Colorizing Infrared Images Through a Triplet Conditional DCGAN Architecture

Patricia L. Suárez[1], Angel D. Sappa[1,2(✉)], and Boris X. Vintimilla[1]

[1] Escuela Superior Politécnica del Litoral, ESPOL, Facultad de Ingeniería en Electricidad y Computación, CIDIS, Campus Gustavo Galindo, 09-01-5863 Guayaquil, Ecuador
{plsuarez,asappa,boris.vintimilla}@espol.edu.ec
[2] Computer Vision Center, Edifici O, Campus UAB, 08193 Bellaterra, Barcelona, Spain

Abstract. This paper focuses on near infrared (NIR) image colorization by using a Conditional Deep Convolutional Generative Adversarial Network (CDCGAN) architecture model. The proposed architecture is based on the usage of a conditional probabilistic generative model. Firstly, it learns to colorize the given input image, by using a triplet model architecture that tackle every channel in an independent way. In the proposed model, the final layer of red channel consider the infrared image to enhance the details, resulting in a sharp RGB image. Then, in the second stage, a discriminative model is used to estimate the probability that the generated image came from the training dataset, rather than the image automatically generated. Experimental results with a large set of real images are provided showing the validity of the proposed approach. Additionally, the proposed approach is compared with a state of the art approach showing better results.

Keywords: CNN in multispectral imaging · Image colorization

1 Introduction

Image acquisition devices have largely expanded in recent years, mainly due to the decrease in price of electronics together with the increase in computational power. This increase in sensor technology has resulted in a large family of images, able to capture different information (from different spectral bands) or complementary information (2D, 3D, 4D); hence, we can have: HD 2D images; video sequences at a high frame rate; panoramic 3D images; multispectral images; just to mention a few. In spite of the large amount of possibilities, when the information needs to be provided to a final user, the classical RGB representation is preferred. This preference is supported by the fact that human visual perception system is sensitive to (400–700 nm); hence, representing the information in that range helps user understanding. In this context, the current paper tackles the near infrared (NIR) image colorization, trying to generate realistic RGB representations. Different applications could take advantage of this

S. Battiato et al. (Eds.): ICIAP 2017, Part I, LNCS 10484, pp. 287–297, 2017.
https://doi.org/10.1007/978-3-319-68560-1_26

contribution—infrared sensors can be incorporated for instance in driving assistance applications by providing realistic colored representations to the driver, while the image processing can be automatically performed by the system in the infrared domain (e.g., semantic segmentation at the material level avoiding classical problems related with the color of the object surface).

The NIR spectral band is the closest in wavelength to the radiation detectable by the human eye; hence, NIR images share several properties with visible images. The interest of using NIR images is related with their capability to segment images according to the object's material. Surface reflection in the NIR spectral band is material dependent, for instance, most pigments used for material colorization are somewhat transparent to NIR. This means that the difference in the NIR intensities is not only due to the particular color of the material, but also to the absorption and reflectance of dyes.

The absorption/reflectance properties mentioned above are used for instance in remote sensing applications for crop stress and weed/pest infestations. NIR images are also widely used in video surveillance applications since it is easier to detect different objects from a given scene. In these two contexts (i.e., remote sensing and video surveillance), it is quite difficult for users to orientate when NIR images are provided, since the lack of color discrimination or wrong color deploy. In this work a neural network based approach for NIR image colorization is proposed. Although the problem shares some particularities with image colorization (e.g., [1,2]) and color correction/transfer (e.g., [3,4]) there are some notable differences. First, in the image colorization domain—gray scale image to RGB—there are some clues, such as the fact that luminance is given by grayscale input, so only the chrominance needs to be estimated. Secondly, in the case of color correction/transfer techniques, in general three channels are given as input to obtain the new representation in the new three dimensional space. In the particular problem tackled in this work (NIR to visible spectrum representation) a single channel is mapped into a three dimensional space, making it a difficult and challenging problem. The manuscript is organized as follows. Related works are presented in Sect. 2. Then, the proposed approach is detailed in Sect. 3. Experimental results with a large set of images are presented in Sect. 4. Finally, conclusions are given in Sect. 5.

