ChromaGAN: An Adversarial Approach for Picture Colorization

Patricia Vitoria, Lara Raad and Coloma Ballester
Department of Information and Communication Technologies
University Pompeu Fabra, Barcelona, Spain
{patricia.vitoria, lara.raad coloma.ballester}@upf.edu

Abstract

The colorization of grayscale images is an ill-posed problem, with multiple correct solutions. In this paper, an adversarial learning approach is proposed. A generator network is used to infer the chromaticity of a given grayscale image. The same network also performs a semantic classification of the image. This network is framed in an adversarial model that learns to colorize by incorporating perceptual and semantic understanding of color and class distributions. The model is trained via a fully self-supervised strategy. Qualitative and quantitative results show the capacity of the proposed method to colorize images in a realistic way, achieving top-tier performances relative to the state-of-the-art.

1. Introduction

Colorization is the process of adding plausible color information to monochrome photographs or videos (we refer to [43] for an interesting historical review). Currently, digital colorization of black and white visual data is a crucial task in areas so diverse as advertising and film industries, photography technologies or artist assistance. Although color hallucination is an easy deal for a human, automatic image colorization still remains a challenge.

Colorization is a highly undetermined problem, requiring mapping a real-valued luminance image to a three-dimensional color-valued one, that has not a unique solution. Before the emergence of deep learning techniques, the most effective methods relied on human intervention, usually through either user-provided color scribbles or a color reference image. Recently, convolutional neural network strategies have benefit from the huge amount of publicly available color images in order to automatically learn what colors naturally correspond to the real objects and its parts. Our work fits in this context.

In this paper we propose an adversarial approach called ChromaGAN that combines the strength of generative adversarial networks (GANs) to learn the probability distribution of natural color images and generate color attributes, with a semantic class distribution learning. As a result, ChromaGAN is able to perceptually colorize a grayscale image from the semantic understanding of the captured scene. To give just some examples, Fig. 1 shows how vibrant and diverse colorizations are frequently achieved. On the other hand, ChromaGAN also shows variability by colorizing differently some objects belonging to the same category, as for example, the birds. The ablation study analyzing the different contributions of the proposed model and the quantitative perceptual results presented in Section 4 show that the effect of the generative adversarial learning is key to obtain those vivid colorizations.

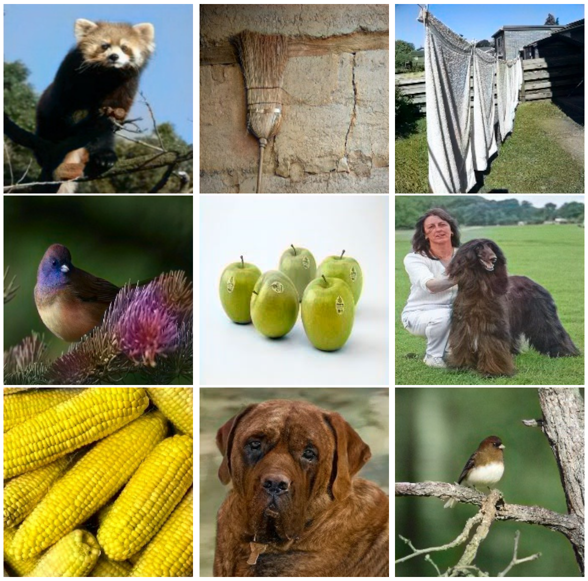

Figure 1. ChromaGAN is able to colorize a grayscale image from the semantic understanding of the captured scene.

The contributions of this work include:

- A fully automatic end-to-end adversarial model able to generate a perceptually plausible colorization without any need of guideline.

- An all-included architecture that integrates the generation of color and semantic distribution with a discriminator module transferring perceptual assessment.
- An ablation study of the importance of an adversarial approach versus classification hints.

The outline of this paper is as follows. Section 2 reviews the related work. In Section 3 the proposed model, architecture and algorithm are detailed. Section 4 presents quantitative and qualitative results. Finally, the paper is concluded in Section 5. The code will be made publicly available.

2. Related Work

In the past two decades several colorization techniques have been proposed. They can be classified in three classes: scribble-based, exemplar-based and deep learning-based methods. The first two classes depend on human intervention.

The third class is based on deep learning leveraging the possibility of creating easily training data from any color image to learn which colors are assigned to which objects.

Scribble-based methods. In these methods the user provides local hints, as for instance color scribbles, which are then propagated to the whole image. They were initiated with the work of Levin *et al.* [25]. They assume that spatial neighboring pixels having similar intensities should have similar colors. They formalize this premise optimizing a quadratic cost function constrained to the values given by the scribbles. Several improvements were proposed. Huang *et al.* [18] improve the bleeding artifact using edge information of the grayscale image. Yatziv *et al.* [43] propose a luminance-weighted chrominance blending to relax the dependency of the position of the scribbles. Then, Luan *et al.* [28] use the input scribbles to segment the grayscale image and thus better propagate the colors. This class of methods suffer from requiring large amounts of user inputs in particular when dealing with complex textures. Moreover, choosing the correct color palette is not an easy task.

