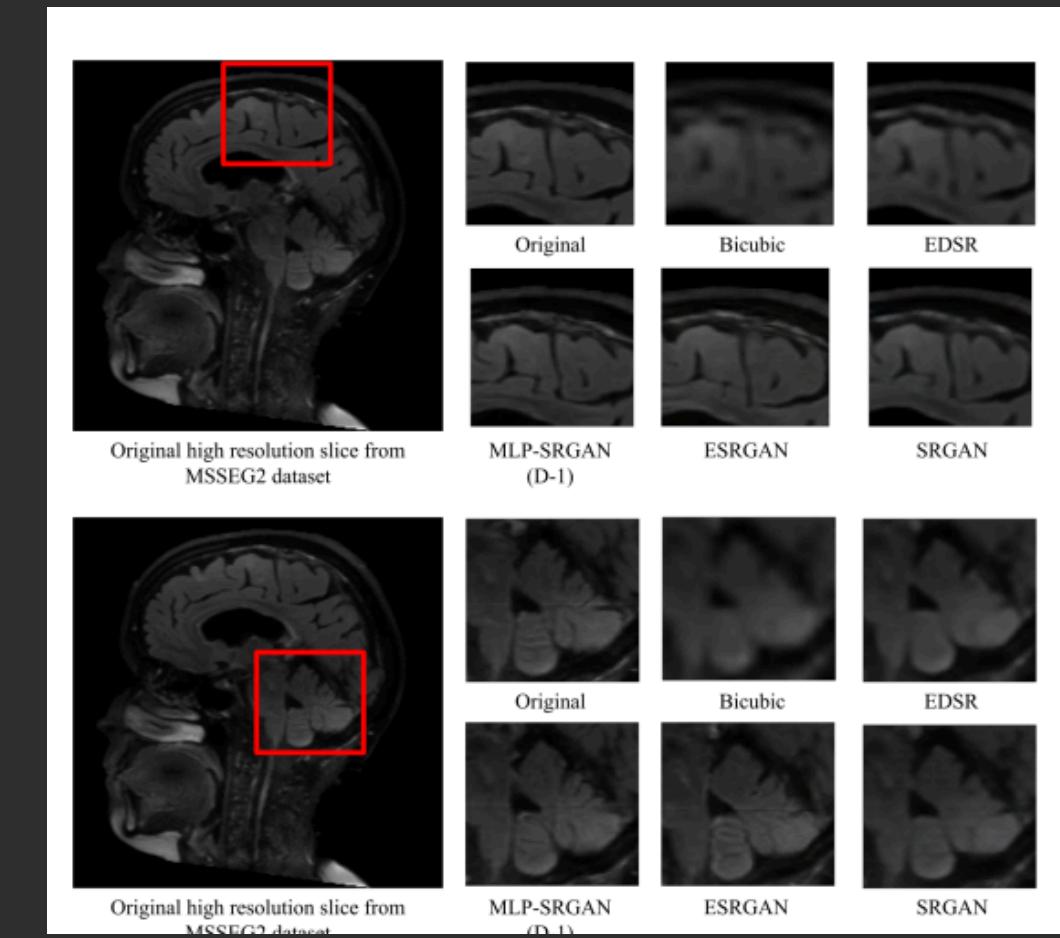
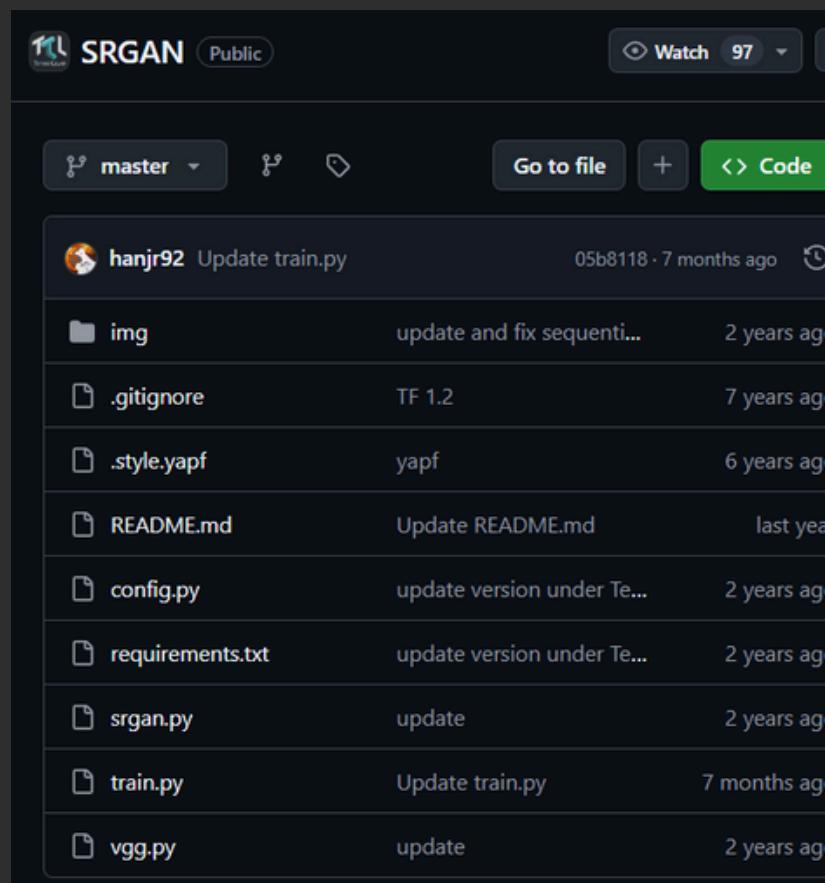
Résultat : Le générateur apprend à créer des images de chiens détaillées et nettes à partir d'images floues.



Stanford Dogs Dataset

Le SRGAN (Super-Resolution GAN) est un modèle récent, datant de 2022, que nous allons utiliser , il est régulièrement mis à jour :

- articles : <https://arxiv.org/pdf/2404.04642.pdf>
- github : <https://github.com/tensorlayer/srgan>



Stanford Dogs Dataset

Lecture de la publication

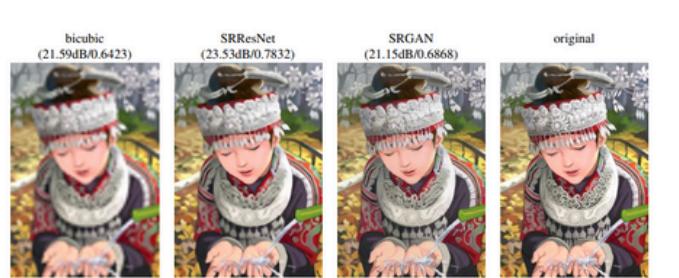


Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

perceptual difference between the super-resolved and original image means that the recovered image is not photo-realistic as defined by Ferweda [16].

In this work we propose a super-resolution generative adversarial network (SRGAN) for which we employ a deep residual network (ResNet) with skip-connection and diverge from MSE as the sole optimization target. Different from previous works, we define a novel perceptual loss using high-level feature maps of the VGG network [49, 33, 5] combined with a discriminator that encourages solutions perceptually hard to distinguish from the HR reference images. An example photo-realistic image that was super-resolved with a 4× upscaling factor is shown in Figure 1.

1.1. Related work

1.1.1. Image super-resolution

Recent overview articles on image SR include Nasrollahi and Moeslund [43] or Yang et al. [61]. Here we will focus

which the corresponding HR counterparts are known. Early work was presented by Freeman et al. [18, 17]. Related approaches to the SR problem originate in compressed sensing [62, 12, 69]. In Glasner et al. [21] the authors exploit patch redundancies across scales within the image to drive the SR. This paradigm of self-similarity is also employed in Huang et al. [31], where self-dictionaries are extended by further allowing for small transformations and shape variations. Gu et al. [25] proposed a convolutional sparse coding approach that improves consistency by processing the whole image rather than overlapping patches.

To reconstruct realistic texture detail while avoiding edge artifacts, Tai et al. [52] combine an edge-directed SR algorithm based on a gradient profile prior [50] with the benefits of learning-based detail synthesis. To super-resolve landmark images, Yue et al. [67] retrieve corresponding HR images with similar content from the web and propose a

2.2. Perceptual loss function

The definition of our perceptual loss function I^{SR} is critical for the performance of our generator network. While I^{SR} is commonly modeled based on the MSE [10, 48], we improve on Johnson et al. [33] and Bruna et al. [5] and design a loss function that assesses a solution with respect to perceptually relevant characteristics. We formulate the perceptual loss as the weighted sum of a content loss ($I_{Content}^{SR}$) and an adversarial loss component as:

$$I^{SR} = \frac{I_{Content}^{SR}}{10} + 10^3 I_{Adversarial}^{SR} \quad (3)$$

frequency content which results in perceptually unsatisfying solutions with overly smooth textures (c.f. Figure 2). Instead of relying on pixel-wise losses we build on the ideas of Gatys et al. [19], Bruna et al. [5] and Johnson et al. [33] and use a loss function that is closer to perceptual similarity. We define the VGG loss based on the ReLU activation layers of the pre-trained 19 layer VGG network described in Simonyan and Zisserman [49]. With $\phi_{i,j}$ we indicate the feature map obtained by the i -th convolution (after activation) before the j -th maxpooling layer within the VGG19 network, which we consider given. We then define the VGG loss as the euclidean distance between the feature representations of a reconstructed image $G_{\theta_O}(I^{LR})$ and the