2 Related Work

The problem tackled in this paper is related with infrared image colorization, as mentioned before, somehow it shares some common problems with monocromatic image colorization that has been largely studied during last decades. Colorization technique algorithms mostly differ in the ways they obtain and treat the data for modeling the correspondences between grayscale and color. There have been a lot of techniques, like spatial and frequency based variational methods, in which obtain perceptually inspired color and contrast enhancement of digital images, and the color logarithmic image processing (CoLIP) and antagonist space, Gavet et al. [5] design a framework that defines a vectorial space for color images.

It illustrates the representation of the chromaticity diagram with color modification application, namely white balance correction and color transfer. Another technique is the grayscale image matting and colorization, Chen et al. [6] present a variation of a matting algorithm with the introduction of alpha's distribution and gradient into the Bayesian framework and an efficient optimization scheme. It can effectively handle objects with intricate and vision sensitive boundaries, such as hair strands or facial organs, plus they combine this algorithm with the color transferring techniques to obtain his colorization scheme. Welsh et al. [7] describe a semi-automatic technique for colorizing a grayscale image by transferring color from a reference color image. They examine the luminance values in the neighborhood of each pixel in the target image and transfer the color from pixels with matching neighborhoods in the reference image. This technique works well on images where differently colored regions give rise to distinct luminance clusters, or possess distinct textures. In other cases, the user must direct the search for matching pixels by specifying swatches indicating corresponding regions in the two images. It is also difficult to fine-tune the outcome selectively in problematic areas. There are other approaches like colorization by example; in [8] an algorithm that colorizes one or more input grayscale images is presented. It is based on a partially segmented reference color image. By partial segmentation they assume that one or more mutually disjoint regions in the image have been established, and each region has been assigned to a unique label.

The approaches presented above have been implemented using classical image processing techniques. However, recently Convolutional Neural Network (CNN) based approaches are becoming the dominant paradigm in almost every computer vision task. CNNs have shown outstanding results in various and diverse computer vision tasks such as stereo vision [9], image classification [10] or even difficult problems related with cross-spectral domains [11] outperforming conventional hand-made approaches. Hence, we can find some recent image colorization approaches based on deep learning, exploiting to the maximum the capacities of this type of convolutional neural networks. As an example, we can mention the work presented in [2]. The authors propose a fully automatic approach that produces brilliant and sharpen image color. They model the unknown uncertainty of the desaturated colorization levels designing it as a classification task and use class-rebalancing at training time to augment the diversity of colors in the result. On the contrary, [12] presents a technique that combines both global priors and local image features. Based on a CNN a fusion layer merges local information, dependent on small image patches, with global priors computed using the entire image. The model is trained in an end-to-end fashion, so this architecture can process images of any resolution. They leverage an existing large-scale scene classification database to train the model, exploiting the class labels of the dataset to more efficiently and discriminatively learn the global priors. In [13], a recent research on colorization, addressing images from the infrared spectrum, has been presented. It uses convolutional neural networks to perform an automatic integrated colorization from a single channel NIR image to RGB images. The approach is based on a deep multi-scale convolutional neural

network to perform a direct estimation of the low RGB frequency values. The main problem with this approach lies on the blur results generated by the multi-scale approach. For that reason it requires a final step that filters the raw output of obtained image from the CNN and transfers the details of the input image to the final output image. Finally, also based on the usage of the CNN framework, [14] proposes a NIR image colorization using a Deep Convolutional Generative Adversarial Network (DCGAN). In that work, a colorization model is obtained based on a flat GAN architecture where all the colors are learned at once from the given input NIR image. This architecture has limitations since all the colors are considered together.