Exemplar-based methods. These methods transfer the colors of a reference image to a grayscale one. Inspired by [16, 34], Welsh *et al.* [42], propose to do it by matching the luminance values and texture information between images. This approach lacks of spatial coherency which yields unsatisfactory results. To overcome this, several improvements have been proposed. Ironi *et al.* [20] transfer some color values from a segmented source image which are then used as scribbles in [25]. In the same spirit, Tai *et al.* [39] construct a probabilistic segmentation of both images to transfer color between any two regions having similar statistics. Charpiat *et al.* [7] deal with the multimodality of the colorization problem estimating for each pixel the conditional probability of colors. Chia *et al.* [9] use the semantic information of the grayscale image. Gupta *et al.* [14] transfer colors based on the features of the superpixel representation of both images. Bugeau *et al.* [4] colorize an image by solving a variational model which allows to select the best color candidate, from a previous selection of color values, while adding some regularization in the colorization. Although this type of methods reduce significantly the user inputs, they are still highly dependent on the reference image which must be similar to the grayscale image.

Deep learning methods. Recently, different approaches have been proposed to leverage the huge amount of grayscale/color image pairs. Cheng *et al.* [8] first proposed a fully-automatic colorization method formulated as a least square minimization problem solved with deep neural networks. A semantic feature descriptor is proposed and given as an input to the network.

In [11], a supervised learning method is proposed through a linear parametric model and a variational autoencoder which is computed by quadratic regression on a large dataset of color images. These approaches are improved by the use of CNNs and large-scale datasets. For instance, Iizuka *et al.* [19] extract local and global features to predict the colorization. The network is trained jointly for classification and colorization in a labeled dataset.

Zhang *et al.* [44] learn the color distribution of every pixel and infer the colorization from the learnt distribution. The network is trained with a multinomial cross entropy loss with rebalanced rare classes allowing for rare colors to appear in the colorized image. In a similar spirit, Larsson *et al.* [24] train a deep CNN to learn per-pixel color histograms. They use a VGG network in order to interpret the semantic composition of the scene as well as the localization of objects and then predict the color histograms of every pixel based on this interpretation. They train the network with the Kullback-Leibler divergence. Again, the colorization is inferred from the color histrograms.

Other CNN based approaches are combined with user interactions. For instance, Zhang *et al.* [45] propose to train a deep network given the grayscale version and a set of sparse user inputs. This allows the user to have more than one plausible solution. Also, He *et al.* [15] propose an exemplar-based colorization method using a deep learning approach. The colorization network jointly learns faithful local colorization to a meaningful reference and plausible color prediction when a reliable reference is unavailable.

Some methods use GANs to colorize grayscale images. Isola *et al.* [21] propose to use conditional GANs to map an input image to an output image using a U-Net based generator. They train their network by combining the L^1-loss with an adapted GAN loss. An extension is proposed by Nazeri *et al.* [30] generalizing the procedure to high resolution images, speeding up and stabilizing the training. Cao *et al.* [5] also use conditional GANs but, to obtain diverse possible colorizations, they sample several times the input noise, which is incorporated in multiple layers in the proposed network architecture, which consists of a fully convolutional non-stride network. Their choice of the LSUN bedroom dataset helps their method to learn the diversity of bedroom colors. Notice, that none of these GANs based methods use additional information such as classification.

3. Proposed Approach

Given a grayscale input image L, our goal is to learn a mapping $\mathcal{G} : L \longrightarrow (a, b)$ such that $I = (L, a, b)$ is a plausible color image and a and b are images representing the chrominance channels in the CIE Lab color space. A plausible color image is one having geometric, perceptual and semantic photo-realism.

In this paper, we learn the mapping $\mathcal{G}$ by means of an adversarial learning strategy. The colorization is produced through a generator $-$equivalent to $\mathcal{G}$ above$-$ that predicts the chrominance channels (a, b). In parallel, a discriminator evaluates how realistic is the proposed colorization $I = (L, a, b)$ of L. To this aim, we propose in Section 3.1 a new adversarial energy that learns the parameters θ and w of the generator $\mathcal{G}_\theta$ and the discriminator D_w, respectively. This is done training end-to-end the proposed network in a self-supervised manner by using a dataset $\mathcal{S}$ of real color images. In particular, given a training image $I_r = (L, a_r, b_r)$ in the CIE Lab color space, a_r and b_r denote the real a and b chrominance channels, respectively.

For the sake of clarity and by a slight abuse of notation, we shall write $\mathcal{G}_\theta$ and D_w instead of θ and w, respectively. Moreover, our generator $\mathcal{G}_\theta$ will not only learn to generate color but also a class distribution vector, denoted by $y \in \mathbb{R}^m$, where m is the number of classes. This provides information about the probability distribution of the semantic content and objects present in the image. The use of a classes' vector was inspired by the work in [19], where they use an additional classification network to better learn global priors. For that, our generator model combines two different modules (see Fig. 2). Let us denote it by $\mathcal{G}_\theta = (\mathcal{G}^1_{\theta_1}, \mathcal{G}^2_{\theta_2})$, where $\theta = (\theta_1, \theta_2)$ stand for all the generator parameters, $\mathcal{G}^1_{\theta_1} : L \longrightarrow (a, b)$, and $\mathcal{G}^2_{\theta_2} : L \longrightarrow y$.

An overview of the model architecture can be seen in Fig. 2 and will be described in Section 3.2. In the next Section 3.1 the proposed adversarial loss is stated.

