Pseudo-Cyclic Network for Unsupervised Colorization with Handcrafted Translation and Output Spatial Pyramids

Rémi Ratajczak*
Univ Lyon, Lyon 2, LIRIS, F-69676
Lyon, France
remi.ratajczak@liris.cnrs.fr

Carlos Crispim-Junior
Univ Lyon, Lyon 2, LIRIS, F-69676
Lyon, France
carlos.crispim-junior@liris.cnrs.fr

Béatrice Fervers
Centre Léon Bérard
Lyon, France
beatrice.fervers@lyon.unicancer.fr

Elodie Faure
Centre Léon Bérard
Lyon, France
elodie.faure@gustaveroussy.fr

Laure Tougne
Univ Lyon, Lyon 2, LIRIS, F-69676
Lyon, France
laure.tougne@liris.cnrs.fr

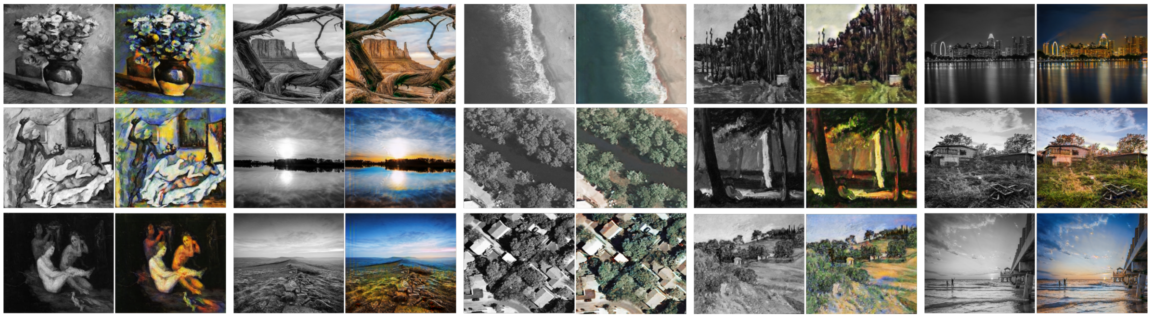


Figure 1: Examples of colorized paintings, landscape and remote sensing images obtained with SpyncoGan.

ABSTRACT

We present a novel pseudo-cyclic adversarial learning approach for unsupervised colorization of grayscale images. We investigate the use of a non-trainable, lightweight and well-defined Handcrafted Translation to enforce the generation of realistic images and replace one of the two deep convolutional generative adversarial neural networks classically used in cyclic models. Additionally, we propose to use Output Spatial Pyramids to jointly constrain the deep latent spaces of an encoder-decoder generator to preserve spatial structures and improve the quality of the generated images. We demonstrate the interest of our approach compared with the state of the art on standard datasets (paintings, landscapes, aerial, thumbnails) that we modified for the purpose of colorization. We evaluate colorization quality of the generated images along the training with deterministic and reproducible criteria. In complement, we demonstrate the ability of our method to generate representations that are prone to make a classification network generalize well to slightly different color spaces. We believe our approach has potential applications in arts and cultural heritage to produce alternative representations without requiring paired data.

CCS CONCEPTS

• **Computing methodologies → Computer vision**; **Unsupervised learning**; **Neural networks**; **Image processing**.

KEYWORDS

Pseudo-Cyclic Network, Handcrafted Translation, Spatial Pyramids, Unsupervised Colorization, GAN

ACM Reference Format:
Rémi Ratajczak, Carlos Crispim-Junior, Béatrice Fervers, Elodie Faure, and Laure Tougne. 2019. Pseudo-Cyclic Network for Unsupervised Colorization with Handcrafted Translation and Output Spatial Pyramids. In *1st Workshop on Structuring and Understanding of Multimedia heritAge Contents (SUMAC '19), October 21, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3347317.3357243

1 INTRODUCTION

In the framework of representation learning, it is often assumed that a neural network could be trained to virtually approximate any function from a set of observations. These automatic methods proved to perform well in practice, but they often disregard early mathematical models derived from physical observations (*e.g.*, handcrafted functions). Semi-physical modeling used in Systems Identification and Control Theory try to find a trade-off between handcrafted and learned functions by regressing theoretically well

[*] author is also with the ADEME, Angers, France, and with the Centre Léon Bérard. This work is also supported by LABEX IMU (ANR-10-LABX-0088/ ANR-11-IDEX-0007).

SUMAC '19, October 21, 2019, Nice, France

ACM ISBN 978-1-4503-6910-7/19/10…$15.00
https://doi.org/10.1145/3347317.3357243

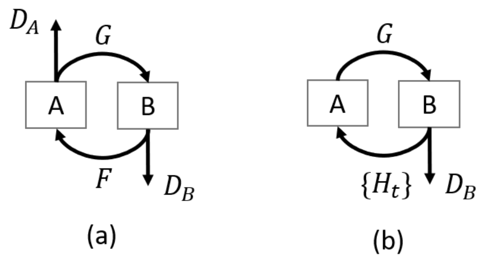


Figure 2: Schematics of (a) a classical Cyclic Network [45] and (b) a Pseudo-Cyclic Network, both in a GAN manifold. G and F are generative networks, D_A and D_B are discriminative networks, $\{H_t\}$ represents a Handcrafted Translation from domain B to A. Note the absence of D_A in (b).

understood models and their parameters [3, 6, 27]. Nonetheless, either handcrafted, identified and fully learned functions could only be defined as sufficient approximations that rely on naturally incomplete observations. Based on this statement, we propose to investigate the following question in a context of colorization: Could we use handcrafted functions to constrain the training of a generative neural network toward a specific latent space without supervision? To this aim, we propose a novel fully automated structure preserving model that we apply to unsupervised colorization. We recall that Unsupervised colorization consists in learning colorization for (and from) grayscale images that are supposed to be not paired with color images, like multi-resolution and multi-dates historical data [31, 32]. While any colorization problem could be formulated as a content preserving style transfer [8, 18, 43], we rather propose to formulate the unsupervised colorization task as an image-to-image translation problem [14]. Image translation methods have proved their effectiveness in unsupervised learning applications by leveraging cyclic constraints [45] on deep convolutional generative adversarial neural networks (DCGAN) [10, 29]. Based on this idea, we introduce a pseudo-cyclic approach built upon empirical priors (Figure 2).