Generative Adversarial Networks (GANs) are a class of neural networks which have gained popularity in recent years. They allow a network to learn to generate data with the same internal structure as other data. GANs are powerful and flexible tools, one of its more common applications is image generation. It is a framework presented on [15] for estimating generative models via an adversarial process, in which simultaneously two models are trained: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to 1/2 everywhere. According to [16], to learn the generator's distribution p_g over data $\boldsymbol{x}$, the generator builds a mapping function from a prior noise distribution ${}_pz(z)$ to a data space $G(z;\theta_g)$. And the discriminator, $D(x;\theta_d)$, outputs a single scalar representing the probability that x came from training data rather than p_g. G and D are both trained simultaneously, the parameters for G are adjusted to minimize $log(1-D(G(z)))$ and for D to minimize $logD(X)$ with a value function $V(G,D)$:

$$\frac{min}{G}\frac{max}{D}V(D,G) = \mathbb{E}_x \sim_{p\ \mathrm{data}_{(x)}}[logD(x)] + \mathbb{E}_z \sim_{p\ \mathrm{data}_{(z)}}[log(1-D(G(z)))]. \tag{1}$$

Generative adversarial nets can be extended to a conditional model if both the generator and discriminator are conditioned on some extra information y. This information could be any kind of auxiliary information, such as class labels or data from other modalities. We can perform the conditioning by feeding y into both discriminator and generator as additional input layer. The objective function of a two-player minimax game would be as :

$$\frac{min}{G}\frac{max}{D}V(D,G) = \mathbb{E}_x \sim_{p\ \mathrm{data}_{(x)}}[logD(x|y)] + \mathbb{E}_z \sim_{p\ \mathrm{data}_{(z)}}[log(1-D(G(z|y)))]. \tag{2}$$

In order to improve the efficiency of the generative adversarial networks, [17] proposes some techniques, one of them named the virtual batch normalization;

it allows to significantly improve the network optimization using the statistics of each set of training batches. The main disadvantage is that this process is computationally expensive. Our proposal is based on designing a generative adversarial deep learning architecture that allows the colorization of images of the near infrared spectrum, so that they can be represented in the visible spectrum. The following section will explain in detail the proposed network model.

3 Proposed Approach

This section presents the approach proposed for NIR image colorization. As mentioned above, a recent work on colorization [14] has proposed the usage of a deep convolutional adversarial generative learning network. It is based on a traditional scheme of layers in a deep network. In the current work we also propose the usage of a conditional DCGAN but in a triplet learning layers architecture scheme. These models have been used to solve other types of problems such as learning local characteristics, feature extraction, similarity learning, face recognition, etc. Based on the results that have been obtained on this type of solutions, where improvements in accuracy and performance have been obtained, we propose the usage of a learning model that allows the multiple representation of each of the channels of an image of the visible spectrum (R, G, B). Therefore, the model will receive as input a near infrared patch (NIR), with a Gaussian noise added in each channel of the image patch to generate the necessary variability to generate more diversity of colors, to be able to generalize the learning of the colorization process. A $l1$ regularization term has been added on a single layer in order to prevent the coefficients to fit so perfectly to overfit, which can improve the generalization capability of the model.

A Conditional DCGAN network based architecture is selected due to several reasons: (i) the learning is conditioned on NIR images from the source domain; (ii) its fast convergence capability; (iii) the capacity of the generator model to easily serve as a density model of the training data; and (iv) sampling is simple and efficient. The network is intended to learn to generate new samples from an unknown probability distribution. In our case, the generator network has been modified to use a triplet to represent the learning of each image channel independently; at the output of the generator network, the three resulting image channels are recombined to generate the RGB image. This will be validated by the discriminative network, which will evaluate the probability that the colorized image (RGB), is similar to the real one that is used as ground truth. Additionally, in the generator model, in order to obtain a true color, the DCGAN framework is reformulated for a conditional generative image modeling tuple. In other words, the generative model $G(z;\theta_g)$ is trained from a near infrared image plus Gaussian noise, in order to produce a colored RGB image; additionally, a discriminative model $D(z;\theta_d)$ is trained to assign the correct label to the generated colored image, according to the provided real color image, which is used as a ground truth. Variables (θ_g) and (θ_d) represent the weighting values for the generative and discriminative networks.