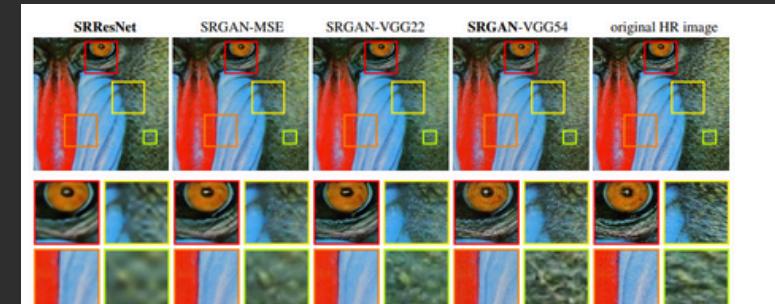


Figure 6: SRResNet (left: a,b), SRGAN-MSE (middle left: c,d), SRGAN-VGG22 (middle: e,f) and SRGAN-VGG54 (middle right: g,h) reconstruction results and corresponding reference HR image (right: i,j). [4× upscaling]

Table 2: Comparison of NN, bicubic, SRCNN [9], SelfExSR [31], DRCN [34], ESPCN [48], SRResNet, SRGAN-VGG54 and the original HR on benchmark data. Highest measures (PSNR [dB], SSIM, MOS) in bold. [4× upscaling]

	Set5	nearest	bicubic	SRCNN	SelfExSR	DRCN	ESPCN	SRResNet	SRGAN	HR
PSNR	26.26	28.43	30.07	30.33	31.52	30.76	32.05	29.40	∞	
SSIM	0.7552	0.8211	0.8627	0.872	0.8938	0.8784	0.9019	0.8472	1	
MOS	1.28	1.97	2.57	2.65	3.26	2.89	3.37	3.58	4.32	
Set14										
PSNR	24.64	25.99	27.18	27.45	28.02	27.66	28.49	26.02	∞	
SSIM	0.7100	0.7486	0.7861	0.7972	0.8074	0.8000	0.8184	0.7397	1	
MOS	1.20	1.80	2.26	2.34	2.84	2.52	2.98	3.72	4.32	
BSD100										
PSNR	25.02	25.94	26.68	26.83	27.21	27.02	27.58	25.16	∞	
SSIM	0.6606	0.6935	0.7291	0.7387	0.7493	0.7442	0.7620	0.6688	1	
MOS	1.11	1.47	1.87	1.89	2.12	2.01	2.29	3.56	4.46	

increase the performance of SRResNet, however, come at the cost of longer training and testing times (c.f. supple-

the adversarial loss and photo-realistic images. We also note that the ideal loss function depends on the application.

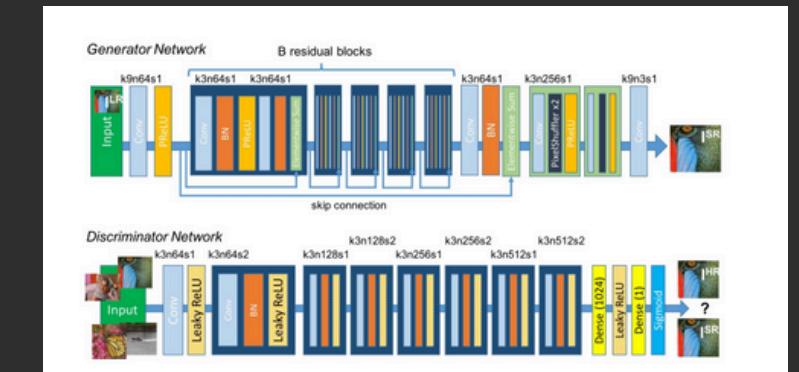


Figure 4: Architecture of Generator and Discriminator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

algorithms have shown excellent performance. In Wang et al. [59] the authors encode a sparse representation prior into their feed-forward network architecture based on the learned iterative shrinkage and thresholding algorithm (LISTA) [23]. Dong et al. [9, 10] used bicubic interpolation to upscale an input image and trained a three layer deep fully convolutional network end-to-end to achieve state-of-the-art SR performance. Subsequently, it was shown that enabling the network to learn the upscaling filters directly can further increase performance both in terms of accuracy and speed [11, 48, 57]. With their deeply-recursive convolutional network (DRCN), Kim et al. [34] presented a highly performant architecture that allows for long-range pixel dependencies while keeping the number of model parameters small. Of particular relevance for our paper are the works by Johnson et al. [33] and Bruna et al. [5], who rely on a loss function closer to perceptual similarity to recover visually more convincing HR images.

1.1.2 Design of convolutional neural networks

The state of the art for many computer vision problems is meanwhile set by specifically designed CNN architectures following the success of the work by Krizhevsky et al. [37].

It was shown that deeper network architectures can be difficult to train but have the potential to substantially increase the network's accuracy as they allow modeling mappings of very high complexity [49, 51]. To efficiently train these deeper network architectures, batch-normalization [32] is often used to counteract the internal co-variate shift. Deeper network architectures have also been shown to increase performance for SISR, e.g. Kim et al. [34] formulate a recursive CNN and present state-of-the-art results. Another powerful design choice that eases the training of deep CNNs is the recently introduced concept of residual blocks [29] and skip-connections [30, 34]. Skip-

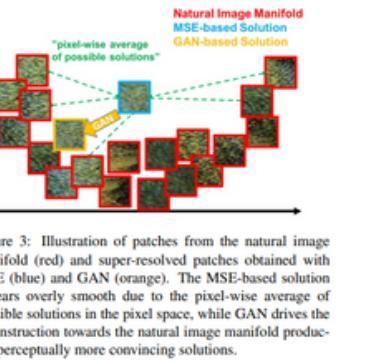


Figure 3: Illustration of patches from the natural image manifold (red) and super-resolved patches obtained with MSE (blue) and GAN (orange). The MSE-based solution appears overly smooth due to the pixel-wise average of possible solutions in the pixel space, while GAN drives the reconstruction towards the natural image manifold producing perceptually more convincing solutions.

quality is exemplified with corresponding PSNR in Figure 2. We illustrate the problem of minimizing MSE in Figure 3 where multiple potential solutions with high texture details are averaged to create a smooth reconstruction.

In Mathieu et al. [42] and Denton et al. [7] the authors tackled this problem by employing generative adversarial networks (GANs) [22] for the application of image generation. Yu and Porikli [66] augment pixel-wise MSE loss with a discriminator loss to train a network that super-resolves face images with large upscaling factors (8×). GANs were also used for unsupervised representation learning in Radford et al. [44]. The idea of using GANs to learn a mapping from one manifold to another is described by Li and Wand [38] for style transfer and Yeh et al. [64] for inpainting. Bruna et al. [5] minimize the squared error in

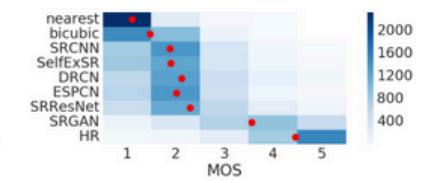


Figure 5: Color-coded distribution of MOS scores on BSD100. For each method 2600 samples (100 images × 26 raters) were assessed. Mean shown as red marker, where the bins are centered around value i . [4× upscaling]

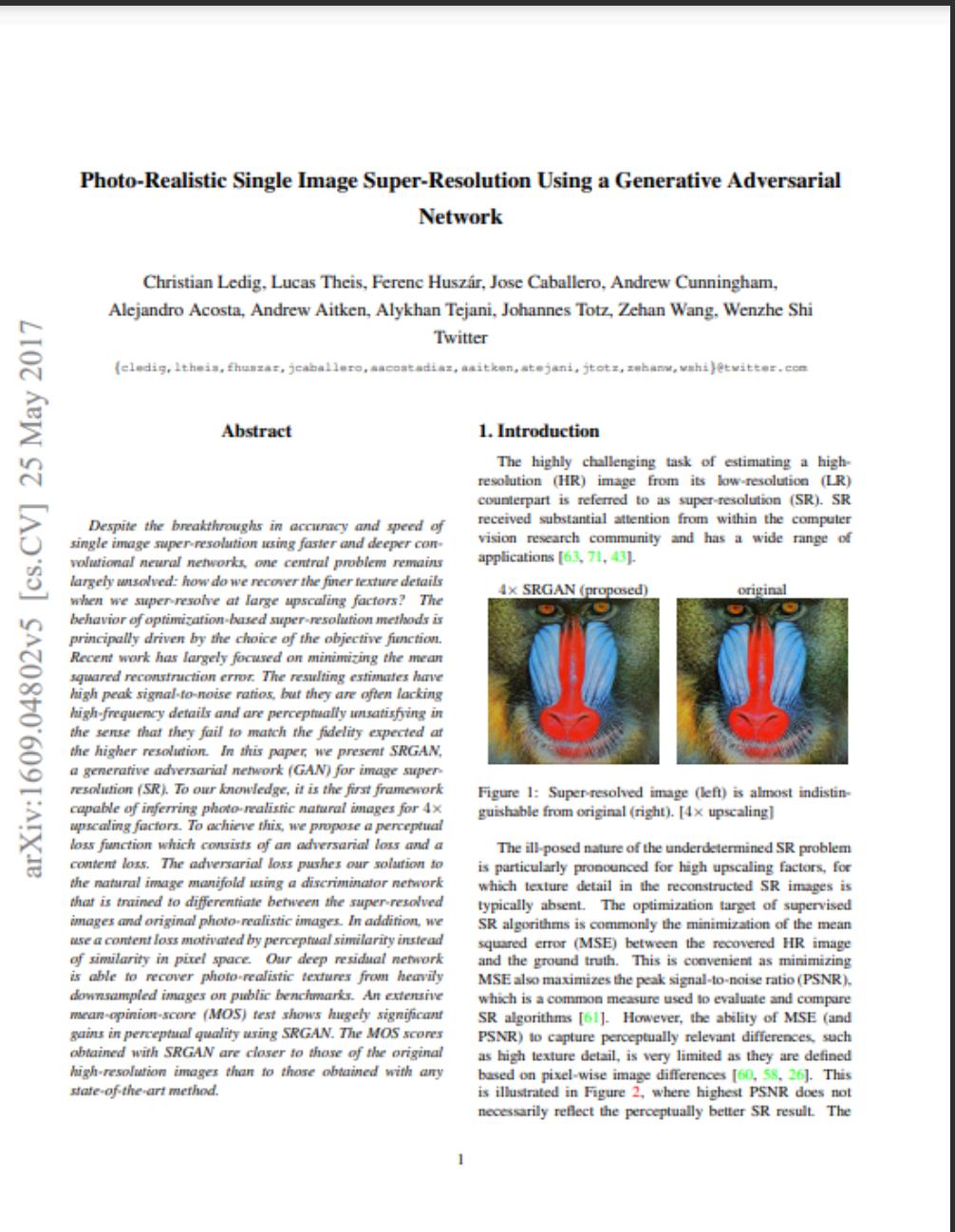
3.5. Performance of the final networks

We compare the performance of SRResNet and SRGAN with NN, bicubic interpolation, and four state-of-the-art methods. Quantitative results are summarized in Table 2 and confirm that SRResNet (in terms of PSNR/SSIM) sets a new state of the art on three benchmark datasets. Please note that we used a publicly available framework for evaluation (c.f. Section 3.1), reported values might thus slightly deviate from those reported in the original papers.

