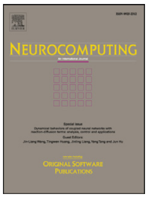
Neurocomputing 398 (2020) 293–303

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Generative collaborative networks for single image super-resolution

Mohamed El Amine Seddik[a,b,*], Mohamed Tamaazousti[a], John Lin[a,c]

[a] CEA/LIST/DIASI/LVIC, Gif-sur-Yvette F-91191, France
[b] CentraleSupelec/L2S, 3 rue Joliot Curie, Gif-sur-Yvette 91192, France
[c] ISIT-UMR6284 CNRS/Auvergne University, Clermont-Ferrand, France

ARTICLE INFO

Article history:
Received 31 August 2018
Revised 3 February 2019
Accepted 27 February 2019
Available online 19 October 2019

Keywords:
Super-resolution
Deep learning
GANs
Perceptual loss

ABSTRACT

A common issue of deep neural networks-based methods for the problem of Single Image Super-Resolution (SISR), is the recovery of finer texture details when super-resolving at large upscaling factors. This issue is particularly related to the choice of the objective loss function. In particular, recent works proposed the use of a VGG loss which consists in minimizing the error between the generated high resolution images and ground-truth in the feature space of a Convolutional Neural Network (VGG19), pre-trained on the very "large" ImageNet dataset. When considering the problem of super-resolving images with a distribution "far" from the ImageNet images distribution (*e.g.,* satellite images), their proposed *fixed* VGG loss is no longer relevant. In this paper, we present a general framework named *Generative Collaborative Networks* (GCN), where the idea consists in optimizing the *generator* (the mapping of interest) in the feature space of a *features extractor* network. The two networks (generator and extractor) are *collaborative* in the sense that the latter "helps" the former, by constructing discriminative and relevant features (not necessarily *fixed* and possibly learned *mutually* with the generator). We evaluate the GCN framework in the context of SISR, and we show that it results in a method that is adapted to super-resolution domains that are "far" from the ImageNet domain.

1. Introduction

The super-resolution problem ($\mathcal{P}_{sr}$) consists in estimating a high resolution (HR) image from its corresponding low resolution (LR) counterpart. $\mathcal{P}_{sr}$ finds a wide range of applications and has attracted much attention within the community of computer vision [2–4]. Generally, the considered optimization objective of supervised methods to solve $\mathcal{P}_{sr}$ is the minimization of the mean squared error (MSE) between the recovered HR image and ground-truth. This class of methods are known to be suboptimal to reconstruct texture details at large upscaling factors. In fact, since MSE consists in a pixel-wise images differences, its ability to recover high texture details is limited [1,5–7]. Furthermore, the minimization of MSE maximizes the Peak Signal-to-Noise-Ratio (PSNR) metric, which is commonly used for the evaluation of $\mathcal{P}_{sr}$ methods [8].

In order to correctly recover finer texture details when super-resolving at large upscaling factors, a recent (state-of-the-art) work [1] defined a perceptual loss which is a combination of an adversarial loss and a VGG loss. The former encourages solutions perceptually hard to distinguish from the HR ground-truth images, while the latter consists in using high-level feature maps of the VGG network [9] pre-trained on ImageNet [10]. When considering the problem of super-resolving images from a target-domain *different* than ImageNet (*e.g.,* satellite images), the features produced by the pre-trained VGG network on the source domain (ImageNet) are suboptimal and no longer relevant for the target domain. In fact, transfer-learning methods are known to be efficient only when the source and target domains are close enough [11–13]. In this work, we present a general framework which we call *Generative Collaborative Networks* (GCN), where the main idea consists in optimizing the generator (*i.e.,* the mapping of interest) in the feature space of a network which we shall refer to as a *features extractor* network. The two networks are said to be *collaborative* in the sense that the features extractor network "helps" the generator by constructing (here, learning) relevant features. In particular, we applied our framework to the problem of single image super-resolution, and we demonstrated that it results in a method that is more adapted (compared to SRGAN [1]) when super-resolving images from a domain that is "far" from the ImageNet domain.

The rest of the paper is organized as follows. In Section 2 we present the state of the art on the problem of single image super-resolution. We describe our Generative Collaborative Networks framework in Section 3. Section 4 presents our proposed method for the super resolution task and related experimental

[*] Corresponding author at: CEA/LIST/DIASI/LVIC, F-91191 Gif-sur-Yvette, France.
E-mail addresses: mohamedelamine.seddik@cea.fr (M.E.A. Seddik), mohamed.tamaazousti@cea.fr (M. Tamaazousti), john.lin@cea.fr (J. Lin).

https://doi.org/10.1016/j.neucom.2019.02.068
0925-2312/© 2019 Published by Elsevier B.V.

results. Section 5 provides some discussions and concludes the article.

2. Related work

The problem of super-resolution has been tackled with a large range of approaches. In the following, we will consider the problem of *single* image super-resolution ($\mathcal{P}_{sisr}$) and thus the approaches that recover HR images from multiple images [14,15] are out of the scope of this paper. First approaches to solve $\mathcal{P}_{sisr}$ were filtering-based methods (*e.g.*, linear, bicubic or Lanczos [16] filtering). Even if these methods are generally very fast, they usually yield overly smooth textures solutions [6]. Most promising and powerful approaches are learning-based methods which consist in establishing a mapping between LR images and their HR counterparts (supposed to be known). Initial work was proposed by Freeman et al. [17]. This method has been improved in [18,19] by using compressed sensing approaches. Patch-based methods combined with machine learning algorithms were also proposed: in [20,21] upsampling a LR image by finding similar LR training patches in a low dimensional space (using neighborhood embedding approaches) and a combination of the HR patches counterparts are used to reconstruct HR patches. A more general mapping of example pairs (using kernel ridge regression) was formulated by Kim and Kwon [22]. Similar approaches used Gaussian process regression [23], trees [24] or Random Forests [25] to solve the regression problem introduced in [22]. An ensemble method-based approach was adopted in [26] by learning multiple patch regressors and selecting the most relevant ones during the test phase.

Convolutional neural networks (CNN)-based approaches outperformed other $\mathcal{P}_{sisr}$ approaches, by showing excellent performance. Authors in [27] used an encoded sparse representation as a prior in a feed-forward CNN, based on the learned iterative shrinkage and thresholding algorithm of [28]. An end-to-end trained three layer deep fully convolutional network, based on bicubic interpolation to upscale the input images, was used in [29,30] and achieved good $\mathcal{P}_{sisr}$ performances. Further works suggested that enabling the network to directly learn the upscaling filters, can remarkably increase performance in terms of both time complexity and accuracy [31,32]. In order to recover visually more convincing HR images, Johnson et al. [33] and Bruna et al. [34] used a closer loss function to perceptual similarity. More recently, authors in [1] defined a perceptual loss which is a combination of an adversarial loss and a VGG loss. The latter consists in minimizing the error between the recovered HR image and ground-truth in the high-level feature space of the pre-trained VGG network [9] on ImageNet [10]. This method notably outperformed CNN-based methods for the problem $\mathcal{P}_{sisr}$.

3. Generative collaborative networks

3.1. Proposed framework

Consider a problem $\mathcal{P}$ of learning a mapping function $\mathcal{F}$, parameterized by $\theta_{\mathcal{F}}$, that transforms images from a domain $\mathcal{X}$ to a domain $\mathcal{Y}$, given a training set of N pairs $\{(x_i, y_i)\}_{i=1}^{N} \in \mathcal{X} \times \mathcal{Y}$. Denote by $p_{\mathcal{X}}$ and $p_{\mathcal{Y}}$ the probability distributions respectively over $\mathcal{X}$ and $\mathcal{Y}$. In addition, we introduce a given *features extractor* function denoted Φ, parameterized by θ_Φ, that maps an image $y \in \mathcal{Y}$ to a certain euclidean feature space $\mathcal{S}_\Phi$ of dimensionality d. The mappings $\mathcal{F}$ and Φ are typically feed-forward Convolutional Neural Networks. The Generative Collaborative Networks (GCN) framework consists in learning the mapping function $\mathcal{F}$ by minimizing a given loss function[1] in the space of features $\mathcal{S}_\Phi$, between the generated images (through $\mathcal{F}$) and ground-truth. Formally,

$$\hat{\theta}_{\mathcal{F}} = \underset{\theta_{\mathcal{F}}}{arg\,min}\, \frac{\lambda_1}{N d} \sum_{i=1}^{N} \sum_{j=1}^{d} \left(\Phi_j(y_i) - \Phi_j(\mathcal{F}(x_i))\right)^2 + \lambda_2\, \Omega(\theta_{\mathcal{F}}), \quad (1)$$

where $\Omega(\theta_{\mathcal{F}})$ is a certain regularization term (detailed below) on the weights $\theta_{\mathcal{F}}$ and λ_1 and λ_2 are summation coefficients. The two networks $\mathcal{F}$ and Φ are collaborative in the sense that, the latter learns specific features of the domain $\mathcal{Y}$ and "helps" the former, as it is learned in the space $\mathcal{S}_\Phi$. An important question arises about how to learn the mapping Φ. In following, we describe different classes of methods depending on the learning strategy of Φ. In fact, the features extractor function Φ can take different forms and be learned by different strategies. In particular, we distinguish two learning strategies (illustrated in Fig. 2), which we shall call *disjoint-learning* and *joint-learning*. The four following cases belong to the *disjoint-learning* strategy:

(1.a) When Φ is the *identity operator* ($\Phi = Id$). In that case, the objective in Eq. (1) becomes a simple pixel-wise MSE loss function. We refer to this class of methods by $\mathcal{P}/mse$.

