Automatic Image Colorization: Comparative Overview

Bigarella Chiara
Student nr. 2004248

Poletti Silvia
Student nr. 1239133

Abstract

The colorization of greyscale images is an ill-posed problem that was approached in different ways in literature. This project provides a comparative analysis concerning five pre-trained colorization models and a cartoonization-based baseline of our invention. The performances are assessed through both quantitative and qualitative metrics, with a final evaluation of the results with respect to image filtering.

1. Introduction

Automatic image colorization consists in assigning colors to greyscale images, in a plausible and realistic way, that can potentially fool a human observer. This task is particularly important for advertisement and film industries, for photography and artist assistance [13] and for reducing both cost and time required in producing comics or animated movies [11].

Despite many advances in deep learning, automatic image colorization is still nowadays considered a difficult task and its applications are still limited. This is mainly due to the fact that colorization is an ill-posed problem that doesn't have a unique solution. Indeed, from a mathematical point of view, it requires to map a 2D greyscale image to a 3D colored one, and there are multiple plausible solutions that can be effectively used to colorize the same object.

Moreover, to obtain satisfactory results, the colorization models can't simply learn the most frequent and dominant colors and generalize over the data, but must accounts also for rare colors.

In our project we want to explore some of the most recent models for automatic image colorization, such as Eccv16 [15], Siggraph17 [10], ChromaGAN [13], InstColorization [12]. We also used an older model developed by Dahl [8] and we implemented a simple model based on a convolutional autoencoder and cartoonization, as a baseline.

To compare all these models, we performed the classification task on the colorized images and measured the classification accuracy achieved by AlexNet. Furthermore, we computed some advanced metrics, such as Learned Perceptual Image Patch Similarity (LPIPS), Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM), and also performed a short Turing Test.

Overall, the best performing model for what concerns the quantitative metrics is the most recent and sophisticated one, i.e. ChromaGAN, as we expected. Finally, according to 124 human partecipants in the Turing Test, the best performing models are ChromaGAN and Siggraph17, while InstColorization performed poorly.

2. Related Work

In the past years, several image colorization models were developed. They belong to three main categories: scribble-based, example-based and deep learning models. Regarding the latter, a common approach was using a CNN based on the VGG-16 architecture. Dahl's model, Zhang's models Eccv16 [15] and Siggraph17 and the more recent InstColorization [12] all belong to this category. Despite some small changes in the models' architectures, the most important difference among those models consists in the loss functions they use.

Other recent and successful deep learning approaches concern the employment of GANs, like in ChromaGAN.

Another approach consists in using memory neural networks that are augmented with an external memory module used to store some critical information. An example of this method is MemoPainter [11], which combines a colorization network with a memory network to store rare instances, thus allowing the colorization network to produce high-quality colorization with a limited amount of data.

Concerning the scribble-based approach, some models exploit user inputs to improve the colorization process. That's the case of Siggraph17 with user local hints, and the LBIE model by [9], which uses the image textual description provided by the user.

The aforementioned methods are just a few examples: there are many more approaches that we won't cover here.

3. Dataset

We considered three types of images: 4023 originally colored images from five different datasets, 18 originally black and white images from various artists and 180 filtered images (see more details in Section 5.4) obtained starting from 18 originally colored images.

Our data includes heterogeneous images, representing different environments, situations and subjects, coming from various sources:

- a subset of ImageNet made of 12 classes (200 images each) taken from [6], ten of which are easily classifiable classes (tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas

pump, golf ball and parachute) while the other two are hard to classify (Samoyed and Rhodesian ridgeback);

- a subset of 100 randomly selected images from Pascal VOC [2] representing realistic scenes in which the subjects could be animals, human beeings, plants, rooms, landscapes, various objects and vehicles;
- a subset of 200 randomly selected images form Places205 [3] about mountain, desert, sea, beach and island landscapes.
- a subset of Bird Species [4] made of 8 classes (100 images each), depicting birds with unusual colors (Cuban Tody, Fire Tailed Myzornis, Flamingo, Nicobar Pigeon and Pink Robin) and best known birds (Bald Eagle, Ostrich and Touchan);
- a subset of Flower Species [1] made of 6 classes (from 50 to 100 images each), depicting flowers with unusual colors and shapes (Purple Coneflower, Grape Hyacinth, Hibiscus) and best known flowers (Rose, Water Lily and Giant White Arum Lily).