We further obtained MOS ratings for SRGAN and all reference methods on BSD100. Examples of images super-resolved with SRResNet and SRGAN are depicted in the supplementary material. The results shown in Table 2 confirm that SRGAN outperforms all reference methods by a large margin and sets a new state of the art for photo-realistic image SR. All differences in MOS (c.f. Table 2) are highly significant on BSD100, except SRCNN vs. SelfExSR. The di-



Stanford Dogs Dataset



- le SRGAN est un réseau adverse génératif qui restaure des images pour un ZOOM x4

- Le critère d'optimisation des algorithmes de super-résolution est généralement de minimiser l'erreur quadratique moyenne (MSE) entre les images restaurées et les images haute résolution réelles.



- Le SRGAN utilise un réseau spécial, appelé réseau résiduel, pour améliorer les images.
- Le critère d'optimisation n'est pas basé sur l'erreur MSE.
- On utilise une fonction de perte perceptuelle qui s'appuie sur un autre réseau (VGG) pour évaluer la qualité.
- on se concentre sur l'amélioration de la résolution d'une seule image.

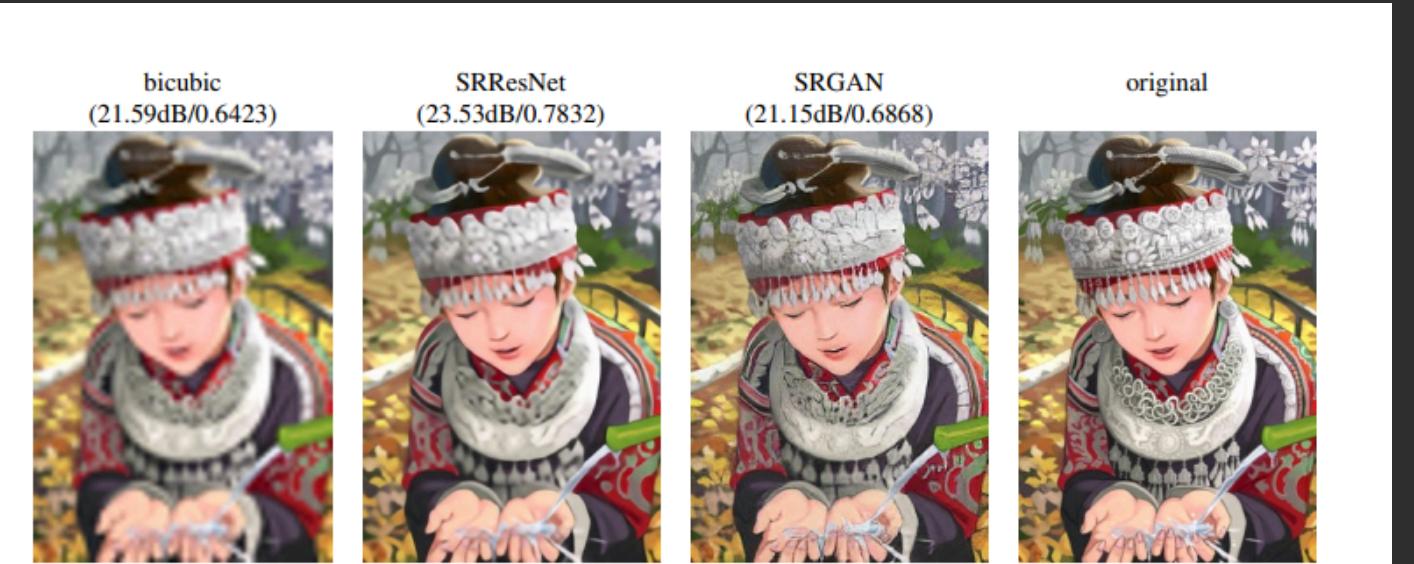


Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

perceptual difference between the super-resolved and original image means that the recovered image is not photo-realistic as defined by Ferwerda [16].

In this work we propose a super-resolution generative adversarial network (SRGAN) for which we employ a deep residual network (ResNet) with skip-connection and diverge from MSE as the sole optimization target. Different from previous works, we define a novel perceptual loss using high-level feature maps of the VGG network [49, 33, 5] combined with a discriminator that encourages solutions perceptually hard to distinguish from the HR reference images. An example photo-realistic image that was super-resolved with a 4× upscaling factor is shown in Figure 1.

1.1. Related work

1.1.1 Image super-resolution

Recent overview articles on image SR include Nasrollahi and Moeslund [43] or Yang et al. [61]. Here we will focus

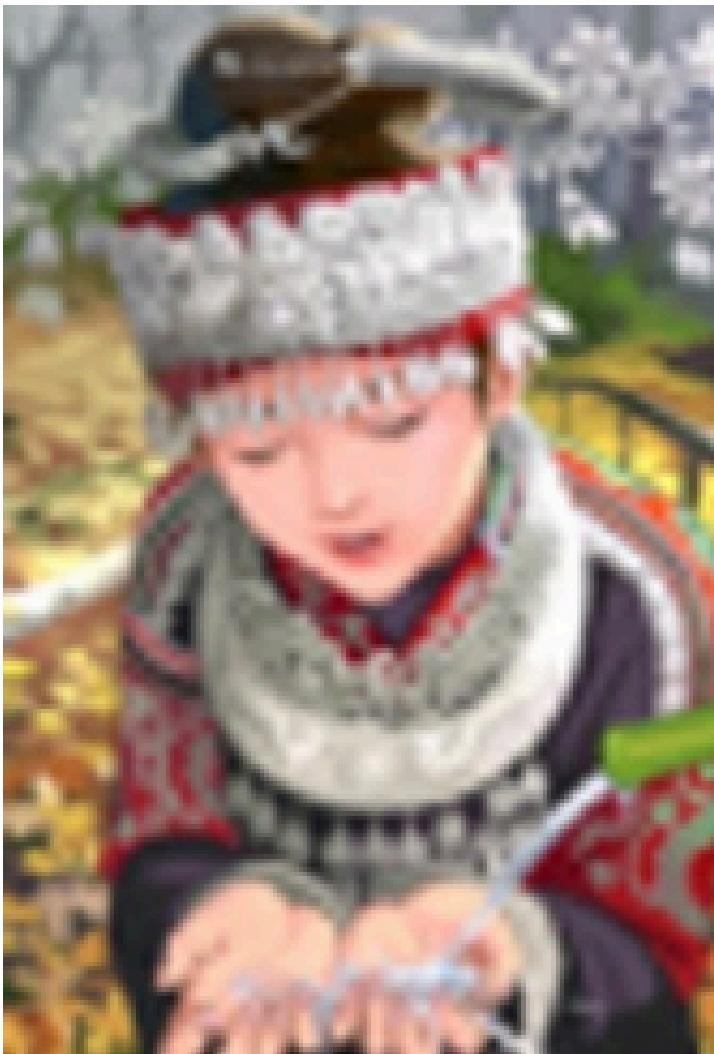
which the corresponding HR counterparts are known. Early work was presented by Freeman et al. [18, 17]. Related approaches to the SR problem originate in compressed sensing [62, 12, 69]. In Glasner et al. [21] the authors exploit patch redundancies across scales within the image to drive the SR. This paradigm of self-similarity is also employed in Huang et al. [31], where self dictionaries are extended by further allowing for small transformations and shape variations. Gu et al. [25] proposed a convolutional sparse coding approach that improves consistency by processing the whole image rather than overlapping patches.