(1.b) When Φ corresponds to a *random feature map* neural network, that is to say, the weights θ_Φ are set randomly according to a given distribution μ. We refer to this class of methods by $\mathcal{P}/ran$.

(1.c) When Φ is a part of a model that solves a *reconstruction problem* (jointly with an auxiliary mapping function $\Psi : \mathcal{S}_\Phi \to \mathcal{Y}$), by minimizing the pixel-wise ℓ_2-loss function between the reconstructed images (through Ψ) and ground-truth:

$$(\hat{\theta}_\Phi, \_) = \underset{(\theta_\Phi, \theta_\Psi)}{arg\,min}\, \frac{1}{N \dim(\mathcal{Y})} \sum_{i=1}^{N} \sum_{j=1}^{\dim(\mathcal{Y})} \times \left((y_i)_j - (\Psi \circ \Phi(y_i))_j\right)^2. \quad (2)$$

Notably, this strategy allows for the learning of reconstruction features which are different from classification-based features. We refer to this class of methods by $\mathcal{P}/rec$.

(1.d) When Φ is trained to solve a *multi-label classification problem* [1], that is to say, when labels are available for the domain $\mathcal{Y}$. More precisely, it exists a dataset $\{(y_i, c_i)\}_{i=1}^{n} \in \mathcal{Y} \times \{1, \ldots, m\}$ of n images labelled among m classes and Φ is learned to minimize the following objective:

$$(\hat{\theta}_\Phi, \_) = \underset{(\theta_\Phi, \theta_\Psi)}{arg\,max}\, \mathbb{P}\{\Psi \circ \Phi(y_i) = c_i \,|\, y_i\,; \; i \in \{1, \ldots, m\}\}, \quad (3)$$

where $\Psi : \mathcal{S}_\Phi \to \{1, \ldots, m\}$. We refer to this class of methods by $\mathcal{P}/cla$.

The features extractor function Φ can also be trained *jointly* with the desired mapping function $\mathcal{F}$. Indeed, as in the GANs paradigm, one can use a discriminator to distinguish the generated images (through $\mathcal{F}$) and ground-truth, and thus learn more relevant and specific features for the problem of interest $\mathcal{P}$. In particular, the *joint-learning* strategy contains two cases:

(2.a) When Φ is a part of a *discriminator*. $\mathcal{D} = \Psi \circ \Phi : \mathcal{Y} \to \{0, 1\}$ that classifies the generated images (through $\mathcal{F}$) and ground-truth. $\mathcal{D}$ is optimized in an alternating manner along with $\mathcal{F}$ to solve the adversarial min-max problem [35]:

$$\min_{\theta_{\mathcal{F}}} \max_{(\theta_\Phi, \theta_\Psi)} \mathbb{E}_{y \sim p_{\mathcal{Y}}}[\log \Psi \circ \Phi(y)] + \mathbb{E}_{x \sim p_{\mathcal{X}}}[\log\{1 - \Psi \circ \Phi \circ \mathcal{F}(x)\}]. \quad (4)$$

The adversarial loss (second term of Eq. (4)) can thus be seen as a regularization of the parameters $\theta_{\mathcal{F}}$ by affecting this quantity to $\Omega(\theta_{\mathcal{F}})$ in Eq. (1). This regularization

[1] ℓ_2-loss is considered in the following.

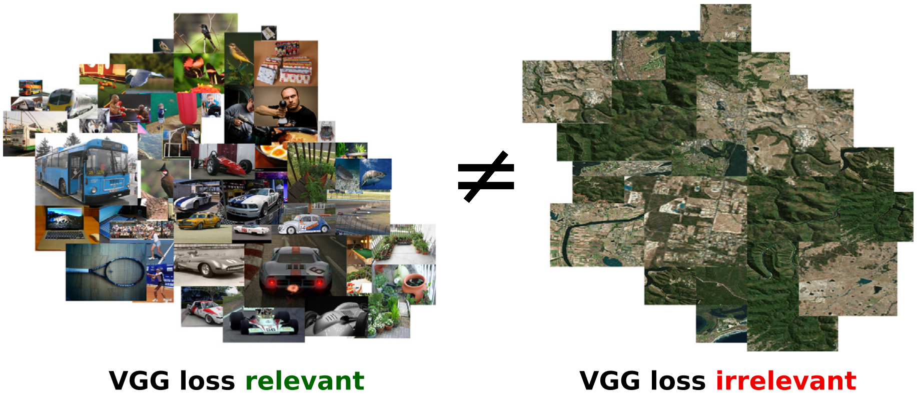


Fig. 1. When super-resolving images from a different domain (*e.g.*, satellite images on the right) than the ImageNet domain (*e.g.*, general objects on the left), the VGG loss introduced by Ledig et al. [1] is no longer relevant. We propose a method that outperforms the SRGAN method [1] when super-resolving satellite images. Our method falls within a large class of methods which constitutes our proposed *Generative Collaborative Networks* framework.

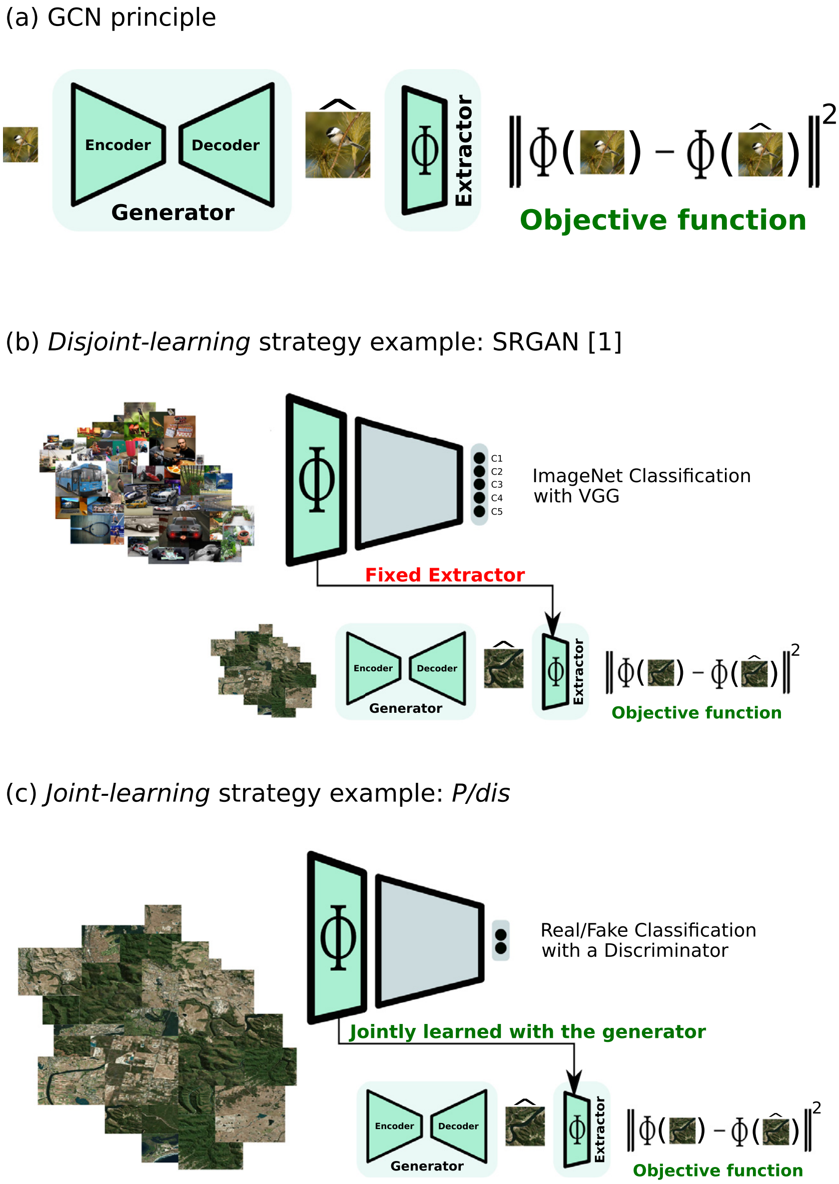

(a) GCN principle
Encoder
Decoder
Generator
Φ
Extractor
Objective function
(b) Disjoint-learning strategy example: SRGAN [1]
Φ
C1
C2
C3
C4
C5
ImageNet Classification with VGG
Fixed Extractor
Encoder
Decoder
Generator
Φ
Extractor
Objective function
(c) Joint-learning strategy example: P/dis
Φ
Real/Fake Classification with a Discriminator
Jointly learned with the generator
Encoder
Decoder
Generator
Φ
Extractor
Objective function

Fig. 2. Overview of the GCN framework with examples of the two learning strategies. The GCN framework consists in optimizing a *generator* in the feature space of an *extractor* as illustrated in (**a**). The extractor can be trained beforehand and used to optimize the generator, which we refer to as *disjoint-learning* strategy (**b**). The extractor can also be optimized jointly with the generator, i.e., using a *joint-learning* strategy (**c**).