These priors are explicitly introduced in the form of a Handcrafted Translation (H_t) that we combine with Output Spatial Pyramids (*OSP*) applied on the latent space. We propose to use the term of Handcrafted Translation in the context of image to image translation. As opposed to learned translations, a Handcrafted Translation is defined upon prior knowledge of the problem. We define Output Spatial Pyramids as consecutive feature maps with same number of channels and increasing resolution (scale) that are all rescaled to the same scale, so that all the feature maps of an OSP have the same volume after rescaling. We argue that this property of Output Spatial Pyramids allows to jointly optimize multiple layers of a deep convolutional neural network toward a single scale objective using a single mapping function between the deep features and the output space. We demonstrate in our experiments that OSP allow to preserve structure similarity while gradually improving the colorization quality of unpaired images.

In summary, we present in this paper a pseudo-cyclic architecture to colorize unpaired grayscale images based on two core components: Handrafted Translation and Output Spatial Pyramids.

2 RELATED WORK

Image to Image translation methods were initially developed to translate images between two representation spaces, A and B, by using one encoder-decoder network per translation and seeking cycle-consistency (*i.e.*, the translation of an image I from A to B to A should be equal to I). In the remainder, we will note G (resp. F) the network performing the translation from A (resp. B) to B (resp. A) and $G(.)$ (resp. $F(.)$) the corresponding function. In [14], authors assumed the existence of data pairs to constrain the generation of realistic images in both domains. Because data pairs may not always be available, unsupervised cyclic networks [19, 45] were proposed by leveraging GAN loss [10, 29]. Long and nest cyclic networks [22] propose different strategies involving multiple generators to enhance the quality of the translation. Integrating attention mechanisms in cyclic networks [25] was also proposed to improve the translation realism of images representing object instances. Similar to the cyclic networks, crossing-domain networks [9, 21] try to learn a shared latent space between the two, or more [5, 39], domains using GAN loss and variational autoencoders (VAE). Note that, by design, the constraints imposed by the VAE formulation are closely related to the identity loss proposed in [45]. In this study, we build our work upon [45] to explore an alternative approach to train an unsupervised cyclic network for colorization using handcrafted translation H_t as a prior constraint instead of a second DCGAN. We refer to this approach as pseudo-cyclic (Figure 2).

Colorization is a particular kind of Image to Image translation: From a grayscale image, we would like to generate a color image without degrading its content. In this work we follow the recent advancements made on fully automated generative methods using deep convolutional neural networks, but hybrid methods using deep learning in a user in the loop framework should also be mentioned [4, 12, 33, 42]. In [41], the authors trained with supervision a network similar to FCN [23] to regress the *AB* channels (from *LAB* color space) of a grayscale image. In [13], the authors proposed to combine multi-level features in an encoder-decoder network, also to regress the *AB* channels. The multi-level features were obtained from different yet linked deep convolutional branches all trained in a supervised end-to-end fashion. In [2], the authors proposed to learn an attention Gated Recurrent Unit encoder to generate color palettes from text and further perform palette-constrained colorization, in a very similar manner to the integration of global hints (*i.e.*, histograms) for deep colorization proposed in [42]. Our work is closer to [32], where the authors learned an unsupervised cyclic network based on [45] to colorize Very High Resolution historical aerial images using recent color acquisitions and texture replacement. However, in this study, we propose to assess whether a Handcrafted Translation could be used as a surrogate component for one of the two DCGANs in such a cyclic network applied to colorization.

Spatial pyramids are commonly used to generate multi-scale representations and perform advanced tasks like object detection or features matching [24, 44]. Recently, perceptual loss [8] was proposed to constrain the training of generative models by integrating the weighted difference of multi-layer representations of two images passed in a pre-trained network. In parallel, Feature Spatial Pyramids (FSP) were developed to combine the predictions made on deep features at multiple resolutions and thus improve object detection accuracy by adding complementary convolutional filters to an existing backbone network [20]. Hypercolumns [11] were proposed to represent an image by stacking rescaled features issued from multiple layers. They were then successfully applied to colorization [18]. Output Spatial Pyramids (Section 3.2) resemble FSP and Hypercolumns: They allow to constrain the optimization of multi-scale features toward an expected representation. However, they do not rely on stacked features nor selected features individually mapped (with 1x1 convolutions in FSP) from the latent space of a backbone model to a pyramid. They are framed in a generative framework assuming that all deep features in the decoder already have the same number of channels to translate them at full resolution using a unique mapping function and constrain the training of the generator.

3 CORE COMPONENTS

3.1 Handcrafted Translation

On the context of colorization, we recall that our goal is to learn a black box model G that translates (*i.e.*, *G(.)* defines the translation) a grayscale image from $\mathbb{R}^{1\times W\times H}$ (domain *A*) to a color image $\mathbb{R}^{3\times W\times H}$ (domain *B*). For this purpose, H_t could be defined as a handcrafted function that can perform the reverse translation, from *B* to *A*. Since our goal is to colorize an image using an approximate Handcrafted Translation to constrain the latent space of a generative network, and not the opposite, we propose to keep H_t as simple as possible by using one of the earliest representations of grayscale intensities: the weighted sum of the *RGB* channels. We recall that, for a pixel $x^{i,j}$ located at i^{th} row and j^{th} column in an image $I \in \mathbb{R}^{3\times W\times H}$, this operation is expressed by Equation (1), where weights per channels roughly mimic human biological vision.

$$x^{i,j}_{gray} = 0.299 \times x^{i,j}_{R} + 0.587 \times x^{i,j}_{G} + 0.114 \times x^{i,j}_{B} \qquad (1)$$

As this function represents a weighted sum of the color channels with constant weights, it could be applied easily through a deterministic non-trainable 1x1 convolution. As a 1x1 convolution, it has the benefit to preserve most of the spatial properties of its input, like shapes, textures and contours. Consequently, formulating this Handcrafted Translation as a non-trainable 1x1 convolution allows to directly constrain the spatial properties of the generated color images in the latent space. This operation should be opposed to the learned translations made of spatial convolutions [45]. When using learned spatial filters optimized without supervision, there is no guarantee that the translation will preserve spatial structures and high frequency properties, because (1) the spatial convolutions tend to generate smooth images [36], and (2) the generators G and F may learn to satisfy a criterion without seeking for spatial structures consistency between the translated domains, thus hallucinating spatial structures that do not exist [14]. Note that by construction, identity loss could help against these phenomenon. These two properties are particularly interesting in applications like denoising [38], semantic segmentation [1] or object morphing [30], but they are undesired when we want both domains to share fine grained spatial properties, like in colorization.