3.1. The Objective Function

Our objective loss is defined by

$$\mathcal{L}(\mathcal{G}_\theta, D_w) = \mathcal{L}_{\mathrm{e}}(\mathcal{G}^1_{\theta_1}) + \lambda_{\mathrm{p}} \mathcal{L}_{\mathrm{p}}(\mathcal{G}^1_{\theta_1}, D_w) + \lambda_{\mathrm{s}} \mathcal{L}_{\mathrm{s}}(\mathcal{G}^2_{\theta_2}). \quad (1)$$

The first term

$$\mathcal{L}_{\mathrm{e}}(\mathcal{G}^1_{\theta_1}) = \mathbb{E}_{(L,a_r,b_r)\sim\mathbb{P}_r} \left[\| \mathcal{G}^1_{\theta_1}(L) - (a_r, b_r) \|_2^2 \right] \quad (2)$$

denotes the *color error loss*, where $\mathbb{P}_r$ stands for the distribution of real color images and $\| \cdot \|_2$ for the Euclidean norm.

Then,

$$\mathcal{L}_{\mathrm{s}}(\mathcal{G}^2_{\theta_2}) = \mathbb{E}_{L\sim\mathbb{P}_{rg}} \left[\mathrm{KL}\left(y_{\mathrm{v}} \,\|\, \mathcal{G}^2_{\theta_2}(L) \right) \right] \quad (3)$$

denotes the *class distribution loss*, where $\mathbb{P}_{rg}$ denotes the distribution of grayscale input images and $y_{\mathrm{v}} \in \mathbb{R}^m$ the output distribution vector of a pre-trained VGG-16 model [38] (more details are given below). $\mathrm{KL}(\cdot\|\cdot)$ stands for the Kullback-Leibler divergence.

Finally, $\mathcal{L}_{\mathrm{p}}$ denotes the *perceptual loss* which consists of an adversarial Wasserstein GAN loss (WGAN) [1]. Let us first remark that leveraging the WGAN instead of other GAN losses favours nice properties such as avoiding vanishing gradients and mode collapse, and achieves more stable training. To compute it, we use the Kantorovich-Rubinstein duality [22, 40]. Moreover, following the variant proposed by [13], we also include a gradient penalty term constraining the L^2 norm of the gradient of the discriminator with respect to its input and, thus, imposing that $D_w \in \mathcal{D}$, where $\mathcal{D}$ denotes the set of 1-Lipschitz functions. To sum up, the perceptual loss is defined by

$$\begin{aligned} \mathcal{L}_{\mathrm{p}}(\mathcal{G}^1_{\theta_1}, D_w) = {} & \mathbb{E}_{\tilde{I}\sim\mathbb{P}_r} \left[D_w(\tilde{I}) \right] \\ & - \mathbb{E}_{(a,b)\sim\mathbb{P}_{\mathcal{G}^1_{\theta_1}}} \left[D_w(L, a, b) \right] \\ & - \mathbb{E}_{\hat{I}\sim\mathbb{P}_{\hat{I}}} [(\| \nabla_{\hat{I}} D_w(\hat{I}) \|_2 - 1)^2]. \end{aligned} \quad (4)$$

where $\mathbb{P}_{\mathcal{G}^1_{\theta_1}}$ is the model distribution of $\mathcal{G}^1_{\theta_1}(L)$, with $L \sim \mathbb{P}_{rg}$. As in [13], $\mathbb{P}_{\hat{I}}$ is implicitly defined sampling uniformly along straight lines between pairs of point sampled from the data distribution $\mathbb{P}_r$ and the generator distribution $\mathbb{P}_{\mathcal{G}^1_{\theta_1}}$. Let us notice that the minus before the gradient penalty term in (4) corresponds to the fact that, in practice, when optimizing with respect to the discriminator parameters, our algorithm minimizes the negative of the loss instead of maximizing it.

From the previous loss (1), we compute the weights of $\mathcal{G}_\theta, D_w$ by solving the following min-max problem

$$\min_{\mathcal{G}_\theta} \max_{D_w \in \mathcal{D}} \mathcal{L}(\mathcal{G}_\theta, D_w), \quad (5)$$

The hyperparameters λ_{p} and λ_{s} are fixed and set to 0.1 and 0.003, respectively. Let us comment more in detail the benefits of each of the elements of our approach.

The adversarial strategy and the GAN loss $\mathcal{L}_{\mathrm{p}}$. The min-max problem (5) follows the usual generative adversarial game. The ability of GANs [12] in learning probability distributions over large, high-dimensional spaces of data such as color images has found widespread use for many tasks in different areas including image processing,