To reconstruct realistic texture detail while avoiding edge artifacts, Tai et al. [52] combine an edge-directed SR algorithm based on a gradient profile prior [50] with the benefits of learning-based detail synthesis. Zhang et al. [70] propose a multi-scale dictionary to capture redundancies of similar image patches at different scales. To super-resolve landmark images, Yue et al. [67] retrieve correlating HR images with similar content from the web and propose a

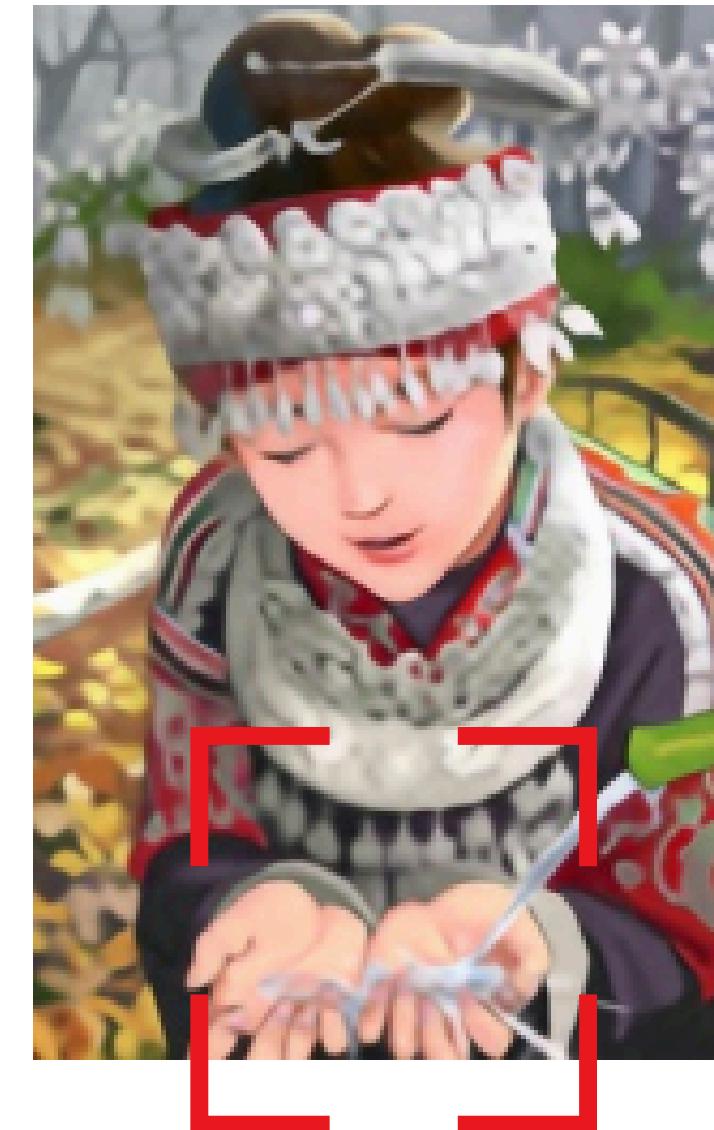


Stanford Dogs Dataset

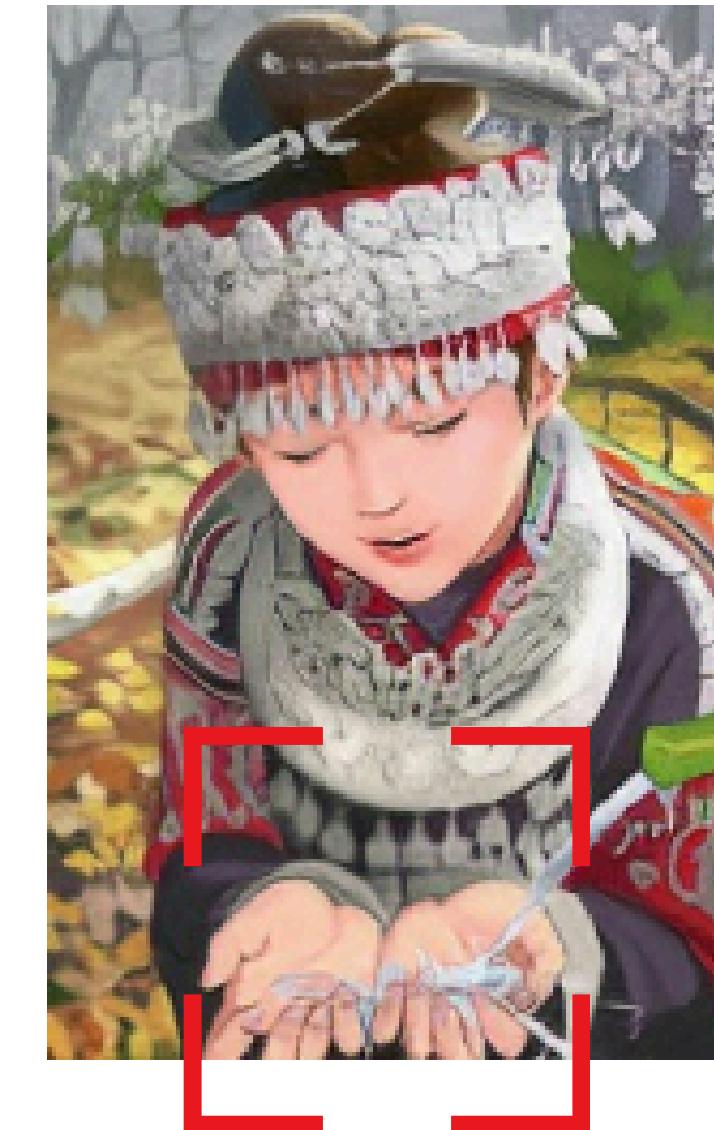
bicubic
(21.59dB/0.6423)



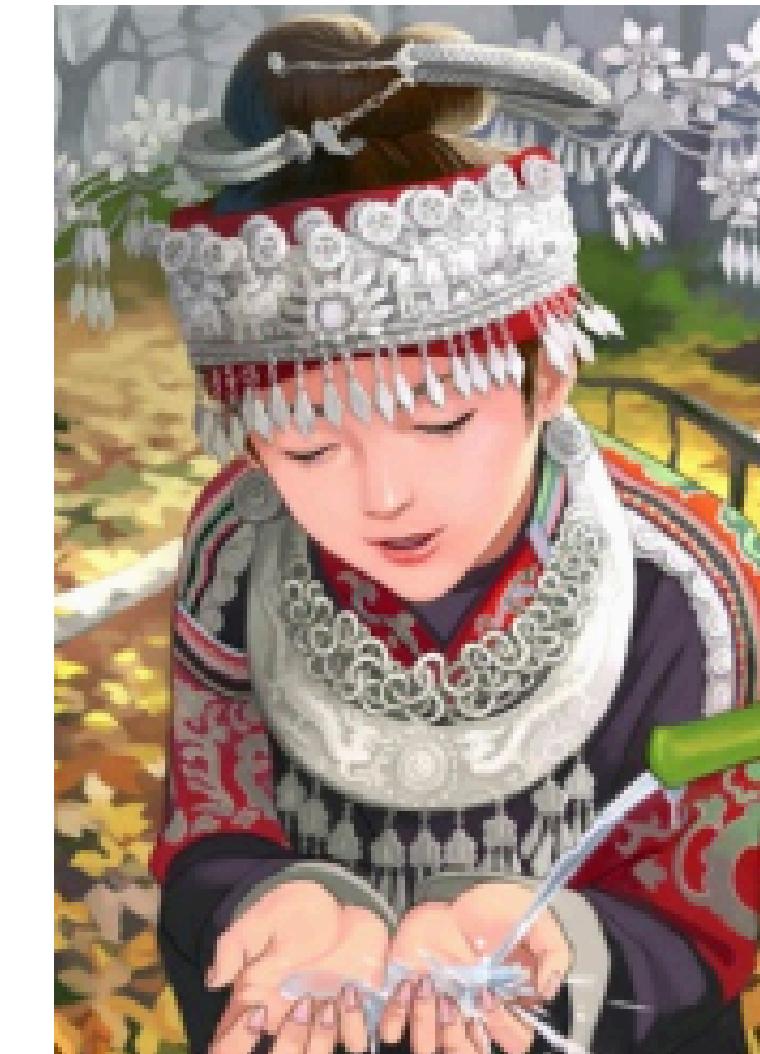
SRResNet
(23.53dB/0.7832)



SRGAN
(21.15dB/0.6868)



original



Stanford Dogs Dataset

algorithms have shown excellent performance. In Wang et al. [59] the authors encode a sparse representation prior into their feed-forward network architecture based on the learned iterative shrinkage and thresholding algorithm (LISTA) [23]. Dong et al. [9, 10] used bicubic interpolation to upscale an input image and trained a three layer deep fully convolutional network end-to-end to achieve state-of-the-art SR performance. Subsequently, it was shown that enabling the network to learn the upscaling filters directly can further increase performance both in terms of accuracy and speed [11, 48, 57]. With their deeply-recursive convolutional network (DRCN), Kim et al. [34] presented a highly performant architecture that allows for long-range pixel dependencies while keeping the number of model parameters small. Of particular relevance for our paper are the works by Johnson et al. [33] and Bruna et al. [5], who rely on a loss function closer to perceptual similarity to recover visually more convincing HR images.

1.1.2 Design of convolutional neural networks

The state of the art for many computer vision problems is meanwhile set by specifically designed CNN architectures following the success of the work by Krizhevsky et al. [37].

It was shown that deeper network architectures can be difficult to train but have the potential to substantially increase the network's accuracy as they allow modeling mappings of very high complexity [49, 51]. To efficiently train these deeper network architectures, batch-normalization [32] is often used to counteract the internal co-variate shift. Deeper network architectures have also been shown to increase performance for SISR, *e.g.* Kim et al. [34] formulate a recursive CNN and present state-of-the-art results. Another powerful design choice that eases the training of deep CNNs is the recently introduced concept of residual blocks [29] and skip-connections [30, 34]. Skip-

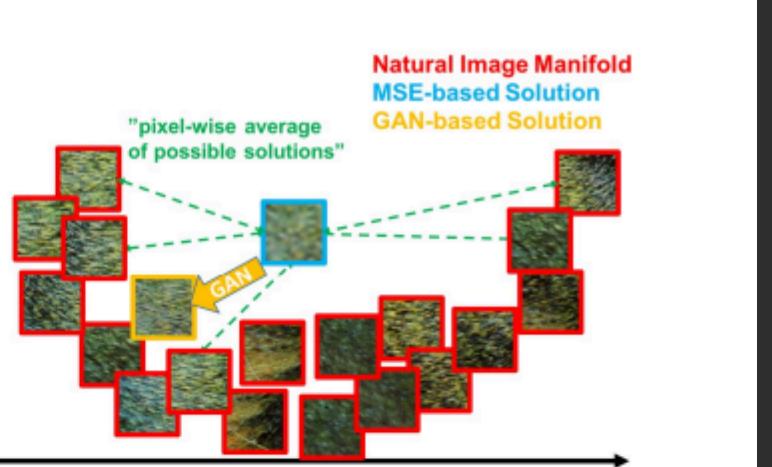


Figure 3: Illustration of patches from the natural image manifold (red) and super-resolved patches obtained with MSE (blue) and GAN (orange). The MSE-based solution appears overly smooth due to the pixel-wise average of possible solutions in the pixel space, while GAN drives the reconstruction towards the natural image manifold producing perceptually more convincing solutions.