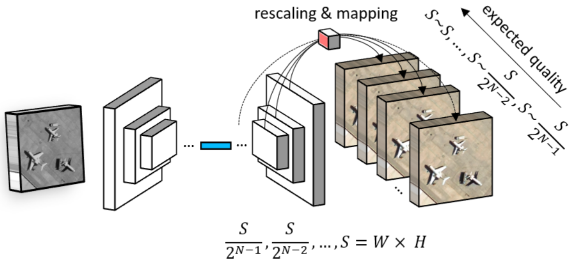

rescaling & mapping
expected quality
$S\sim S, \ldots, S\sim \frac{S}{2^{N-2}}, S\sim \frac{S}{2^{N-1}}$
$\frac{S}{2^{N-1}}, \frac{S}{2^{N-2}}, \ldots, S = W \times H$

Figure 3: Schematic of the Output Spatial Pyramids. S **denotes the scale of the input image (**$W\times H$**). Rescaling (upsampling) is performed with a classical interpolation. Mapping is performed from feature spaces to output domain** B**.**

Finally, one may observe that the above formulation targets the generation of *RGB* images, while previous studies on colorization demonstrated the efficiency to learn generative models targeting the *Lab* or the *HCL* color spaces by decoupling intensity and chroma [13, 18]. We made this choice because these color spaces do not allow the use of a simple linear Handcrafted Translation from (generated) color components (*e.g.*, *ab* channels) to a lightness/grayscale component (*e.g.*, *L* channel) as defined by equation (1), which is the purpose of our study.

3.2 Output Spatial Pyramids

In this study, we propose Output Spatial Pyramids (*OSP*) to jointly constrain the deep features of a generative model toward a single scale objective (Figure 3).

Let G be a convolutional encoder-decoder network from domain A to B, framed in the context of image-to-image translation. Let S be the scale $W \times H$ (width × height) of an image $I \in A$. A general practice is to optimize the weights of G based on the gradient of a loss function $\mathcal{L}$ calculated from the final/most outer output O_{d_1}, whose scale equals S. O_{d_1} is generated from layer l_{d_1} of the decoder (subscript d) of G, and it is expected to be represented in domain B. At the end of the training stage, all the weights of G should have been optimized to produce an O_{d_1} that is as realistic as possible according to an optimization criterion. However, reaching an optimal state in the inner layers of the decoder $\{l_{d_2}, ..., l_{d_N}\}$ may be difficult when a large quantity of parameters is involved.

To help the training of G, and thus the generation of realistic images, we propose to also integrate the early outputs of the decoder, $\{O_{d_2}, ..., O_{d_N}\}$ in the loss functions (Section 4.2). However, since G is phrased in an encoder-decoder manner, successive outputs O_{d_i} and O_{d_j}, with $i \in \{1, .., N-1\}, j = i + 1$, differ by a scale factor. In the following, we will assume a typical scale factor equals to 2. To prevent the loss of details that would occur on proxy ground truth I by gradually downsampling it to match the scales of each of the early outputs, we rather prefer to upsample the early outputs to I's scale before translating them in domain B. Our approach allows to use a single, scale consistent, discriminator for all the outputs, as opposed to [7, 37]. We define an upsampling function $up(.)$ that

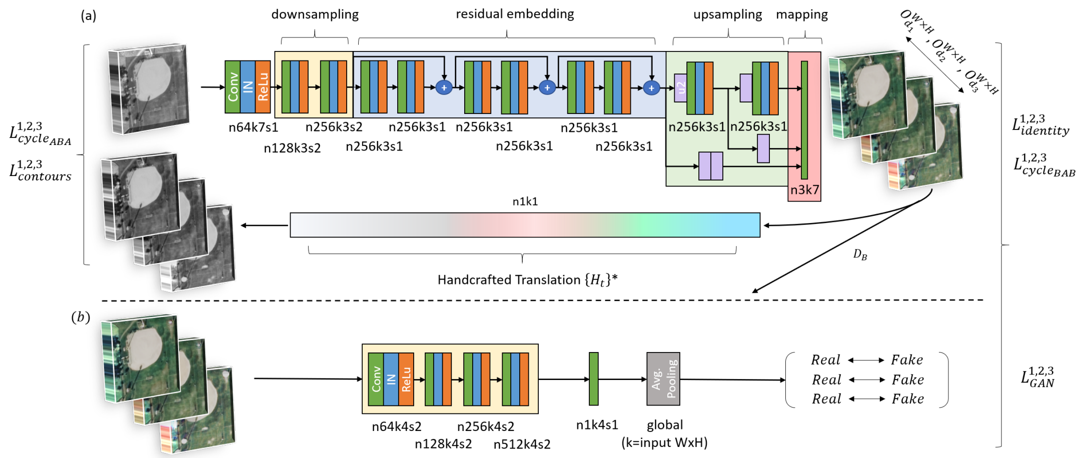


Figure 4: Schematic of our SpyncoGan with $N = 3$ in OSP and a non-trainable Handcrafted Translation H_t from color to grayscale domain. (a) pseudo-cyclic generator, (b) discriminator. Parameters n denote the number of convolutional filters (*e.g.*, n256 indicates 256 filters), k their square spatial dimension, s the stride value and u the upsampling value.

transforms an image from a scale $\frac{S}{2}$ to S. This function could be implemented either with a trained super-resolution network or with a more classical interpolation (*e.g.*, nearest neighbors or bilinear interpolation). Equation (2) denotes the upsampling process applied on O_{d_i} using function composition notations (application of $up(.)$ i times).

$$O_{d_i}^{W\times H} = up^{i-1}(O_{d_i}), i \in \{1, ..., N\} \tag{2}$$

Once early outputs are upsampled in the deep features space, we need to map them to domain B to further calculate a loss function and backpropagate the gradient. Such a mapping could be performed with per-output convolutional layers, similarly to the hypercolumns [11]. However, we do not try to combine deep features together, but we rather want to generate a realistic image from each deep feature separately. Whether different convolutional filters would be used for different outputs, we cannot be sure that the mappings they will learn will be similar. By extension, we cannot ascertain that the deep representations would be targeting the same objective at different scales by looking at the outputs of the mappings.

Since we want to constrain early features toward a same plausible result starting from the embedding to gradually improve the final output, we instead propose to use a single convolutional mapping layer with shared weights for all the $O_{d_i}^{W\times H}$. While we borrow the single mapping idea from the shared regression used FSP [20], we propose to keep the number of features n constant all along the decoder instead of relying on intermediate 1x1 convolutions to handle different number of features at different layers. This approach has two main advantages in the context of image translation: (1) early deep representations are spatially constrained, and (2) we may ascertain that they are representing relevant characteristics for a final task at hand (*e.g.*, colorization, segmentation) by visualizing them through the mapping without introducing additional complexity. In practice, OSP could be understood as a joint constraint on the generator whose intermediate representations target the same objective. In other words, if early features already permit to obtain a perfectly generated image from a discriminator viewpoint, deeper layers would only need to super-resolute the features.

