2021/11/21 v0.6 IPOL article class

Published in Image Processing On Line on 2026–11–00.
Submitted on 2025–07–23, accepted on 2026–10–19.
ISSN 2105–1232 © 2026 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2026.632

A Survey of Metrics Used for Image Colorization Assessment

Nicolas Maignan, Fabien Pierre, Frédéric Sur

Université de Lorraine, CNRS, Inria, LORIA, Nancy, France
{nicolas.maignan,fabien.pierre,frederic.sur}@univ-lorraine.fr

Communicated by Lara Raad *Demo edited by* Lara Raad

Abstract

The objective evaluation of image and video colorization remains challenging. Metrics such as PSNR and SSIM have limited ability to capture perceptual quality and typical colorization artifacts. Furthermore, they require a reference color image, which is generally unavailable for the colorization task. Alternatively, referenceless evaluation metrics have been proposed, aiming to better reflect human perception, yet few studies have analyzed their effectiveness. In this paper, we provide a comprehensive review of the evaluation metrics employed by recent colorization methods. We introduce a framework to assess these metrics based on their sensitivity to synthetic transformations that simulate common colorization defects, such as color bleeding or non-semantic colors. This evaluation is conducted both in the static case of images and in the temporal dimension for video colorization. Besides, we propose extending to color images two computationally lightweight perceptual metrics previously defined in the literature for grayscale images.

Source Code

The reviewed Python source code and documentation for this algorithm are available from the web page of this article[1] and from the GitLab repository of Inria[2]. Compilation and usage instructions are included in the `README.md` file of the archive.

Keywords: quality assessment; perceptual metrics; colorization

[1] https://doi.org/10.5201/ipol.2026.632

[2] https://gitlab.inria.fr/tangram/anr-arce/published-codes/colorization-metrics/-/tree/IPOL?ref_type=heads

1 Introduction

Image and video colorization refer to the process of adding color to grayscale images, often sourced from historical photographs or movies captured on black-and-white photographic film. Mathematically speaking, colorization is an ill-posed inverse problem, as several realistic color frames may correspond to the same grayscale image. For example, the same grayscale image of an apple could correspond to an apple of several different colors. Therefore, any colorization is an interpretation guided by expert knowledge, or at least perceptual plausibility. While image colorization consists of adding realistic colors to a single image independently of any other, video colorization introduces additional challenges: each frame must be convincingly colorized in isolation, while also maintaining chromatic consistency with other frames depicting the same scene or content.

Manual video colorization dates back to the early 20th century. For example, Berthe and Élisabeth Thuillier led a workshop of 200 women to colorize films by Georges Méliès and Pathé during the period from 1890 to 1910 [38]. Initially, this process involved painting directly onto individual film frames. Over time, more sophisticated techniques emerged, such as stencil coloring—commonly known as Pathécolor [45]. This method relied on cutting film copies into stencils corresponding to different colors, enabling precise paint application and resulting in sharp-edged colorization.

Simultaneously, efforts were made to capture natural colors directly on film. Kinemacolor, for instance, utilized two filmstrips with red and green filters to record limited color information. However, such systems required specialized projectors for playback. By the 1930s, advancements like the three-color Technicolor process rendered these methods obsolete, as they allowed natural color capture and playback without the need for specialized equipment.

Despite the widespread availability of color footage, colorization of legacy films remains an active field of interest. Artists and production companies often enhance black-and-white footage to improve audience appeal or to add realism to archival material. Additionally, research in colorization has contributed to advancements in image compression and transmission technologies.

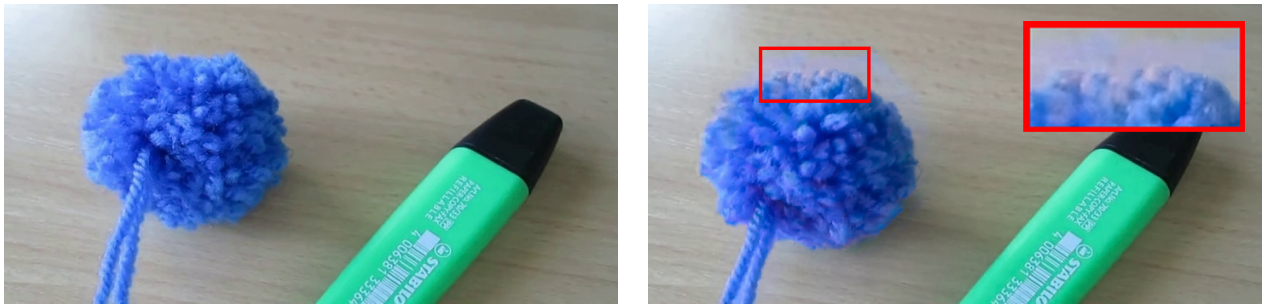
When working with grayscale frames from legacy movies, a common assumption is made: the known grayscale values are treated as the luminance component of the colorized image. While grayscale values can be approximated as a weighted average of the red, green, and blue channels in the RGB color space, most modern algorithms operate in the CIELAB color space [36]. Various color spaces have been developed to address specific challenges. For instance, RGB and its variants use additive synthesis for screen displays, while HSV and HSL [42] provide intuitive axes for modifying color properties. CIELAB and CIELUV [36], on the other hand, encode colors in a non-linear yet perceptually uniform manner aligned with human vision.

In the CIELAB space, the grayscale frame is directly mapped to the luminance channel L*, while the chrominance channels a* and b* encode color information along the green-red and blue-yellow axes, respectively. The CIELAB color space is particularly well-suited for colorization tasks because its design ensures perceptual uniformity, meaning that changes in a* and b* values correspond to consistent perceptual changes in color. This separation of luminance and chromaticity simplifies the problem, allowing algorithms to focus on accurately predicting the chrominance components while preserving the known luminance from the grayscale input.

Over the past two decades, numerical algorithms have revolutionized image and video colorization. One seminal work is the method introduced by Levin et al. [24], which established principles still used in modern colorization. These algorithms typically propagate color information across spatial and temporal domains based on prior input, such as user-defined scribbles. Nonetheless, producing such an input is tedious for users.

To remove the need for numerous scribbles, another family of methods proposes replacing scribbles with a single previously colorized reference frame, whose colors are then propagated to an image with similar content when colorizing images or to the remaining frames when colorizing a video. Sub-

sequent video methods, such as Deep Video Color Propagation (DVCP) [31], Deep Exemplar-Based Video Colorization (DEB) [59], and Learned Variational Video Color Propagation (LVVCP) [11], use an already-colored first frame or a reference image with similar semantics to sequentially colorize frames. However, these techniques often produce artifacts that get worse with temporal distance from the reference. Common issues include objects losing their colors after being occluded or absent in the reference, colors overflowing between moving objects, also called color bleeding, and progressive desaturation of colors. An example of color bleeding is shown in Figure 1, where a pompom rolls on a wooden table. As the pompom moves downward, its blue color persists on the pixels it previously occupied, leaving a blue trace on the table. This artifact arises from errors in optical flow estimation at the pompom's boundary.



$t = 5$ $t = 28$

Figure 1: Colorization using LVVCP [11] of a sequence featuring a rolling pompom. In frame $t = 28$, we observe *color bleeding* (boxed) caused by the movement of the pompom. Since the optical flow estimate is most prone to errors at object boundaries, the blue color remains at the same pixel locations as the pompom moved away, creating an expanding blue patch that persists on the table.

Because achieving full automation (i.e. without any input from the user) while preserving temporal consistency proved considerably harder for video than for still images, early efforts toward automatic colorization concentrated primarily on the image case, which produced substantial research on fully automatic image colorization before video-specific automatic methods matured. Following this trend, convolutional networks have been trained in various image colorization methods such as Colorful Image Colorization (CIC) [60] and DeOldify [39] to generate colors directly on images using learned natural color features. However, these networks often produce dull colors and may assign incorrect hues to objects. Moreover, they usually fail to ensure temporal coherence when processing video frames independently. For a more comprehensive review of colorization algorithms, the interested reader is referred to [4].

As video colorization inherits most of its per-frame visual artifacts from the image case, and given the comparative scarcity of metrics designed specifically for temporal defects, evaluation in video colorization studies has relied heavily on frame-level, image-based metrics. The introduction of these metrics in colorization studies coincided with the rise of deep learning approaches. Indeed, the colorization errors arising with the handcrafted methods were explainable, whereas the errors produced by deep learning approaches are not. A common way to evaluate deep learning approaches is through objective metrics computed on large datasets. Previously, articles relied primarily on qualitative evaluations, presenting image results without quantitative measures [24, 56]. However, these metrics inherently measure different aspects of visual artifacts. As a result, the use of diverse combinations of metrics across studies complicates direct comparisons between articles. A list of evaluation metrics is provided in Section 6 of [4].

To address these challenges, we propose a unified measurement tool that enables reproducible evaluation of colorization methods with configurable parameters. This tool accommodates various types of metrics from the literature, including comparisons with ground truth, semantic classification,

and temporal analysis.

The main contributions of this work are as follows:

- Adaptation of existing grayscale metrics, BRISQUE [32] and NIQE [33], to color images.
- Analysis of multiple metrics used in image and video colorization studies, either reference-based metrics when ground-truth data is available, or referenceless metrics based on human perception.
- Development of a comprehensive and modular software tool for colorization evaluation with reproducible parameterization.
- Assessment of the metrics on a large open image dataset.

In the following, we present and analyze the metrics in Section 2. Artificial defects simulating colorization artifacts are detailed in Section 3.1. The relevance of the metrics is then experimentally assessed on a large image dataset in Section 3.2, followed by an evaluation on video data in Section 3.3.

2 Metrics Studied

Although industrial applications concern both video and image colorization, we focus on the presentation of metrics defined for images in Sections 2.1 and 2.2. Since video colorization presents specific temporal defects in addition to all issues arising in image colorization, Section 2.3 will therefore extend the image metrics to video and focus on the assessment of temporal coherence.

In this section, we use the following notations: I represents the colorized output images, $\hat{I}$ denotes the reference color images (ground truth, when available), H and W refer to the height and width of the images, respectively, and T is the total number of frames in a video sequence. To denote a quantity at time t, we add a superscript. For example, the t-th colorized frame is denoted by I^t. Finally, when the metric is not explicitly designed for video processing, the score for a video is defined as the mean across all frames in the video sequence.