The images have been preprocessed by using OpenCV (ChromaGAN and InstColorization) or Pillow combined with Skimage (Baseline, Dahl, Zhang, Siggraph) and have been reshaped to various formats. Dahl also required center cropping and desaturation. Despite the preliminar reshape, the outputs generated by Zhang, Siggraph and ChromaGAN have the same shape of the original image.

Given an RGB image, we obtain the corresponding image in the *Lab* color space, which is composed of 3 channels: L for perceptual lightness ($L = 0$ is white, $L = 100$ is black), a and b for four primary colors ($a = \pm 100$ are red and green, $b = \pm 100$ are yellow and blue). Our models get only the L channel as input (greyscale images) with the goal of predicting the a and b channels.

In particular, the classification with AlexNet required the images to be normalized in the range $[0, 1]$ and a further standardization according to the mean and standard deviation of the training set. On the other hand, the LPIPS metric required image normalization in the range $[-1, 1]$.

4. Methods

In this section, we propose a simple autoencoder based on cartoonization as baseline. Then, we introduce some state-of-the-art pre-trained models taken from the literature. The architectures of the proposed methods are reported in the Appendix.

4.1. Baseline

As a baseline, we built with Keras a simple autoencoder having 8 Convolutional layers for the encoding part (ReLU activations, zero-padding, 3×3 kernels and sometimes 2×2 strides), while the decoding part consists in the combination of 5 Convolutional layers (ReLU activations except for the last layer, zero-padding and 3×3 kernel) and 3 UpSampling layers of size 2×2. The encoder learns a compact representation of the black and white input image and the decoder generates the corresponding novel coloured image.

The model was trained (50 epochs) on a heterogeneous dataset containing all the data available.

Moreover, we enriched this model with a novel approach: instead of using the original dataset, we fed the model with the cartoonized (black and white) version of the images, computed with the pre-trained GAN-based cartoonization model by [14]. This cartoonization provides fine-grained informative results and synthesizes the original images in order to exclude noisy elements that could interfere with the colorization task. The model produces cartoonized colored images whose a and b channels are combined with the L channel of the original *Lab* images. Therefore, we mantain the original details of the pictures, while producing a more precise and sectorial colorization.

For comparison, we also include in our experiments the Baseline without cartoonization (Baseline w/c).

4.2. Dahl

Dahl's model consists in an autoencoder from black and white images to colored ones, with residuals connections. As shown in Figure 5, the encoder is a VGG-16 network with ReLU activations, that takes in input greyscale images and infers some color information at each layer. This information is added up in the decoder thanks to the residual connections, until a $224 \times 224 \times 3$ tensor is constructed. In the last layer of the decoder, a sigmoid activation is used to squash the values between 0 and 1.

Finally, the model's loss function is the average of the following three Euclidean Distances:

- the Euclidean Distance computed between the original colored image and the network output;
- the Euclidean Distance computed between the original colored image and the network output, blurred with a 3 pixel gaussian kernel;
- the Euclidean Distance computed between the original colored image and the network output, blurred with a 5 pixel gaussian kernel.

The biggest disadvantages of this model are the desaturated results and the possible crop and reshape during the preprocessing, leading to some information loss.

4.3. Eccv16

The innovation introduced by Zhang's colorization model is not the model's architecture (a CNN made of 8

blocks of two or three repeated Convolutional and ReLU layers followed by a BatchNorm layer, as shown in Figure 6b) but rather a more suitable loss function for saturated colorization, combined with class rebalancing, which allows to increase the diversity of colors in the results.

Since an object can potentially have several plausible colorization, the model accounts for the multimodal distribution of possible colors for each pixel. The intrinsic multimodal nature of the colorization problem can't be captured by a simple Euclidean loss between the target and the predicted colors. Instead, the model learns a map $\mathcal{G} : \mathbb{R}^{H\times W\times 1} \rightarrow [0,1]^{H\times W\times Q}$ from the grayscale input to a probability distribution $\hat{Z} = P(a,b)$ over $Q = 313$ possible (a,b) pairs (i.e. colors), which were obtained through the quantization of the *ab* output space, as shown in Figure 6a. Then, the multimomial crossentropy los is defined as:

$$\mathcal{L}_{cl}(\hat{Z}, Z) = -\sum_{h,w} v(Z_{h,w}) \sum_{q} Z_{h,w,q} log(\hat{Z}_{h,w,q})$$

where Z is the soft-encoded target (obtained by taking the 5-nearest neighbors in the quantized *ab* space for each groundtruth pixel, and weighting them according to their distance from the groundtruth) and v is a weighting function for class rebalancing, in order to emphasize rare colors.