4 SPYNCOGAN

We present SpyncoGan (Spynco for Spatial PYramids and haNdcrafted translation COmbined), a pseudo cyclic network using Handcrafted Translation and Output Spatial Pyramids (OSP) as building blocks. The architecture of a SpyncoGan is presented on Figure 4 with an application to colorization. It could be applied to other generative tasks by carefully redefining H_t (*i.e.*, one may need the re-define H_t for other applications).

4.1 Model Architecture

SpyncoGan is composed of convolutional blocks, each made of a convolutional layer, an instance normalization layer (*IN*) and a ReLu unit. Instance normalization was chosen for its interesting properties in generative tasks compared to batch normalization [35]. Padding, matching half the filter's size k, is systematically applied before a convolution. Downsampling is performed using the stride value when applying convolutional filters (Figure 4). Upsampling is performed with an interpolation before applying a convolution instead of a transposed convolution to reduce the checkerboard artifact effect in the outer outputs [28]. We use separable convolutions [26] in downsampling and upsampling layers of SpyncoGan to reduce the number of learnable parameters. We keep classical convolutions in the residual layers because of their ability to learn an identity mapping through the skipped connections. We did not apply depthwise nor grouped convolutions. For the sake of reducing the number of hyperparameters, we use a constant number of output ($N = 3$) in the Output Spatial Pyramids of SpyncoGan. We use a constant number of convolutional filters $n = 256$ in the residual layers and in the decoder/upsampling layers. We implement the mapping from the deep feature spaces to domain B with a single convolution, which shares its weights with all the outputs of the pyramid to constrain the relationships between successive deep feature spaces.

As opposed to classical cyclic networks, we point out that SpyncoGan relies on a single discriminator and a single generator (instead of two), making it fairly similar to a classical DCGAN, except for the cyclic and multi-scale constraints we impose with proposed H_t and *OSP*. In particular, we replace the second generator of a cyclic network by the Handcrafted Translation defined in section 3, and we discard the second discriminator since we assume the prior knowledge contained in non-trainable H_t is enough. SpyncoGan has a total of ≈ 7.063 million parameters to optimize, including ≈ 4.978 million parameters for the generator and ≈ 2.085 million parameters for the discriminator. Handcrafted Translation is made of 3 non-trainable parameters that are shared among all the outputs of *OSP*.

As described above, we would like to draw the attention of the reader to the fact that we are using a fixed number of convolutional filters n in the upsampling layers (decoder), which is a requirement to learn a single mapping function with the *OSP*. By design, this approach increases the memory needed to train the generator compared with the classical depth/scale ratio (the higher the scale, the less the number of features). This increase is partially absorbed by the separable convolutions and H_t, but it may prevent the training of SpyncoGan-like networks with very deep architectures.

4.2 Loss Functions

In this section, we define the loss functions we used to guide the optimization of SpyncoGan.

We recall that $I \in A$ and $J \in B$ are two images of scale $S = W \times H$. From the *OSP* of $G(I)$, we get N outputs $O_{d_i}^{W\times H}$ as defined by Equation (3). These outputs all have the same scale as I and J after consecutive rescalings. The whole point of these outputs is to optimize the intermediate deep feature spaces by integrating them in the loss functions.

$$\{O_{d_1}^{W\times H}, ..., O_{d_N}^{W\times H}\} = \{G_1(I^{W\times H}), ..., G_N(I^{W\times H})\} = G(I^{W\times H}) \tag{3}$$

where $\forall i \in \{1, ..., N\}, O_{d_i}^{W\times H} \in B$, and G_i represents the output i of the *OSP* of G (*i.e.*, $O_{d_i}^{W\times H}$).

For the sake of concision, the remaining of this paper will omit the superscript $W \times H$, supposed always present for I and J, and notation $G_i(.)$ will be used instead of $O_{d_i}^{W\times H}$ when appropriate. Additionally, we will consider α_i, β_i, γ_i and ζ_i as constant multiplicative factors weighting the contribution of each output in the loss functions. The values used for these parameters in our experiments are detailed in section 5.2, alongside the empirical considerations behind them.

Identity loss [45] aims to avoid mode collapse by seeking inter-domain identity. In Equation (4), we re-frame it as a sum of loss functions calculated on the Output Spatial Pyramid. Note that to apply the identity loss, we have to consider images from domain A and B with the same number of channels (for colorization, this is achieved by artificially replicating single channel grayscale data over three channels).

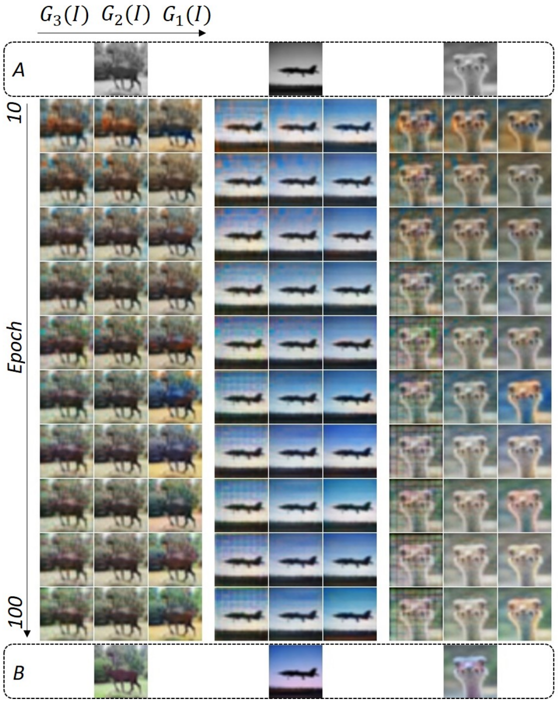


Figure 5: Qualitative results obtained along the training of SpyncoGan on Cifar-10.

$$\mathcal{L}_{identity}^{1,...,N}(G) = \sum_{i=1}^{N} \alpha_i \mathbb{E}_B[\|G_i(J) - J\|_1] \tag{4}$$

Similarly, cycle consistency is defined with the cycle loss of Equation (5), sum of (6) and (7). It aims to constrain the latent spaces using inputs from both domains as proxy ground truths.