Colorization metrics can be divided into three categories: reference-based image metrics, referenceless image metrics and video metrics.

Concerning the code organization, all metrics are defined within a Python module named `colorization_metrics`, located in `src/colorization_metrics`. This module includes the following key components:

- `evaluate.py`: A module that can be directly called by other Python software to evaluate colorization results.
- `main_cli.py`: A script to call metrics using a command line interface.
- `utils.py`: Utility script providing shared functionality.
- `metrics/`: A directory containing a separate Python file for each metric.

For clarity, all subsequent file paths in this section will be given relative to the `src/colorization_metrics` directory.

2.1 Reference-based Metrics

In this section, the metrics are used to compare colorized images with their ground truth counterparts to quantify potential colorization defects. They are mostly the so-called Image Quality Assessment (IQA) metrics [50] that have been initially developed to detect noise and artifacts introduced by compression or signal transmission. We present these metrics in the context of colorization.

PSNR. The Peak Signal to Noise Ratio (PSNR) [13] is a reference-based metric that compares the pixels of colorized images with those of reference images. For this metric, a higher score indicates a less distorted image.

The PSNR of a colorized frame is expressed as

$$\mathrm{PSNR}(I^t, \hat{I}^t) = 10 \log_{10} \left(\frac{r^2}{\mathrm{MSE}(I^t, \hat{I}^t)} \right), \tag{1}$$

with $r = 255$ the range of the intensity values in an 8-bit encoding, where the Mean Squared Error (MSE) is defined as the average squared difference between the colorized image and the ground-truth

$$\mathrm{MSE}_{RGB}(I^t, \hat{I}^t) = \frac{1}{3HW} \sum_{c \in \{R,G,B\}} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(I_c^t(i,j) - \hat{I}_c^t(i,j) \right)^2. \tag{2}$$

When assessing video sequences, the PSNR is calculated by averaging Equation (1) over the frames. Each pixel is compared independently of its spatial and temporal neighbors with its counterpart pixel in the reference image. Moreover, this comparison is performed independently for each color channel. The color space commonly used is RGB [23, 15, 49, 55, 3], but recent papers present evaluation on CIELAB [31, 11] space, which involves averaging MSE over chrominance channels a* and b* only, defined in the following equation

$$\mathrm{MSE}_{a*b*}(I^t, \hat{I}^t) = \frac{1}{2HW} \sum_{c \in \{a*,b*\}} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(I_c^t(i,j) - \hat{I}_c^t(i,j) \right)^2. \tag{3}$$

The computation of Equation (1) is implemented in `metrics/psnr.py` with the `_compute_psnr_rgb` and `_compute_psnr_lab` functions for, respectively, the computation on RGB (Equation (2)) and CIELAB (Equation (3)) images. For a video, the computation is performed with the `compute_psnr_dir` function. The implementation relies on the scikit-image [47] PyPI module to compute the PSNR on each channel.

As a result of averaging, PSNR will diminish the influence of pixels that differ strongly from others, making it impossible to correctly measure localized colorization defects within an object. Also, the relevance of the evaluation depends on the chosen color space. For colorization algorithms where luminance remains unchanged, using metrics like PSNR in RGB can mask colorization errors detection. This is because the three RGB channels encode luminance, which is identical between the prediction and the ground truth, thereby minimizing the impact of chromatic inaccuracies. Evaluating PSNR in chrominance-only channels, such as a* and b* in the CIELAB color space, avoids this issue, as these channels are independent of luminance. Furthermore, being based on a reference, PSNR does not take into account the existence of multiple semantically valid colorizations. This tends to give a better score to a dull colorization than one that is semantically correct but divergent from the reference.

In the end, PSNR only provides an implicit assessment of temporal coherence in videos. Since the metric is computed by averaging the scores across all frames, it does not directly enforce temporal consistency. Moreover, as with spatial artifacts, PSNR is insensitive to isolated temporal defects. For example, if a single frame in a long sequence is poorly colorized, the overall score will be only marginally affected, even though the visual flaw will be quite noticeable, especially given a standard frame rate of 24 frames per second.

PSNR is widely used in image processing, easy to calculate and often used in the literature. However, it is not sensitive to the defects of image or video colorization, so it cannot be used to compare colorizations reliably. Its main use is as an initial rough measure when colorization must correspond to a well-defined ground truth.

SSIM. The Structural Similarity (SSIM) index [50] is a reference-based quantitative measure that studies the structure of images. Unlike PSNR, which is based on MSE and therefore a pixel-by-pixel difference, SSIM analyzes an 11×11 neighborhood around each pixel. In this metric, a greater score indicates a less noisy image.

Thus, rather than comparing the values of corresponding pixels directly, the means and standard deviations of their respective neighborhoods are compared. The measurement is based, for each channel of the image under consideration, on three a priori independent quantities: luminance l (different from CIELAB space L* luminance), contrast c and structure s. These quantities are calculated at each pixel (i, j) as follows

$$\mathrm{l}(I^t, \hat{I}^t, i, j) = \frac{2\,\mu_{I^t}(i,j)\,\mu_{\hat{I}^t}(i,j) + C_1}{\mu^2_{I^t}(i,j) + \mu^2_{\hat{I}^t}(i,j) + C_1}, \tag{4}$$

$$\mathrm{c}(I^t, \hat{I}^t, i, j) = \frac{2\,\sigma_{I^t}(i,j)\,\sigma_{\hat{I}^t}(i,j) + C_2}{\sigma^2_{I^t}(i,j) + \sigma^2_{\hat{I}^t}(i,j) + C_2}, \tag{5}$$

$$\mathrm{s}(I^t, \hat{I}^t, i, j) = \frac{\sigma_{I^t\hat{I}^t}(i,j) + C_3}{\sigma_{I^t}(i,j)\,\sigma_{\hat{I}^t}(i,j) + C_3}, \tag{6}$$

with, for two images x and y, $C_1 = 2.55^2$, $C_2 = 7.65^2$ and $C_3 = C_2/2$ and a neighborhood of 11×11 pixels, for a total of $N = 121$ pixels

$$\mu_x(i,j) = \frac{1}{N} \sum_{l=-5}^{5} \sum_{m=-5}^{5} w(l,m)\; x(i+l, j+m), \tag{7}$$

$$\sigma_x(i,j) = \left(\frac{1}{N-1} \sum_{l=-5}^{5} \sum_{m=-5}^{5} w(l,m)\;\; \big(x(i+l,j+m) - \mu_x(i,j)\big)^2 \right)^{\frac{1}{2}}, \tag{8}$$

$$\sigma_{xy}(i,j) = \frac{1}{N-1} \sum_{l=-5}^{5} \sum_{m=-5}^{5} w(l,m)\; \big(x(i+l,j+m) - \mu_x(i,j)\big)\; \big(y(i+l,j+m) - \mu_y(i,j)\big). \tag{9}$$

These various quantities are weighted by w, a symmetrical, circular Gaussian kernel with a standard deviation of 1.5. The final measurement on the whole image is computed as the mean of the product of l, c and s

$$\mathrm{SSIM}(I^t, \hat{I}^t) = \frac{1}{3HW} \sum_{c \in \{R,G,B\}} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \mathrm{l}\left(I^t_c, \hat{I}^t_c, i, j\right)\; \mathrm{c}\left(I^t_c, \hat{I}^t_c, i, j\right)\; \mathrm{s}\left(I^t_c, \hat{I}^t_c, i, j\right). \tag{10}$$

The computation of Equation (10) is implemented in `metrics/ssim.py` with the `compute_ssim` function. The computation on a video is done with the `compute_ssim_dir` function. It computes the mean of Equation (10) over the frames. As for the PSNR, it uses the scikit-image [47] module to compute the SSIM.

A first improvement over PSNR is that the neighborhood of each pixel is taken into account, leading to a better understanding of local aspects rather than isolated pixel-level variations. Indeed, the standard deviations allow, in particular, the magnitude s (Equation (6)) to penalize more severely textures added or removed by colorization.

However, averaging over the image channels prevents SSIM from taking into account correlations between channels. Also, luminance comparison is still present, as the measurement is done in RGB space by default, even if the colorization shares the same luminance as the ground truth. Finally, the measurement does not take into account the temporal consistency, so it does not reflect the quality of video colorization.

In the end, SSIM offers a small improvement over PSNR and should be used in the same cases: a preliminary assessment that relies more on the quality of reference images than on the correct colorization of frames.

LPIPS. While PSNR just compares image gray levels pixelwise and SSIM compares basic local perceptual features, Learned Perceptual Image Patch Similarity (LPIPS) [61] uses features from deep neural networks supposed to extract valuable perceptual information. Here, a lower score for this metric indicates a less distorted image.

Basically, this metric uses features from AlexNet [20], though VGG [41] and SqueezeNet [14] can also be employed. These features, extracted from both the colorized and reference images, are normalized and compared. Let $f^l \in \mathbb{R}^{H_l \times W_l \times C_l}$ represent the extracted features — of spatial resolution $H_l \times W_l$ and C_l channels — at layer l of the deep network from the colorized image I^t, and $\hat{f}^l \in \mathbb{R}^{H_l \times W_l \times C_l}$ those from the reference image $\hat{I}^t \in \mathbb{R}^{H_l \times W_l \times C_l}$, both unit-normalized along the channel dimension. The LPIPS score is then computed as follows

$$\mathrm{LPIPS}(I^t, \hat{I}^t) = \sum_l \frac{1}{H_l W_l} \sum_{i=1}^{H_l} \sum_{j=1}^{W_l} \left\| w_l \odot \left(f^l(i,j) - \hat{f}^l(i,j) \right) \right\|_2^2, \tag{11}$$

where $\odot$ is the Hadamard product and $w_l \in \mathbb{R}^{C_l}$ are learned weight vectors applied at different layers of the network.