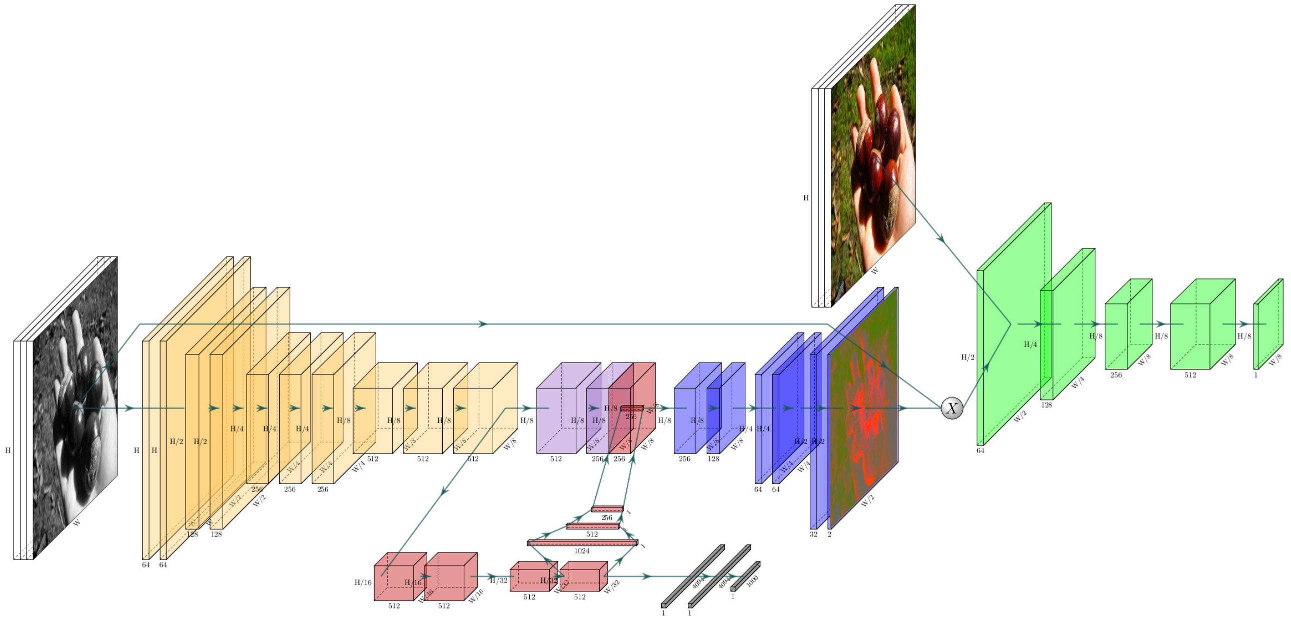

Figure 2. Overview of our model, ChromaGAN, able to automatically colorize grayscale images. It combines a Discriminator network, D_w (in green), and a Generator network, $\mathcal{G}_\theta$. $\mathcal{G}_\theta$ consists of two subnetworks: $\mathcal{G}^1_{\theta_1}$ (yellow, purple, red and blue layers) that outputs the chrominance information $(a, b) = \mathcal{G}^1_{\theta_1}(L)$, and $\mathcal{G}^2_{\theta_2}$ (yellow, red and gray layers) which outputs the class distribution vector, $y = \mathcal{G}^2_{\theta_2}(L)$.

computer vision, text generation, and natural language processing (*e.g.*, [6, 21, 23, 26, 32, 41, 47]). GAN learning strategy is based on a game theory scenario between two networks, the generator and the discriminator, having adversarial objectives and aiming to converge to a Nash equilibrium [3, 17, 29, 31, 37]. The generator usually maps a source of noise from a latent space to the input space and the discriminator receives either a generated or a real data and must distinguish between both. The goal of this training procedure is to learn the parameters of the generator, G, so that the probability distribution of the generated data is as close as possible to the one of the real data. To do so, the discriminator, D, is trained to maximize the probability of assigning the correct label to both real examples and samples from the generator G, while G is trained to fool D by generating realistic examples. The authors of [33] introduced convolutional layers to the GANs architecture. However, these initial proposals optimize the Jensen-Shannon divergence that can be non-continuous with respect to the generator parameters. Besides, the WGAN [1, 2] minimizes an approximation of the Earth-Mover distance or Wasserstein-1 metric between two probability distributions. It is known to be a powerful tool to compare probability distributions with non-overlapping supports, in contrast to the Kullback-Leibler divergence and the Jensen-Shannon divergence which produce the vanishing gradients problem. Also, the WGAN alleviates the mode collapse problem which is interesting when aiming to be able to capture multiple possible colorizations.

As the experiments show in Section 4 and has been also noticed by some authors in different contexts [21], the adversarial GAN model produces sharp and colorful images favouring the emergence of a perceptually real palette of colors instead of ochreish outputs produced by colorization using only terms such as the L^2 or L^1 color error loss.

Color Error Loss. In some colorization methods [24, 44] the authors propose to learn a per-pixel color probability distribution allowing them to use different classification losses. Instead, we chose to learn two chrominance values per-pixel using the L^2 norm. As mentioned, only using this type of loss yields ochreish outputs. However, in our case the use of the perceptual GAN-based loss relaxes this effect making it sufficient to obtain notable results (Section 4).

Class Distribution Loss. The KL-based loss $\mathcal{L}_\text{s}(\mathcal{G}^2_{\theta_2})$ (3) compares the generated density distribution vector $y = \mathcal{G}^2_{\theta_2}(L)$ to the ground truth distribution $y_\text{v} \in \mathbb{R}^m$. The latter is computed using the VGG-16 [38] pre-trained on ImageNet dataset [10]. The VGG-16 model was trained on color images, thus, in order to use it without any further training, we re-shape the grayscale image as (L, L, L). The class distribution loss adds semantic interpretation of the scene. The effect of this term is analyzed in Section 4.

3.2. Detailed Model Architecture

The proposed GAN architecture is conditioned by the grayscale image L through the loss (1) proposed in Section 3.1, and contains three distinct parts. The first and second one, belonging to the generator, focus on geometrically and semantically generating a color image (i.e., the chrominance channels (a, b)) and classifying its semantic content. The third one belongs to the discriminator network. As pointed out above, the discriminator learns to distinguish between real and fake data. Moreover, the generator does benefit from the feedback of the discriminator in order to generate realistic color images. An overview of the model is shown in Fig. 2. In the remaining of the section we will describe the architecture of the generator and discriminator. More details are available in the supplementary material.