$$\mathcal{L}_{cycle}^{1,...,N}(G) = \mathcal{L}_{cycle_{BAB}}^{1,...,N}(G) + \mathcal{L}_{cycle_{ABA}}^{1,...,N}(G) \tag{5}$$

$$\mathcal{L}_{cycle_{BAB}}^{1,...,N}(G) = \sum_{i=1}^{N} \beta_i \mathbb{E}_B[\|(G_i(H_t(J)) - J\|_1] \tag{6}$$

$$\mathcal{L}_{cycle_{ABA}}^{1,...,N}(G) = \sum_{i=1}^{N} \beta_i \mathbb{E}_A[\|H_t(G_i(I)) - I\|_1] \tag{7}$$

GAN loss is defined by Equation (8)[1]. It penalizes the generator and rewards the discriminator when the discriminator is successful at classifying generated images from real images [10]. It constrains the generation of images similar to domain B images. Since we are using a Handcrafted Translation from domain B to domain A, there is no need for a discriminator applied on domain A images: if the images generated by G are able to fool the discriminator, we suppose *a priori* that the Handcrafted Translation will successfully translate them to domain A.

$$\mathcal{L}_{GAN}^{1,...,N}(G, D) = \sum_{i=1}^{N} \gamma_i \mathbb{E}_A[\|1 - D(G_i(I))\|_2^2] + \mathbb{E}_B[\|D(J)\|_2^2] \tag{8}$$

[1] Errata: correct E_B to E_A ; add ^{2} to better match concrete implementation.

Additionally, we make use of the direct spatial relationship permitted by the Handcrafted Translation to constrain the generation of realistic contours. To this aim, we introduce contours loss (Equation (9)). Contours loss is similar to a classical spatial gradient loss (*i.e.*, total variation), but using the Sobel kernel $S_k(.)$ to retrieve the contours. $S_k(.)$ is easily applicable with convolutions. It is also symmetric, defined horizontally as well as vertically, so that it gives more importance to the central pixels resulting in contours that should be more precisely located than with a total variation loss.

$$\mathcal{L}^{1,\dots,N}_{contours}(G) = \sum_{i=1}^{N} \zeta_i \mathbb{E}_A[\|S_k(H_t(G_i(I))) - S_k(I)\|_1] \qquad (9)$$

Total loss $\mathcal{L}^{1,\dots,N}$ is a raw sum of above Equations (4), (5), (8), (9).

5 EXPERIMENTS

Our goal is to assess the effectiveness of the Handcrafted Translation as a replacement for one of the generator-discriminator in a cyclic model, tasked with an unsupervised colorization problem. We also seek to assess the contribution of the early outputs generated by the *OSP*. Experiments were carried out on 2 GPUs GeForce 1080 Ti with Pytorch, Scikit, Caffe and OpenCV libraries.

5.1 Datasets

To evaluate the quality of the colorization, we adapt classical datasets used in classification problems: Cifar-10 [15] and UCMerced Land Use [40]. Cifar-10 dataset contains 60 000 image instances of 10 objects in a thumbnails fashion (32 × 32 pixels). UCMerced Land Use images represent 24 remotely sensed structures from a bird's eye viewpoint (256 × 256 pixels). Complementary experiments are carried out on Cezanne paintings and Landscape photos [14] (no classification, 256 × 256 pixels). Since these datasets represent color images, we first translate them to grayscale. However, because our approach is defined in an unsupervised setup, we recall that the images are not explicitly paired during training nor during testing to simulate unsupervised learning. In practice, colorization training is performed on the train data. Evaluation is performed on test data unseen during training. For Cifar-10, Cezanne paintings and Landscape photos datasets, provided train/test splits are used. For UCMerced Land Use dataset, we randomly sampled 80% of the images for training and 20% for testing (no default train/test splits).

5.2 SpyncoGan parameters

Loss function parameters of SpyncoGan (α_i, β_i, γ_i, ζ_i) were fixed empirically to give more importance to the final output. This choice was made to counterbalance the contribution of the most inner layers which is taken into account multiple times in the loss functions due to *OSP* (*i.e.*, $O^{W\times H}_{d_i}$ is calculated from $O_{d_{i+1}}$). They have been set as follow by considering $N = 3$ as stated in section 4: $\alpha_{i\in\{1,2,3\}} = \{5, 3, 2\}$, $\beta_{i\in\{1,2,3\}} = \{10, 6, 4\}$, $\gamma_{i\in\{1,2,3\}} = \{1, 1, 1\}$ and $\zeta_{i\in\{1,2,3\}} = \{1, 0, 0\}$, where i is the decoder layer index. Note that because of ζ_i, only the final output explicitly constrains the contours. This choice was made to not taking contours loss into account for the most inner outputs, that are more prone to visual artifacts (*e.g.*, checkerboard).

5.3 Metrics

Evaluation of colorization quality is performed every 10 epochs to quantify the evolution of the metrics during training. We calculate the Mean Square Error (MSE) and the Structural Similarity Measure (SSIM) averaged in the (three) channels dimension between colorized and real color images. The MSE allows to roughly determine how different two images are (the lower the better): this metric is commonly used to evaluate the results of regression algorithms, and its monotonic variant (root MSE, RMSE) was already applied to evaluate colorization algorithms [18]. The SSIM indicates the perceived quality of an image relative to another (the higher the better), with a focus on the structural differences. Overall, these metrics provide an insight about colorization quality when real color images are available (case of our datasets).