The version of the metric used here is the so-called linear variant as described in Zhang et al. [61] in which pre-trained networks are used to extract features from images. The authors have trained the weights w_l on their own dataset of human judgments, the Berkeley-Adobe Perceptual Patch Similarity (BAPPS) [61] dataset. In the training part of this dataset, a 64×64 patch of an image is distorted by traditional methods (blur, noise, shift, ...) and by CNN denoising auto-encoders to give two patches. Two participants then indicate which noisy patch is perceptually closer to the original patch, and their choices are used to compute the two-alternative forced-choice (2AFC) score. The LPIPS metric weights w_l have been trained on those 2AFC scores so that the final metric mimics human judgment.

The computation of Equation (11) uses the official implementation by Zhang et al. [61] available in the `lpips` PyPI module. The `compute_lpips` function within the `metrics/lpips.py` file can be used for computation. To compute the score on a video, the `compute_lpips_dir` function should be used instead.

Similarly to SSIM, LPIPS compares images at a local level by evaluating patches rather than individual pixels. However, LPIPS extracts features from convolutional layers, which allows it to correlate information across multiple channels. This enables the metric to assess the relationships between color channels within image patches, unlike SSIM, which evaluates color channels independently.

Let us note that [61] presents an assessment of colorization on the ImageNet dataset [37]. Table 5 of their ArXiv v2 preprint [62] reports that LPIPS based on a pretrained AlexNet achieves 2AFC scores 1.8 and 7.2 percentage points higher than those obtained with the L_2 norm (closely related to PSNR) and SSIM, respectively. LPIPS achieves a 2AFC score of 65.3% on the colorization task. These results suggest that LPIPS agrees with human perceptual judgments of colorization quality.

FID and FVD. The Fréchet Inception Distance (FID) [10] is a reference-based metric that compares neural network features between two groups of images. With this metric, lower scores represent images with fewer artifacts.

Developed to measure results of generative networks trained with a GAN procedure [7], this metric uses a pre-trained Inception V3 classifier network [44] to recover the features $\mathcal{F}$ and $\hat{\mathcal{F}}$ from

the final average pooling layer of that neural network respectively from the colorized images $\mathcal{I}$ and reference images $\hat{\mathcal{I}}$. Then the means $(\mu, \hat{\mu})$ and the covariances $(\Sigma, \hat{\Sigma})$, defined as the empirical mean vectors and covariance matrices of $\mathcal{F}$ and $\hat{\mathcal{F}}$ respectively, are used to compute the metric score

$$\mathrm{FID}(\mathcal{F}, \hat{\mathcal{F}}) = \|\mu - \hat{\mu}\|_2^2 + \mathrm{Tr}\left(\Sigma + \hat{\Sigma} - 2(\Sigma\hat{\Sigma})^{\frac{1}{2}}\right), \tag{12}$$

which is the Fréchet distance between the two multivariate Gaussian distributions.

The computation of Equation (12) uses the PyTorch implementation, recommended by the authors of FID [10], available in the `pytorch-fid` PyPI module[3]. To this end, the `compute_fid_dir` function is given within the `metrics/fid.py` file.

To obtain the required set of features, each image is processed by the Inception network, and the features are extracted from one of its layers. While the final average pooling layer is the most commonly used, other layers can also be chosen: the `pytorch-fid` PyPI module gives access to four layers in total, corresponding to the first max pooling layer, the second max pooling layer before the Inception modules, the auxiliary classifier output and the final average pooling layer, giving tensors of 64, 192, 768 or 2048 channels respectively. The precise architecture is presented in Table 1 of [44]. Since the first three outputs are not one-dimensional vectors but three-dimensional tensors, an average pooling is performed along the spatial dimensions to recover respectively 64, 192 and 768 scalar values. Then, the mean and the covariance are computed on the chosen layer features.

The metric is designed to be computed for a set of images, not a single image. Since it computes statistical moments of features, a sufficiently large sample is required to beat the curse of dimensionality. Otherwise, the covariance matrix cannot be accurately estimated and can simply be singular, which can lead to complex numbers or NaN when computing the square root. By default, the Inception layer used and studied in [10] gives 2048 features, which explains why a minimum of 10000 images are recommended by the authors to ensure a correct estimation within this approach.

Since it is encouraged to use a higher number of images than the number of features used, it could be argued that the first output available, giving 64 features, is the one to use to evaluate colorizations. Indeed, most colorization articles in the literature use short sequences of around thirty frames for comparison. For example, the authors of [11] use videos of twenty-five frames. However, the first layers in convolutional networks evaluate correlations between pixels on a short range because of the limited receptive field, which corresponds to local features. This is why deeper layers are used, to take into account longer-range correlations, which are supposed to be closer to human perception. Consequently, further analysis will be done here on the original output of 2048 features while keeping in mind that it can lead to numerical errors.

Besides, several articles using this metric do not specify which reference image set is used, which prevents meaningful comparison of results across publications. For instance, the video colorization article TVCV [64] explicitly computes FID between colorized and ground-truth images. In contrast, the article DEB [59] does not clarify whether the reference set corresponds to the original color frames or to the default ImageNet [37] distribution.

To evaluate temporal coherence, the authors of [46] have introduced a new metric named Fréchet Video Distance (FVD) that is essentially FID but with a trained 3D convolutional network I3D [2] in place of the 2D convolutional network Inception. This helps capture temporal inconsistencies in addition to image quality, since it applies Equation (12) to feature vectors extracted from entire videos rather than to features extracted independently from individual frames. However, this formulation makes FVD most effective on large video datasets, which are not available for evaluating legacy video colorization.

In summary, FID is effective for evaluating colorized images when a large reference dataset is available, providing a single score that reflects the overall quality of the dataset. It offers valuable

[3]pytorch-fid, https://github.com/mseitzer/pytorch-fid

insights into higher-level semantic features and can be applied to video colorization, provided a sufficient number of frames is available. However, its use as a colorization metric is made difficult by its original design for evaluating generative models, which assumes the availability of large datasets. FVD, while extending FID to capture temporal coherence, remains challenging for assessing video colorization of legacy footage due to the lack of suitable datasets for such videos.

2.2 Referenceless Metrics

In this section, we explore metrics that do not require ground truth to produce scores, which is specifically the application domain of legacy image and video colorization where no ground truth is available. Those types of metrics mainly aim at estimating characteristics of natural images and then use those features for evaluation.

Colorfulness measure. The Colorfulness measure [8] is a metric based on human judgment which uses correlation between human scores and basic image statistics to give a score. It was developed to assess the quality of compressed and transmitted images. A larger score for this metric reflects a less deteriorated image.

The Colorfulness metric corresponds in fact to three different metrics that are linear combinations of quantities computed from the CIELAB version of the image

$$\text{Colorfulness}_1 = \sigma_{ab} + 0.37\ \mu_{ab}, \tag{13}$$
$$\text{Colorfulness}_2 = \sigma_{ab} + 0.94\ \mu_{C}, \tag{14}$$
$$\text{Colorfulness}_3 = \sigma_{rgyb} + 0.3\ \mu_{rgyb}. \tag{15}$$

The first variant (Equation (13)) uses the mean and standard deviation of the image in the a* and b* channels of the CIELAB space. The quantities used in Equations (13), (14), and (15) are computed as follows, with μ_C the mean of the CIELAB chroma C

$$\begin{aligned}
\mu_{ab} &= \sqrt{\mu_a^2 + \mu_b^2}, \\
\sigma_{ab} &= \sqrt{\sigma_a^2 + \sigma_b^2}, \\
C &= \sqrt{a^2 + b^2}, \\
rg &= R - G, \\
yb &= \frac{1}{2}(R + G) - B, \\
\mu_{rgyb} &= \sqrt{\mu_{rg}^2 + \mu_{yb}^2}, \\
\sigma_{rgyb} &= \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2}.
\end{aligned}$$

The implementation of Equations (13), (14), and (15) can be found in the `metrics/colorfulness.py` file. The `compute_colorfulness` function is available to compute colorfulness, with its `metric` argument accepting the values 1, 2, or 3 to apply Equations (13), (14), and (15), respectively. For videos, the `compute_colorfulness_dir` function provides the same functionality, also requiring the `metric` parameter. This function computes the mean of Equations (13), (14), and (15) over the frames.

The coefficients and equations used to define colorfulness are set so that the metric correlates with human appreciation, evaluated by twenty subjects as an integer score between 1 and 7, corresponding to colorfulness categories ranging from “not colorful” and “slightly colorful” to “highly colorful” and “extremely colorful”.

One difference between the third variant and the first two lies in their treatment of luminance. By using the RGB channels, the third variant accounts for luminance, while the first two do not. Consequently, the first two variants evaluate only chrominance dynamics without ensuring their proper distribution. However, the third variant also does not explicitly enforce the luminance-chrominance correlation.

Concerning the first variant, Colorfulness_1, the use of the standard deviation σ_{ab} measures variations in hue and saturation. However, adding the weighted mean μ_{ab} will favor monochrome images and penalize images with balanced hues. Indeed, a saturated image with an equivalent quantity of red, yellow, blue and green - which are the extreme values of the CIELAB chrominance space - will have a mean corresponding to a desaturated image.

The second variant, Colorfulness_2, uses the same standard deviation σ_{ab} but a different weighted mean, the mean of the chroma μ_C. The advantage of using this mean is that it does not penalize images with all possible hues balanced in the ab chrominances. In fact, by applying the mean function after computing the L_2-norm of the ab chrominances, it does not measure the mean chrominance but the mean distance of all chrominances from the absence of hue. This means that the mean will provide a more accurate measure of the saturation than in the first variant.

The final variant, Colorfulness_3, uses a different color space based on RGB. It penalizes gray colors and will give a better score to red or green images over blue ones. Indeed, the first part of the metric, the rg channel will penalize images which have a balance between red and green values without looking at the blue. Meanwhile, the second part, the yb channel, requires a blue component which is opposed to the mean of red and green ones. Since both components of the metric measure the predominance of red or green, whereas only the second component measures the predominance of blue, the metric tends to assign higher scores to red- or green-dominated images than to blue-dominated images.

Despite their ability to assess overall saturation, these metrics neither ensure semantic accuracy nor detect local colorization artifacts. Therefore, a high colorfulness score does not guarantee faithful or coherent colorization.

BRISQUE and NIQE. The Blind Referenceless Image Spatial QUality Evaluator (BRISQUE) [32] and the Natural Image Quality Evaluator (NIQE) [33] are referenceless metrics for evaluating image quality using features extracted from the LIVE IQA dataset [40]. The lower the values of these metrics, the better the perceptual quality of the corresponding images.