To conclude, the final predicted colorization $\hat{Y}$ is the annealed-mean of the distribution $\hat{Z}$, which consists in taking the mean of the softmax distribution $\sigma_T(\hat{Z}) = \sigma(\hat{Z}/T)$ adjusted according the temperature parameter T. This avoids desaturated or spatially inconsistent results.

4.4. Siggraph17

This other Zhang's model is an interactive method that is trained on grayscale images integrated by concatenation with some sparse user inputs. Then, the model can be tested in its automatic colorization version.

The model has a U-Net architecture $\mathcal{F}_\theta$ made of 10 convolutional blocks, each followed by a BatchNorm layer (Figure 7 in blue). Symmetric shortcuts are added to upsampling Convolution layers and make it easier for deeper layers to access to important low-level information. Note that the subnetwork consisting of the first 8 convolutional blocks without shortcut connections, corresponds to the Eccv16 architecture.

The Local Hints Network (Figure 7 in red) incorporates the user points U_l and predicts (as a side task, for user recommendation) a color distribution $\hat{Z}$. The Global Hints Network (Figure 7 in green) transforms the global hint U_g by 1×1 Convolutional layers, and adds the result into the main colorization network.

The network learns to minimize the following loss:

$$\mathcal{L}_\delta(\mathcal{F}_\theta(X,U), Y) = \sum_{h,w} \sum_{q} \ell_\delta(\mathcal{F}_\theta(X,U)_{h,w,q}, Y_{h,w,q})$$

where Y and X are the groundtruth image and its greyscale version, $\mathcal{F}_\theta(X,U)$ is the network output based on the trainable weights θ and the user local or global hint U, and ℓ_δ is a smooth-ℓ_1 loss that doesn't consider the vibrant but artifact-prone colorization with class-rebalancing.

4.5. ChromaGAN

The strength of ChromaGAN is to use the semantic understanding of the depicted scene combined with a generative adversarial network (GAN). In fact, the semantic class distribution learning makes ChromaGAN capable of variability (it can provide different colors for objects belonging to the same category, as it happens in reality) while the generative adversarial learning leads to vivid and vibrant colorizations.

The generator $\mathcal{G}_\theta$ is divided into two jointly trained subnetworks: the first one outputs the chrominance information $\mathcal{G}^1_{\theta_1}(L) = (a,b)$ (Figure 8 in blue) and the second one is a classification network giving in output the class distribution vector $\mathcal{G}^2_{\theta_2}(L) = y$ (Figure 8 in grey) that is trained to be close to the VGG-16 output, in order to generate useful information for the colorization process. The inital layers (Figure 8 in yellow) are shared and initialized with the pretrained VGG-16 weights. Then, both the subnetworks split into two tracks (Figure 8 in purple and red for $\mathcal{G}^1_{\theta_1}$, and in red and grey for $\mathcal{G}^2_{\theta_2}$). The results are fused by concatenation and used to generate the colors.

The discriminator $\mathcal{D}_w$ focuses on the local patches of the generated image and classifies each of them as real or fake. The ultimate goal is to find the optimum of:

$$\min_{\mathcal{G}_\theta} \max_{\mathcal{D}_w} \mathcal{L}(\mathcal{G}_\theta, \mathcal{D}_w) = \mathcal{L}_e(\mathcal{G}^1_{\theta_1}) + \lambda_g \mathcal{L}_g(\mathcal{G}^1_{\theta_1}, \mathcal{D}_w) + \lambda_s \mathcal{L}_s(\mathcal{G}^2_{\theta_2})$$

where $\mathcal{L}_e$ is the expectation of the Euclidean distance between the colorization and the real colors, $\mathcal{L}_s$ is the expectation of the Kullback-Leibler divergence of the predicted class distribution and the VGG-16 pre-trained class distribution, both computed on the greyscale images, and $\mathcal{L}_g$ is an adversarial loss. Note that backpropagation with respect to $\mathcal{L}_s$ only affects $\mathcal{G}^2_{\theta_2}$, while backpropagation with respect to $\mathcal{L}_e$ affects the whole network.

4.6. InstColorization

Instead of just performing learning and colorization on the entire image, InstColorization learns meaningful object-level semantics within the bounding boxes localized by an object detector. Then, it uses two colorization networks: the first colorizes the whole image and the second the patches (resized to 256×256) in the bounding boxes. These networks have different weights but share the same architecture, which is the one used by Zhang's Eccv16, as well as the loss function. Once the first network is trained, its learned weights are used to inizialize the second network.