Generator Architecture. The generator $\mathcal{G}_\theta$ is made of two subnetworks (denoted by $\mathcal{G}^1_{\theta_1}$ and $\mathcal{G}^2_{\theta_2}$) divided in three stages with some shared modules between them. Both of them will take as input a grayscale image of fixed size $H \times W$. The subnetwork $\mathcal{G}^1_{\theta_1}$ outputs the chrominance information, $(a, b) = \mathcal{G}^1_{\theta_1}(L)$, and the subnetwork $\mathcal{G}^2_{\theta_2}$ outputs the computed class distribution vector, $y = \mathcal{G}^2_{\theta_2}(L)$.

The first stage (displayed in yellow in Fig. 2) is shared between both subnetworks. It has the same structure as the VGG-16 with key differences that include the removal of the three last fully-connected layers at the top of the network. Moreover, we initialize them with pre-trained VGG-16 weights which are not frozen during training.

From this first stage on, both subnetworks, $\mathcal{G}^1_{\theta_1}$ and $\mathcal{G}^1_{\theta_2}$, split into two distinct tracks. The first one (displayed in purple in Fig. 2) process the data by using two modules of the form Convolution-BatchNorm-ReLu. The second track (displayed in red in Fig. 2), present in the two subnetworks, first processes the data by using four modules of the form Convolution-BatchNorm-ReLu, followed by three fully connected layers (shown in red in Fig. 2). This second path (displayed in gray in Fig. 2) outputs $\mathcal{G}^2_{\theta_2}$ providing the class distribution vector. To generate the probability distribution $y = \mathcal{G}^2_{\theta_2}(L)$ of the m semantic classes, we use a softmax function. Notice that the path going from the input layer to this node is a classification network and is initialized with pre-trained classification weights. However, as part of this path is shared with the generator $\mathcal{G}^1_{\theta_1}$, once the network is trained, this path not only has learned to give a class distribution close to the output of the VGG-16, but also to generate useful information to help the colorization process. This could be understood as fine tuning the network in order to learn to perform two tasks at once.

In the third stage both branches are fused (in red and purple in Fig. 2) by concatenating the output features predicting the channels (a, b). This is achieved by processing the information through six modules of the form Convolution-ReLu with two up-sampling layers in between.

Note that while performing back propagation with respect to the class distribution loss, only the second subnetwork $\mathcal{G}^2_{\theta_2}$ will be affected. In the case of the color error loss, the entire network will be affected.

Discriminator Architecture. The discriminator network D_w is based on the Markovian discriminator architecture (PatchGAN [21]). The PatchGAN discriminator keeps track of the high-frequency structures of the generated image compensating the fact that the L^2 loss $\mathcal{L}_{\text{e}}(\mathcal{G}^1_{\theta_1})$ fails in capturing high-frequency structures but succeeds in capturing low-level ones. In order to model the high-frequencies, the PatchGAN discriminator focuses on local patches. Thus, instead of penalizing at the full image scale, it tries to classify each patch as real or fake. Hence, rather than giving a single output for each input image, it generates a value for each patch. We follow the architecture defined in [21] where the input and output are of size $H \times W$ and $H/8 \times W/8$, respectively, and where both of them are defined in the CIE Lab color space.

4. Experimental results and Discussion

In this section we evaluate the proposed method both quantitatively and qualitatively. Notice that evaluating the quality of a colorized image quantitatively is a challenging task and an output equal to the ground truth would be only one of the several potential solutions. For instance, for some objects, different colors could perfectly suit to the same single object. To give an example, a ball could be painted in any color and still would look realistic to the human eye. Therefore, quantitative measures reflecting how close the outputs are to the ground truth data are not the best measures for this type of problem. Thus, in order to quantify the quality of our method in comparison with other methods, we will not only use a metric based on a distance with respect to the ground truth, but we will also perform a perceptual study to quantify the realism of the colorized images regarding the perception in the human visual system.

To assess the effect of each term of our loss function in the entire network, we perform an ablation study by evaluating the following variants of our method.

- **ChromaGAN.** The proposed method where the adversarial and classification approach are used.
- **ChromaGAN w/o Class.** $\lambda_s = 0$: Our method without class distribution loss.
- **Chroma Network.** $\lambda_p = 0$: Our method without adversarial approach.

4.1. Dataset

We train each variant of the network end-to-end on 1.3M images from the subset of images [35] taken from ImageNet [10]. It contains objects from 1000 different categories and color conditions, including grayscale images.

This could be seen as a dropout method to prevent overfitting. Due to the presence of fully connected layers in our network, the input size to the classification branch has to be fixed. We chose to work with input images of 224 × 224 pixels as is done when training the VGG-16 [38] on ImageNet. Nonetheless, the input size of our network is not restricted to the input size of the trained VGG-16. Therefore, we have resized each image in the training set and convert it to a three channels grayscale image by triplicating the luminance channel L.

4.2. Implementation Details

We train the network for a total of five epochs and set the batch size to 10, on the 1.3M images from the ImageNet training dataset resized to 224 × 224. A single epoch takes approximately 23 hours on a NVIDIA Quadro P6000 GPU. The prediction of the colorization of a single image takes an average of 4.4 milliseconds. We minimize our objective loss using Adam optimizer with learning rate equal to $2e-5$ and momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We alternate the optimization of the generator $\mathcal{G}_\theta$ and discriminator D_w. The first stage of the network (displayed in yellow in Fig. 2), takes as input a grayscale image of size 224 × 224, and is initialized using the pre-trained weights of the VGG-16 [38] trained on ImageNet.