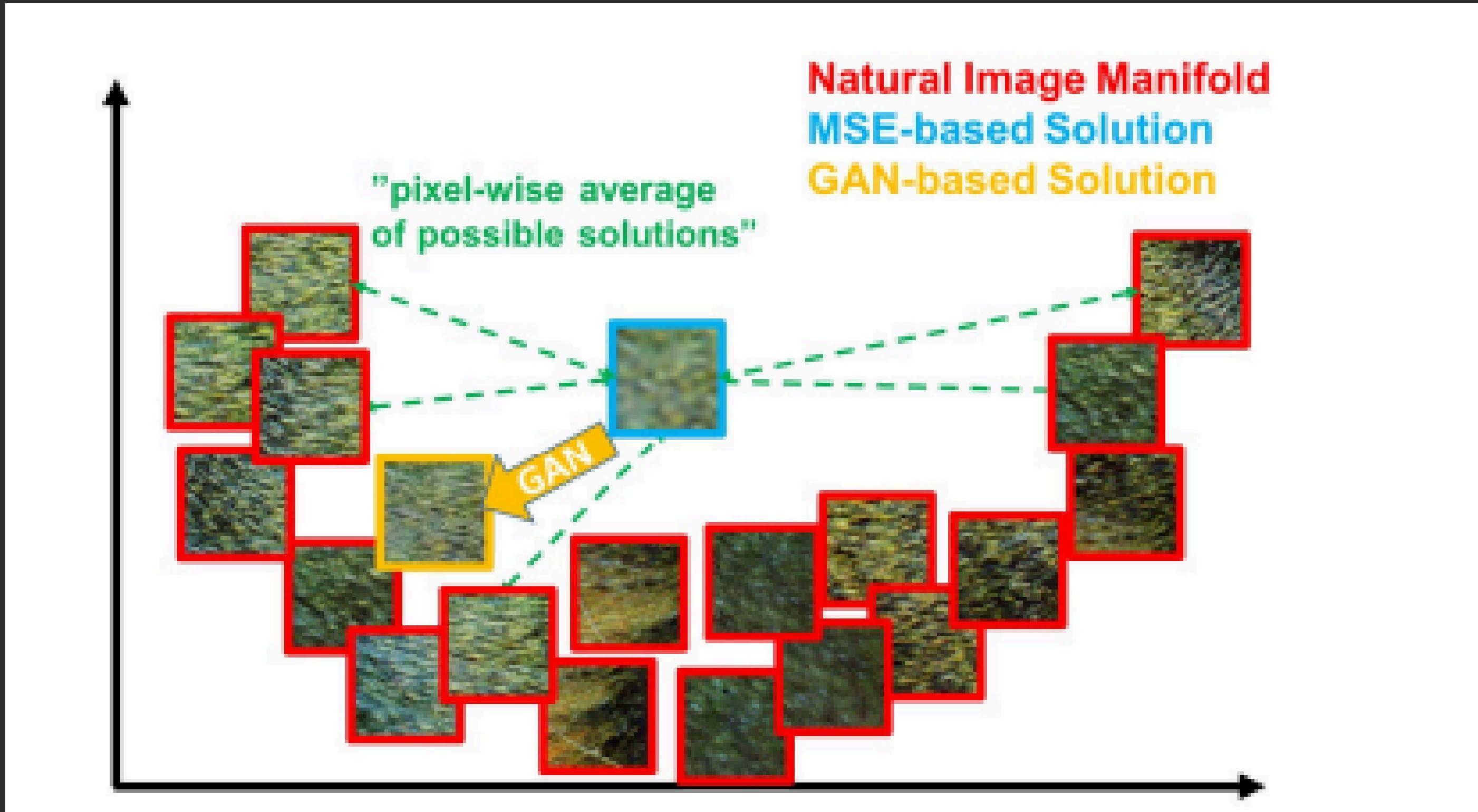
quality are exemplified with corresponding PSNR in Figure 2. We illustrate the problem of minimizing MSE in Figure 3 where multiple potential solutions with high texture details are averaged to create a smooth reconstruction.

In Mathieu et al. [42] and Denton et al. [7] the authors tackled this problem by employing generative adversarial networks (GANs) [22] for the application of image generation. Yu and Porikli [66] augment pixel-wise MSE loss with a discriminator loss to train a network that super-resolves face images with large upscaling factors ($8\times$). GANs were also used for unsupervised representation learning in Radford et al. [44]. The idea of using GANs to learn a mapping from one manifold to another is described by Li and Wand [38] for style transfer and Yeh et al. [64] for inpainting. Bruna et al. [5] minimize the squared error in

- La fonction de perte MSE a du mal à préserver les détails fins, ce qui rend les images floues et de qualité inférieure.
- Cela est dû au fait qu'elle moyennise les différentes solutions possibles pour reconstruire l'image.
- L'utilisation d'un GAN aide à surmonter ce problème.



Stanford Dogs Dataset



Stanford Dogs Dataset

Table 1: Performance of different loss functions for SR-ResNet and the adversarial networks on Set5 and Set14 benchmark data. MOS score significantly higher ($p < 0.05$) than with other losses in that category*. [4× upscaling]

Set5	SRResNet-		SRGAN-		
	MSE	VGG22	MSE	VGG22	VGG54
PSNR	32.05	30.51	30.64	29.84	29.40
SSIM	0.9019	0.8803	0.8701	0.8468	0.8472
MOS	3.37	3.46	3.77	3.78	3.58
Set14					
PSNR	28.49	27.19	26.92	26.44	26.02
SSIM	0.8184	0.7807	0.7611	0.7518	0.7397
MOS	2.98	3.15*	3.43	3.57	3.72*

- SRGAN-MSE: l_{MSE}^{SR} , to investigate the adversarial network with the standard MSE as content loss.
- SRGAN-VGG22: $l_{VGG/2.2}^{SR}$ with $\phi_{2,2}$, a loss defined on feature maps representing lower-level features [68].
- SRGAN-VGG54: $l_{VGG/5.4}^{SR}$ with $\phi_{5,4}$, a loss defined on feature maps of higher level features from deeper network layers with more potential to focus on the content of the images [68, 65, 40]. We refer to this network as **SRGAN** in the following.

We also evaluate the performance of the generator network without adversarial component for the two losses l_{MSE}^{SR} (**SRResNet-MSE**) and $l_{VGG/2.2}^{SR}$ (**SRResNet-VGG22**). We refer to SRResNet-MSE as **SRResNet**. Note, when training SRResNet-VGG22 we added an additional total variation loss with weight 2×10^{-8} to $l_{VGG/2.2}^{SR}$ [2, 33]. Quantitative

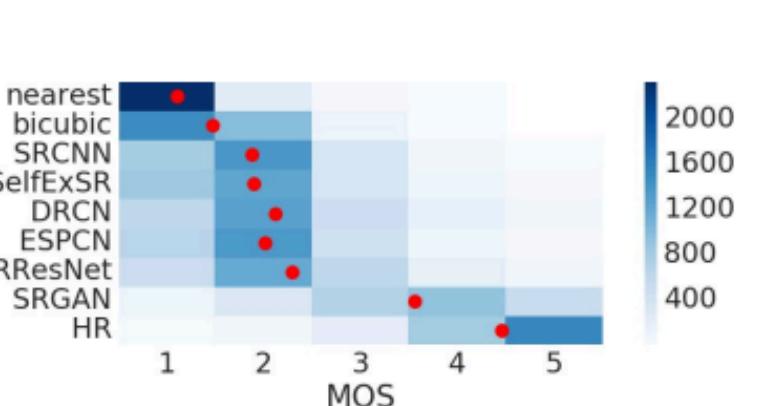


Figure 5: Color-coded distribution of MOS scores on **BSD100**. For each method 2600 samples (100 images \times 26 raters) were assessed. Mean shown as red marker, where the bins are centered around value i . [4× upscaling]

3.5. Performance of the final networks

We compare the performance of **SRResNet** and **SRGAN** to NN, bicubic interpolation, and four state-of-the-art methods. Quantitative results are summarized in Table 2 and confirm that **SRResNet** (in terms of PSNR/SSIM) sets a new state of the art on three benchmark datasets. Please note that we used a publicly available framework for evaluation (c.f. Section 3.1), reported values might thus slightly deviate from those reported in the original papers.

We further obtained MOS ratings for **SRGAN** and all reference methods on **BSD100**. Examples of images super-resolved with **SRResNet** and **SRGAN** are depicted in the supplementary material. The results shown in Table 2 confirm that **SRGAN** outperforms all reference methods by a large margin and sets a new state of the art for photo-realistic image SR. All differences in MOS (c.f. Table 2) are highly significant on **BSD100**, except SRCNN vs. SelfExSR. The distribution of all collected MOS ratings is summarized in Figure 5.

- La fonction de perte MSE a du mal à préserver les détails fins, ce qui rend les images floues et de qualité inférieure.
- Cela est dû au fait qu'elle moyennise les différentes solutions possibles pour reconstruire l'image.
- L'utilisation d'un GAN aide à surmonter ce problème.



- On cherche à ajuster les paramètres du générateur pour minimiser la perte perceptuelle entre les images générées (SR) et les images réelles (HR).
- L'objectif est de créer des images haute résolution très réalistes avec le générateur, tout en utilisant le discriminateur pour différencier les images générées des réelles. L'optimisation se fait en parallèle sur les deux réseaux, qui se défient dans un problème minimax.