	Original	B&W	Baseline w/c	Baseline	Dahl	Eccv16	Siggraph17	ChromaGAN	InstColorization
Pre-trained	74.5%	43.1%	32.5%	34.0%	39.8%	42.7%	43.2%	46.8%	49.5%
Feature Extraction	97.2%	84.2%	79.4%	80.4%	87.8%	87.6%	88.9%	90.2%	90.0%
Finetuning	99.7%	63.0%	58.4%	63.6%	64.4%	80.9%	79.9%	77.9%	73.7%

Table 1: Summary of the classification accuracy of Alexnet in three different settings: Alexnet pre-trained on Imagenet, Alexnet feature extraction for the ImageNet subset, AlexNet finetuning for the Birds and Flowers dataset.

Figure 1: Colorization comparison on two images from ImageNet Church (first row) and Bird Species Flamingo (second row).

At the end of the second network's training, the resulting full-image features and object-level features have to be combined in a consistent way by a fusion module. This allows to obtain better results on scenes with multiple objects in a cluttered background. The whole process is reported in Figure 9a.

In particular, the fusion (Figure 9b) takes place at multiple layers of the colorization networks. For each layer, the full-image feature and the N object-level features (N is the number of detected objects) are processed by a small CNN and then combined by taking the weighted sum of the stack composed by the full-image weight map and the patches' weight maps, which have been previously reshaped (and zero padded) using the size and location of the bounding boxes for each object.

5. Experiments

To compare the results of each model, we computed several metrics: classification with AlexNet, LPIPS, PSNR and SSIM (all quantitative metrics) and a Turing test on a few images (qualitative metric). Finally, we applied image filtering to evaluate possible improvements in the performances.

5.1. Classification with AlexNet

First, we considered the AlexNet classifier pre-trained on ImageNet and tested on the ImageNet subset in its original, black and white and re-colorized versions. Table 1 reports the AlexNet classification accuracy in this setting and in other two settings that we will discuss later in this section. Note that the Baseline without cartoonization (Baseline w/c) always reaches a slightly worse accuracy than the Baseline combined with cartoonization which can actually improve the colorization performance.

The great gap in the accuracies computed on the original and the black and white versions of the images suggests that colors play an important role in image classification.

The best colorizations according to this experiment are given by ChromaGAN and InstColorization, while the Baseline and Dahl are not even able to improve the accuracy with respect to the black and white images.

Overall, the accuracy on the models' colorizations is much lower than the one computed on the original images and the latter is relatively low. Therefore we applied feature extraction to better focus on our ImageNet subset: we used the pre-trained AlexNet as a fixed feature-extractor, and only updated the final layer (for 2 epochs) in order to consider just our 12 ImageNet classes. This resulted in more reliable accuracy values and all the models except the Baseline are able to outperform the black and white images.

For a further comparison, we applied finetuning to perform classification on the birds and flowers images, which present more vibrant and diverse colors than our ImageNet subset: we updated (for 2 epochs) all the AlexNet parameters for the new task. In this new setting we have, as expected, a greater gap than before between the original and the black and white accuracies, meaning that the color is

	Baseline w/c	Baseline	Dahl	Eccv16	Siggraph17	ChromaGAN	InstColorization
PSNR	-	-	16.87	20.88	22.19	21.66	21.41
SSIM	-	-	0.56	0.86	0.87	0.87	0.88
LPIPS	0.28	0.28	0.36	0.23	0.20	0.21	0.23

Table 2: Summary of the PSNR, SSIM and LPIPS metrics computed on the different models.

much more relevant. Indeed, all the models including the Baseline with cartoonization are able to improve the accuracy with respect to the black and white images.

The best colorizations according to this experiment are given by the Eccv16 and Siggraph17 models, which are able to generalize better across different datasets.

In this last setting, we can notice a general decreasing in the accuracy (except for the original images) with respect to the feature extraction using the ImageNet subset. This is due to the fact that our pre-trained models have been trained on Image-Net and their colorization of birds and flowers images are overall bad. However, looking at our results, a badly colored image generally seems more distinguishable than its black and white version.

To conclude, the colorizations of two images are reported as an example in Figure 1.