4.3. Quantitative Evaluation

We quantitatively assess our method in terms of *peak signal to noise ratio* (PSNR) and perceptual realism. We compute the PSNR of the obtained (a, b) images with respect to the ground truth and compare them to the ones obtained for other fully automatic methods as shown in Table 1. The table shows the average of this measure over all the test images. One can observe that, in general, our PSNR values are higher than those obtained in [19, 24, 44]. Moreover, comparing the PSNR of the three variants of our method the highest one is achieved by Chroma Network. This is not surprising since the training loss of this method gives more importance to the quadratic color error term compared to the losses of ChromaGAN and ChromaGAN w/o Class.

Method	PSNR (dB)
ChromaGAN	24.84
ChromaGAN w/o Class	25.04
Chroma Network	**25.57**
Iizuka *et al.* [19]	23.69
Larsson *et al.* [24]	24.93
Zhang *et al.* [44]	22.04

Table 1. Comparison of the average PSNR values for automatic methods, some extracted from the table in [45]. The experiment is performed on 1000 images of the ILSVRC2012 challenge set [36].

Method	Naturalness
Real images (method 0)	72.6%
ChromaGAN (method 1)	66.9%
ChromaGAN w/o Class (method 2)	62.0%
Chroma Network (method 3)	58.4%
Iizuka *et al.* [19] (method 4)	48.9%

Table 2. Numerical detail of the curve in Fig. 3. The values shows the mean naturalness over all the experiments of each method.

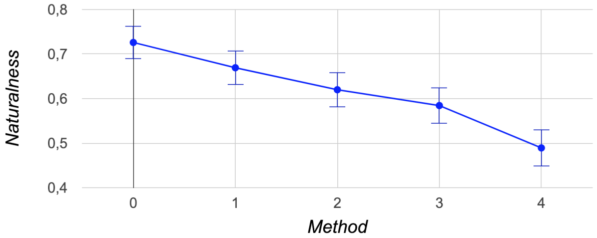

Figure 3. Results of the perceptual study. Method 0 corresponds to the real images, 1 to ChromaGAN, 2 to ChromaGAN w/o classification, 3 to Chroma Network, and 4 the method by [19].

Regardless the PSNR value of Table 1 we would have expected the opposite given the qualitative results. In order to verify our intuition we perform the following perceptual realism study on our colorization results. Images were shown to non-expert participants, where some are natural color images and others are the result of a colorization method such as ChromaGAN, ChromaGAN w/o classification, Chroma Network and Iizuka *et al.* [19]. We include the latter to our study since their loss is similar to Chroma Network differing in the architecture. For each image the participant shall indicate if the colorization is realistic or not in a pre-attentive observation. The set of 50 images is taken randomly from a set of 1000 images composed of 200 ground truth (from both ImageNet [10] and Places datasets [46]), 200 ChromaGAN results, 200 Chroma Network results, 200 ChromaGAN w/o classification results and 200 results of [19]. The study was performed 62 times. In Fig. 3 and Table 2 the results of perceptual realism are shown for each method. The mean and standard deviation are indicated for each test. One can observe that in the case of our method, the one that is perceptually more realistic is ChromaGAN which corresponds to what we expected. For all the variants of our algorithm the perceptual results are better compared to Iizuka's *et al.* [19] results. Moreover, by comparing the results of Chroma Network and ChromaGAN w/o Class, we can see that the adversarial approach plays a more important role than using class distribution while generating natural images.

4.4. Qualitative Evaluation

We compare our results with the results obtained in [19, 24, 44] by using the publicly available online demos. The