To give a referenceless evaluation of the quality of images, those two metrics are based on human perception by using two different datasets. The LIVE IQA dataset [40], used by BRISQUE, was created by altering 29 images with JPEG and JPEG2000 compressions, white Gaussian noise, and Rayleigh fast-fading, resulting in a dataset of 779 images. These images were evaluated by approximately twenty individuals, resulting in a Difference Mean Opinion Score (DMOS), where higher values indicate lower perceived quality. The Pristine dataset [33] consists of 125 images considered to be of high quality and is used by the NIQE metric.

The two metrics work by extracting a set of features used to predict a quality score correlated with human perception. The feature extraction process begins by normalizing the light intensity of image I^t to obtain $\tilde{I}^t$. For any pixel (i, j),

$$\tilde{I}^t(i,j) = \frac{I^t(i,j) - \mu_{I^t}(i,j)}{\sigma_{I^t}(i,j) + 1}, \tag{16}$$

where

$$\mu_{I^t}(i,j) = \sum_{l=-3}^{3} \sum_{m=-3}^{3} w(l,m)\, I^t(i+l, j+m), \tag{17}$$

and

$$\sigma_{I^t}(i,j) = \left(\sum_{l=-3}^{3} \sum_{m=-3}^{3} w(l,m) \big(I^t(i+l, j+m) - \mu_{I^t}(i+l, j+m) \big)^2 \right)^{\frac{1}{2}}, \tag{18}$$

where w is a 2D circularly symmetric 7×7 Gaussian kernel.

The first set of features is derived by fitting a Generalized Gaussian Distribution (GGD) to the $\tilde{I}^t$ values

$$\mathrm{GGD}\left(\tilde{I}^t, \alpha, \sigma^2\right) = \frac{\alpha}{2\beta\,\Gamma(1/\alpha)} \exp\left(- \left(\frac{\left|\tilde{I}^t\right|}{\beta} \right)^{\alpha} \right) \quad \text{where} \quad \beta = \sigma \sqrt{\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}}. \tag{19}$$

Values kept as features are the shape parameter α and the variance σ^2.

Subsequently, four directional autocorrelations are calculated by translating $\tilde{I}^t$ in four directions— horizontal, vertical, and along both diagonals— and multiplying $\tilde{I}^t$ with these translations.

$$\mathrm{H}(i,j) = \tilde{I}(i,j)\tilde{I}(i,j+1) \tag{20}$$

$$\mathrm{V}(i,j) = \tilde{I}(i,j)\tilde{I}(i+1,j) \tag{21}$$

$$\mathrm{D1}(i,j) = \tilde{I}(i,j)\tilde{I}(i+1,j+1) \tag{22}$$

$$\mathrm{D2}(i,j) = \tilde{I}(i,j)\tilde{I}(i+1,j-1) \tag{23}$$

The other sets of features are derived from an Asymmetrical Generalized Gaussian Distribution (AGGD) fitted to each correlation, providing four parameters for each product. For BRISQUE, the retained features are the shape parameter ν, mean η, left variance σ_l^2, and right variance σ_r^2, while for NIQE they are ν, η, β_l and β_r. This step results in a total of sixteen features from the AGGD model

$$\mathrm{AGGD}\left(x, \nu, \eta, \sigma_l^2, \sigma_r^2\right) = \begin{cases} \frac{\nu}{(\beta_l+\beta_r)\,\Gamma(1/\nu)} \exp\left(-\left(-x/\beta_l\right)^{\nu}\right) & \text{where} \quad \beta_l = \sigma_l \sqrt{\frac{\Gamma(1/\nu)}{\Gamma(3/\nu)}}, \text{ if } x < 0 \\ \frac{\nu}{(\beta_l+\beta_r)\,\Gamma(1/\nu)} \exp\left(-\left(x/\beta_r\right)^{\nu}\right) & \text{where} \quad \beta_r = \sigma_r \sqrt{\frac{\Gamma(1/\nu)}{\Gamma(3/\nu)}}, \text{ if } x \geq 0 \end{cases} \tag{24}$$

$$\eta = (\beta_r - \beta_l) \frac{\Gamma(2/\nu)}{\Gamma(1/\nu)}. \tag{25}$$

Finally, this entire process is repeated on a version of the image, $I^t_{1/2}$, that is bicubically downscaled to half its original resolution, extracting additional features at a lower resolution, leading to eighteen features per scale, for a total of thirty-six features.

BRISQUE estimates parameters over the entire image, but this has the disadvantage of losing local information. To overcome this drawback, NIQE estimates features on image blocks of size 96×96 pixels, each image being described by as many descriptors as blocks. Once the features have been extracted from the LIVE IQA images for BRISQUE and from the Pristine image patches for NIQE, the metrics can be computed as follows.

BRISQUE trains an ϵ-Support Vector Regressor (SVR) to predict the DMOS with the extracted features. The BRISQUE score of an image is predicted by the trained SVR. NIQE, for its part, estimates a MultiVariate Gaussian model (MVG) from the features of Pristine image blocks. The NIQE score is given as the distance between the MVG fitted on Pristine features and the MVG fitted on I^t features

$$\mathrm{NIQE}(I^t) = \sqrt{(\hat{\mu} - \mu)^{\top} \left(\frac{\hat{\Sigma} + \Sigma}{2} \right)^{-1} (\hat{\mu} - \mu)}, \tag{26}$$

with the mean μ and the covariance Σ of the block features of the evaluated image and the mean $\hat{\mu}$ and the covariance $\hat{\Sigma}$ of the block features of the Pristine dataset.

The BRISQUE implementation is located in `metrics/brisque.py` within the `compute_brisque` function, using the original ϵ-SVR model trained by the authors. Similarly, the NIQE implementation is available in `metrics/niqe.py` with the `compute_niqe` function, relying on the original MVG estimated by the authors. For processing videos, `compute_brisque_dir` and `compute_niqe_dir` functions are provided.

Concerning the computation of Equation (24), our implementation shows numerical problems with the metric. Despite following the same masking procedure for positive and negative features, our Python implementation produces a slightly different binarization compared to the original MATLAB implementation. This discrepancy, on the order of 10^{-14}, appears when calculating the local means of Equation (17) and local standard deviations of Equation (18). Those differences are enough to lead to differences in the final scores between implementations on the order of 10^{-1}.

These two metrics have significant limitations for colorization evaluation, which were overlooked by the authors of [51]. Indeed, BRISQUE and NIQE are only defined on grayscale images and were not designed to work on artifacts specific to colorization. While there are numerous methods for converting a color image to grayscale (a topic of study in its own right; see Liu et al. [26] for a review), the MATLAB `rgb2gray` function[4] produces different results compared to extracting the L* channel from the CIELAB color space. Although different colorizations yield slight variations in grayscale conversions, these differences are negligible and do not meaningfully reflect colorization quality. Thus, this approach is unsuitable for evaluating colorization. Furthermore, the dataset used includes artifacts that are not characteristic of colorization errors. As a result, the subset of low-quality images is not representative of unsatisfactory colorization, rendering these metrics inadequate for assessing colorization quality. However, the metrics can be adapted to treat color images as we propose in the rest of this section and in a prior conference paper [30].

A possibility is to add new correlations, not along the spatial axes but along the color axis. First, each RGB channel is normalized

$$\tilde{\mathrm{R}}(i,j) = \frac{I_R^t(i,j) - \mu_{I_R^t}(i,j)}{\sigma_{I_R^t}(i,j) + 1}, \tag{27}$$

$$\tilde{\mathrm{G}}(i,j) = \frac{I_G^t(i,j) - \mu_{I_G^t}(i,j)}{\sigma_{I_G^t}(i,j) + 1}, \tag{28}$$

$$\tilde{\mathrm{B}}(i,j) = \frac{I_B^t(i,j) - \mu_{I_B^t}(i,j)}{\sigma_{I_B^t}(i,j) + 1}. \tag{29}$$

Next, interchannel correlations are computed, giving four additional features extracted for each AGGD on channels RG, RB and GB

$$\mathrm{RG}(i,j) = \tilde{\mathrm{R}}(i,j)\,\tilde{\mathrm{G}}(i,j), \tag{30}$$

$$\mathrm{RB}(i,j) = \tilde{\mathrm{R}}(i,j)\,\tilde{\mathrm{B}}(i,j), \tag{31}$$

$$\mathrm{GB}(i,j) = \tilde{\mathrm{G}}(i,j)\,\tilde{\mathrm{B}}(i,j). \tag{32}$$

This results in new BRISQUE and NIQE variants, using sixty features instead of thirty-six. We call those variants "BRISQUE - Correl" and "NIQE - Correl" and they will be studied in later sections.

A key advantage of these metrics is their ability to operate with smaller datasets to extract features from natural images, compared to FID for instance. This has been extensively explored by Mittal et al. [33] and presents a promising avenue for research in the development of legacy video colorization metrics.

[4]MATLAB Help Center, https://www.mathworks.com/help/matlab/ref/rgb2gray.html (28 November 2024)

MANIQA. The Multi-dimension Attention Network for No-Reference Image Quality Assessment (MANIQA) [54] is a referenceless metric using features extracted from Visual Transformers (ViT) [6] and other machine learning networks to give a score corresponding to the perceptual quality of the image, like BRISQUE and NIQE from the previous section. Note that higher scores correspond to better perceptual quality.

MANIQA uses consecutive neural networks to extract features from 8×8 patches of the evaluated image, giving quality scores and importance weights to each of those patches and then computing the final score as the weighted mean of the patch scores (see Equation (39)).

More specifically, a sub-image $\tilde{I}$ of resolution 224×224 is randomly cropped from the evaluated image and fed to a pretrained Vision Transformer (ViT) [6] using patches of size 8×8, yielding $N = h \times w = 784$ patches. At each of the 12 transformer layers, features $\tilde{F}_l \in \mathbb{R}^{C_l \times N}$ are computed, where C_l is the channel dimension of the ViT l layer. The features from layers 7 to 10 are then concatenated to form the final representation $\tilde{F} \in \mathbb{R}^{C \times N}$ with $C = \sum_{l=7}^{10} C_l$.