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I^{HR} \sim p_{\text{train}}(I^{HR})} [\log D_{\theta_D}(I^{HR})] + \\ \mathbb{E}_{I^{LR} \sim p_G(I^{LR})} [\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})))] \quad (2)$$

$$\hat{\theta}_G = \arg \min_{\theta_G} \frac{1}{N} \sum_{n=1}^N l^{SR}(G_{\theta_G}(I_n^{LR}), I_n^{HR}) \quad (1)$$



Stanford Dogs Dataset

- La fonction de perte perceptuelle comprend une perte sur le contenu et une perte adverse.
- Au lieu de la MSE, la perte sur le contenu est définie en utilisant un réseau VGG19, mesurant la différence entre les cartes de caractéristiques des images générées et réelles.

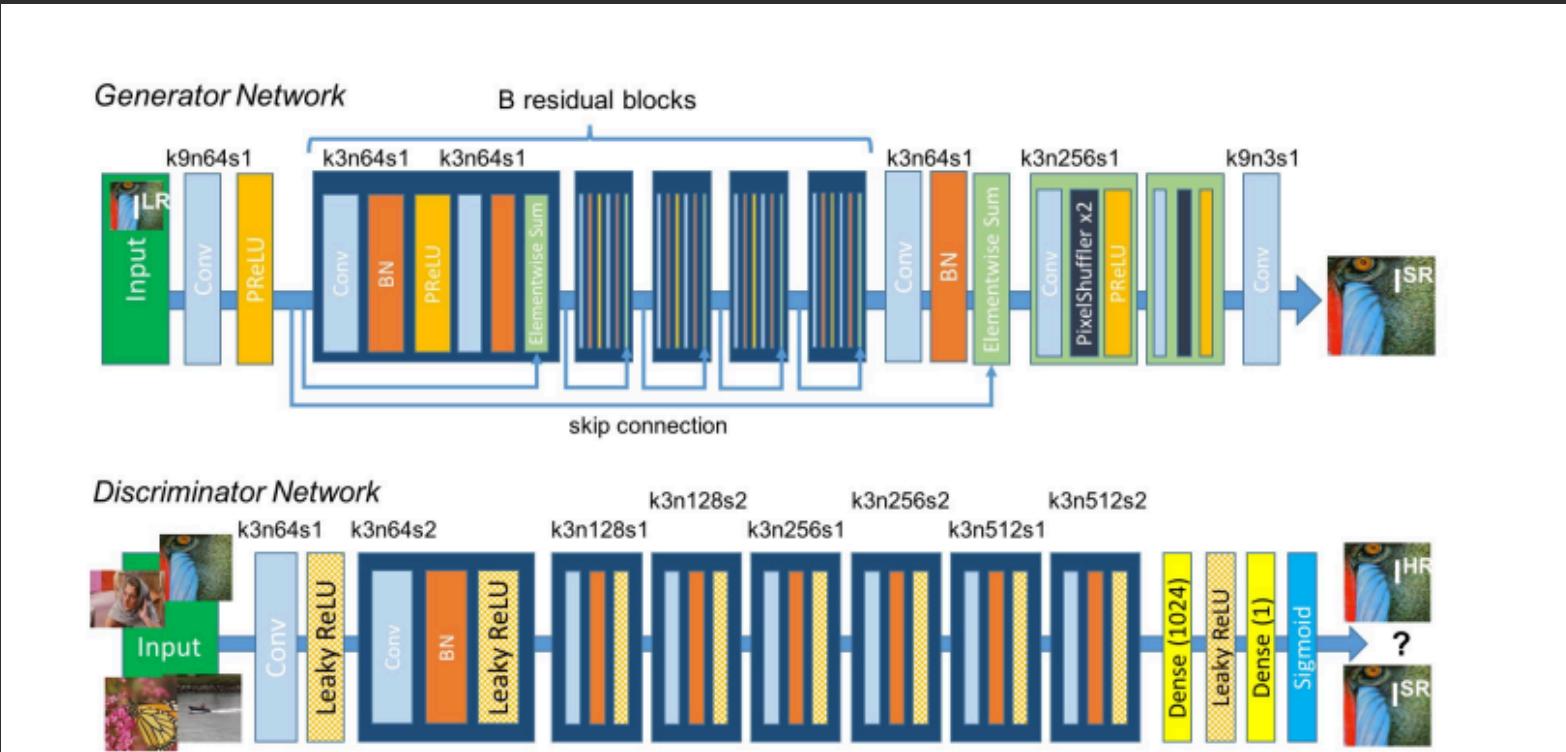


Figure 4: Architecture of Generator and Discriminator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

2.2. Perceptual loss function

The definition of our perceptual loss function l^{SR} is critical for the performance of our generator network. While l^{SR} is commonly modeled based on the MSE [10, 48], we improve on Johnson et al. [33] and Bruna et al. [5] and design a loss function that assesses a solution with respect to perceptually relevant characteristics. We formulate the perceptual loss as the weighted sum of a content loss (l_X^{SR}) and an adversarial loss component as:

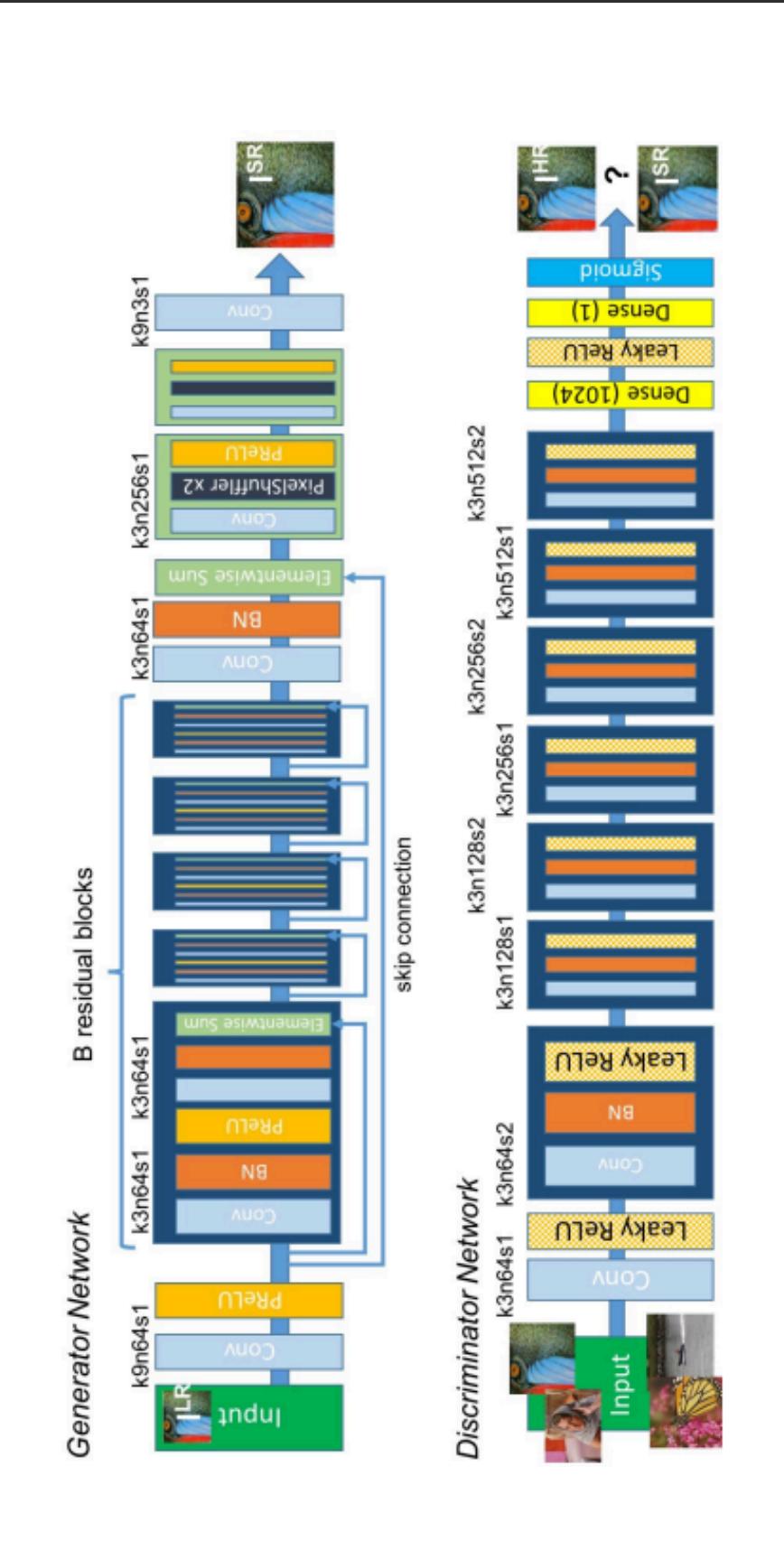
$$l^{SR} = \underbrace{l_X^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{Gen}^{SR}}_{\text{adversarial loss}} \quad (3)$$

frequency content which results in perceptually unsatisfying solutions with overly smooth textures (c.f. Figure 2).

Instead of relying on pixel-wise losses we build on the ideas of Gatys et al. [19], Bruna et al. [5] and Johnson et al. [33] and use a loss function that is closer to perceptual similarity. We define the **VGG loss** based on the ReLU activation layers of the pre-trained 19 layer VGG network described in Simonyan and Zisserman [49]. With $\phi_{i,j}$ we indicate the feature map obtained by the j-th convolution (after activation) before the i-th maxpooling layer within the VGG19 network, which we consider given. We then define the VGG loss as the euclidean distance between the feature representations of a reconstructed image $G_{\theta_G}(I^{LR})$ and the original image I^{HR} .