5.2. PSNR, SSIM and LPIPS metrics

We computed the following metrics:

- Peak Signal-to-Noise Ratio (PSNR) [5] computes the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. This is insufficent for assessing structured outputs such as images, as it assumes pixel-wise independence.

- Structural Similarity (SSIM) [5] is based on human-like perceived changes in structural information, based on the fact that the pixels have strong inter-dependencies that carry important information about the structure of the objects in the visual scene. However, human judgments of similarity are context-dependent and may not actually constitute a distance metric [16].

- Learned Perceptual Image Patch Similarity (LPIPS) [7] evaluates the distance between image patches, so that small values indicates that the colorized images are similar to the corresponding original images. This innovative approach exploits the internal activations of networks trained for high-level classification tasks (as AlexNet), corresponding to human perceptual judgments. Figure 2 shows the computations needed.

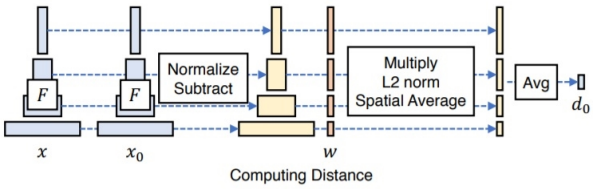


Figure 2: To evaluate the distance d_0 between two images, x and x_0, given a network $\mathcal{F}$, LPIPS first computes deep embeddings, then normalizes the activations in the channel dimension, scales each channel by a vector w, and takes the ℓ_2 distance. Then it performs the average across spatial dimension and across all layers.

In order to get general results, the metrics were computed on the ri-colorized images from a heterogeneous dataset containing all the data available. Since PSNR and SSIM require a heavy computation, we decided to evaluate the baseline only with LPIPS, which is indeed the most significant metric.

As we can see from Table 2, the best models according to LPIPS are Siggraph17 and ChromaGAN, which are indeed the highest-scored model in the Turing Test with human partecipants (see next section). This evaluation is also confirmed by PSNR. On the other hand, SSIM indicates InstColorization as best-performing model, followed by ChromaGAN and Siggraph17.

Dahl's model is the worst-performing for all the metrics, due to its desaturated results. Notably, the baseline is able to outperform it and in general there's a grat gap between its scores and the other models' scores. However, this is most probably due to the reshaping of its colorized images ($224 \times 224 \times 3$) to match the original images ($256 \times 256 \times 3$) causing a little corruption of the results.

5.3. Turing Test

The Turing test is a qualitative metric based on human perceptions. Due to our limited resources we just elaborate a short survey on Google Forms, divided into two sections: we asked the participants to first evaluate the colorizations of three black and white photographs and then evaluate the ri-colorizations of four originally colored images.

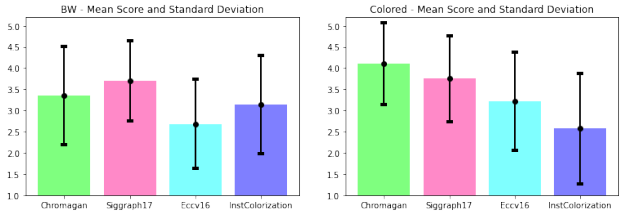
A preliminary test has been conducted on few parte-

Figure 3: ChromaGAN colorization (second row) on some filtered images (first row).

cipants, who were always be able to dicriminate between the original version of the images and the ri-colorizations. Therefore, we decided to not show the original colored images in the final Turing test in order to avoid biases on the different and equally plausible choices of colors.

The 124 participants had to score how realistic was each colorization in a scale from 1 (not realistic at all) to 5 (very realistic). The test includes only the best colorizers, namely Eccv16, Siggraph17, ChromaGAN and InstColorization.

Figure 4 summarizes the mean scores for each model. Clearly, the models performed differently on originally colored images and greyscale photographs, since these ones have a low-level image statistics which are quite different from those of the modern-day photos on which the models were trained [15]. Overall, the best average scores are reached by Siggraph17 and ChromaGAN.



B&W Colored

Figure 4: Mean scores obtained with the colorization on black and white photographs and originally colored images.

5.4. Image filtering

Our last experiment consists in applying some filters to a selection of 18 ImageNet images to evaluate possible changes in the image colorization output. We considered a blurring filter with kernel sizes of 3, 7 or 11, a cartoonizing filter from [14] and the increasing or decreasing of contrast and luminance.

In general, a blurred image is harder to colorize, and the more blurred the image is, the worse the final colorization we get. Indeed, in the example reported Figure 3 the dogs have some weird pink/orange spots on their fur. On the contrary, images with a higher contrast and luminance get a better colorization by all the models, with just a few exceptions depending on the specific image.