The features $\tilde{F}$ are then passed through two stages, each composed of two consecutive Transposed Attention Blocks (TAB) followed by a Scale Swin Transformer Block (SSTB). We describe TAB and SSTB in the next paragraphs.

The authors designed TAB to implicitly encode the context by using an attention over channels, inspired by Restormer [57]. The previous features $\tilde{F}$ are linearly projected to the query Q, the key K and the value V by three affine linear transformations

$$Q = \tilde{F} A_Q^\top + b_Q, \tag{33}$$

$$K = \tilde{F} A_K^\top + b_K, \tag{34}$$

$$V = \tilde{F} A_V^\top + b_V, \tag{35}$$

with $[A_Q, A_K, A_V]$ learned matrices of size (N, N) and $[b_Q, b_K, b_V]$ learned vectors of size N. The attention is then computed on (Q, K, V) and summed to $\tilde{F}$ to give the new features $\tilde{F}_{TAB}$

$$\text{Attention}(Q, K, V) = V \,\text{Softmax}\left(\frac{QK^\top}{N}\right), \tag{36}$$

$$\tilde{F}_{TAB} = \tilde{F} + \text{Attention}(Q, K, V). \tag{37}$$

Since $\frac{QK^\top}{h}$ is a (C, C) matrix, the softmax function is applied along the matrix rows.

The SSTB proposed by the authors extends the Swin Transformer [28] by sequentially applying its first two stages and adding a convolutional layer after each stage. This gives the final features $\tilde{F}_{SSTB}$.

Once the final features are computed through the two stages of TAB and SSTB, perceptual scores s_i and importance weights w_i are predicted from them using two independent linear projections, applied separately to each of the N patch features of $\tilde{I}$.

Finally, the score for the cropped image $\tilde{I}$ is given by the mean of the s_i values weighted by w_i

$$\text{MANIQA}_{\tilde{I}} = \frac{\sum_{i=1}^{N} w_i s_i}{\sum_{i=1}^{N} w_i}. \tag{38}$$

The final MANIQA score is computed by randomly cropping twenty sub-images $\tilde{I}_i$ and averaging all the scores

$$\text{MANIQA}(I^t) = \frac{1}{20} \sum_{i=1}^{20} \text{MANIQA}_{\tilde{I}_i}. \tag{39}$$

The complete network is trained on the PIPAL dataset [17] which is composed of 250 original images degraded by 116 different distortions, resulting in 29000 images evaluated by multiple human

subjects who give a Mean Opinion Score (MOS). The network is trained to predict the MOS values, and its predictions are used as MANIQA scores.

The computation of Equation (39) uses the official implementation by Zhang et al. [61] available in the GitHub repository[5]. The `compute_maniqa` function within the `metrics/maniqa.py` file can be used for computation. To compute the score on a video, the `compute_maniqa_dir` function should be used instead.

Apart from the difficulty of interpreting what is computed by the network at each layer, two limitations deserve attention. First, MANIQA operates on image crops of size 224×224, preventing the direct evaluation of smaller images. Upsampling these images may alter their perceived quality. Therefore, our implementation of MANIQA returns NaN when an image is too small. Second, MANIQA uses a fixed number of 20 random crops, regardless of the original image resolution. Consequently, the degree of overlap between crops depends on image size. Additional experiments would be needed to establish a heuristic for selecting an appropriate number of crops.

2.3 Video Metrics

Studies on the Human Visual System (HVS) focus on human perception of shapes, objects, colors, and texture. Some of these studies have examined the perception of temporal changes, showing that the human eye can detect high-frequency flickering, where apparent luminance varies rapidly. Humans are sensitive to changes of more than 63 Hz (e.g., [5] Figure 2), even in the blue color where the human eye is the less sensitive. Since legacy movies are usually available in less than 24 frames per second, ensuring a temporally consistent result is therefore of utmost importance in the case of video colorization.

Non-temporal metrics. Since the metrics presented in Section 2.1 and in Section 2.2 have been developed for single image evaluation and not for video measurement, they do not take into account the temporal consistency of colorization. To compute FID (see Section 2.1), we use the set of colorized frames as $\mathcal{I}$ and the corresponding ground-truth frames as $\hat{\mathcal{I}}$, obtaining a single score for the entire video sequence. For the other metrics, we compute the score for each frame and then average these scores over the entire video sequence.

CDC. The Color Distribution Consistency index (CDC) [27] is a referenceless metric that works exclusively on videos and aims to measure the disparities in the evolution of color between frames. A lower value for this metric represents a less distorted image.

This measure works by estimating, for all frames, the distance between their color distribution and the distribution in later frames. For a time step t, the CDC is computed as follows

$$\mathrm{CDC}_t = \frac{1}{3(T-t)} \sum_{c\in\{R,G,B\}} \sum_{i=1}^{T-t} \mathrm{JS}(P_c(I^i), P_c(I^{i+t})), \tag{40}$$

where P_c is the distribution of pixel values in color channel c of image I, T is the total number of frames in the video and JS is the Jensen-Shannon divergence between two normalized distributions x and y defined as

$$\mathrm{JS}(x,y) = \frac{1}{2}\left(\mathrm{D}(x,m) + \mathrm{D}(y,m)\right) \quad \text{with} \quad m = \frac{x+y}{2}, \tag{41}$$

[5] https://github.com/IIGROUP/MANIQA

where x and y are normalized distributions and D is the Kullback-Leibler divergence, also known as the relative Shannon entropy. The complete CDC measure is computed by averaging the time steps 1, 2 and 4

$$\mathrm{CDC} = \frac{\mathrm{CDC}_1 + \mathrm{CDC}_2 + \mathrm{CDC}_4}{3}. \tag{42}$$

The computation of Equation (42) is available in the `cdc` PyPI module by using the `compute_cdc_dir` function.

The measure uses pixel distributions, which means it is not sensitive to the spatial localization of the colors, so it is unable to detect spatial phenomena such as a static halo. However, it can measure color bleeding caused by movements. Indeed, those defects cause the colors to stay where the object was a few frames before, which increases the number of pixels covered by those colors (thus changing the color distribution), especially 2 or 4 frames later.

This metric is computed in the RGB space, which means that changes in luminance are also measured. Consequently, a video with uniform and constant chrominance will still yield a non-zero score, driven by variations in the luminance distribution. This score represents the minimal value that the metric can achieve for the chosen sequence. Furthermore, the metric is sensitive to occlusions: for example, if an object with a uniform hue disappears from the frame, the luminance distribution may be significantly altered, potentially resulting in a worse score than for a duller or more uniformly colored sequence. However, this sensitivity is intended to be mitigated by the multi-temporal aspect of the metric, which considers distances across different time intervals.

This metric is, to the best of our knowledge, the only one that has been published specifically to measure video colorization. However, no extensive testing is reported in [27].

2.4 Metrics Usage

Table 1 presents a comprehensive overview of the metrics employed across a wide range of state-of-the-art studies in colorization research. Each row corresponds to a specific evaluation metric, while the columns categorize studies into two domains: image colorization and video colorization. The bibliographic references indicate the adoption of the respective metric in a given study.

Evaluation metrics play a critical role in assessing the quality and effectiveness of colorization models. Metrics such as PSNR and SSIM are widely used due to their ability to quantify pixel-level and structural similarity between the generated output and the ground truth. These traditional metrics are particularly valued for their computational efficiency and interpretability. However, recent advancements in perceptual evaluation have highlighted the importance of metrics like LPIPS and FID, which better align with human visual perception and assess higher-level features. Their increasing adoption emphasizes a shift toward perceptually driven evaluation criteria in colorization tasks.

Furthermore, domain-specific metrics such as FVD and CDC have emerged to address the temporal consistency challenges inherent to video colorization. These metrics complement traditional image-based measures, ensuring that the temporal coherence and smoothness of color transitions are adequately evaluated in dynamic sequences.

Table 1 not only highlights the diversity in the use of metrics but also illustrates trends in the field, such as the growing interest in perceptual and temporal evaluation metrics. By providing a unified reference for the metrics employed in state-of-the-art studies, it facilitates the comparison of approaches and helps researchers select appropriate metrics for future work.

Concerning BRISQUE and NIQE, they have also been used in [49] to evaluate the restoration task on legacy images before their colorization by the algorithm described in the article.

Metrics	Image colorization articles	Video colorization articles
	Image metrics	
PSNR on RGB	[22] (2016), [63] (2017), [9] (2018), [43] (2020), [19] (2021), [52] (2022)	[23] (2019), [49, 55] (2022), [3, 29, 58] (2023), [51] (2024)
PSNR on CIELAB	[48] (2020)	[31] (2018), [15] (2019), [11] (2022)
SSIM	[43] (2020), [19] (2021), [52] (2022)	[49, 55] (2022), [29, 58] (2023), [51] (2024)
LPIPS	[43] (2020), [52] (2022)	[23] (2019), [11, 49, 55] (2022), [3, 58] (2023)
FID	[21] (2021), [16, 52, 53] (2022)	[59] (2019), [49] (2022), [18], [64, 3, 25] (2023), [51] (2024)
Colorfulness	[16, 52, 53] (2022)	[59] (2019), [3] (2023)
BRISQUE		[51] (2024)
NIQE		[51] (2024)
	Video metrics	
FVD	Not defined	[51] (2024)
CDC	Not defined	[55] (2022), [18, 64, 25] (2023)

Table 1: Metrics used for state-of-the-art comparison in colorization literature.

3 Experiments

The experimental assessment of the metrics is divided into two sections: image and video. In the image section, we degrade images using various methods described in Section 3.1 to reproduce typical artifacts that appear in state-of-the-art colorization results. It is followed by an extensive analysis of how metrics capture these artifacts in Section 3.2. In the video section, we evaluate metrics designed for video content and assess whether image metrics perform effectively on video sequences, as detailed in Section 3.3.

3.1 Proposed Alterations

Before looking at the methods used to generate common artifacts present in modern results, we first introduce the original image as a baseline in Figure 2. This image, together with others available in the `data/pictures` directory, was obtained from the pixabay[6] website where photographers publish their images to be used freely. All image sources are recorded in the `data/pictures/references.md` file.