Table 1
Existent loss functions of the proposed GCN framework.

Standard methods	$\mathcal{P}/mse$	$\mathcal{P}/cla$	$\mathcal{P}/rec$	$\mathcal{P}/dis$	$\mathcal{P}/dis, rec$
Existence	√[5]	√[41]	✗	✗	✗
Adversarial methods	$\mathcal{P}/adv, mse$	$\mathcal{P}/adv, cla$	$\mathcal{P}/adv$	$\mathcal{P}/adv, rec$	
Existence	√[40]	√[1]	✗	✗	

"pushes" the solution of the problem in Eq. (1) to the manifold of the images in the domain $\mathcal{Y}$. We refer to this class of methods by $\mathcal{P}/adv$. When $\lambda_2 = 0$, we refer to it by $\mathcal{P}/dis$.

(2.b) When Φ is a part of a *discriminator* and an *auto-encoder*. Namely, by optimizing its weights θ_Φ to solve simultaneously, an *adversarial problem* as in Eq. (4); through $\mathcal{D} = \Psi_1 \circ \Phi : \mathcal{Y} \to \{0, 1\}$, and a *reconstruction problem* as in Eq. (2); through a mapping $\Psi_2 : \mathcal{S}_\Phi \to \mathcal{Y}$. We refer to this class of methods by $\mathcal{P}$/*adv,rec* or $\mathcal{P}$/*dis,rec* depending on the value of λ_2 in Eq. (1).

3.2. Existing loss functions

The natural way to learn a mapping from a manifold to another is to use $\mathcal{P}/mse$ methods. It is well known [1,5–7] that this class of methods lead to overly-smooth and poor perceptual quality solutions. In order to handle the mentioned perceptual quality limitation, a variety of methods have been proposed in the literature. First methods used generative adversarial networks (GANs) for generating high perceptual quality images [36,37], style transfer [38] and inpainting [39], namely the class of methods $\mathcal{P}/adv$ with $\lambda_1 = 0$. Authors in [40] proposed to use $\mathcal{P}/mse$ with an adversarial loss ($\lambda_1 > 0$ and $\lambda_2 > 0$) to train a network that super-resolves face images with large upscaling factors. Authors in [33,34] and in [41] used $\mathcal{P}/cla$ by considering respectively Φ =VGG19 and Φ =AlexNet networks as fixed features extractors (learned *disjointly* from the mapping of interest), which result in a more perceptually convincing results for both super-resolution and artistic style-transfer [42,43]. More recently, authors in [1] used $\mathcal{P}$/*cla,adv* by considering Φ =VGG19 as a fixed features extractor combined with an adversarial loss ($\lambda_2 > 0$). To the best of our knowledge, as summarized in Table 1, the use of the other learning strategies of Φ; namely (1.c), (2.a) and (2.b), have not been explored in the literature. We particularly apply these strategies in the context of Single Image Super-Resolution, which results in methods that are more suitable (comparing to the SRGAN method [1]) to super-resolution domains that differ from the ImageNet domain (see Figure 1). The proposed methods as well as the corresponding experiments are presented in the following section.

4. Application of GCN to single image super-resolution

4.1. Proposed methods

In this section, we consider the problem of Single Image Super-Resolution ($\mathcal{P}_{sisr}$). In particular, we suppose we are given N pairs $\{(I_i^{LR}, I_i^{HR})\}_{i=1}^N$ of low-resolution images and their high-resolution counterparts. Recalling our GCN framework (presented in Section 3) the proposed methods for the problem $\mathcal{P}_{sisr}$ are: $\mathcal{P}_{sisr}$*rec*, $\mathcal{P}_{sisr}$ *dis*, $\mathcal{P}_{sisr}$ *dis,rec*, $\mathcal{P}_{sisr}$ *adv* and $\mathcal{P}_{sisr}$ *adv,rec*. We show in the following that the most convincing results are given by $\mathcal{P}_{sisr}$*adv,rec*. In particular, we show on a dataset of satellite images (different from the ImageNet domain) that our method $\mathcal{P}_{sisr}$*adv,rec* outperforms the SRGAN method [1] by a large margin on the considered domain. Note that, as our goal is to show the irrelevance of the VGG loss for some visual domains (different from ImageNet), we do not consider the well-known SR benchmarks (*e.g.*, Set5, Set14, B100, Urban100) for the evaluation, as these benchmarks are relatively close to the ImageNet domain. The evaluation of the different methods is based on *perceptual metrics* [44] which we recall in the following section.

4.2. Evaluation metrics

The evaluation of super-resolution methods (more generally image regression-based methods) requires comparing visual patterns which remains an open problem in computer vision. In fact, classical metrics such as L2/PSNR, SSIM and FSIM often disagree with human judgments (*e.g.*, blurring causes large perceptual change but small L2 change). Thus, the definition of a *perceptual metric* which agrees with humans perception is an important aspect for the evaluation of $\mathcal{P}_{sisr}$ methods. Zhang et al. [44] recently evaluated deep features across different architectures (Squeeze [45], AlexNet [46] and VGG [9]) and tasks (supervised, self-supervised and unsupervised networks) and compared the resulting metrics with traditional ones. They found that deep features outperform all classical metrics (*e.g.*, L2/PSNR, SSIM and FSIM) by large margins on their introduced dataset. As a consequence, deep networks seem to provide an embedding of images which agrees surprisingly well with humans judgments.

Zhang et al. [44] compute the distance between two images x, y with a network[2] Φ in the following way:

$$d_\Phi(x, y) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \| w_l \odot (\Phi^l(x)_{hw} - \Phi^l(y)_{hw}) \|_2^2, \qquad (5)$$

where $\Phi^l(\cdot)$ are the extracted features from layer l and unit-normalized in the channel dimension. w_l is a re-scaling vector of the activations channel-wise at layer l. H_l and W_l are respectively the height and width of the lth feature map.

Thus, we compute the *perceptual error* (PE) of a $\mathcal{P}_{sisr}$ method (a mapping $\mathcal{F}$) on a given test-set of N low-resolution images and their high-resolution counterparts $\Pi = \{(I_i^{LR}, I_i^{HR})\}_{i=1}^N$ as the mean distances between the generated images (through $\mathcal{F}$) and ground-truth as follows:

$$\mathrm{PE}_\Phi(\Pi) = \frac{1}{N} \sum_{i=1}^{N} d_\Phi(\mathcal{F}(I_i^{LR}), I_i^{HR}). \qquad (6)$$

Note that we use the implementation of [44] to compute the perceptual distances $d_\Phi(\cdot, \cdot)$ using six variants which are based on the networks Squeeze [45], AlexNet [46] and VGG [9] and their "perceptual calibrated" versions. The best method is considered to be the one which minimizes the maximum amount of PEs across different networks $\Phi \in$ {Squ, Squ-l, Alex, Alex-l, VGG, VGG-l}.

4.3. Experiments

The overall goal of this section is to validate our statement about the relevance of the VGG loss when super-resolving images from a different domain than the ImageNet domain. To highlight this aspect, we first present the considered datasets, architectures and training details. Then we select the more appropriate method (across the GCN framework methods) for the $\mathcal{P}_{sisr}$ problem based on perceptual metrics [44]. Finally, we compare our proposed method to some baselines and the state-of-the-art SRGAN method [1], on three different datasets (detailed in the following section). We show in particular that our method outperforms SRGAN on the satellite images domain.

[2] The considered networks are Squeeze[45], AlexNet [46] and VGG [9] and their "perceptual calibrated" versions which we refer to respectively as Squeeze-l, AlexNet-l and VGG-l. See [44] and the provided github project within for further details.