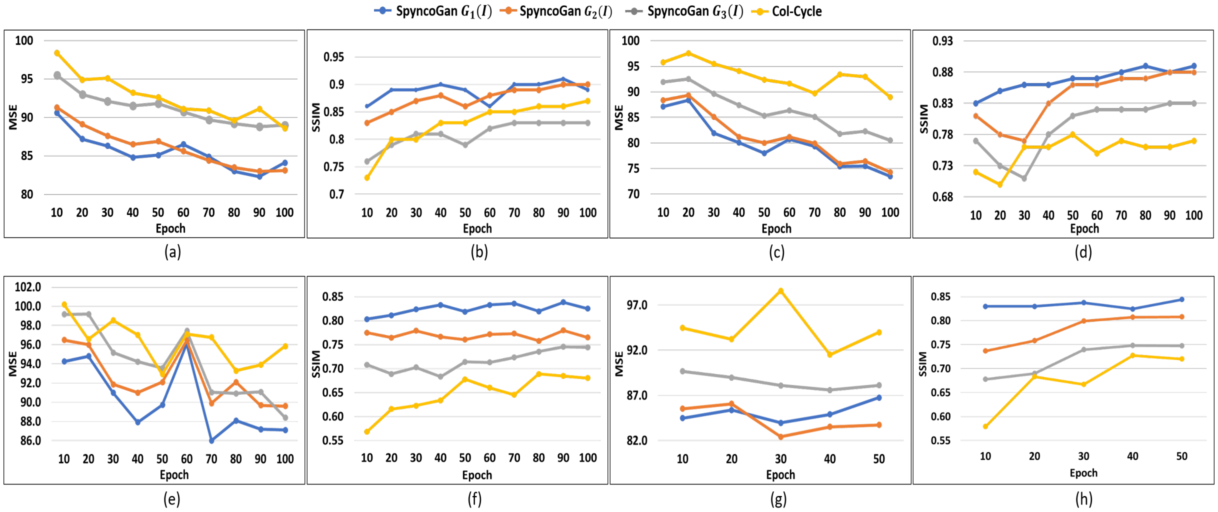
In addition, we evaluate whether colorization with a pseudo-cyclic network could result in accuracy gain in the context of classification. In particular, we propose to assess how robust a classification network would be to color spaces when trained on colorized images by supposing the images generated by SpyncoGan at every 10 epochs as lying on slightly different color manifolds (*i.e.*, domains). This observation can be verified empirically on Figure 5 and in the supplementary materials[2]. To evaluate it, we train standard classification networks on colorized, real color and grayscale training sets of the classification datasets, and we evaluate them in a cross-domain (color space) fashion on all the real and generated test sets. We use AlexNet with batch normalization [16] on Cifar-10 and VGG-16 [34] on UCMerced Land Use to evaluate different architectures while keeping a relatively small computation time. We feed these networks with images rescaled to 256x256 pixels. Learning rate was fixed to 0.0001 with a 0.1 step decay applied at 33% and 66% of training advancement for a total of 20 and 40 epochs respectively.

6 RESULTS AND DISCUSSION

6.1 Qualitative results for SpyncoGan

Figure 1 presents the qualitative results obtained on the test sets of Cezanne paintings, Landscape photos and UCMerced Land Use datasets. We believe these results look very realistic for an unsupervised approach. More results are available in the supplementary materials. Figure 5 shows the qualitative results obtained with SpyncoGan on three image samples of Cifar-10. From top to bottom, images represent grayscale (domain A), colorized and real color (domain B) images. Colorized images on different rows have been generated at different epochs (from 10 to 100 with a step of 10). From left to right, we present *OSP*'s results, namely $G_3(I)$, $G_2(I)$ and $G_1(I)$. From an overall perspective, we observe checkerboard artifacts on $G_3(I)$ images (most left image) that remains along the training. They seem to have been filtered out by the deeper layers, which is the expected behavior for our network. However, since $G_3(I)$ was directly obtained from the deep residual layers after rescaling and a spatial convolution; whose weights are shared between all outputs; we believe that the residual layers were either unable to learn a sufficient representation to discard the artifacts caused by down and up sampling, or caused the artifact themselves.

[2] http://liris.univ-lyon2.fr/SpyncoGan/files/ratajczak-SpyncoGan19supp.pdf



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6: Mean Square Error (MSE) and Structural Similarity Measure (SSIM) between generated images and real color images from the test sets of Cifar-10 (a,b), UCMerced Land Use (c,d), Cezanne paintings (e,f) and Landscape photos (g,h).

Table 1: Results of output ablation with SpyncoGan ($G_1(I)$). Scores are averaged over 50 epochs.

Dataset	Loss function	Avg. MSE ↓	Avg. SSIM (%) ↑
Cezanne paintings	$\mathcal{L}^1$	92.9	82
Cezanne paintings	$\mathcal{L}^{1,2,3}$	91.5	82
Landscape photos	$\mathcal{L}^1$	85.7	83
Landscape photos	$\mathcal{L}^{1,2,3}$	85.1	83
UCMerced Land Use	$\mathcal{L}^1$	85.5	86
UCMerced Land Use	$\mathcal{L}^{1,2,3}$	83.1	85
Cifar-10	$\mathcal{L}^1$	87.2	89
Cifar-10	$\mathcal{L}^{1,2,3}$	86.8	89

Both cases are highly interesting knowing that the residual layers (6 convolutional blocks) are not prone to generate such artifacts since they do not rescale their input features.

6.2 Quantitative Evaluation

6.2.1 Output and Loss Ablation Study. We assess the usefulness of *OSP* by training SpyncoGan considering $N = 1$ in the loss functions (*i.e.*, $\mathcal{L}^1$) instead of $N = 3$ (*i.e.*, $\mathcal{L}^{1,2,3}$). We name this evaluation output ablation. To this end, we use the architecture of SpyncoGan presented in Section 4, meaning that all 3 outputs from the *OSP* are available, but only last output $G_1(I)$ is used for training. We also perform loss ablation to investigate the contributions of contours loss $\mathcal{L}^{1,\ldots,N}_{contours}$ to total loss $\mathcal{L}^{1,\ldots,N}$. Table 1 and Table 2 presents the average MSE and SSIM scores from epochs 10 to 50 with a step of 10 epochs. We observe on Table 1 that using all the $N = 3$ outputs to constrain SpyncoGan improves MSE by 1.2 points in average at the cost of a small SSIM decrease of 0.25% in average. Overall, giving less freedom to the inner layers through the *OSP* seems to increase colorization consistency with respect to real color images. Despite the small contribution of $\mathcal{L}^{1,\ldots,N}_{contours}$ in total loss induced by a relatively small ζ_i, we observe on Table 2 that removing the constraint imposed on the contours significantly reduce structural similarity. As expected, constraining the generation of realistic contours, as permitted by H_t, helps to preserve structural properties even without *OSP*. However, we observed unrealistic (deep) visualization for the inner outputs of SpyncoGan trained with $\mathcal{L}^1$, as opposed to the somewhat realistic visualization obtained for $G_2(I)$ and $G_3(I)$ with $\mathcal{L}^{1,2,3}$ (see supplementary materials).

Table 2: Results of contours loss ablation with SpyncoGan ($G_1(I)$) on Cezanne paintings averaged over 50 epochs.