This image establishes a clear reference point for evaluating the nature and impact of the degradations introduced by colorization algorithms. The alterations, which highlight the limitations of these methods, are categorized as follows:

- Luminance and chrominance inconsistencies: They occur when the luminance and chrominance components of the image become decorrelated, resulting in unnatural or visually disturbing effects.

- Non-semantic colors: The assigned colors may fail to align with the semantic context of the image, resulting in unrealistic depictions of objects.

- Unsatisfactory saturation: Over-saturated images appear unrealistic and exaggerated, while under-saturated images resemble partially colorized or faded visuals.

[6] https://pixabay.com/



Figure 2: Original image.

- Monochromatic colorizations: In some cases, the process introduces minimal color diversity, producing images dominated by a single, flat tone.
- Geometric transformations: They represent false defects that do not degrade the quality of the colorization and, therefore, should not be penalized by evaluation metrics.

To generate the following alterations, we use the code given in `experiments/utils/image_deterioration.py` which can be called on one image using the `experiments/scripts/apply_deteriorations.py` script.

Luminance and chrominance inconsistencies. In natural images, there is a certain correlation between luminance L* and chrominances a* and b*. Indeed, the object borders are often present in the three CIELAB channels because different objects have a high probability of exhibiting both luminance and chrominance differences. To prevent color bleeding artifacts, some methods explicitly enforce this correlation by considering edge-aware algorithms, as done in [1] and [19].

Color bleeding artifacts occur when colors propagate beyond object edges, either sharply or gradually. These artifacts are simulated by translating the luminance or chrominance channels in the CIELAB version of our image, creating a defect similar to the color bleeding shown in Figure 1. This will effectively decorrelate the luminance from the chrominance channels.

Partial translations, as illustrated in Figure 3 and defined in Algorithm 1, involve translating a central subregion of the chrominance channels a* and b* to another location. Also, global translations are applied to the entire image as presented in Figure 4.

These alterations should be detectable by appropriate metrics, as they introduce decorrelation between the luminance and chrominance channels. This decorrelation produces the color bleeding artifact visible in Figure 3 and Figure 4. However, partial translations are expected to yield better scores than global translations since they require fewer corrections to achieve acceptable quality. Artifacts are less noticeable in Figure 3 than in Figure 4, especially when viewed briefly, resulting in less severe image degradation.

Finally, it is worth noting that the two images of Figure 4 should yield similar results if the metric correctly identifies the strength of the color bleeding. Perceptually, translating either the luminance

Algorithm 1: Partial Translation

Input: RGB Image I of size $h \times w$, displacement d, direction δ, percentage p
Output: RGB Image with partially translated chrominance channels

$(L^*, a^*, b^*) \leftarrow \text{RGB2Lab}(I)$; *// Convert I to CIELAB color space*
// Compute sub-image bounding-box
$h' \leftarrow \lfloor h \cdot p \rfloor$, $w' \leftarrow \lfloor w \cdot p \rfloor$;
$x_1 \leftarrow \lfloor (w - w')/2 \rfloor$, $y_1 \leftarrow \lfloor (h - h')/2 \rfloor$;
$x_2 \leftarrow x_1 + w'$, $y_2 \leftarrow y_1 + h'$;
// Crop chrominance channels
$C_{a^*} \leftarrow a^*[y_1 : y_2, x_1 : x_2]$;
$C_{b^*} \leftarrow b^*[y_1 : y_2, x_1 : x_2]$;
// Translate the cropped chrominances
// $\delta_x, \delta_y \in \{-1, 0, 1\}$ depending on direction δ: if $\delta ==$ NORTHEAST, then $\delta_y == -1$ and $\delta_x == 1$
$C_{a^*}[y, x] \leftarrow C_{a^*}[(y - \delta_y \cdot d) \bmod h',\ (x - \delta_x \cdot d) \bmod w']$;
$C_{b^*}[y, x] \leftarrow C_{b^*}[(y - \delta_y \cdot d) \bmod h',\ (x - \delta_x \cdot d) \bmod w']$;
$a^*[y_1 : y_2,\ x_1 : x_2] \leftarrow C_{a^*}$;
$b^*[y_1 : y_2,\ x_1 : x_2] \leftarrow C_{b^*}$;
return $\text{Lab2RGB}\left((L^*, a^*, b^*)\right)$; *// Convert result to RGB*



(a) 20-pixel shift to the left.

(b) 40-pixel shift to the right and top.

Figure 3: Partial translations of the chrominances. A central subregion of the chrominance channels a* and b* is copied to another location, as indicated in the captions of Figure 3a and Figure 3b. This effect is visible on the flags and hair in Figure 3a-left and on the faces in Figure 3b-right.

or chrominance components by the same number of pixels produces similar color-bleeding artifacts.

Halo effects result from the diffusion of colors with a different sharpness than the luminance edges, creating smooth transitions between distinct objects. To simulate these artifacts, we apply a central zoom in the chrominance channels to create smoother chroma edges in the central region, as defined in Algorithm 2 and shown in Figure 5a. To obtain a global transformation, we apply a Gaussian blur to the chrominance channels, as in Figure 5b.

These artifacts are harder to detect, as they do not attempt to decorrelate the luminance and chrominance edges but instead smooth the chrominance boundaries. However, a metric specialized in detecting image colorization artifacts should be able to detect these issues.



(a) Translation of luminance L*.



(b) Translation of a* and b* chrominances.

Figure 4: Complete translations of CIELAB channels. Specifically, in Figure 4a-left, only the luminance channel is shifted 20 pixels to the left while in Figure 4b-right, both chrominance channels a* and b* are shifted 20 pixels to the left. The color bleeding is visible in both images, on the flags, faces and hair.

Algorithm 2: Central Zoom

Input: RGB Image I of size $h \times w$, zoom factor z
Output: RGB Image with zoomed chrominance channels
$(L^*, a^*, b^*) \leftarrow \text{RGB2Lab}(I)$; *// Convert I to CIELAB color space*
// Compute sub-image bounding-box
$h' \leftarrow \lfloor h/z \rfloor$, $w' \leftarrow \lfloor w/z \rfloor$;
$x_1 \leftarrow \lfloor (w - w')/2 \rfloor$, $y_1 \leftarrow \lfloor (h - h')/2 \rfloor$;
$x_2 \leftarrow x_1 + w'$, $y_2 \leftarrow y_1 + h'$;
// Crop chrominance channels
$C_{a^*} \leftarrow a^*[y_1 : y_2, x_1 : x_2]$;
$C_{b^*} \leftarrow b^*[y_1 : y_2, x_1 : x_2]$;
// Overwrite chrominances with bilinearly upsampled crops
$a^* \leftarrow \text{resize}(C_{a^*}, h, w)$;
$b^* \leftarrow \text{resize}(C_{b^*}, h, w)$;
return $\text{Lab2RGB}\left((L^*, a^*, b^*)\right)$; *// Convert result to RGB.*



(a) Central zoom on the chrominances.



(b) Gaussian blur on the chrominances.

Figure 5: Halo simulation. Figure 5a-left presents the same features as the previous alterations with a global displacement of colors. For Figure 5b-right, the artifacts are subtler but can be observed when zooming into the boundaries of objects. This is noticeable on the sign in the left part of the image, where colors from the letters of the panel blend into their surroundings, smoothing the original distinct colors.

Non-semantic colors. To obtain a realistic colorization, it is important to ensure that each object will be assigned only realistic colors. For instance, an apple can be colorized in red, yellow, green but not blue. This semantic consistency is particularly important when restoring historical images.

To ensure that artifacts are applied only to the hues and do not alter the luminance/chrominance correlations, we perform "Hue shifted" and "Hue altered" transformations in the hue channel H of the HSV color space. The hue channel in HSV is a circular channel with values ranging from 0° to 360°. Specifically, the alteration in Figure 6a is done by adding a constant angle to the entire channel as described in Equation (43) and in Algorithm 3, while the alteration in Figure 6b applies an affine transformation of the form $a(x+b)-b$ to the hue channel as described in Equation (44) and in Algorithm 4, introducing new variations in originally similar hues.

$$H_{\text{shifted}} \equiv H_{\text{original}} + s \mod 360, \tag{43}$$

$$H_{\text{altered}} \equiv f\left(H_{\text{original}} - o \mod 360\right) + o \mod 360, \tag{44}$$

with $s = 120°$, $o = 50°$ and $f = 5$.

Algorithm 3: Shift in hue

Input: RGB Image I of size $h \times w$, shift s
Output: RGB Image with shifted hue
$(H, S, V) \leftarrow$ RGB2HSV(I) ; *// Convert I to HSV color space*
$H_{shifted} \leftarrow (H + s) \mod 360$;
return HSV2RGB$(H_{shifted}, S, V)$; *// Reconstruct RGB image from $(H_{shifted}, S, V)$*

Algorithm 4: Affine transformation in hue

Input: RGB Image I of size $h \times w$, offset o, factor f
Output: RGB Image with altered hue
$(H, S, V) \leftarrow$ RGB2HSV(I) ; *// Convert I to HSV color space*
$H_{altered} \leftarrow (f \cdot ((H - o) \mod 360) + o) \mod 360$;
return HSV2RGB$(H_{altered}, S, V)$; *// Reconstruct RGB image from $(H_{altered}, S, V)$*

These artifacts should be detected by the metrics as inappropriate color choices, but they are correctly saturated since we do not alter the saturation channel, and there are no spatial artifacts. Therefore, metrics that detect only these defects would consider the images as well-colored.

Unsatisfactory saturation. A common limitation of image colorization algorithms is their inability to produce colorful colorizations for every type of scene. For example Colorful Image Colorization [60, 63] produces dull versions of our image as shown in Figure 7 where the "CIC - ECCV" Figure 7a-left corresponds to the first version of the algorithm [60] and the "CIC - SIGGRAPH" Figure 7b-right corresponds to the second version [63]. These colorizations are performed on color images using the CIC implementation, which extracts the luminance channel L^* from the images converted to the CIELAB color space.

Our alterations are designed to simulate dull colorizations. One way to desaturate the image is by reducing the saturation in the HSV color space, as described in Algorithm 5. We apply this transformation in two ways: in the central region of the image (Figure 8a), by setting the saturation to zero, and across the entire image (Figure 8b), by reducing the saturation by 80%. In both cases, the hue remains unchanged. Therefore, metrics that assess color semantics should not detect these artifacts unless they also account for saturation.