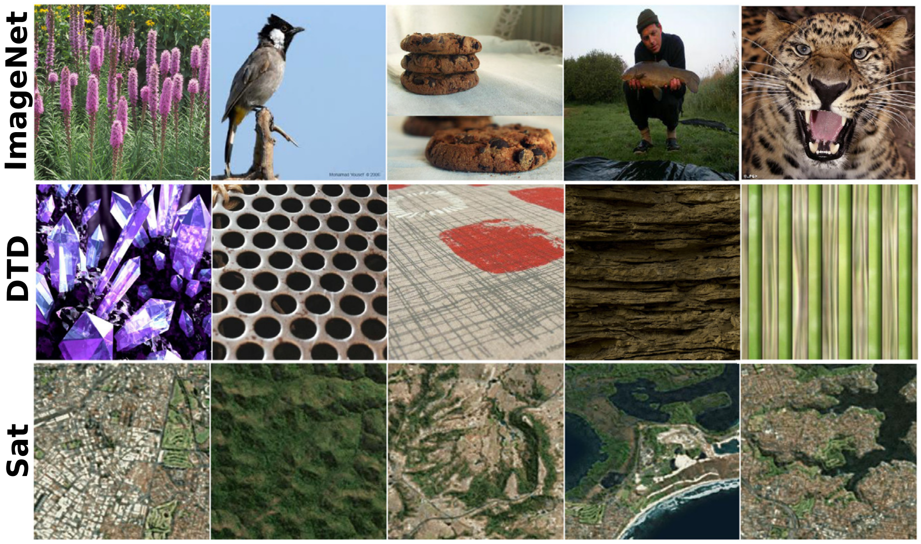


Fig. 3. Examples of images from the considered datasets.

4.3.1. Datasets

The idea of replacing the MSE pixel-wise content loss on the image by a loss function that is closer to perceptual similarity is not new. Indeed, [1] defined a VGG loss on the feature map obtained by a specific layer of the pre-trained VGG19 network and shows that it fixes the inherent problem of overly smooth results which comes with the pixel-wise loss. Nevertheless, VGG19 being trained on ImageNet, their method would not perform particularly well on different images, the distribution of which is far away from that of ImageNet. Therefore, we propose a similar method where the difference is that our features extractor is not pre-trained, but trained jointly with the generator. This removes the aforementioned limitation since the features extractor is trained on the same dataset as the generator and thus extract relevant features.

To show that, we trained our different networks (*i.e.,* with different features extractors) on three distinct datasets (examples of images of these datasets are shown in Fig. 3):

- A subset of *ImageNet* [10], for which we sampled 70,000 images. Since VGG19 was trained on ImageNet for many (more than 300K) iterations, we expect to have similar or worse results than the state-of-the-art method SRGAN from [1] on this database.
- *The Describable Textures Dataset (DTD)* [47], containing 5600 images of textural patterns. These data are relatively close to ImageNet and we show that our method gives convincing results relatively close to SRGAN.
- A dataset containing satellite images[3], which we generated by randomly cropping 256 × 256 images on a 7205 × 7205 satellite image which result in 235,183 images. We particularly show that our method significantly outperforms SRGAN on this dataset. We refer to this dataset by *Sat*.

All experiments are performed with a scale factor of 4× between low- and high-resolutions images and the formers are obtained during the training by down-scaling the original images by a factor 1/4.

4.3.2. Architectures

Our overall goal is to prove that the proposed GCN framework, is adapted to train a generative mapping model and that it surpasses the MSE loss in keeping perceptual similarity in the generated image (whereas the MSE loss tends to smooth things out and lose high frequency details). As opposed to [1]'s work, our framework does not require to have a pre-trained network, like VGG, to extract helpful features for training. In this paper, we focus on the Super Resolution problem. Therefore, we chose our mapping function $\mathcal{F}$, or generator, to be that of Ledig et al. [1]: a feed-forward CNN parametrized by $\theta_{\mathcal{F}}$, composed of 10 residual blocks. These blocks are made of two convolutional layers with 3 × 3 kernels and 64 features maps, each followed by batch normalization and PReLU as activation. The image's size is then increased of a factor 4 by two trained upsamplings. The architecture of all the used discriminators follows the guidelines of Radford et al. [48] as it is composed of convolutional layers, followed by a batch normalization and a LeakyReLU ($\alpha = 0.2$) activation. This block is repeated eight times and each time the number of 3 × 3 kernels increases by a factor 2 (ranging from 64 to 512), a strided convolution is used to reduce the image resolution by 2. Two dense layers and a sigmoid activation then return the discrimination probability. In the case of an auto-encoder (every *Reconstruction* problem), we follow the same architecture for the encoder and a symmetric one for the decoder. Fig. 4 depicts an overview of the architectures for both the generator and the discriminator.

4.3.3. Training details and parameters

All networks were trained[4] on a NVIDIA Geoforce GTX 1070 GPU using the datasets described in Section 4.3.1, which do not contain the (1000) testing images shown as results. We scaled the range of both the LR input images and the HR images to $[-1, 1]$, which explains the tanh activation for the last layer of the generator. All variants of our networks, which differ in their features extractor, were trained from scratch (for the generator and the features extractor) with mini batches of 10 images. We used the Adam optimizer with a learning rate of $2 \cdot 10^{-4}$ and a decay of 0. The generator and the feature extractor are updated alternatively. As we realized training was stable and quite fast, we trained with only 5000 update iterations to pinpoint the best method among the different GCNs. Finally, the regularization parameters in our global loss are set by default as $\lambda_1 = 1$ and $\lambda = 10^{-3}$. As a reminder, our goal here is, given a generator architecture (or mapping function $\mathcal{F}$), to find the best strategy to train it, following

[3] Can be found in http://www.terracolor.net/sample_imagery.html.

[4] A Keras implementation is provided in https://github.com/melaseddik/GCN.

Table 2
Results of the proposed $\mathcal{P}_{sisr}$ methods in terms of traditional metrics (L2 and SSIM) and the *perceptual error* (PE) given by Eq. (6) on different datasets. As we can notice, the method $\mathcal{P}_{sisr}/adv, rec$ outperforms the other methods in the datasets ImageNet and Sat, while $\mathcal{P}_{sisr}/dis, rec$ gives the best results on DTD.

	Methods	Low-level		Perceptual metrics					
		L2	SSIM	Squ	Squ-l	Alex	Alex-l	VGG	VGG-l
ImageNet	$\mathcal{P}_{sisr}/dis$	0.018	$\underline{0.147}$	1.606	0.279	1.470	0.398	2.088	0.358
	$\mathcal{P}_{sisr}/rec$	0.020	0.162	1.723	0.301	1.595	0.425	2.243	0.388
	$\mathcal{P}_{sisr}/dis, rec$	$\underline{0.017}$	$\underline{0.147}$	$\underline{1.587}$	0.279	$\underline{1.420}$	0.382	$\underline{2.052}$	$\underline{0.353}$
	$\mathcal{P}_{sisr}/adv$	0.028	0.202	1.820	**0.222**	1.554	**0.322**	2.598	0.432
	$\mathcal{P}_{sisr}/adv, rec$	**0.016**	**0.141**	**1.533**	$\underline{0.263}$	**1.362**	$\underline{0.368}$	**1.994**	**0.340**
DTD	$\mathcal{P}_{sisr}/dis$	$\underline{0.027}$	0.184	1.873	0.327	1.739	0.440	2.401	0.421
	$\mathcal{P}_{sisr}/rec$	$\underline{0.027}$	$\underline{0.183}$	$\underline{1.851}$	0.320	$\underline{1.726}$	0.438	$\underline{2.398}$	$\underline{0.420}$
	$\mathcal{P}_{sisr}/dis, rec$	**0.023**	**0.167**	**1.703**	0.292	**1.576**	0.404	**2.260**	**0.392**
	$\mathcal{P}_{sisr}/adv$	0.036	0.227	2.077	$\underline{0.281}$	1.812	$\underline{0.375}$	2.770	0.473
	$\mathcal{P}_{sisr}/adv, rec$	0.046	0.236	2.089	**0.277**	1.793	**0.344**	2.796	0.481
Sat	$\mathcal{P}_{sisr}/dis$	**0.011**	**0.129**	$\underline{1.484}$	0.210	1.508	0.356	2.121	$\underline{0.355}$
	$\mathcal{P}_{sisr}/rec$	0.060	0.168	1.705	0.245	1.762	0.423	2.260	0.395
	$\mathcal{P}_{sisr}/dis, rec$	**0.011**	$\underline{0.138}$	1.493	0.215	$\underline{1.435}$	0.351	**2.108**	0.372
	$\mathcal{P}_{sisr}/adv$	0.030	0.214	1.719	$\underline{0.181}$	1.627	$\underline{0.306}$	2.711	0.419
	$\mathcal{P}_{sisr}/adv, rec$	$\underline{0.018}$	0.183	**1.359**	**0.140**	**1.310**	**0.220**	$\underline{2.115}$	**0.344**

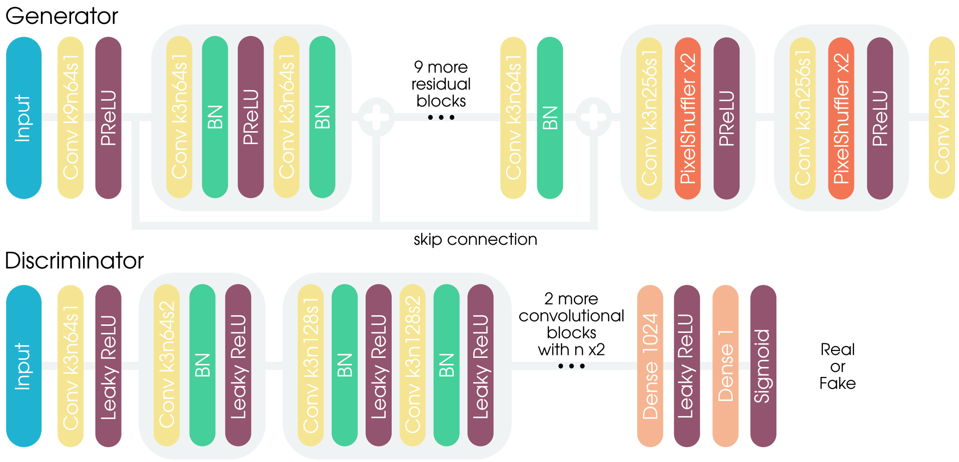


Fig. 4. Overview of the used architectures for the generator and the discriminator. We have considered the same architectures as that of Ledig et al. [1].

our GCNs paradigms. The best method is then further compared to baselines.