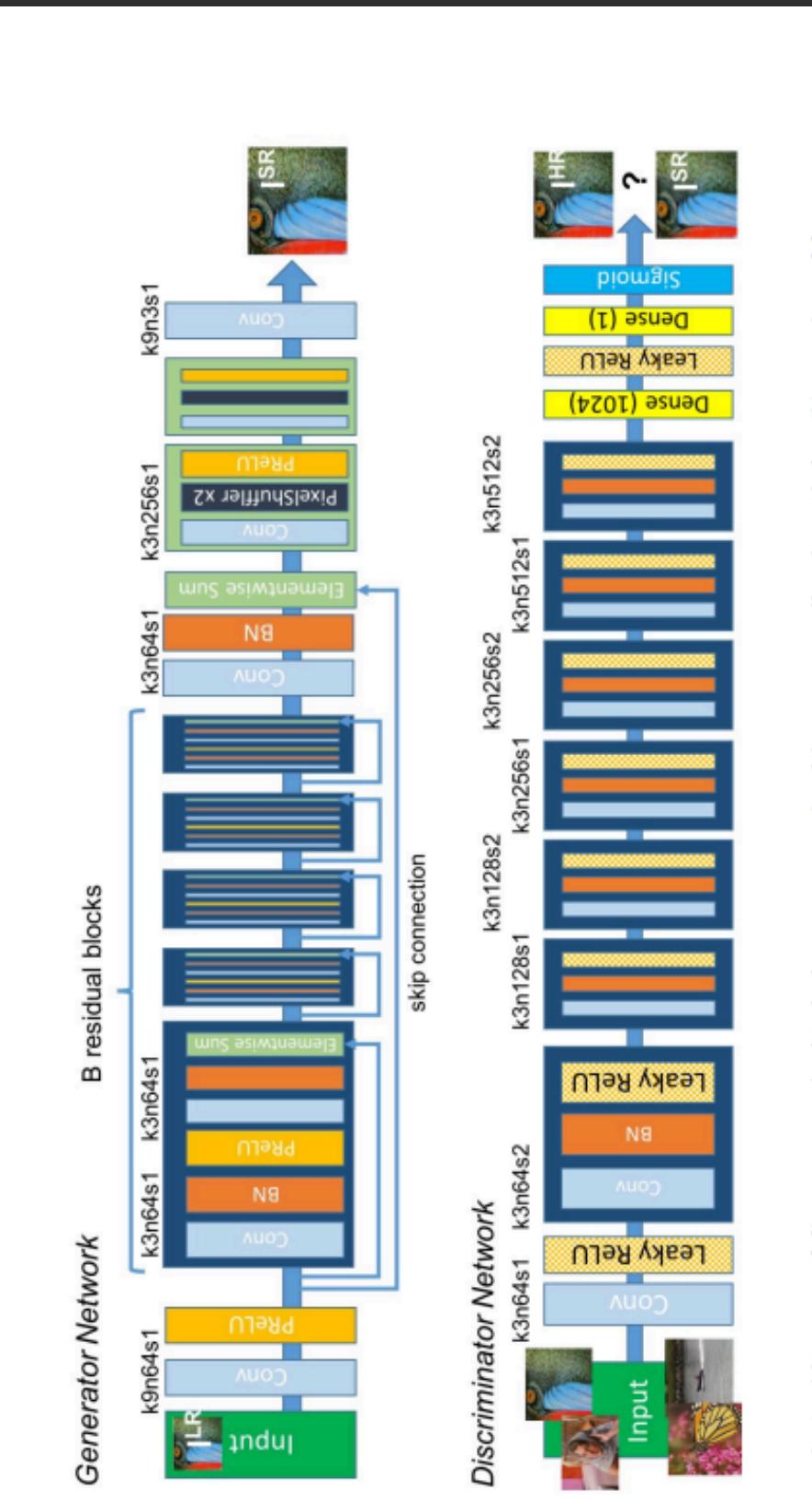


- La fonction de perte de perception comprend une perte sur le contenu et une perte adversaire.
- Au lieu de la MSE, la perte sur le contenu est définie en utilisant un réseau VGG19, mesurant la différence entre les cartes de caractéristiques des images générées et réelles.



Stanford Dogs Dataset

- L'entraînement utilise 350 000 images aléatoires de la base de données ImageNet.
- Les images sont réduites par interpolation bicubique.
- L'entraînement se fait par minibatch de 6, sur des images de 96x96 pixels.
- Le générateur est pré-entraîné avec une fonction de perte MSE pour initialiser ses poids avant l'entraînement avec le GAN



Stanford Dogs Dataset

Les tests montrent que le SRGAN, avec sa combinaison de pertes de contenu et adversaire, est meilleur que les autres modèles et est considéré comme le meilleur disponible.

<https://github.com/krasserm/super-resolution>

<https://github.com/tensorlayer/srgan>

<https://github.com/leftthomas/SRGAN>

<https://www.youtube.com/watch?v=TpMIsRdhco>

<https://www.youtube.com/watch?v=nbRkLE2fiVI>

<https://blog.paperspace.com/super-resolution-generative-adversarial-networks/>

<https://chatgpt.com/>

....



Stanford Dogs Dataset

1

Collecter des données

2

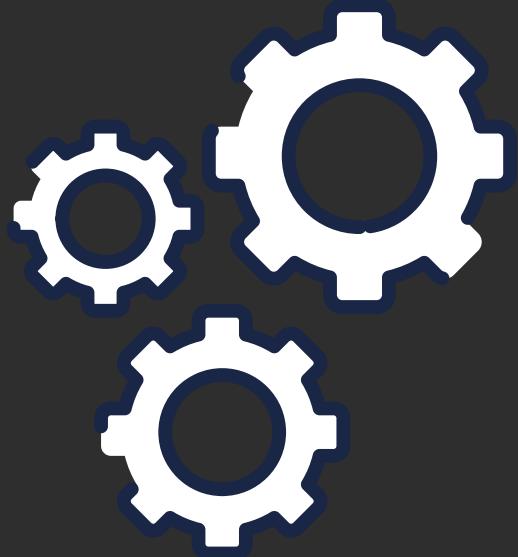
Préparer et explorer

3

Modéliser

4

Comprendre



Stanford Dogs Dataset

**120 dossiers d'images :
nombre de races : 120
nombre d'images : 20 580**

Pour ce projet nous allons d'abord faire une étude de faisabilité sur 3 races puis rajouter plus de race si l'étude de faisabilité est positive

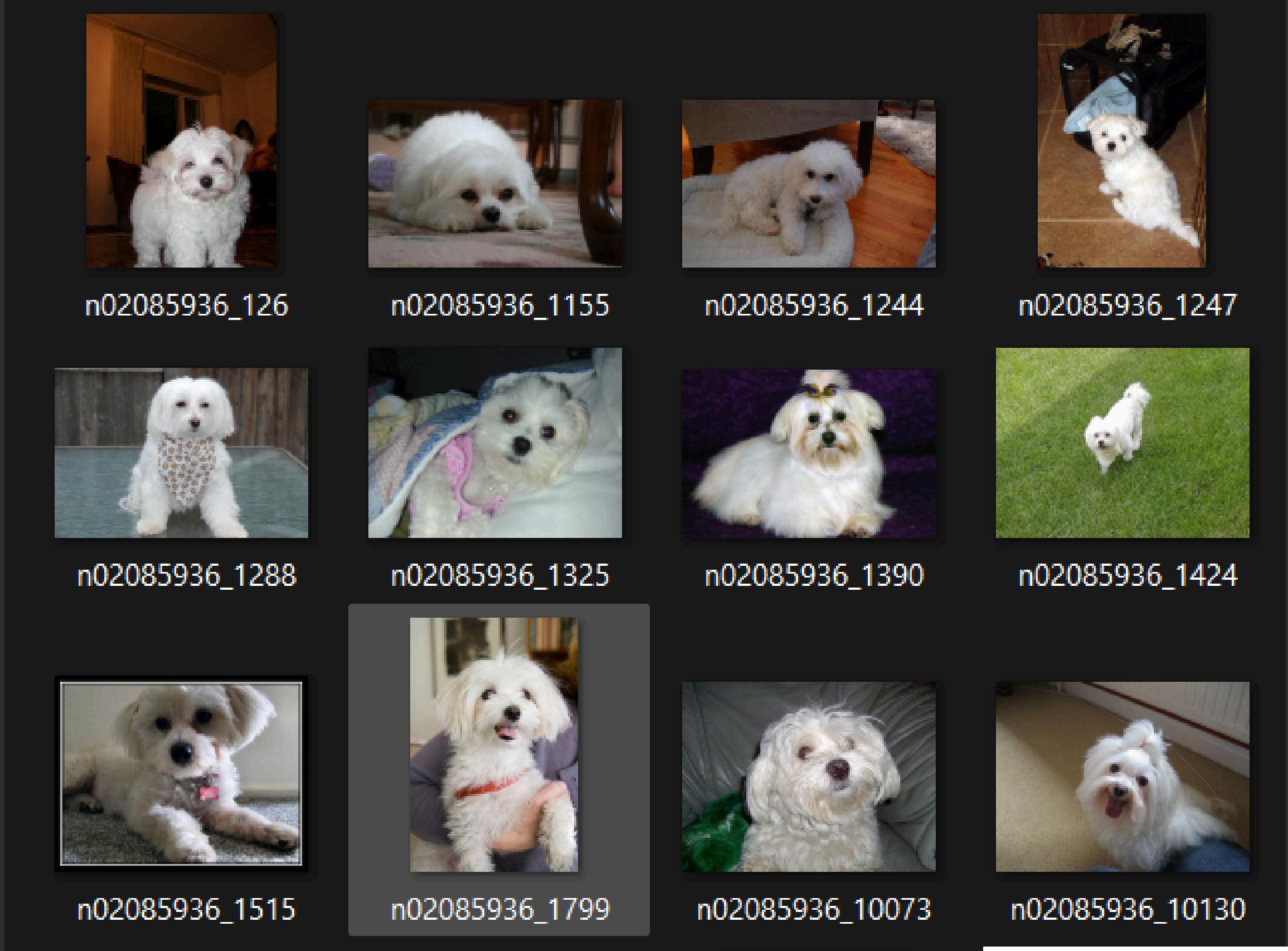
- n02107683-Bernese_mountain_dog
- n02107908-Appenzeller
- n02107312-miniature_pinscher
- n02107574-Greater_Swiss_Mountain_dog
- n02106662-German_shepherd
- n02107142-Doberman
- n02106382-Bouvier_des_Flandres
- n02106550-Rottweiler
- n02106030-collie
- n02106166-Border_collie
- n02105505-komondor
- n02105641-Old_English_sheepdog
- n02105855-Shetland_sheepdog
- n02105251-briard
- n02105412-kelpie
- n02105056-groenendael
- n02105162-malinois
- n02100583-vizsla
- n02099712-Labrador_retriever
- n02099849-Chesapeake_Bay_retriever
- n02099429-curly-coated_retriever
- n02099601-golden_retriever
- n02098286-West_Highland_white_terrier
- n02098413-Lhasa
- n02099267-flat-coated_retriever
- n02097658-silky_terrier
- n02098105-soft-coated_wheaten_terrier
- n02097298-Scotch_terrier
- n02097474-Tibetan_terrier
- n02097130-giant_schnauzer
- n02097209-standard_schnauzer
- n02096585-Boston_bull
- n02097047-miniature_schnauzer
- n02096204-Australian_terrier