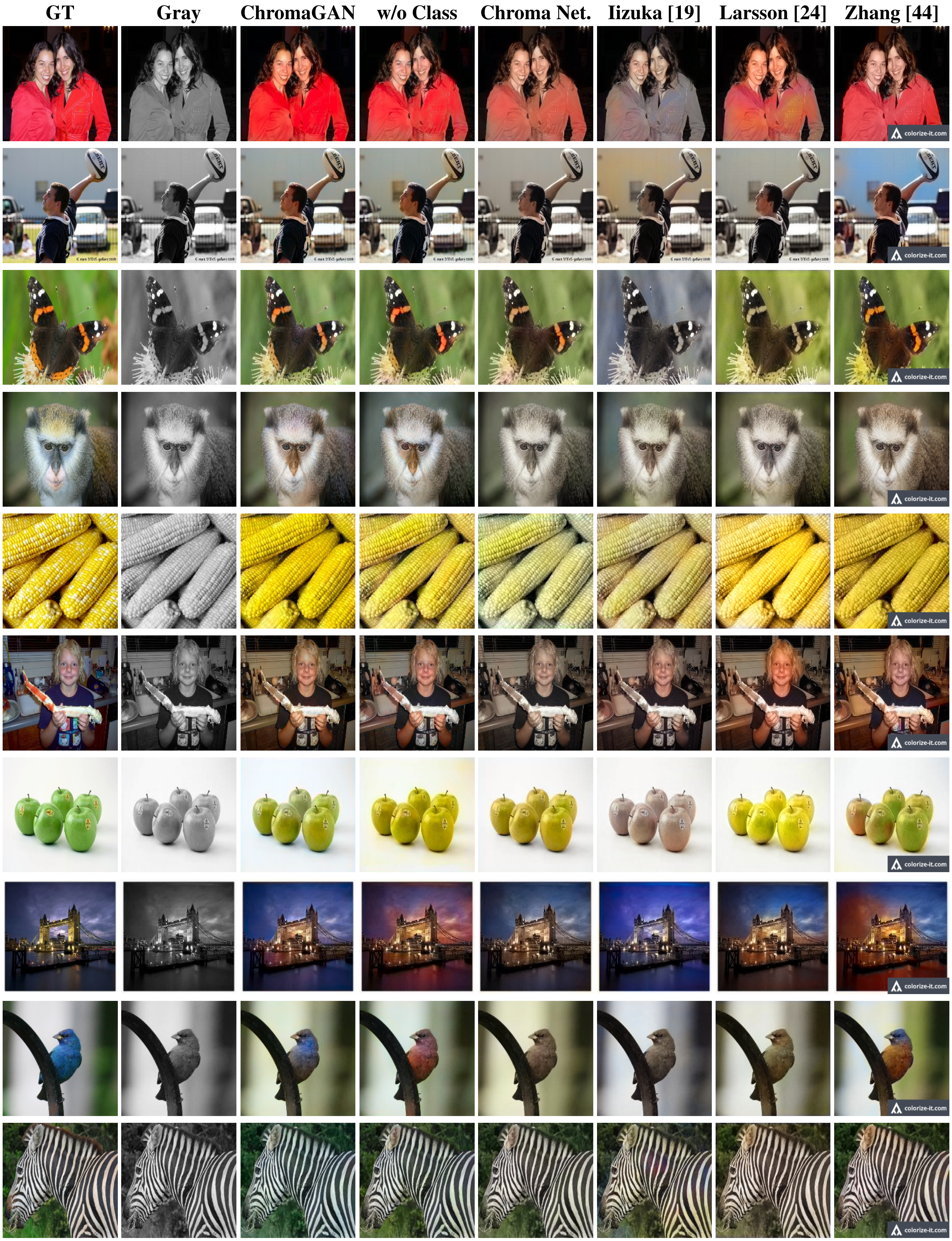

Figure 4. Some qualitative results using, from right to left: Ground truth, Gray scale, ChromaGAN, ChromaGAN w/o Classification, Chroma Network, Iizuka *et al*. [19], Larsson *et al*. [24] and Zhamg *et al*. [44]

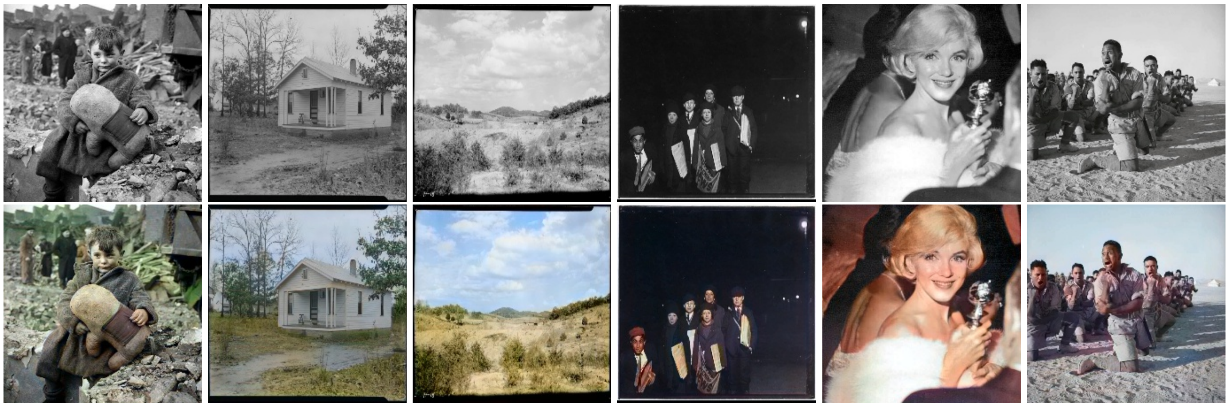

Figure 5. Colorization results of historical black and white photographs using the proposed ChromaGAN. Note that old black and white photographs are statistically different than actual ones, thus, making the process of colorize more difficult.

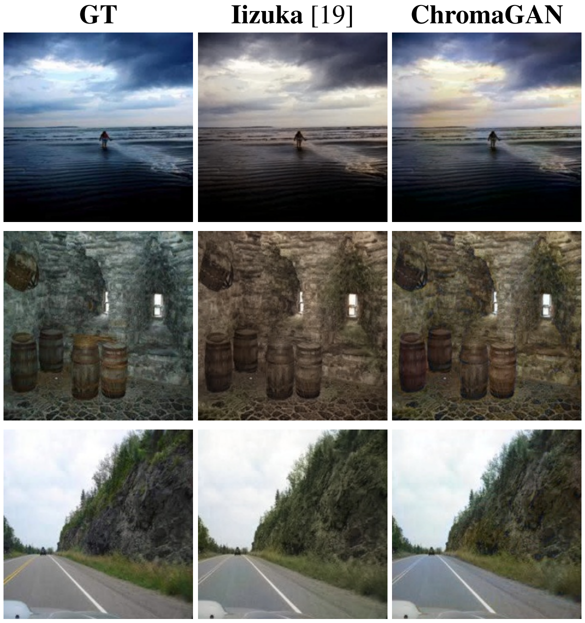

Figure 6. Results in some of the images from the validation set of the Places Dataset. Left: Ground truth, middle: Iizuka et al [19], right: ChromaGAN. Notice that the model of [19] is trained using the Places Dataset. On the contrary, we use our model trained on the ImageNet dataset. Results are comparable.