4.3.4. Features extractor selection

As we said above, we investigated the ability of different features extractor to construct relevant perceptual feature maps for training and improving the rendering quality of the generator. In order to select the best learning strategy given a certain dataset, we train the generator on each dataset (presented in Section 4.3.1) using the different learning strategies: $\mathcal{P}_{sisr}/rec$, $\mathcal{P}_{sisr}/dis$, $\mathcal{P}_{sisr}/dis,rec$, $\mathcal{P}_{sisr}/adv$ and $\mathcal{P}_{sisr}/adv,rec$. Note that, the features extractor for all the considered methods correspond to the first layer of the discriminators (or encoder-decoders). In fact, as the problem $\mathcal{P}_{sisr}$ consists in recovering low-level perceptual cues, we limited our study to the first layer.

Table 2 summarizes the results of the proposed $\mathcal{P}_{sisr}$ methods in terms of low-level metrics (L2 and SSIM) and perceptual metrics [44] which are given by Eq. (6). We notice from this table that the method $\mathcal{P}_{sisr}/adv, rec$ performs relatively well on the datasets ImageNet and Sat in terms of perceptual metrics. While $\mathcal{P}_{sisr}/dis, rec$ gives better results on the DTD dataset. The main difference between these two methods is that the former considers an adversarial loss on the objective function while the latter does not consider the adversarial term. This explains the reason why $\mathcal{P}_{sisr}/adv, rec$ does not perform well on DTD. In fact, texture images belong to a complex manifold and their distribution is relatively hard to fit by a generative model.

Fig. 5 shows qualitative results of the different proposed methods on the different presented datasets. Generally, the methods which were trained with an additional adversarial loss ($\mathcal{P}_{sisr}/adv$ and $\mathcal{P}_{sisr}/adv,rec$) output images of higher quality (on the datasets ImageNet and Sat) as GANs were introduced to do just so: generate images that follow the distribution of the dataset. Among these two *adversarial* methods, it seems to us (as suggested by the quantitative results of Table 2) that $\mathcal{P}_{sisr}/adv,rec$ (column (c) of Fig. 5) is able to detect and render more details, due to its ability to generate more relevant features as the features extractor Φ is learned to solve a *multi-task* problem; namely a *discrimination* and a *reconstruction* problem, in particular, this method allows for the learning of both classification and reconstruction-based features. We will thus further investigate the $\mathcal{P}_{sisr}/adv,rec$ method for the comparison to the baseline and the state-of-the-art method SRGAN [1], on the satellite images domain.

4.3.5. against baseline methods on the satellite images domain

Our main objective is to show that the VGG loss function (namely, the SRGAN method [1]) is no longer relevant when super-resolving images from a domain different than the ImageNet domain. In particular, by considering the satellite images domain, we show in this section that the selected method from the previous section ($\mathcal{P}_{sisr}/adv, rec$) outperforms some baselines, which

Table 3
Comparison of our method $\mathcal{P}_{sisr}/adv, rec$ with baselines and the SRGAN method [1] on the satellite images domain, in terms of classical metrics (L2 and SSIM) and perceptual metrics [44].

	Methods	Low-level		Perceptual metrics					
		L2	SSIM	Squ	Squ-l	Alex	Alex-l	VGG	VGG-l
Sat	$\mathcal{P}_{sisr}/mse$	**0.011**	**0.134**	1.873	0.245	1.855	0.411	2.536	0.419
	$\mathcal{P}_{sisr}/adv, mse$	0.082	0.197	$\underline{1.458}$	$\underline{0.205}$	1.466	0.352	$\underline{2.125}$	$\underline{0.347}$
	SRGAN [1]	0.228	0.188	1.510	0.220	$\underline{1.361}$	$\underline{0.282}$	2.230	0.412
	$\mathcal{P}_{sisr}/adv, rec$	$\underline{0.018}$	$\underline{0.183}$	**1.359**	**0.140**	**1.310**	**0.220**	**2.115**	**0.344**

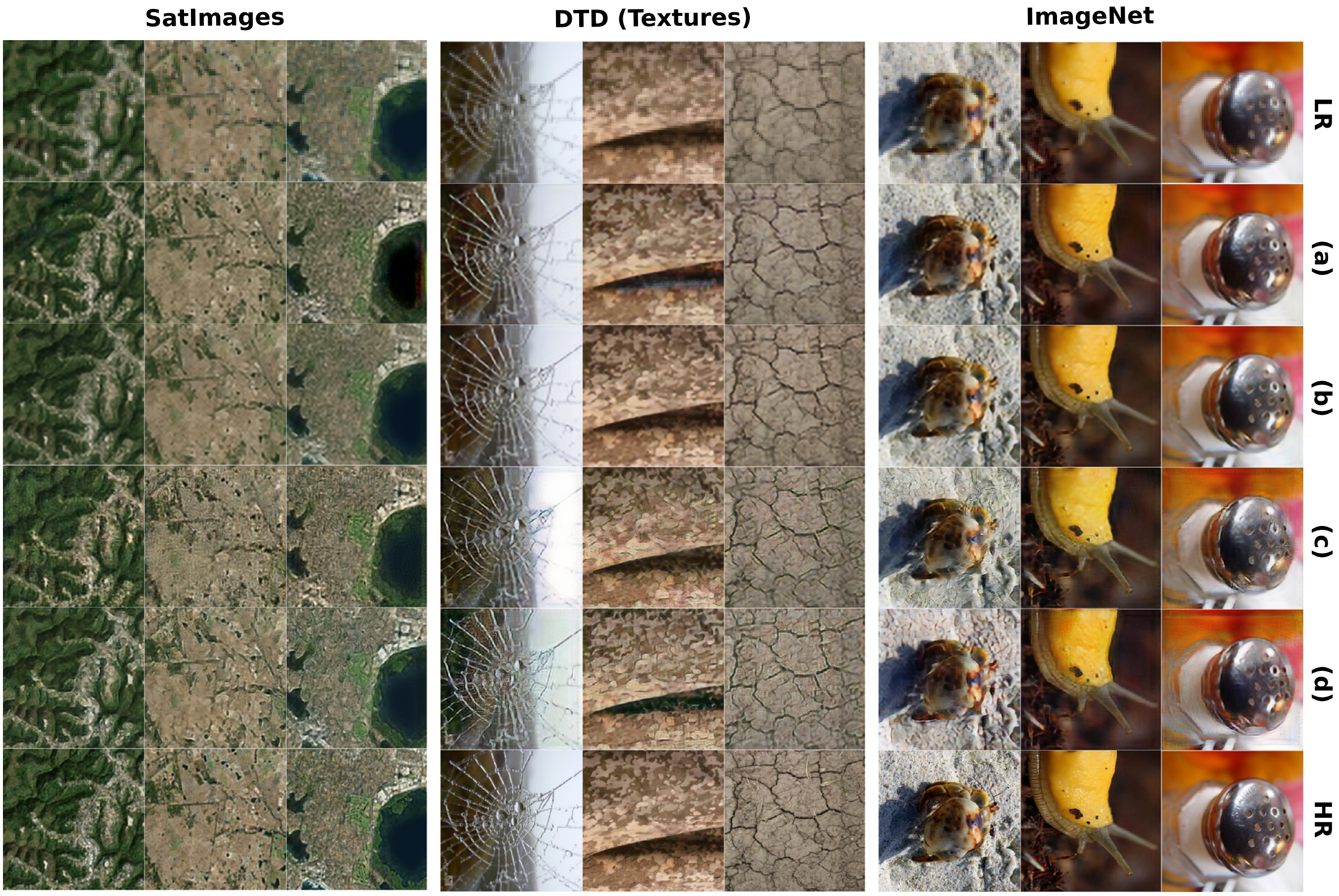


Fig. 5. Rows refer to the different considered Datasets. Columns refer to methods and ground-truth images: **LR** and **HR** refer to the low- and high-resolution pairs. The different used methods are: **(a)** $\mathcal{P}_{sisr}/rec$, **(b)** $\mathcal{P}_{sisr}/dis,rec$, **(c)** $\mathcal{P}_{sisr}/adv$ and **(d)** $\mathcal{P}_{sisr}/adv,rec$. Best view in PDF.

are $\mathcal{P}_{sisr}/mse$ (pixel-wise MSE loss) and $\mathcal{P}_{sisr}/adv,mse$ (pixel wise MSE loss combined with an adversarial loss), and the state-of-the-art super-resolution method, SRGAN [1]. Note that all the methods use the same architectures (depicted in Fig. 4) for the generator and discriminator and are trained on the same domain (here, on satellite images). Our purpose being to show the relevance of the proposed method on a domain "far" from the ImageNet domain, we do not consider standard SR benchmarks, which are raltively "close" to the ImageNet domain.

