Automatic image colorization: a comparative overview

Bigarella Chiara
Student nr. 2004248

Poletti Silvia
Student nr. 1239133

Abstract

The ABSTRACT is to be in fully-justified italicized text, at the top of the left-hand column, below the author and affiliation information. Use the word "Abstract" as the title, in 12-point Times, boldface type, centered relative to the column, initially capitalized. The abstract is to be in 10-point, single-spaced type. Leave two blank lines after the Abstract, then begin the main text. Abstract should be no longer than 300 words.

1. Introduction

Introduction (10%): describe the problem you are working on, why it's important, and an overview of your results.

Colorization is a highly undetermined problem without a unique solution. Indeed, it can be mathematically formalized as a real-valued map from 2D-greyscale images to 3D-colored ones.

2. Related Work

Related Work (10%): discuss published work or similar apps that relates to your project. How is your approach similar or different from others?

Papers are: [13], [12], [15],[10], [8],[11],[14],[9].

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 Image filtering section) obtained starting from 18 originally colored images.

Our data includes heterogeneous images, representing many 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] reguarding 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 Flowers [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 ($256 \times 256 \times 3$ for Baseline, Zhang, Siggraph and InstColorization and $224 \times 224 \times 3$ for Dahl and ChromaGAN). Dahl also required center cropping and desaturation. Despite the preliminar reshape, Zhang, Siggraph and ChromaGAN provides for results having the original image shape.

Given an RGB image we obtain the corrisponding image in the *Lab* color space, in which colors are expressed through 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. Then, the resulting images are projected again in the RGB color space.

In particular, the classification with AlexNet required the normalization of the images' RGB channels in the range $[0, 1]$ and a further standardization of the images according to the mean and standard deviation of the training set. On the other hand, the LPIPS metric required the normalization of the images' RGB channels in the range $[-1, 1]$ and the dataset reshaping from $N \times H \times W \times 3$ to $N \times 3 \times H \times W$, where N is the number of images.

4. Methods

In order to carry out a comparative overview about automatic image colorization, we propose a simple autoencoder based on cartoonization as baseline. Then, we tested 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 consisted in a 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 cartoonization model by [14]. This cartoonization provides fine-grained results (we don't miss much information) and synthesizes the original images in order to exclude noisy elements that could interfer 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

4.3. Zhang

The innovation introduced by the Zhang 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 5b) 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. Indeed, 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 throug the quantization of the *ab* output space, as shown in Figure 5a. 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. Siggraph

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 provides 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 7 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 7 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 7 in yellow) are shared and initialized with the pre-trained VGG-16 weights. Then, both the subnetworks split into two tracks (Figure 7 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 grayscale 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, we have two colorization networks: the first colorizes the whole image and the second the patches (resized to 256×265) in the bounding boxes. These networks have different weights but share the same architecture: the chosen architecture is the same as Zhang, as well as the loss function. Once the first networks is trained, its learned weights are used to inizialize the second network. 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 8a.

In particular, the fusion module (Figure 8b) 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 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, meaning that our approach is valid and 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 various 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 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 Zhang and Siggraph 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 the 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. LPIPS, PSNR and SSIM Metrics

[5] and [7]

5.3. Turing Test

124 subjects, 3 B&W + 4 colored.

i risultati sono diversi perchè le foto B%W sono scattate con una diversa tecnologia rispetto alle foto a colori

Figure 2

5.4. Image filtering

possiamo prendere ad esempio ChromaGAN (uno dei migliori modelli): l'erba la fa verde perchè è più real-

	Original	B&W	Baseline w/c	Baseline	Dahl	Zhang	Siggraph	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.