methods are trained with ImageNet dataset in the case of [44, 24] and with Places dataset in the case of [19]. We show several colorization results on the validation set of ImageNet dataset in Fig. 4 and on Places in Fig. 6. As we can observe, the method of [19] and Chroma Network tend to output muted colours in comparison to the lively colors obtained with ChromaGAN, ChromaGAN w/o class and [24, 44]. Also, ChromaGAN is able to reconstruct color information by adding natural and vivid colors in almost all the examples (specially, the first, fifth, seventh, ninth and tenth rows). Desaturated results are mainly obtained by [19] and with our method without using the adversarial approach (specially in the first, second, third, fourth, fifth and ninth rows), in some cases also by [24] (second, fourth and ninth rows). Also, color boundaries are not clearly separated generally in the case of [19] and sometimes by our model without class (seventh row) and [24] (third, fourth and ninth rows). Inconsistent chromaticities can be found in the second and seventh row by [44] where the wall is blue and the apples green and red at the same time. Third and eighth rows display some failure cases of our method: the bottom-right butterfly wing is colored in green. In fact, the case of the eighth row shows a difficult case for all the methods. Additional examples on the Imagenet and COCO dataset [27] can be found in the supplementary material. For the sake of comparison, we also show some results of Places dataset [46] by using ChromaGAN trained on ImageNet, together with the results of [19] trained on Places dataset in Fig. 6.

Legacy Black and White Photographs. ChromaGAN is trained using color images where the chrominance information is removed. Due to the progress in the field of photography, there is a great difference in quality between old black and white images and modern color images. Thus, generating color information in original black and white images is a challenging task. Fig. 5 shows some results. Additional examples can be found in the supplementary material, where we also include results applied on paintings.

5. Conclusion

In this paper, a novel colorization method is detailed. The proposed ChromaGAN model is based on an adversarial strategy that captures geometric, perceptual and semantic information. A variant of ChromaGAN which differs in whether the learning of the distribution of semantic classes is incorporated or not in the training process is also encouraging. Both cases prove that our adversarial technique provides photo-realistic colorful images. The quantitative and qualitative comparison with state-of-the-art methods show that our method outperforms them in terms of perceptual realism and PSNR.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv:1701.07875*, 2017.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[3] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 224–232. JMLR. org, 2017.

[4] A. Bugeau, V.-T. Ta, and N. Papadakis. Variational exemplar-based image colorization. *IEEE Transactions on Image Processing*, 23(1):298–307, 2014.

[5] Y. Cao, Z. Zhou, W. Zhang, and Y. Yu. Unsupervised diverse colorization via generative adversarial networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 151–166. Springer, 2017.

[6] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros. Everybody dance now. *arXiv:1808.07371*, 2018.

[7] G. Charpiat, M. Hofmann, and B. Schölkopf. Automatic image colorization via multimodal predictions. In *European conference on computer vision*, pages 126–139. Springer, 2008.

[8] Z. Cheng, Q. Yang, and B. Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423, 2015.

[9] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin. Semantic colorization with internet images. In *ACM Transactions on Graphics (TOG)*, volume 30, page 156. ACM, 2011.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[11] A. Deshpande, J. Rock, and D. Forsyth. Learning large-scale automatic image colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 567–575, 2015.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Adv in neural inf processing systems*, pages 2672–2680, 2014.

[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Adv in Neural Inf Processing Systems*, pages 5769–5779, 2017.

[14] R. K. Gupta, A. Y.-S. Chia, D. Rajan, E. S. Ng, and H. Zhiyong. Image colorization using similar images. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 369–378. ACM, 2012.

[15] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)*, 37(4):47, 2018.

[16] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340. ACM, 2001.

[17] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[18] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu. An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 351–354. ACM, 2005.

[19] S. Iizuka, E. Simo-Serra, and H. Ishikawa. 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 (TOG)*, 35(4):110, 2016.

[20] R. Ironi, D. Cohen-Or, and D. Lischinski. Colorization by example. In *Rendering Techniques*, pages 201–210. Citeseer, 2005.

[21] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[22] L. Kantorovitch. On the translocation of masses. *Management Science*, 5(1):1–4, 1958.

[23] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.

[24] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision*, pages 577–593. Springer, 2016.

[25] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM transactions on graphics (tog)*, volume 23, pages 689–694. ACM, 2004.

[26] K. Lin, D. Li, X. He, Z. Zhang, and M.-T. Sun. Adversarial ranking for language generation. In *Advances in Neural Information Processing Systems*, pages 3155–3165, 2017.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[28] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 309–320. Eurographics Association, 2007.

[29] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[30] K. Nazeri and E. Ng. Image colorization with generative adversarial networks. *arXiv preprint arXiv:1803.05400*, 2018.

[31] H. Prasad, P. LA, and S. Bhatnagar. Two-timescale algorithms for learning nash equilibria in general-sum stochastic

games. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1371–1379. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[32] A. Pumarola, A. Agudo, A. Martinez, A. Sanfeliu, and F. Moreno-Noguer. GANimation: Anatomically-aware Facial Animation from a Single Image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[33] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[34] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[37] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[39] Y.-W. Tai, J. Jia, and C.-K. Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 747–754. IEEE, 2005.

[40] C. Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[41] P. Vitoria., J. Sintes., and C. Ballester. Semantic image inpainting through improved wasserstein generative adversarial networks. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP,*, pages 249–260. INSTICC, SciTePress, 2019.

[42] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 277–280. ACM, 2002.

[43] L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *IEEE transactions on image processing*, 15(5):1120–1129, 2006.

[44] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[45] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017.

[46] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2018.

[47] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.