Table 3 presents quantitative results, in terms of classical metrics (L2 and SSIM) and perceptual metrics given by Eq. (6), of the different methods on the Sat dataset. As we can notice, our method $\mathcal{P}_{sisr}/adv, rec$ outperforms the other methods in terms of perceptual metrics. Knowing that the perceptual metrics agree with human judgments [44], these results validate the effectiveness of the $\mathcal{P}_{sisr}/adv, rec$ method. Note also that even if SRGAN [1] is optimized to minimize a VGG loss, it does not give the lowest perceptual errors in terms of the perceptual metrics VGG and VGG-l, this is due to the fact that the VGG features are not relevant for the satellite images domain. In addition, $\mathcal{P}_{sisr}/adv, rec$ gives the lowest perceptual errors in terms of the perceptual metrics Alex and Alex-l which agrees with a human perception. In fact, AlexNet network may more closely match the architecture of the human visual cortex [49].

Fig. 6 shows some qualitative results of different methods on a patch of an image from the Sat dataset. As we can notice, the $\mathcal{P}_{sisr}/adv, rec$ method gives the perceptually closest result to the ground-truth image, which agrees with the quantitative results of Table 3.

4.3.6. Further results

In this section, we provide further qualitative and quantitative comparisons to the considered baselines of the previous section. In particular, we consider all the presented datasets for the comparisons. Qualitative results are provided in Fig. 7. SRGAN performs better on ImageNet, which is not that surprising considering our features extractor was trained much less than VGG19 used in [1] and the VGG features being more relevant for images from the

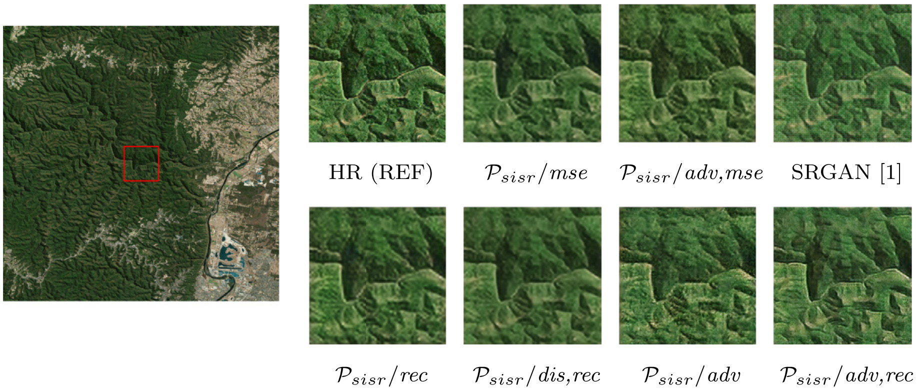


Fig. 6. Results of different $\mathcal{P}_{sisr}$ methods on a patch of an image from the Sat dataset.

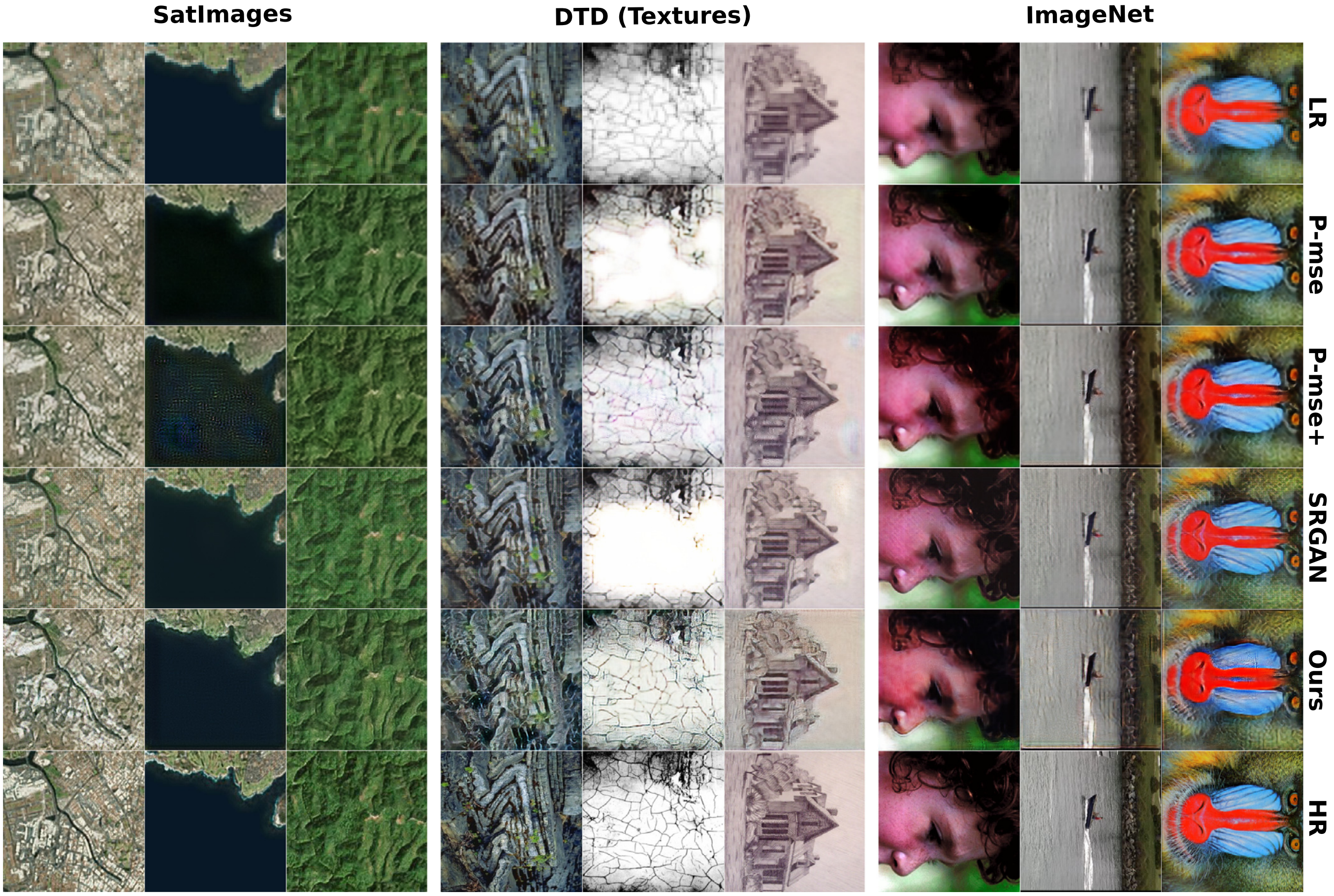


Fig. 7. Rows refer to the different Datasets. Columns refer to methods and ground-truth images: **LR** and **HR** refer to the low- and high-resolution pairs. **P-mse+** refers to the method $\mathcal{P}_{sisr}/mse$ with an adversarial loss ($\lambda_2 > 0$), **SRGAN** for the method in [1] and our method $\mathcal{P}_{sisr}/adv,rec$. Best view in PDF.

ImageNet domain. Nonetheless, we do have sharper images than the MSE based methods, although we show some artifact (especially on the boat) which we attribute to the competition between the content and adversarial losses. On DTD though, we can see the benefit of our method over a pre-trained VGG loss. Indeed, SRGAN is blurrier on both the house (first row) and the cliff (third row), in spite of having less artifacts than our method. On the "cracks" example (second row), SRGAN even totally obliterates the details in the center. Finally, results on the dataset Sat, which is the most different dataset compared to ImageNet, are the most compelling. Our method generates super resolved images that are really close to the real high resolution images, while we can clearly see imperfections on SRGAN's results because of VGG19 which was not trained to detect perceptual features on satellite images.