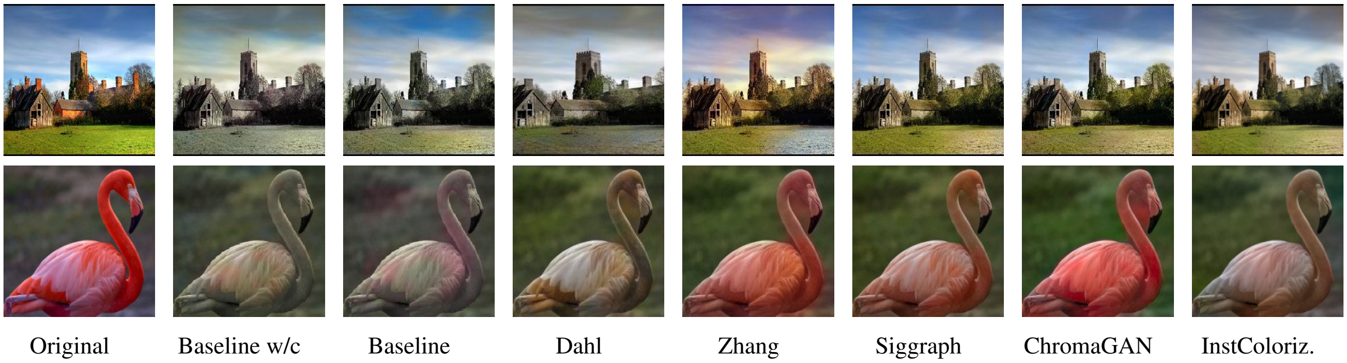
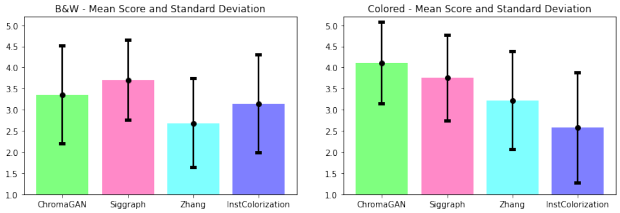


Figure 1: Colorization comparison on two images from ImageNet Church (first row) and 325 Birds Species Flamingo (second row).



B&W

Colored

Figure 2: Mean scores obtained with the colorization on black and white photographs and originally colored images.

istica (si potrebbe pensare che ha colori anche migliori dell'originale), cartoon fa schifo perchè il modello non ha fatto un training su tali immagini, altrimenti potremmo supporre che performerebbe meglio. Infatti non riconosce l'erba come tale e sembra la colori di blu, scambiandola per acqua. Figure 3

6. Conclusion

Conclusion (5%): summarize your key results; what have you learned? Suggest ideas for future extensions.

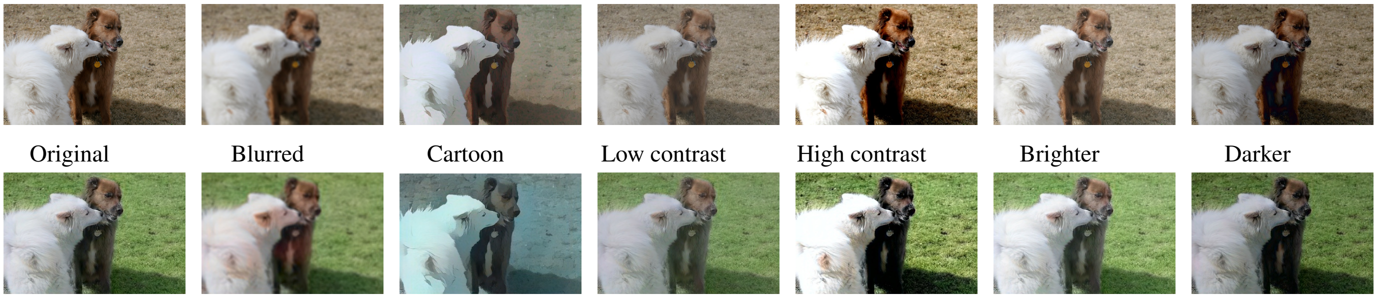


Figure 3: ChromaGAN colorization (second row) on some filtered images (first row).

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.