Loss function	Ablation	Avg. MSE ↓	Avg. SSIM (%) ↑
$\mathcal{L}^1$	$\mathcal{L}^1_{contours}$	92.6	79
$\mathcal{L}^1$	/	92.9	82
$\mathcal{L}^{1,2,3}$	$\mathcal{L}^{1,2,3}_{contours}$	92.0	77
$\mathcal{L}^{1,2,3}$	/	91.5	82

6.2.2 Unsupervised Colorization Quality. Figure 6 displays the MSE and SSIM scores along the training for SpyncoGan ($G_3(I)$, $G_2(I)$, $G_1(I)$) and Col-Cycle [32]. We selected Col-Cycle as our baseline because it has a relatively small (fully) cyclic architecture that is comparable with SpyncoGan. Both networks use nearest neighbor interpolation before convolution for upsampling, and they have the same number of layers with same number of filters but for H_t and *OSP*: Col-Cycle is composed of two DCGANs and has a decreasing number of filters in its decoders, like CycleGan [45]. The networks were trained in the same conditions for 100 epochs (but 50 for Landscape photos) with learning rate of 0.0002 and linear learning rate decay applied after half the epochs have passed. Overall, by considering both metrics and all epochs, we observe that the outputs of SpyncoGan result in lower MSE and higher SSIM than Col-Cycle. They allow to obtain colorized images that are more satisfying than Col-Cycle from a realistic colorization

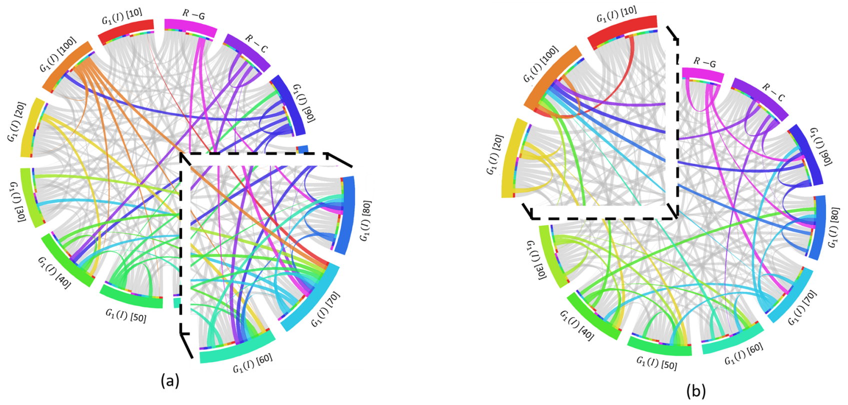


Figure 7: Chord diagram highlighting first quarter (color chords) of top-1 accuracy in cross-domain classification on (a) UCMerced Land Use and (b) Cifar-10 datasets with $G_1(I)$. R-G is Real-Gray. R-C is Real-Color. [Numbers] indicate epochs.

perspective, as well as from a structure preserving viewpoint. In particular, we observe that the quality of $G_i(I)$ generally increases with $\frac{1}{i}$. In accordance with the results presented in Sections 6.1 and 6.2.1, we may conclude the following two points. First, in the context of colorization, Handcrafted Translation is an effective lightweight alternative to a second generator-discriminator in cyclic networks, thus showing the interest of pseudo-cyclic architectures. Second, the Output Spatial Pyramids are, as expected, constraining early deep features in the same directions, permitting a gradual improvement of the results. However, Figure 6 (a), (c) and (g) shows that $G_2(I)$ sometimes achieves equal or smaller MSE scores than $G_1(I)$. Based on Figure 5, we point out that there there is less color diversity on $G_2(I)$ than on $G_1(I)$, which may explain MSE score variations (less diversity reduce potential errors). We believe these results are due to the intrinsic nature of the Output Spatial Pyramids, that generate more constraints on the inner layers in regard to real color images. Nonetheless, since the SSIM is higher on $G_1(I)$ than on $G_2(I)$, these constraints seem not to be always sufficient to improve the structural similarly without additional convolutional filters after upsampling.

6.2.3 Classification accuracy. We study how a classification network generalizes to different colors spaces (*i.e.*, domains) when trained on colorized images. We report cross-domain classification results (*i.e.*, classification network trained on a single color space, tested on all color spaces) for $G_1(I)$ on Figure 7 with chord diagrams [17]. Each chord indicates a classification relationship between two image sets (arcs). Only the 25% highest accuracy rates are represented with color chords. A chord attached to an arc indicates the image set (arc) used for training. A chord separated with a blank space from an arc indicates the image set (arc) used for testing. This visualization allows to quickly identify the image sets that are more prone to make a network generalize well to slightly different color spaces, as well as on which image sets it does generalize well: one only needs to count the number of color chords starting or ending from each image set. The training set with the highest number of starting chords allows the best generalization. We observe that training VGG-16 with $G_1(I)$ at colorization epoch 70 allows a better generalization on UCMerced Land Use than training it with any other color space, including the real color space. For Cifar-10, $G_1(I)$ at colorization epoch 100 results in a better generalization of AlexNet. We ascertain these observations by calculating the average accuracy obtained by a trained network when classifying all the other datasets on Table 3. These results show that training a classification network on colorized images tend to improve its overall robustness to different color spaces compared to training only on real color images.

Table 3: Top-1 accuracy (%) in cross domain setup with VGG-16 for 40 epochs on UCMerced Land Use (1) and AlexNet for 20 epochs on Cifar-10 (2). Accuracy is averaged on all real and colorized datasets.

Training Set	Col. Epoch	Avg. % (1)	Avg. % (2)
SpyncoGan ($G_1(I)$)	70	97.0	78.7
SpyncoGan ($G_1(I)$)	100	95.1	81.0
Real Color	/	92.4	75.7
Real Gray	/	92.5	22.1

7 CONCLUSION

In this study, we presented a novel unsupervised representation learning approach for colorization by introducing the Handcrafted Translation and Output Spatial Pyramids in a Pseudo-Cyclic Network that jointly constrain the optimization of the deep features. We demonstrated the effectiveness of our approach in the generation of realistic and structurally-preserved color images along the training. We also showed that the generated images are prone to make a classification network generalize to slightly different color spaces. In future work, we will investigate the possibility to guide a generative network toward multi-modal spaces with Handcrafted Translation, resulting in different and always more varied representations. In particular, we will investigate the applicability of our approach to other generative tasks including hyper-spectral images generation and semantic segmentation.

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 12 (2017), 2481–2495.

[2] Hyojin Bahng, Seungjoo Yoo, Wonwoong Cho, David Keetae Park, Ziming Wu, Xiaojuan Ma, and Jaegul Choo. 2018. Coloring with Words: Guiding Image Colorization Through Text-based Palette Generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 431–447.

[3] Stephen A Billings. 2013. *Nonlinear System Identification*. John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118535561

[4] Changjian Chen, Yi Xu, and Xiaokang Yang. 2019. User tailored colorization using automatic scribbles and hierarchical features. *Digital Signal Processing* 87 (apr 2019), 155–165. https://doi.org/10.1016/j.dsp.2019.01.021

[5] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 8789–8797.