The CDCGAN network has been trained using Stochastic AdamOptimazer since it prevents overfitting and leads to convergence faster. Furthermore, it is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters. Our image dataset was normalized in a (−1,1) range and an additive Gaussian Distribution noise with a standard deviation of 0.00011, 0.00012, 0.00013 added to each image channel of the proposed triplet model. The following hyper-parameters were used during the learning process: learning rate 0.0002 for the generator and the discriminator networks respectively; epsilon = 1e-08; exponential decay rate for the 1st moment momentum 0.5 for discriminator and 0.4 for the generator; weight initializer with a standard deviation of 0.00282; $l1$ weight regularizer; weight decay 1e-5; leak relu 0.2 and patch's size of 64×64.

The Triplet architecture of the baseline model is conformed by convolutional, de-convolutional, relu, leak-relu, fully connected and activation function tanh and sigmoid for generator and discriminator networks respectively. Additionally, every layer of the model uses batch normalization for training any type of mapping that consists of multiple composition of affine transformation with element-wise nonlinearity and do not stuck on saturation mode. It is very important to maintain the spatial information in the generator model, there is not pooling and drop-out layers and only the stride of 1 is used to avoid downsize the image shape. To prevent overfitting we have added a l1 regularization term (λ) in the generator model, this regularization has the particularity that the weights matrix end up using only a small subset of their most important inputs and become quite resistant to noise in the inputs; this characteristics is very useful when the network try to learn which features are contributing to the learning process. Park and Kang [18], present a color restoration method that estimates the spectral intensity of the NIR band in each RGB color channel to effectively restores natural colors. According to the spectral sensitivity of conventional cameras with the IR cut-off filter, the contribution of the NIR spectral energy in each RGB color channel is greater in the red channel, hence our architecture add the NIR band at the final red channel layer, this improve the details of generated images, color and hue saturation. Figure 1 presents an illustration of the proposed Triplet GAN architecture.

The generator (G) and discriminator (D) are both feedforward deep neural networks that play a min-max game between one another. The generator takes as an input a near infrared image blurred with a Gaussian noise patch of 64×64 pixels, and transforms it into the form of the data we are interested in imitating, in our case a RGB image. The discriminator takes as an input a set of data, either real image (z) or generated image ($G(z)$), and produces a probability of that data being real ($P(z)$). The discriminator is optimized in order to increase the likelihood of giving a high probability to the real data (the ground truth given image) and a low probability to the fake generated data (wrongly colored NIR image), as introduced in [16]; thus, the conditional discriminator network it is updated as follow:

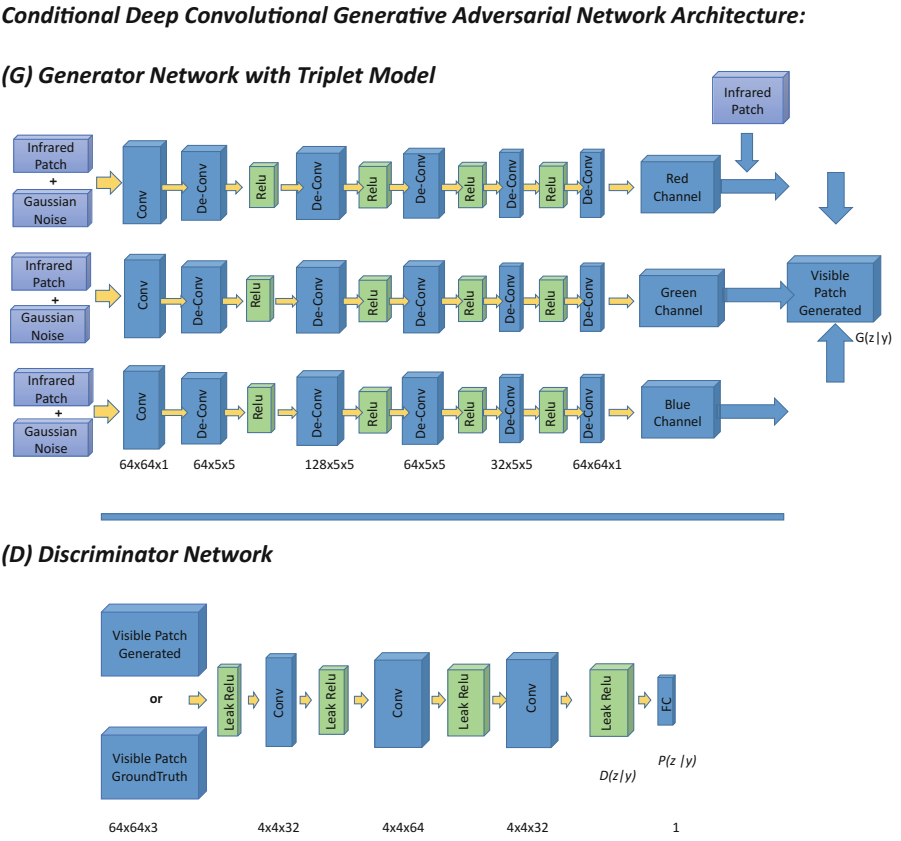

Fig. 1. Illustration of the network architecture used for NIR image colorization.

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^{(i)}) + \log(1 - D(G(y^{(i)}, z^{(i)})))], \tag{3}$$

where m is the number of patches in each batch, x is the ground truth image and y is the colored NIR image generated by the network and z is the randomly Gaussian sampled noise. The weights of the discriminator network (D) are updated by ascending its stochastic gradient. On the other hand, the generator is then optimized in order to increase the probability of the generated data being highly rated, it is updated as follow:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(y^{(i)}, z^{(i)}))), \tag{4}$$

where m is the number of samples in each batch and y is the colored NIR image generated by the network and z is the randomly Gaussian sampled noise. Like

in the previous case, the weights of the generator network (G) are updated by descending its stochastic gradient.

4 Experimental Results

The proposed approach has been evaluated using NIR images and their corresponding RGB obtained from [19]. The *urban* and *old-building* categories have been considered for evaluating the performance of the proposed approach. Figure 2 presents two pairs of images from each of these categories. The *urban* category contains 58 pairs of images of (1024×680 pixels), while the *old-building* contains 51 pairs of images of (1024×680 pixels). From each of these categories 250.000 pairs of patches of (64×64 pixels) have been cropped both, in the NIR images as well as in the corresponding RGB images. Additionally, 2500 pairs of patches, per category, of (64×64 pixels) have been also generated for validation. It should be noted that images are correctly registered, so that a pixel-to-pixel correspondence is guaranteed.

Fig. 2. Pair of images (1024×680 pixels) from [19], *urban* category (the two images in the left side) and *old-building* category (the two images in the right side): (*top*) NIR images to colorize; (*bottom*) RGB images used as ground truth. (Color figure online)

The CDCGAN network proposed in the current work for NIR image colorization has been trained using a 3.2 eight core processor with 16 GB of memory with a NVIDIA GeForce GTX970 GPU. On average every training process took about 28 h. Results from the proposed architecture have been compared with those obtained with the GAN model presented in [14].

Colored images are referred to as (RGB_{NIR}) while the corresponding RGB images, provided in the given data sets, are referred to as (RGB_{GT}) and used as ground truth. The quantitative evaluation consists of measuring at every pixel the angular error (AE) between the obtained result (RGB_{NIR}) and the corresponding ground truth value (RGB_{GT}):

$$AngularError = \cos^{-1}\left(\frac{\text{dot}(RGB_{NIR}, RGB_{GT})}{\text{norm}(RGB_{NIR}) * \text{norm}(RGB_{GT})}\right). \quad (5)$$

This angular error is computed over every single pixel of the whole set of images used for validation. Table 1 presents the average angular errors (AE) obtained with the proposed approach for the two categories together with the results obtained with [14] for the same categories. It can be appreciated that in all the cases the results with the proposed CDCGAN are better that those obtained with [14].