Quantitative results are summarized in Table 4. As shown in [1,44], the standard quantitative measures such as L2 and SSIM fail to highlight image quality according to the human visual system. In fact, while the results of $\mathcal{P}_{sisr}/mse$ are overly smooth perceptually, it has the lowest L2 and SSIM errors on Sat. However, perceptual metrics agree with what we assess qualitatively: SRGAN performs best on ImageNet but not on Sat, the distribution of which is the farthest from ImageNet. Actually, SRGAN ranks third of all four methods on Sat, just before $\mathcal{P}_{sisr}/adv,mse$, while still performing best on DTD which still is pretty close to ImageNet. This shows that the VGG features become less and less relevant as

Table 4
Comparison of our methods $\mathcal{P}_{sisr}/adv, rec$ and $\mathcal{P}_{sisr}/dis, rec$ with baselines and the SRGAN method [1] on the datasets ImageNet (a subset of 200,000 randomly selected images) and DTD, in terms of classical metrics (L2 and SSIM) and perceptual metrics [44].

	Methods	Low-level		Perceptual metrics					
		L2	SSIM	Squ	Squ-l	Alex	Alex-l	VGG	VGG-l
ImageNet	$\mathcal{P}_{sisr}/mse$	0.017	0.146	1.568	0.280	1.435	0.391	2.064	0.349
	$\mathcal{P}_{sisr}/adv, mse$	0.020	0.156	1.634	0.241	1.397	0.329	2.223	0.384
	SRGAN	0.028	0.170	**1.303**	**0.177**	**1.084**	**0.225**	2.045	0.342
	$\mathcal{P}_{sisr}/adv, rec$	**0.016**	**0.141**	1.533	0.263	1.362	0.368	**1.994**	**0.340**
DTD	$\mathcal{P}_{sisr}/mse$	0.029	0.185	1.972	0.342	1.856	0.470	2.479	0.434
	$\mathcal{P}_{sisr}/adv, mse$	0.025	0.188	1.880	0.268	1.586	0.349	2.512	0.430
	SRGAN	0.031	0.191	**1.557**	**0.209**	**1.298**	**0.241**	2.308	0.393
	$\mathcal{P}_{sisr}/dis, rec$	**0.023**	**0.167**	1.703	0.292	1.576	0.404	**2.260**	**0.392**

Table 5
Comparison of the proposed $\mathcal{P}_{sisr}$ methods in terms of traditional metrics (L2 and SSIM) and the *perceptual error* (PE) given by Eq. (6) on all the considered datasets. In terms of perceptual metrics, the proposed $\mathcal{P}_{sisr}$ methods rank in the second position after SRGAN [1] on the datasets ImageNet and DTD, while they outperform all the baselines on the satellite images domain which is far from the ImageNet domain.

	Methods	Low-level		Perceptual metrics					
		L2	SSIM	Squ	Squ-l	Alex	Alex-l	VGG	VGG-l
ImageNet	$\mathcal{P}_{sisr}/mse$	0.017	0.146	1.568	0.280	1.435	0.391	2.064	0.349
	$\mathcal{P}_{sisr}/adv, mse$	0.020	0.156	1.634	0.241	1.397	0.329	2.223	0.384
	SRGAN	0.028	0.170	**1.303**	**0.177**	**1.084**	**0.225**	2.045	0.342
	$\mathcal{P}_{sisr}/dis$	0.018	0.147	1.606	0.279	1.470	0.398	2.088	0.358
	$\mathcal{P}_{sisr}/rec$	0.020	0.162	1.723	0.301	1.595	0.425	2.243	0.388
	$\mathcal{P}_{sisr}/dis, rec$	0.017	0.147	1.587	0.279	1.420	0.382	2.052	0.353
	$\mathcal{P}_{sisr}/adv$	0.028	0.202	1.820	0.222	1.554	0.322	2.598	0.432
	$\mathcal{P}_{sisr}/adv, rec$	**0.016**	**0.141**	1.533	0.263	1.362	0.368	**1.994**	**0.340**
DTD	$\mathcal{P}_{sisr}/mse$	0.029	0.185	1.972	0.342	1.856	0.470	2.479	0.434
	$\mathcal{P}_{sisr}/adv, mse$	0.025	0.188	1.880	0.268	1.586	0.349	2.512	0.430
	SRGAN	0.031	0.191	**1.557**	**0.209**	**1.298**	**0.241**	2.308	0.393
	$\mathcal{P}_{sisr}/dis$	0.027	0.184	1.873	0.327	1.739	0.440	2.401	0.421
	$\mathcal{P}_{sisr}/rec$	0.027	0.183	1.851	0.320	1.726	0.438	2.398	0.420
	$\mathcal{P}_{sisr}/dis, rec$	**0.023**	**0.167**	1.703	0.292	1.576	0.404	**2.260**	**0.392**
	$\mathcal{P}_{sisr}/adv$	0.036	0.227	2.077	0.281	1.812	0.375	2.770	0.473
	$\mathcal{P}_{sisr}/adv, rec$	0.046	0.236	2.089	0.277	1.793	0.344	2.796	0.481
Sat	$\mathcal{P}_{sisr}/mse$	**0.011**	0.134	1.873	0.245	1.855	0.411	2.536	0.419
	$\mathcal{P}_{sisr}/adv, mse$	0.082	0.197	1.458	0.205	1.466	0.352	2.125	0.347
	SRGAN	0.228	0.188	1.510	0.220	1.361	0.282	2.230	0.412
	$\mathcal{P}_{sisr}/dis$	**0.011**	**0.129**	1.484	0.210	1.508	0.356	2.121	0.355
	$\mathcal{P}_{sisr}/rec$	0.060	0.168	1.705	0.245	1.762	0.423	2.260	0.395
	$\mathcal{P}_{sisr}/dis, rec$	**0.011**	0.138	1.493	0.215	1.435	0.351	**2.108**	0.372
	$\mathcal{P}_{sisr}/adv$	0.030	0.214	1.719	0.181	1.627	0.306	2.711	0.419
	$\mathcal{P}_{sisr}/adv, rec$	0.018	0.183	**1.359**	**0.140**	**1.310**	**0.220**	2.115	**0.344**

the dataset's distribution part from ImageNet. On the other hand, our training framework allows to construct relevant features on any (never seen) dataset. Thus our method $\mathcal{P}_{sisr}/adv,rec$ performs best on Sat. Our method performing better than $\mathcal{P}_{sisr}/adv,mse$ also shows that our framework helps finding detail preserving features. Fig. 7 provides the results of the different baselines and our method on some examples of the considered datasets. We notice from these images that our method $\mathcal{P}_{sisr}/adv, rec$ recovers finer details on the different datasets while it outperforms the considered baselines on satellite images. Table 5 summarizes the results of the different methods on the considered datasets through the paper. From these results, we make the following conclusions:

- When the considered domain is far enough from the ImageNet domain, the VGG loss introduced by Ledig et al. [1] is no longer relevant.
- The VGG network cannot be fine-tuned when considering a domain for which there is no available labels for the images (*e.g.*, satellite images). Thus, the SRGAN method cannot be exploited efficiently in this case.
- Our framework results in a method ($\mathcal{P}_{sisr}/adv, rec$) that outperforms some baselines and the SRGAN method on the satellite images domain.
- Even on a domain close to the ImageNet domain (*e.g.*, texture images), one can find within our framework methods which give almost similar results to the SRGAN method, while the later is based on VGG features and thus need to train the VGG network on the whole ImageNet dataset.

5. Conclusion and perspectives

In this paper, we propose a general framework named Generative Collaborative Networks (GCN) which generalizes the existing methods for the problem of learning a mapping between two domains. The GCN framework highlights that there is a learning strategy of mappings that is not explored in the literature. In particular, the optimization of these mappings in the feature space of a features extractor network, which is mutually learned at the same time as the considered mapping (*joint-learning* strategy). The GCN framework was evaluated in the context of super-resolution on three datasets (ImageNet [10], *DTD* [47] and satellite images). We have shown that the proposed *joint-learning* strategy leads to a method that outperforms the state of the art [1] which uses a pre-trained features extractor network (VGG19 on ImageNet). Specifically, this holds when the domain of interest is "far" from the ImageNet domain (*e.g.*, satellite images or images from the medical domain[5]). However, note that even for domains close to the ImageNet domain, the proposed method gives convincing (almost similar to [1]) results without using the whole ImageNet dataset to learn the features extractor network (as performed in [1]).

In this work, we systematically designed the proposed methods by using the first layer of the features extractor networks, while it could be interesting to evaluate in more detail the impact of this choice regarding the learning strategy. Moreover, the impact of the selected layer may also depend on the considered dataset. More generally, the GCN framework offers a large vision on the wide variety of existing loss functions used in the literature of learning mappings-based problems (*e.g.,* super-resolution, image completion, artistic style transfer, etc.). In fact, we show that these loss functions can be simply reformulated, in the proposed framework, as a certain combination of a particular type of features extractor networks ($\mathcal{P}|rec$, $\mathcal{P}|dis$, $\mathcal{P}|dis,rec$, $\mathcal{P}|adv$ and $\mathcal{P}|adv,rec$) and a particular learning strategies (*joint-learning* or *disjoint-learning*). Therefore it will be interesting to explore this promising framework in other learning mappings-based problems.