Appendix

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

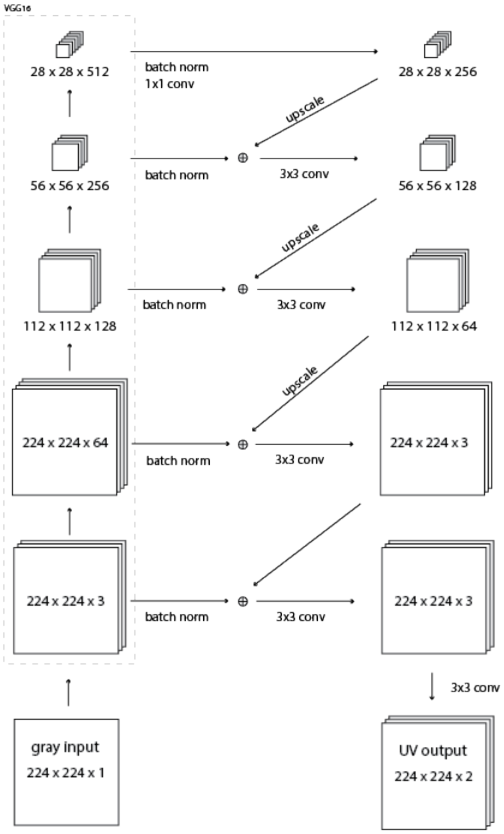
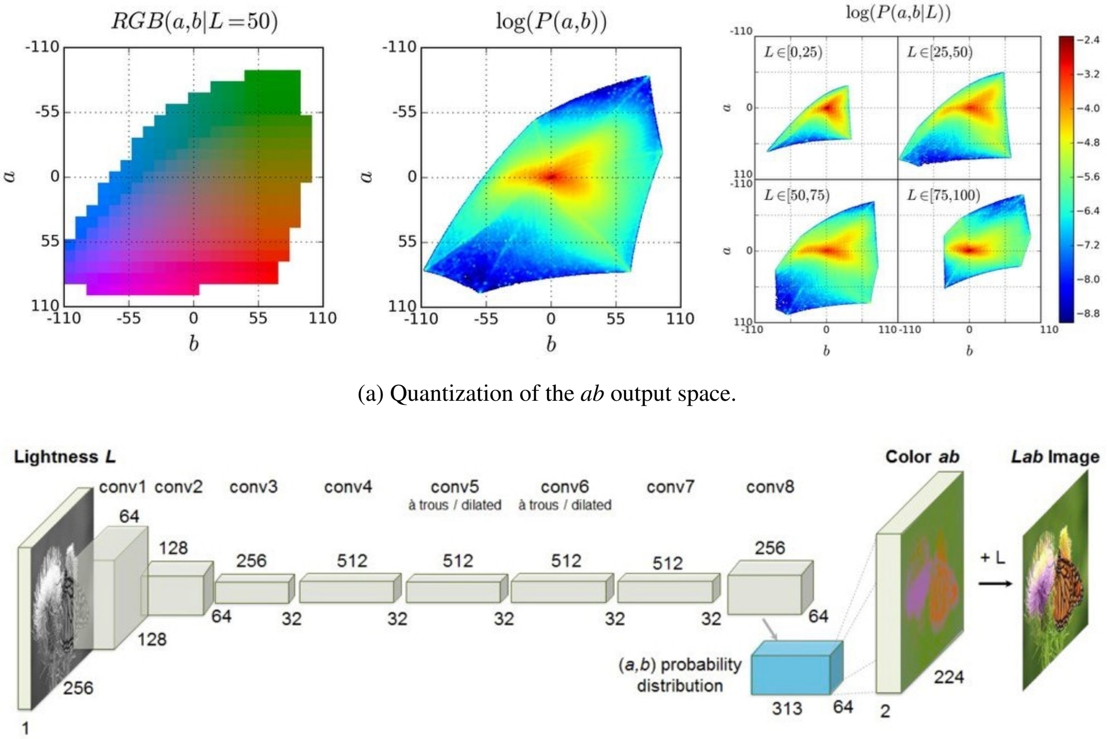


Figure 4: Dahl architecture.



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

(b) Zhang architecture.

Figure 5: Zhang approach and architecture.

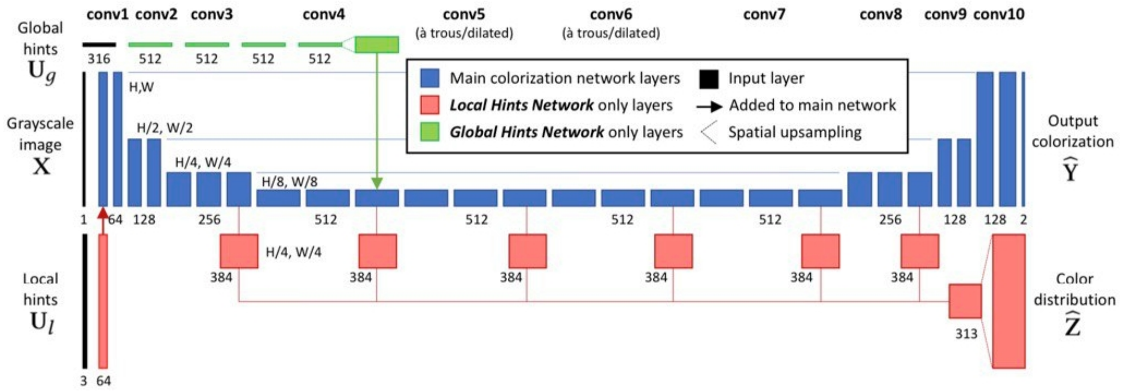


Figure 6: Siggraph architecture.