[6] U. Forssell and P. Lindskog. 1997. Combining Semi-Physical and Neural Network Modeling: An Example ofIts Usefulness. *IFAC Proceedings Volumes* 30, 11 (jul 1997), 767–770. https://doi.org/10.1016/s1474-6670(17)42938-7

[7] Yan Gan, Junxin Gong, Mao Ye, Yang Qian, Kedi Liu, and Su Zhang. 2018. GANs with Multiple Constraints for Image Translation. *Complexity* 2018 (dec 2018), 1–12. https://doi.org/10.1155/2018/4613935

[8] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*. 2414–2423. https://doi.org/10.1109/CVPR.2016.265

[9] Abel Gonzalez-Garcia, Joost van de Weijer, and Yoshua Bengio. 2018. Image-to-image translation for cross-domain disentanglement. In *Advances in Neural Information Processing Systems (NeurIPS)*. 1287–1298.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems (NIPS)*. Curran Associates, Inc., 2672–2680.

[11] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. 2015. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 447–456.

[12] Mingming He, Dongdong Chen, Jing Liao, Pedro V. Sander, and Lu Yuan. 2018. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)* 37, 4 (jul 2018), 1–16. https://doi.org/10.1145/3197517.3201365

[13] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2016. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2016)* 35, 4 (2016), 110:1–110:11.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-To-Image Translation With Conditional Adversarial Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1125–1134.

[15] Alex Krizhevsky. 2009. *Learning multiple layers of features from tiny images*. Technical Report.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12)*. Curran Associates Inc., USA, 1097–1105. http://dl.acm.org/citation.cfm?id=2999134.2999257

[17] Martin I Krzywinski, Jacqueline E Schein, Inanc Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J Jones, and Marco A Marra. 2009. Circos: An information aesthetic for comparative genomics. *Genome Research* (2009). https://doi.org/10.1101/gr.092759.109

[18] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. 2016. Learning Representations for Automatic Colorization. In *European Conference on Computer Vision (ECCV)*. 577–593.

[19] Daoyu Lin, Kun Fu, Yang Wang, Guangluan Xu, and Xian Sun. 2017. MARTA GANs: Unsupervised Representation Learning for Remote Sensing Image Classification. *IEEE Geoscience and Remote Sensing Letters* (2017).

[20] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature Pyramid Networks for Object Detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2117–2125.

[21] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised Image-to-Image Translation Networks. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 700–708.

[22] Yu Liu, Yanming Guo, Wei Chen, and Michael S. Lew. 2018. An Extensive Study of Cycle-Consistent Generative Networks for Image-to-Image Translation. In *24th International Conference on Pattern Recognition (ICPR)*. IEEE, 219–224. https://doi.org/10.1109/icpr.2018.8545089

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3431–3440.

[24] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 2 (2004), 91–110.

[25] Shuang Ma, Jianlong Fu, Chang Wen Chen, and Tao Mei. 2018. DA-GAN: Instance-level image translation by deep attention generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5657–5666.

[26] Franck Mamalet and Christophe Garcia. 2012. Simplifying convnets for fast learning. In *International Conference on Artificial Neural Networks*. Springer, 58––65.

[27] Henrik Aalborg Nielsen and Henrik Madsen. 2006. Modelling the heat consumption in district heating systems using a grey-box approach. *Energy and Buildings* 38, 1 (jan 2006), 63–71. https://doi.org/10.1016/j.enbuild.2005.05.002

[28] Augustus Odena, Vincent Dumoulin, and Chris Olah. 2016. Deconvolution and checkerboard artifacts. *Distill* 1, 10 (2016), e3.

[29] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *CoRR* abs/1511.06434 (2015). arXiv:1511.06434 http://arxiv.org/abs/1511.06434

[30] R Raghavendra, Kiran B Raja, Sushma Venkatesh, and Christoph Busch. 2017. Transferable Deep-CNN features for detecting digital and print-scanned morphed face images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 1822–1830.

[31] Rémi Ratajczak, Carlos F Crispim-Junior, Élodie Faure, Béatrice Fervers, and Laure Tougne. 2019. Automatic Land Cover Reconstruction From Historical Aerial Images: An Evaluation of Features Extraction and Classification Algorithms. *IEEE Transactions on Image Processing (TIP)* (2019). https://doi.org/10.1109/TIP.2019.2896492

[32] Rémi Ratajczak, Carlos F Crispim-Junior, Élodie Faure, Béatrice Fervers, and Laure Tougne. 2019. Toward an Unsupervised Colorization Framework for Historical Land Use Classification. In *International Geoscience and Remote Sensing Symposium (IGARSS 2019)*. IEEE, Yokohama, Japan.

[33] Patsorn Sangkloy, Jingwan Lu, Chen Fang, FIsher Yu, and James Hays. 2017. Scribbler: Controlling Deep Image Synthesis with Sketch and Color. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 5400–5409.

[34] K Simonyan and A Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[35] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022* (2016).

[36] D Ulyanov, A Vedaldi, and Lempitsky V S. 2017. Deep Image Prior. *CoRR* abs/1711.10925 (2017).

[37] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 8798–8807.

[38] Junyuan Xie, Linli Xu, and Enhong Chen. 2012. Image denoising and inpainting with deep neural networks. In *Advances in neural information processing systems (NIPS)*. 341–349.

[39] Xuewen Yang, Dongliang Xie, and Xin Wang. 2018. Crossing-Domain Generative Adversarial Networks for Unsupervised Multi-Domain Image-to-Image Translation. In *Proceedings of the 26th ACM International Conference on Multimedia (MM '18)*. ACM, New York, NY, USA, 374–382. https://doi.org/10.1145/3240508.3240716

[40] Yi Yang and Shawn Newsam. 2010. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 270–279.

[41] Richard Zhang, Phillip Isola, and Alexei A Efros. 2016. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*. Springer, 649–666.

[42] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. 2017. Real-Time User-Guided Image Colorization with Learned Deep Priors. *ACM Transactions on Graphics (TOG)* 9, 4 (2017).

[43] Yuheng Zhi, Huawei Wei, and Bingbing Ni. 2018. Structure Guided Photorealistic Style Transfer. In *Proceedings of the 26th ACM International Conference on Multimedia (MM '18)*. ACM, New York, NY, USA, 365–373. https://doi.org/10.1145/3240508.3240637

[44] Huiyu Zhou, Yuan Yuan, and Chunmei Shi. 2009. Object tracking using SIFT features and mean shift. *Computer Vision and Image Understanding (CVIU)* 113, 3 (2009), 345–352.

[45] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *IEEE International Conference on Computer Vision (ICCV)*. 2223–2232.