(a) Shift in hue.



(b) Affine transformation in hue.

Figure 6: Hue artifacts in HSV color space. In Figure 6a-left, the change in hue is visible in the bottom banner, where orange has changed to light green. In Figure 6b-right, there are more variations in the hues since an affine transformation has been applied. This is especially visible on the faces and in the background, where the makeup and the foliage have more varied hues than in Figure 6a.



(a) ECCV version [60].



(b) SIGGRAPH version [63].

Figure 7: Colorization using CIC. These two images exhibit dull colors, representative of automatic image colorizations by convolutional networks.



(a) Total desaturation in the center.



(b) Partial desaturation of the image.

Figure 8: Partial desaturation in the HSV color space. In Figure 8b-right, the colorization becomes more than just dull and in the center of Figure 8a-left, colors are absent. The metrics should detect these artifacts as lacking color entirely.

Algorithm 5: Desaturation

Input: RGB Image I of size $h \times w$, central patch percentage o, factor f
Output: RGB Image with altered saturation
$(H, S, V) \leftarrow$ RGB2HSV(I) ; *// Convert I to HSV color space*
$h' \leftarrow h \cdot o$, $w' \leftarrow w \cdot o$;
$x_1 \leftarrow \lfloor (w - w')/2 \rfloor$, $y_1 \leftarrow \lfloor (h - h')/2 \rfloor$;
$x_2 \leftarrow x_1 + w'$, $y_2 \leftarrow y_1 + h'$;
$S_{altered} \leftarrow S$;
$S_{altered}[y_1 : y_2, x_1 : x_2] \leftarrow (1 - f) \cdot S_{altered}[y_1 : y_2, x_1 : x_2]$;
return HSV2RGB$(H, S_{altered}, V)$; *// Reconstruct RGB image from $(H, S_{altered}, V)$*

To further evaluate how well metrics can distinguish between dull colorization and absence of colorization, we also create grayscale versions of the image by retaining only the red channel from the RGB color space (Figure 9a), an alternative grayscale representation to the luminance from the CIELAB space (Figure 9b). These images serve as examples of grayscale versions that could be extracted from the color image, providing insight into how metrics differentiate between unsaturated colorization and colorless images. Notably, the red channel representation produces less variety in chromatic regions where colors share similar red proportions but differ in luminance, such as in the rainbow banner visible at the bottom of Figure 9a, which shows only two distinct gray levels, compared to five in Figure 9b.



(a) Extracted red channel. (b) Extracted luminance.

Figure 9: Grayscale versions of the image. These two images contain no color information and are represented by a single channel.

However, desaturation is not the only issue related to saturation. A colorization can also be considered unsatisfactory if it is oversaturated. To evaluate whether any metric can detect this artifact, we also introduce oversaturation in the image shown in Figure 10 by multiplying the saturation channel by three, as described in Algorithm 6.

It can be seen that these images have unnatural colors because the saturation is too high and while artists can choose to apply this type of treatment, automatic colorization needs to produce realistic results before being retouched by a professional.

Monochromatic colorizations. An issue that can arise independently of inadequate saturation is the limited diversity of colors. It is visible in Figure 7a where the main color is yellow. As with other colorization methods, CIC produces a uniform color across the entire image, often resembling a sepia effect. To study how metrics handle these artifacts, we propose transformations that result in



Figure 10: Oversaturation in the HSV color space.

Algorithm 6: Oversaturation

Input: RGB Image I of size $h \times w$, factor f
Output: RGB Image with altered saturation
$(H, S, V) \leftarrow$ RGB2HSV(I) ; *// Convert I to HSV color space*
$S_{altered} \leftarrow (1 + f) \cdot S$;
$S_{altered} \leftarrow \text{clip}(S_{altered}, 0, 1)$; *// Clip $S_{altered}$ values outside [0,1]*
return HSV2RGB$(H, S_{altered}, V)$; *// Reconstruct RGB image from $(H, S_{altered}, V)$*

monochrome images. One such transformation involves applying a sepia filter, as shown in Figure 11a, which creates a uniform brown and orange hue. It is performed by multiplying the RGB color image by the following rank-1 matrix inspired by the matrix proposed by the W3C[7]

$$\text{sepia_filter} = \begin{bmatrix} 1.44 \cdot 0.272 & 1.44 \cdot 0.534 & 1.44 \cdot 0.131 \\ 1.28 \cdot 0.272 & 1.28 \cdot 0.534 & 1.28 \cdot 0.131 \\ 0.272 & 0.534 & 0.131 \end{bmatrix}$$

To jointly examine how metrics handle CIELAB channel correlations and their tendency to favor averaged colors, we alter the chrominance channels by replacing them with their mean value, as demonstrated in Figure 11b and described in Algorithm 7.

Algorithm 7: Mean Chrominances

Input: RGB Image I of size $h \times w$
Output: Desaturated RGB image where chrominances are replaced by their mean values
$(L, a, b) \leftarrow$ RGB2Lab(I) ; *// Convert I to CIELAB color space*
$\bar{a} \leftarrow \frac{1}{hw} \sum_{i,j} a_{i,j}$;
$\bar{b} \leftarrow \frac{1}{hw} \sum_{i,j} b_{i,j}$;
return Lab2RGB $\left((L, \bar{a}, \bar{b})\right)$; *// Convert result to RGB*

Additionally, we compute the mean by pixel of the RGB channels and project it onto a single channel: red, green, or blue. The results of these projections are illustrated in Figures 12a, 12b, and 12c, respectively. These transformations, while monochromatic, provide valuable insights into

[7]W3C Working Draft "Filter Effects Module Level 1", https://www.w3.org/TR/filter-effects-1/#sepiaEquivalent.



(a) Sepia filter applied to the image.

(b) Image with mean chrominance applied.

Figure 11: One-dimensional color filters. Those two images, despite being coded in the three-channel RGB color space, can also be represented as a one-channel image that lies on a different color axis than the gray axis, without losing any information.

the sensitivity of perceptual metrics to color. As discussed in [34], the Human Visual System shows higher sensitivity to variations in the red and green channels, and lower sensitivity to variations in the blue channel.



(a) Mean projected to the red channel.

(b) Mean projected to the green channel.

(c) Mean projected to the blue channel.

Figure 12: Projection of the mean RGB values onto individual color channels.

Geometric transformations. It is important to recall that alterations depicted in Figures 3 to 12 leave the luminance channel unaltered. To assess the robustness of the metrics—especially reference-less ones—against a spurious detection of quality drop, we test them with transformations that do not inherently alter the quality of colorization. Specifically, we consider the following transformations: image translation (Figure 13a), horizontal flipping (Figure 13b), and vertical flipping (Figure 13c).

The translation introduces spatial discontinuities at the image edges but does not affect the internal colorization, meaning metrics should ideally ignore such changes. Horizontal flipping preserves the overall appearance and color consistency, so metrics should identify it as a perfect match. In contrast, vertical flipping may challenge perceptual metrics based on neural networks if the training process did not include data augmentation for this transformation.

Summary. Table 2 summarizes all previously described alterations also giving the titles by which they are designated in the following figures. To facilitate the analysis, the alterations are color-coded by category: "Luminance and chrominance inconsistencies" are in red, "Non-semantic colors" are in magenta, "Unsatisfactory saturation" are in gray, "Monochromatic colorizations" are in blue and "Geometric transformations" are in green. The two colorizations performed with CIC, classified as "Unsatisfactory colorization", are in black since they do not use any additional prior from the user



(a) Translation of the image. (b) Horizontal flip. (c) Vertical flip.

Figure 13: Transformations without colorization-related issues.

to replicate the same hues as the ground truth color image. Also, these colorizations are usually not colorful, with both a lack of hue variety and low saturation.

Alteration	**Figure link**	**Color code**
Partial translation 1	Figure 3a	Red
Partial translation 2	Figure 3b	Red
Luminance translated	Figure 4a	Red
Chrominance translated	Figure 4b	Red
Halo by zoom	Figure 5a	Red
Chrominances edges blurred	Figure 5b	Red
Hue shifted	Figure 6a	Magenta
Hue altered	Figure 6b	Magenta
CIC - ECCV	Figure 7a	Black
CIC - SIGGRAPH	Figure 7b	Black
Desaturated in the center	Figure 8a	Gray
Desaturated	Figure 8b	Gray
Gray from red channel	Figure 9a	Gray
Only luminance	Figure 9b	Gray
Oversaturated	Figure 10	Gray
Sepia filter	Figure 11a	Blue
Mean chrominance	Figure 11b	Blue
Red filter	Figure 12a	Blue
Green filter	Figure 12b	Blue
Blue filter	Figure 12c	Blue
Image translated	Figure 13a	Green
Horizontal flip	Figure 13b	Green
Vertical flip	Figure 13c	Green

Table 2: List of alterations. "Luminance and chrominance inconsistencies" are in red, "Non-semantic colors" are in magenta, "Unsatisfactory colorization" are in black, "Unsatisfactory saturation" are in gray, "Monochromatic colorizations" are in blue and "Geometric transformations" are in green.

3.2 Image Results

To evaluate the metrics on a significant number of images, we used the Unsplash Lite dataset[8], available in its 1.2.2 version at the time of evaluation. Images with URLs specified in `data/unsplash/removed.txt` were excluded because of their unavailability, leaving a total of 24,969 images. This dataset, comprising 25,000 nature-themed images, was selected for its ease of use and permissive licensing, which facilitate reproducible research.

[8]Unsplash Lite dataset, 1.2.2 version, https://github.com/unsplash/datasets/releases/tag/1.2.2

For the evaluation, the following experiment is conducted: each image of the dataset is processed with the numerous alterations described in Section 3.1 and then alterations are evaluated using each metric explained in Section 2. This is summarized in Algorithm 8.