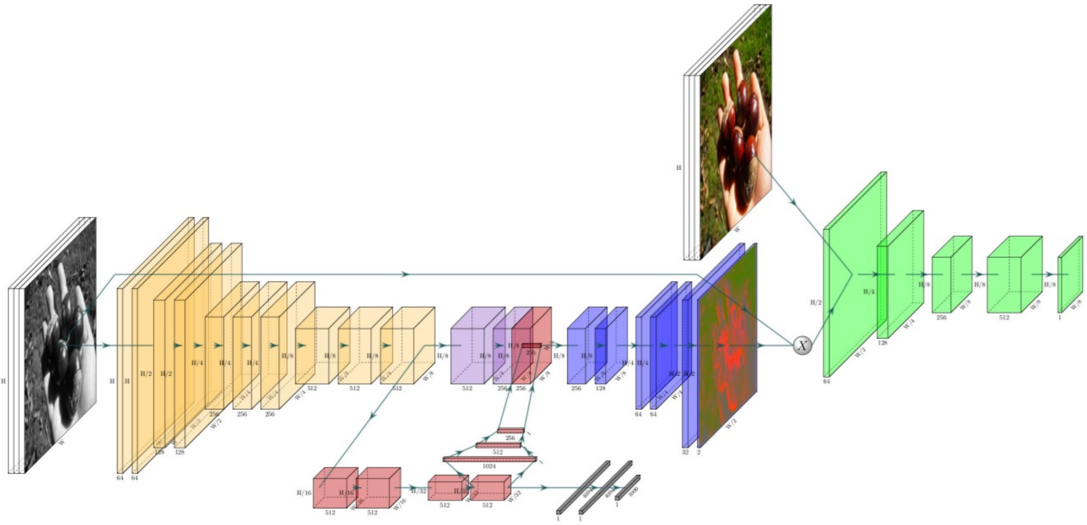
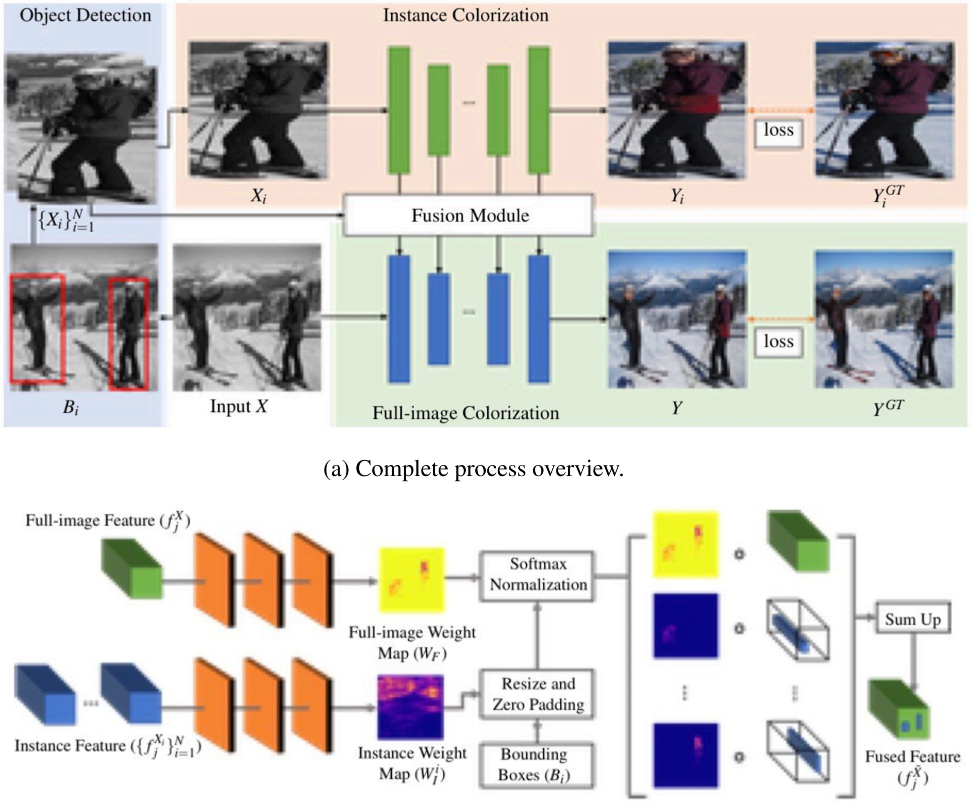


Figure 7: ChromaGAN architecture. The discriminator (green) is combined with the generator, which consists of two sub-networks: one for chrominance (yellow, purple, red, blue) and the other for class distribution (yellow, red, gray).



(a) Complete process overview.

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

Figure 8: InstColorization approach and architecture.