Clearly, with cartoonization we reach very unrealistic colors. In the example, it looks like the model didn't recognize the grass and probably mistook it for water. Probably, this is not due to the fact that cartoonization discards some details from the original images, because blurring does the same. This could be rather due to the fact that the models are not trained on cartoonized images and expect a completely different "image style" in input.

In conclusion, Figure 3 is a good example of how the colorization is an ambiguous task: the model colors the grass with a plausible green tone, which actually could result more realistic than the original brown tone.

6. Conclusion

As expected, the model reaching the overall best performances is the state-of-the-art ChromaGAN, followed by Siggraph17. Eccv16 produces vibrant colorizations with the use of class-rebalancing, but at the expense of some over-aggressively and visibly artificial colorizations. In general, InstColorization is able to improve the results of Eccv16, except for those classes showing bright and rare colors (low accuracy when tested with AlexNet finetuned on birds and flowers images). Moreover, its segmentated colorization can lead to incoherent results when object detection doesn't work properly, and this came out in the Turing test scores.

We have also seen that some image filtering concerning luminance and contrast can improve the final colorization.

In conclusion, our baseline performs very poorly due to the limited amount of data for the training, the simple architecture and the absence of a colorization-adapted loss function. However, the introduction of cartoonization in the training and testing phases seems to slightly improve the final results, meaning that more sectorial and definite areas are easier to colorize. This opens up new perspectives for future works based on the combination of state-of-the-art models with cartoonization.

Source Code

Our code and data are available at the Github repository https://github.com/ChiaraBi/vision-project.

References

[1] 102 category flowes dataset. https://www.robots.ox.ac.uk/~vgg/data/flowers/102/, 2008.

[2] Pascal voc dataset. https://deepai.org/dataset/pascal-voc, 2012.

[3] Places205 dataset. https://paperswithcode.com/dataset/places205, 2014.

[4] 325 bird species dataset. https://www.kaggle.com/gpiosenka/100-bird-species, 2019.

[5] Peak signal-to-noise ratio and structural similarity metrics. https://cvnote.ddlee.cc/2019/09/12/psnr-ssim-python, 2019.

[6] Imagenette and imagewoof datasets. https://github.com/fastai/imagenette, 2021.

[7] Learned perceptual image patch similarity metric. https://github.com/richzhang/PerceptualSimilarity, 2021.

[8] Ryan Dahl. Automatic colorization. https://tinyclouds.org/colorize/, 2016.

[9] Jianbo Chen et al. Language-based image editing with recurrent attentive models, 2018.

[10] Richard Zhang et al. Real-time user-guided image colorization with learned deep priors, 2017.

[11] Seungjoo Yoo et al. Coloring with limited data: Few-shot colorization via memory-augmented networks, 2019.

[12] Jheng-Wei Su, Hung-Kuo Chu, and Jia-Bin Huang. Instance-aware image colorizationl, 2020.

[13] Patricia Vitoria, Lara Raad, and Coloma Ballester. Chromagan: Adversarial picture colorization with semantic class distribution, 2020.

[14] Xinrui Wang and Jinze Yu. Learning to cartoonize using white-box cartoon representations, 2020.

[15] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization, 2016.

[16] Richard Zhang, Phillip Isola, and Alexei A. Efros et al. The unreasonable effectiveness of deep features as a perceptual metric, 2018.

Appendix

This section contain the graphical representation of the proposed models' architectures.