<http://vision.stanford.edu/aditya86/ImageNetDogs/>



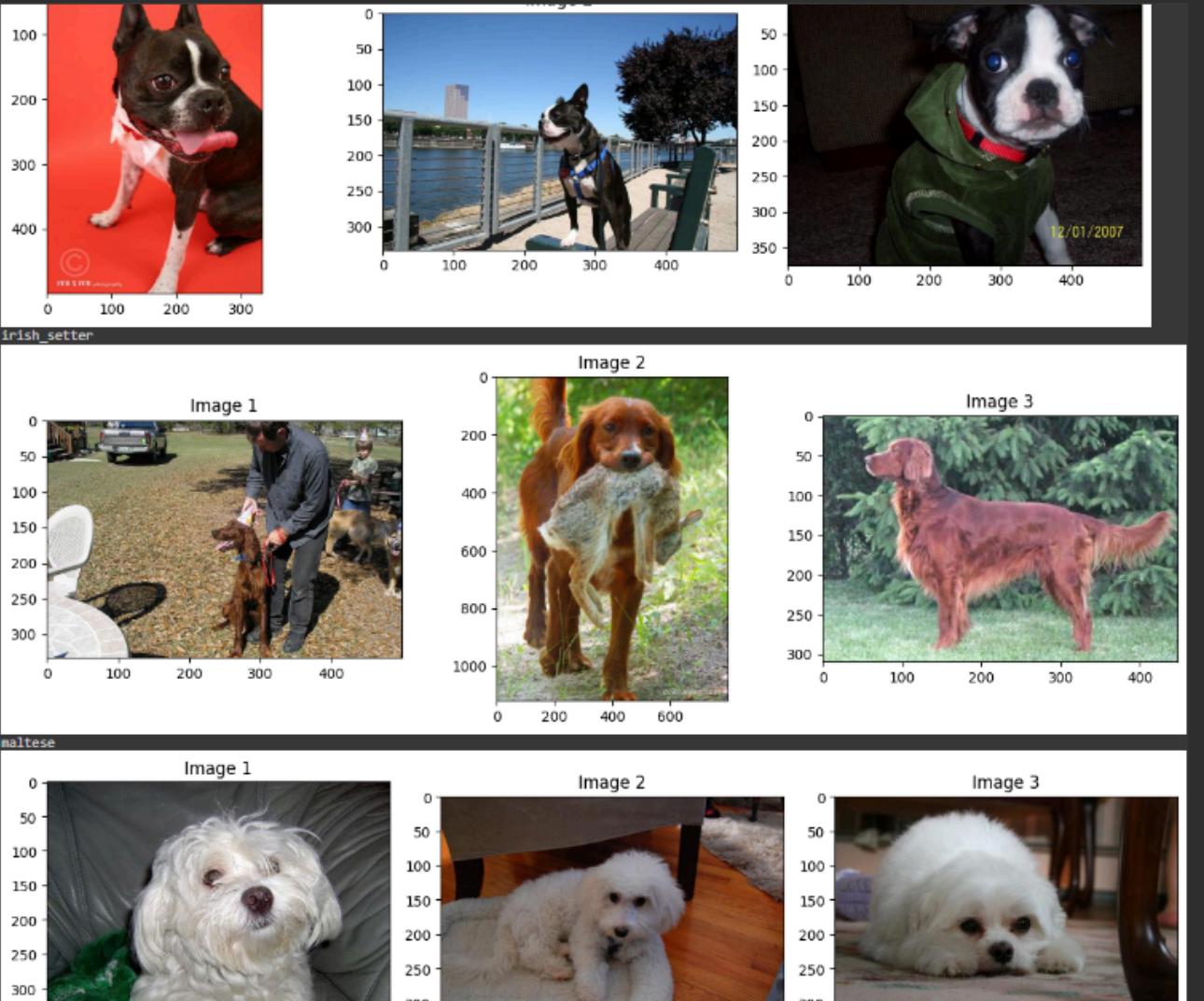
Stanford Dogs Dataset

n02085936-Maltese_dog



Stanford Dogs Dataset

- Les images sont stockées dans le folder **.stfdogs20580**
- Le dataset est téléchargé et préparé automatiquement grâce à une fonction personnalisée dans la classe **STFDogs20580**.



- La préparation consiste à diviser N images en jeux d'entraînement et de validation, chacun ayant des versions HR et LR.
- La réduction de dimension est réalisée par interpolation bicubique avec OpenCV.



Stanford Dogs Dataset

Preprocessing / Séparation du jeu de données

On crée ici le dataset d'entraînement et de validation

```
stfdogs20580_train = STFDOGS20580(subset='train', n_images=N_IMAGES)
stfdogs20580_valid = STFDOGS20580(subset='valid', n_images=N_IMAGES)
```

```
stfdogs20580_train.dataset(batch_size=16, random_transform=True)
stfdogs20580_valid.dataset(batch_size=16, random_transform=True, repeat_count=1)
```



Stanford Dogs Dataset

Entrainement du générateur sur perception et discriminateur entropie croisée binaire

```
from train2 import SrganTrainer, SrganGeneratorTrainer

gan_generator = generator()
gan_generator.load_weights(weights_file('pre_generator.weights.h5'))

: gan_trainer = SrganTrainer(generator=gan_generator, discriminator=discriminator(
    gan_trainer.train(train_ds,
                       steps=5000)

50/5000, perceptual loss = 0.3369, discriminator loss = 1.5018
100/5000, perceptual loss = 0.3322, discriminator loss = 0.5175
150/5000, perceptual loss = 0.3242, discriminator loss = 0.0723
200/5000, perceptual loss = 0.3449, discriminator loss = 0.1604
250/5000, perceptual loss = 0.3309, discriminator loss = 0.0941
300/5000, perceptual loss = 0.3268, discriminator loss = 0.0109
```



Stanford Dogs Dataset

Preprocessing / Séparation du jeu de données

On crée ici le dataset d'entraînement et de validation

```
stfdogs20580_train = STFDOGS20580(subset='train', n_images=N_IMAGES)
stfdogs20580_valid = STFDOGS20580(subset='valid', n_images=N_IMAGES)
```

```
stfdogs20580_train.dataset(batch_size=16, random_transform=True)
stfdogs20580_valid.dataset(batch_size=16, random_transform=True, repeat_count=1)
```



Stanford Dogs Dataset

Démonstration de faisabilité



Stanford Dogs Dataset

Conclusion

Nous avons démontré la faisabilité après avoir étudié des articles sur la super-résolution. L'implémentation du SRGAN fonctionne, mais il reste des optimisations possibles en ajustant les hyperparamètres, en réalisant un entraînement complet, et en augmentant les images d'entraînement.



Stanford Dogs Dataset

On utilise HuggingFace pour la démonstration



Hugging Face fournit une plateforme permettant de partager, héberger et déployer facilement des modèles de machine learning, en particulier pour le traitement du langage naturel.

SRGAN: Super-Resolution GAN

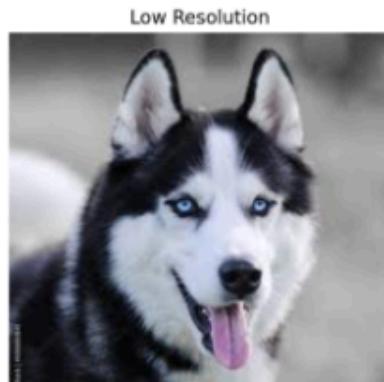
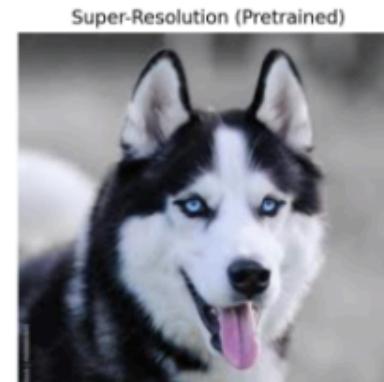
Cette application permet d'améliorer la résolution d'images de chiens à l'aide de SRGAN.

Téléchargez une image basse résolution

Drag and drop file here
Limit 200MB per file • JPG, PNG, JPEG

Browse files

Résultats de quelques images prédefinies:

 Low Resolution	 Super-Resolution (Pretrained)	 Super-Resolution (GAN)
 Low Resolution	 Super-Resolution (Pretrained)	 Super-Resolution (GAN)