Table 1. Average angular errors obtained with the approach presented in [14] (flat DCGAN) and with the proposed Triplet based CDCGAN architecture.

Category	[14]	Prop. Approach (CDCGAN)
Urban	6.15	5.94
Old-building	6.95	5.71

Qualitative results are presented in Figs. 3 and 4. Figure 3 shows NIR images from the *urban* category colorized with the proposed CDCGAN network and with the approach presented in [14]; ground truth images (last column) are depicted to appreciate the similarity reached with the proposed approach. Similar results have been obtained when images from the *old-building* category are colorized

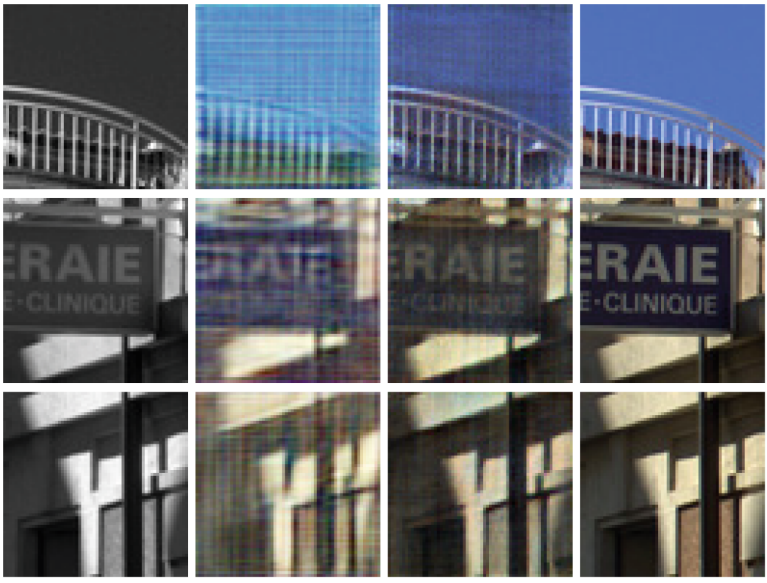

Fig. 3. (1*st.Col*) NIR patches from the ***Urban category***. (2*nd.Col*) Results from the approach presented in [14] (flat DCGAN). (3*rd.Col*) Colorization obtained with the proposed approach (CDCGAN network). (4*th.Col*) Ground truth images. (Color figure online)

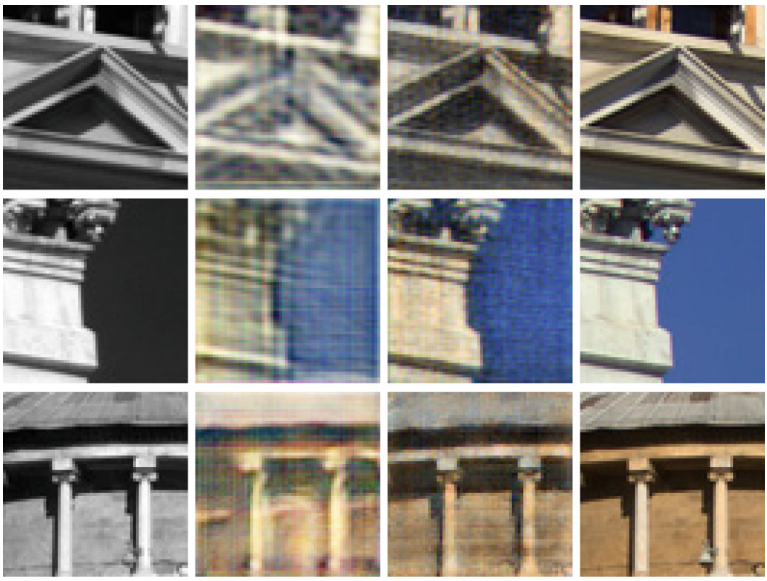

Fig. 4. (1*st.Col*) NIR patches from the ***Old-Building category***. (2*nd.Col*) Results from the approach presented in [14] (flat DCGAN). (3*rd.Col*) Colorization obtained with the proposed approach (CDCGAN network). (4*th.Col*) Ground truth images. (Color figure online)

with the proposed CDCGAN network (see Fig. 4). As mentioned above, the usage of a conditional triplet model allows to improve results with respect to the flat model [14]. This improvement can be particularly appreciated in both the color and the edges of the colorized images.