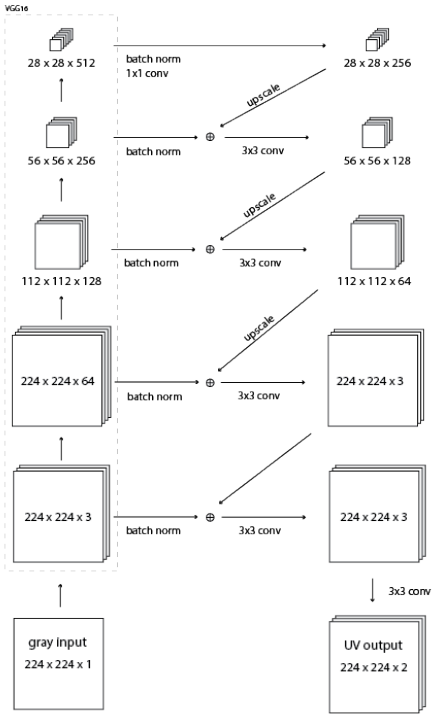
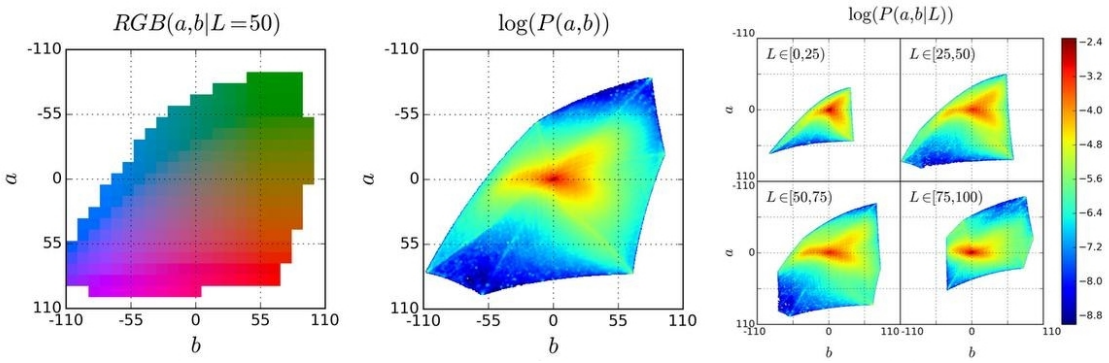
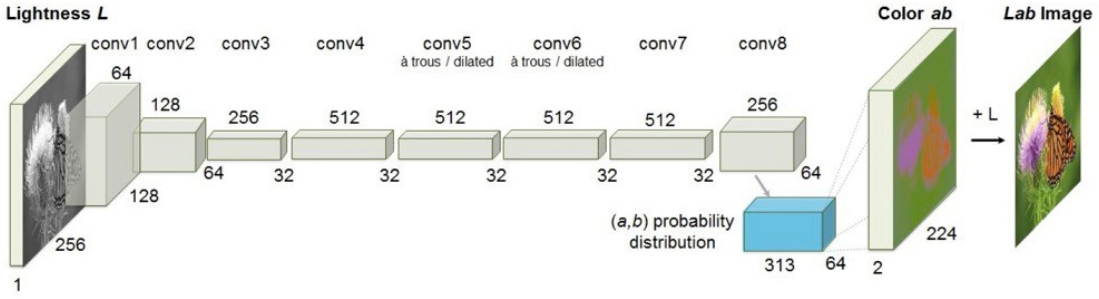


Figure 5: Dahl architecture.



(a) Quantization of the *ab* output space.



(b) Zhang architecture.

Figure 6: Zhang approach and architecture.

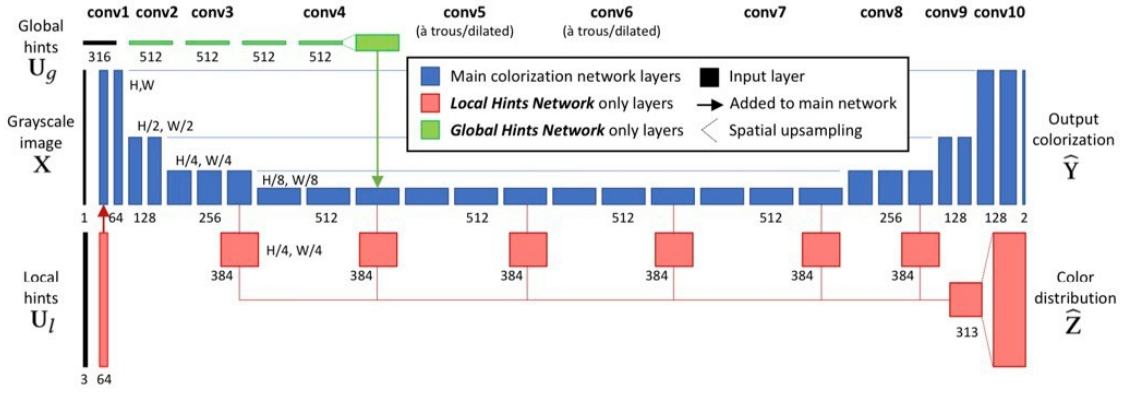


Figure 7: Siggraph architecture.