Declaration of Competing Interest

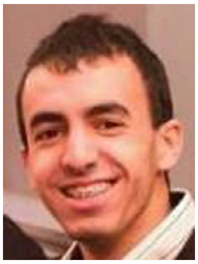
None.

References

[1] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, CVPR (2016).
[2] K. Nasrollahi, T.B. Moeslund, Super-resolution: a comprehensive survey, Mach. Vis. Appl. 25 (6) (2014) 1423–1468.
[3] Q. Yang, R. Yang, J. Davis, D. Nistér, Spatial-depth super resolution for range images, in: Computer Vision and Pattern Recognition, CVPR., IEEE, 2007, pp. 1–8.
[4] W.W. Zou, P.C. Yuen, Very low resolution face recognition problem, IEEE Trans. Image Process. 21 (1) (2012) 327–340.
[5] P. Gupta, P. Srivastava, S. Bhardwaj, V. Bhateja, A modified psnr metric based on hvs for quality assessment of color images, in: 2011 International Conference on Communication and Industrial Application (ICCIA), IEEE, 2011, pp. 1–4.
[6] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[7] Z. Wang, E.P. Simoncelli, A.C. Bovik, Multiscale structural similarity for image quality assessment, in: 2004 Conference Record of the Thirty-Seventh Asilomar Conference onSignals, Systems and Computers, 2, Ieee, 2003, pp. 1398–1402.
[8] C.-Y. Yang, C. Ma, M.-H. Yang, Single-image super-resolution: a benchmark, in: European Conference on Computer Vision, Springer, 2014, pp. 372–386.
[9] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CVPR (2016).
[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, CVPR 2009., IEEE, 2009, pp. 248–255.
[11] Y. Tamaazousti, H. Le Borgne, C. Hudelot, Mucale-net: multi categorical-level networks to generate more discriminating features,
[12] Y. Tamaazousti, H. Le Borgne, C. Hudelot, M.E.A. Seddik, M. Tamaazousti, Learning more universal representations for transfer-learning, TPAMI (2019).
[13] A. Karbalayghareh, X. Qian, E.R. Dougherty, Optimal bayesian transfer learning, IEEE Trans. Signal Process. 66 (2018) 3724–3739.
[14] S. Borman, R.L. Stevenson, Super-resolution from image sequences-a review, in: 1998 Midwest Symposium on Circuits and Systems, 1998. Proceedings, IEEE, 1998, pp. 374–378.
[15] S. Farsiu, M.D. Robinson, M. Elad, P. Milanfar, Fast and robust multiframe super resolution, IEEE Trans. Image Process. 13 (10) (2004) 1327–1344.
[16] C.E. Duchon, Lanczos filtering in one and two dimensions, J. Appl. Meteorol. 18 (8) (1979) 1016–1022.
[17] W.T. Freeman, T.R. Jones, E.C. Pasztor, Example-based super-resolution, IEEE Comput. Graph. Appl. 22 (2) (2002) 56–65.
[18] W. Dong, L. Zhang, G. Shi, X. Wu, Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization, IEEE Trans. Image Process. 20 (7) (2011) 1838–1857.
[19] R. Zeyde, M. Elad, M. Protter, On single image scale-up using sparse-representations, in: International Conference on Curves and Surfaces, Springer, 2010, pp. 711–730.
[20] R. Timofte, V. De, L. Van Gool, Anchored neighborhood regression for fast example-based super-resolution, in: Computer Vision (ICCV)., IEEE, 2013, pp. 1920–1927.
[21] R. Timofte, V. De Smet, L. Van Gool, A+: Adjusted anchored neighborhood regression for fast super-resolution, in: Asian Conference on Computer Vision, Springer, 2014, pp. 111–126.
[22] K.I. Kim, Y. Kwon, Single-image super-resolution using sparse regression and natural image prior, IEEE Trans. Pattern Anal. Mach. Intell. 32 (6) (2010) 1127–1133.
[23] H. He, W.-C. Siu, Single image super-resolution using gaussian process regression, in: Computer Vision and Pattern Recognition (CVPR)., IEEE, 2011, pp. 449–456.
[24] J. Salvador, E. Perez-Pellitero, Naive bayes super-resolution forest, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 325–333.
[25] S. Schulter, C. Leistner, H. Bischof, Fast and accurate image upscaling with super-resolution forests, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3791–3799.
[26] D. Dai, R. Timofte, L. Van Gool, Jointly optimized regressors for image super-resolution, in: Computer Graphics Forum, 34, Wiley Online Library, 2015, pp. 95–104.
[27] Z. Wang, D. Liu, J. Yang, W. Han, T. Huang, Deep networks for image super-resolution with sparse prior, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 370–378.
[28] K. Gregor, Y. LeCun, Learning fast approximations of sparse coding, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, Omnipress, 2010, pp. 399–406.
[29] C. Dong, C.C. Loy, K. He, X. Tang, Learning a deep convolutional network for image super-resolution, in: European Conference on Computer Vision, Springer, 2014, pp. 184–199.
[30] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 295–307.
[31] C. Dong, C.C. Loy, X. Tang, Accelerating the super-resolution convolutional neural network, in: European Conference on Computer Vision, Springer, 2016, pp. 391–407.
[32] W. Shi, J. Caballero, F. Huszár, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1874–1883.
[33] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: European Conference on Computer Vision, Springer, 2016, pp. 694–711.
[34] J. Bruna, P. Sprechmann, Y. LeCun, Super-resolution with deep convolutional sufficient statistics, ICLR (2015).
[35] C.K. Sønderby, J. Caballero, L. Theis, W. Shi, F. Huszár, Amortised map inference for image super-resolution, ICLR (2016).
[36] E.L. Denton, S. Chintala, R. Fergus, et al., Deep generative image models using a laplacian pyramid of adversarial networks, in: Advances in Neural Information Processing Systems, 2015, pp. 1486–1494.
[37] M. Mathieu, C. Couprie, Y. LeCun, Deep multi-scale video prediction beyond mean square error, ICLR (2015).
[38] C. Li, M. Wand, Combining markov random fields and convolutional neural networks for image synthesis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2479–2486.
[39] R. Yeh, C. Chen, T.Y. Lim, M. Hasegawa-Johnson, M.N. Do, Semantic image inpainting with perceptual and contextual losses, CoRR (2016).
[40] X. Yu, F. Porikli, Ultra-resolving face images by discriminative generative networks, in: European Conference on Computer Vision, Springer, 2016, pp. 318–333.
[41] A. Dosovitskiy, T. Brox, Generating images with perceptual similarity metrics based on deep networks, in: Advances in Neural Information Processing Systems, 2016, pp. 658–666.
[42] L. Gatys, A.S. Ecker, M. Bethge, Texture synthesis using convolutional neural networks, in: Advances in Neural Information Processing Systems, 2015, pp. 262–270.
[43] L.A. Gatys, A.S. Ecker, M. Bethge, Image style transfer using convolutional neural networks, in: Computer Vision and Pattern Recognition (CVPR)., IEEE, 2016, pp. 2414–2423.
[44] R. Zhang, P. Isola, A.A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, IEEE/CVF (2018).
[45] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, Squeezenet: alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size, ICLR (2017).
[46] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[47] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[48] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, ICLR (2015).

[5] This domain is particularly relevant for the proposed framework as it seems very far from the ImageNet domain. Unfortunately, we have not found a big amount of publicly available data (to the best of our knowledge) for medical images which prevented us from considering this domain through the paper.

[49] D.L. Yamins, J.J. DiCarlo, Using goal-driven deep learning models to understand sensory cortex, Nat. Neurosci. 19 (3) (2016) 356.



Mohamed El Amine Seddik received a Master of Engineering in Data Science from Institut Mines-Telecom de Lille (with the final year completed Telecom Paris Tech) and a Master Degree in Vision and Machine Learning from ENS Cachan in 2017. He is currently a PhD student in the computer vision lab at the CEA LIST, interested in random matrix theory for machine learning, deep learning and image analysis.

Mohamed Tamaazousti received his Master's Degree in applied mathematical from the University of Orléans in 2009 and the Ph.D. degree in computer vision from the University Blaise Pascal in 2013. He is currently a permanent researcher at CEA LIST. His main research interests include structure from motion for rigid scenes, real time vision-based localization and reconstruction (SLAM) for autonomous system. He is also interested in augmented and diminished reality applications.

John Lin is a PhD student of the Université Clermont Auvergne, in Clermont-Ferrand, France in partnership with the ENCOV research group at the Institut Pascal and the LVIC laboratory at CEA-LIST under the supervision of Professor Adrien Bartoli. He received his Master's Degree in Optics and Computer Science from the Institut d'Optique Graduate School in Palaiseau, France, in 2017. His research objectives are to model the light sources to improve the rendering of augmented reality applications through a learning-based approach, making his interests reach out to domains such as Computer Graphics, Generative Neural Networks and light material interactions.