5 Conclusions

This paper tackles the challenging problem of NIR image colorization by using a novel Conditional Generative Adversarial Network model. Results have shown that in most of the cases the network is able to obtain a reliable RGB representation of the given NIR image. Comparisons with a previous approach shows considerable improvements. Future work will be focused on evaluating others network architectures, like auto-encoders, cycle-consistent adversarial networks, which have shown appealing results in recent works. Additionally, increasing the number of images to train, in particular the color variability, will be considered. Finally, the proposed approach will be tested in other image categories.

Acknowledgments. This work has been partially supported: by the ESPOL projects PRAIM and KISHWAR; by the Spanish Government under Project TIN2014-56919-C3-2-R; and by the "CERCA Programme / Generalitat de Catalunya".

References

1. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: IEEE International Conference on Computer Vision (ICCV), pp. 415–423 (2015)
2. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 649–666. Springer, Cham (2016). doi:10.1007/978-3-319-46487-9_40
3. Oliveira, M., Sappa, A.D., Santos, V.: Unsupervised local color correction for coarsely registered images. In: Computer Vision and Pattern Recognition (CVPR), pp. 201–208. IEEE (2011)
4. Oliveira, M., Sappa, A.D., Santos, V.: A probabilistic approach for color correction in image mosaicking applications. IEEE Trans. Image Process. **24**, 508–523 (2015)
5. Gavet, Y., Debayle, J., Pinoli, J.-C.: The color logarithmic image processing (CoLIP) antagonist space. In: Celebi, M.E., Lecca, M., Smolka, B. (eds.) Color Image and Video Enhancement, pp. 155–182. Springer, Cham (2015). doi:10.1007/978-3-319-09363-5_6
6. Chen, T., Wang, Y., Schillings, V., Meinel, C.: Grayscale image matting and colorization. In: Asian Conference on Computer Vision (2004)
7. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. ACM Trans. Graph. (TOG) **21**, 277–280 (2002)
8. Ironi, R., Cohen-Or, D., Lischinski, D.: Colorization by example. In: Rendering Techniques, pp. 201–210 (2005)
9. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. J. Mach. Learn. Res. **17**, 2 (2016)
10. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. CoRR abs/1409.4842 (2014)
11. Aguilera, C.A., Aguilera, F.J., Sappa, A.D., Aguilera, C., Toledo, R.: Learning cross-spectral similarity measures with deep convolutional neural networks. In: Conference on Computer Vision and Pattern Recognition Workshops
12. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ACM Trans. Graph. **35**, 110 (2016). Proceedings of SIGGRAPH 2016
13. Limmer, M., Lensch, H.: Infrared colorization using deep convolutional neural networks. arXiv preprint (2016). arXiv:1604.02245
14. Suarez, P.L., Sappa, A.D., Vintimilla, B.X.: Learning to colorize infrared images. In: 15th International Conference on Practical Applications of Agents and Multi-Agent Systems (2017)
15. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
16. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint (2014). arXiv:1411.1784
17. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: Advances in Neural Information Processing Systems, pp. 2226–2234 (2016)
18. Park, C., Kang, M.G.: Color restoration of RGBN multispectral filter array sensor images based on spectral decomposition. Sensors **16**, 719 (2016)
19. Brown, M., Süsstrunk, S.: Multi-spectral SIFT for scene category recognition. In: Computer Vision and Pattern Recognition (CVPR), pp. 177–184. IEEE (2011)