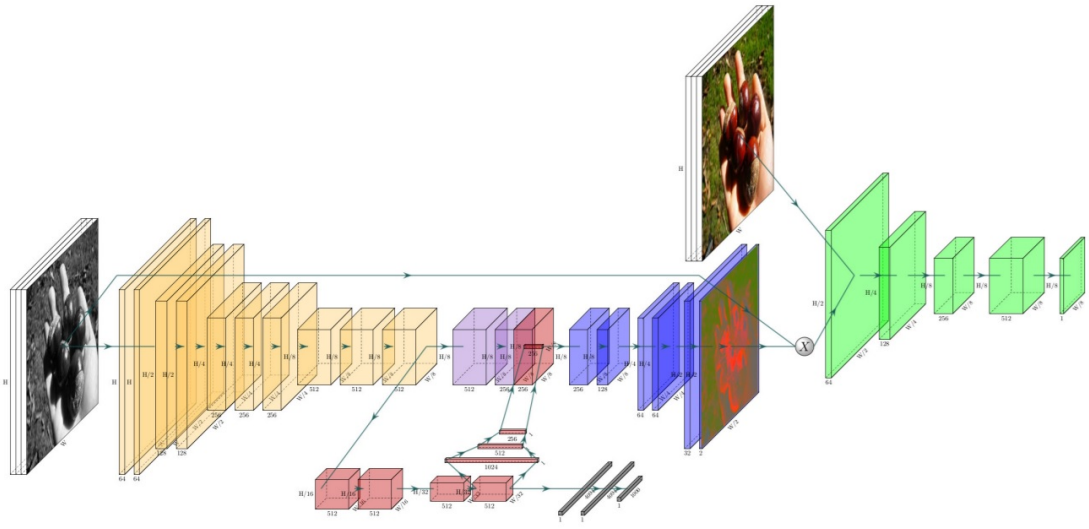
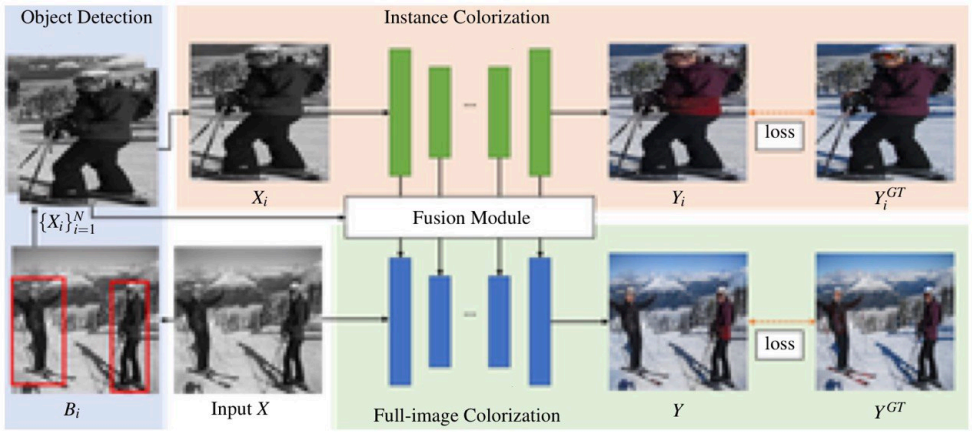
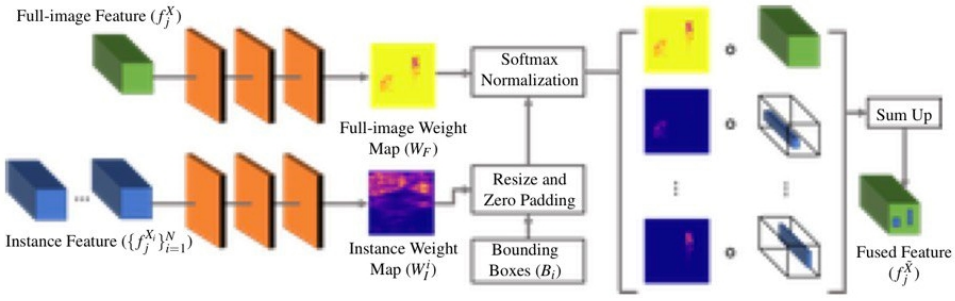


Figure 8: ChromaGAN architecture. The discriminator (green) is combined with the generator, which consists of two sub-networks: one for chrominance (yellow, purple, red, blue) and the other for class distribution (yellow, red, gray).



(a) Complete process overview.



(b) Fusion module at the j-th layer.

Figure 9: InstColorization approach and architecture.