Algorithm 8: Metrics evaluation on images

Input: Images $\mathcal{I}$, alteration functions $\mathcal{F}$, metric functions $\mathcal{M}$
Output: Dictionary *Dict* of scores per image, alteration and metric, saved in HDF5 format
foreach *image* I *in* $\mathcal{I}$ ***in parallel*** **do**
  foreach *alteration function* f *in* $\mathcal{F}$ **do**
    if I_{alt} *not already saved on disk* **then**
      $I_{alt} \leftarrow f(I)$;
      Save I_{alt} to disk;
    foreach *metric function* m *in* $\mathcal{M}$ **do**
      if $Dict[I, f, m]$ *not already computed* **then**
        $Dict[I, f, m] \leftarrow m(I, I_{alt})$;
return *Dict*

We use box and whisker plots to summarize the results across the 24,969 images. Such a plot summarizes the distribution of results through five key values: minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum. The box represents the interquartile range (IQR), spanning from Q1 to Q3, while the median (Q2) is marked with a red line inside the box. The whiskers extend from the box, showing the range of data within 1.5 times the IQR from the quartiles. These whiskers stop at the actual minimum and maximum values within this range. Boxplots were used to compare the distribution of image metrics across different alterations of the dataset.

The following boxplots present the results, ordered from the best median score on the left to the worst on the right, with the median displayed above each box. This means that alterations on the left are considered better colorizations than those on the right. The boxplots were generated using the `pyplot.boxplot` function from the `matplotlib` [12] PyPI module, with outliers deactivated to improve the readability of the graphs. To generate those figures, we use the code given in the following path: `experiments/compute/exp2_dataset.py`.

PSNR. According to the classical RGB-based PSNR metric (see Figure 14), PSNR is insensitive to color bleeding of "Luminance and chrominance inconsistencies" (red labels), since partial or total translations of chrominances, chrominance zooming and blurring are the alterations that obtain the highest scores. This is due to the averaging over all pixels which reduces the influence of color-bleeding and halo artifacts on the overall score. The following group of defects is the "Unsatisfactory saturation" (gray labels), with a surprising result: a complete desaturation with the **Only luminance** alteration achieves a higher score than an 80 % desaturation with the **Desaturated** alteration. This confirms our analysis of the metric: luminance has a greater impact than colors. This metric favors dull colors over more varied and saturated colors. This is especially visible with the scores of hue alterations in the "Non-semantic colors" (magenta labels), "Unsatisfactory colorizations" (black labels) and **Oversaturated** that are worse than the desaturation scores. Concerning the "Monochromatic colorizations" (blue labels), the alteration with the highest score is the **Mean chrominance**. Scoring only 20 % below the best result, this alteration, which assigns a single hue to the entire image, ranks surprisingly high, further supporting the tendency of PSNR to favor averaged outputs. Among the worst images according to PSNR are **Luminance Translated** and "Geometric transformations" (green labels), which modify heavily the luminance, indicating once again the

importance of luminance over chrominance in PSNR in RGB space. Finally, the three primary-color filters from the "Monochromatic colorizations" (blue labels) category obtain the worst scores, while the sepia filter achieves an intermediate score. This suggests that dull colors, with contributions from all three RGB channels, are preferred over colors dominated by a single channel.

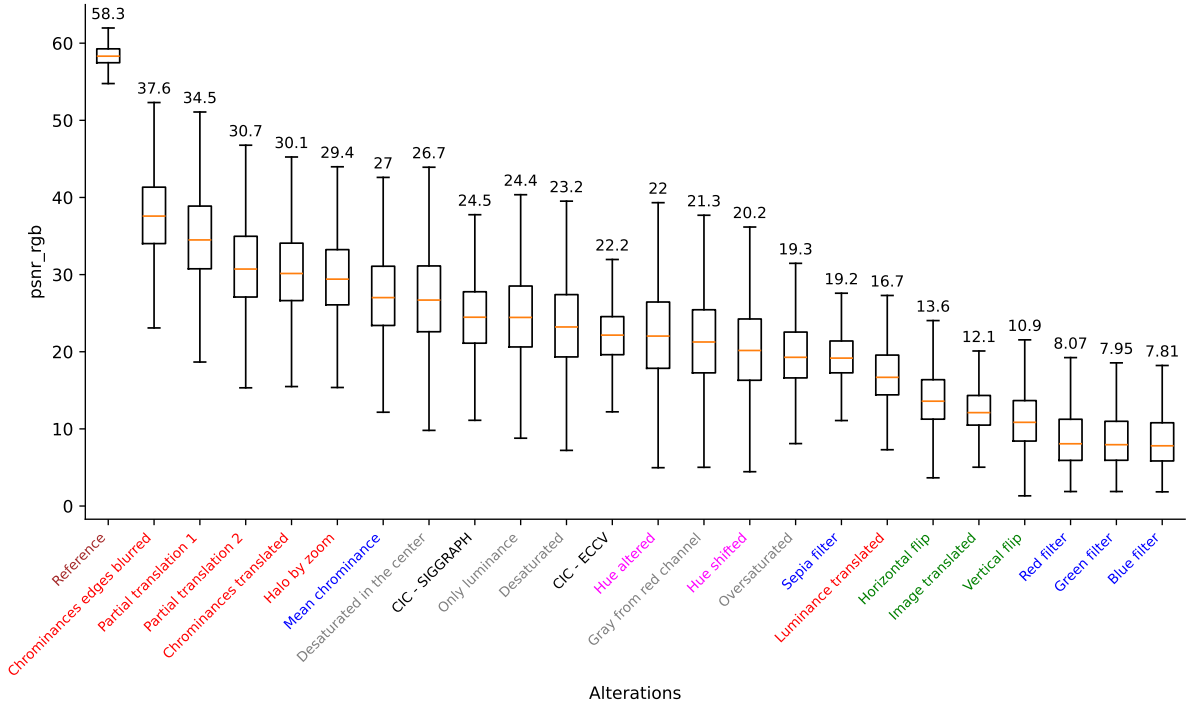


Figure 14: PSNR evaluation in the RGB space on the Unsplash dataset. Higher is better.

Concerning the computation of PSNR on the ab channels from the CIELAB space (Figure 15), the results are similar, except for the **Luminance translated** alteration (red label) which does not modify the a* and b* channels.

The **Mean chrominances** alteration (blue label) still has a relatively high PSNR, indicating that the metric remains overly favorable to drab colorizations that are close in color to the ground truth.

A noteworthy aspect of this CIELAB PSNR is the relatively high score of "Geometric transformations" (green labels). This can be explained by the lower variations in chrominances compared to luminance since textures of objects are mostly captured in the latter channel.

SSIM. This metric, which scores the ideal image as 1 and the most degraded image as 0, is more sensitive to some alterations than PSNR. Indeed, in Figure 16, "Luminance and chrominance inconsistencies" (red labels) alterations that translate the luminance, "Geometric transformations" (green labels) alterations and "Monochromatic colorizations" (blue labels) primary color filters are heavily penalized. Concerning the results for color space projections **Red/Green/Blue filters**, they are understandable by the nature of the SSIM computation: it is computed by applying Equation (10) independently on each RGB channel and averaging the results. Removing the information from two of the three channels reduces the score by two-thirds of its maximum value. For the "Luminance and chrominance inconsistencies" (red labels) alteration, **Luminance translated**, and the "Geometric transformations" (green labels) alterations **Image translated** and the two flips, SSIM appears more sensitive than PSNR to luminance misalignment, possibly because it compares local neighborhoods in the RGB color space. These alterations also exhibit the greatest variability across the dataset:

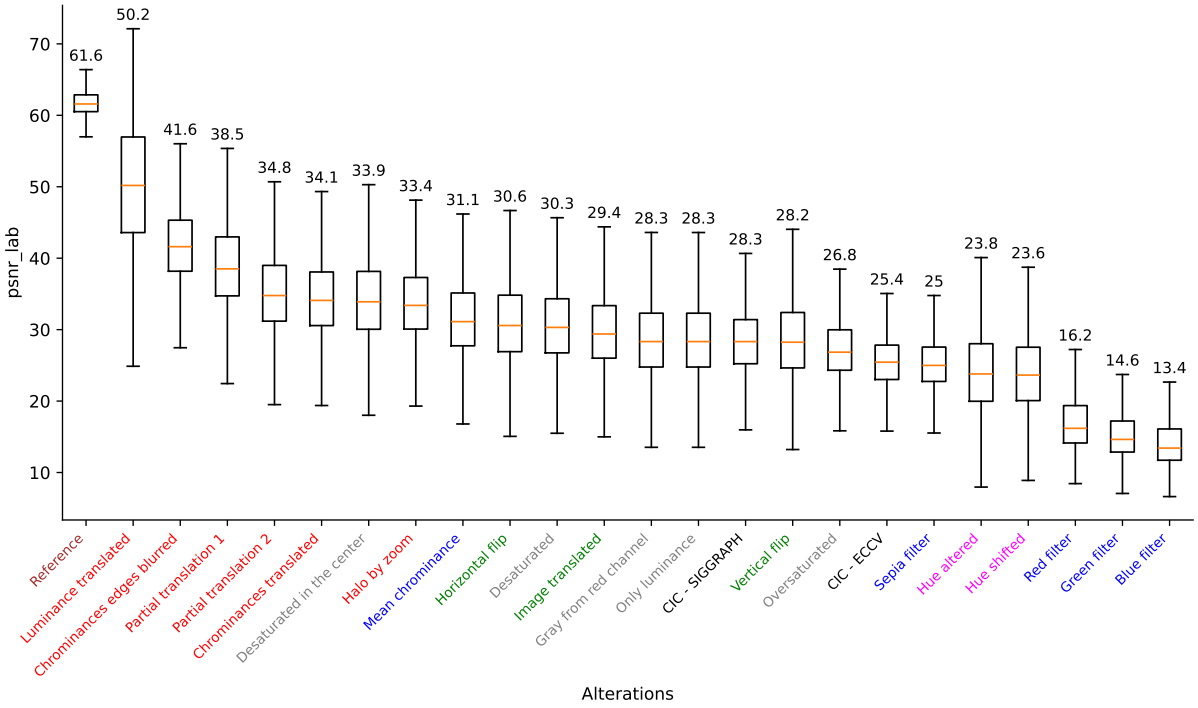


Figure 15: PSNR evaluation in the CIELAB space on the Unsplash dataset. Higher is better.

their whiskers span more than 80 % of the possible score range. This suggests that the lack of spatial correspondence between the luminance of the ground-truth and altered images affects the stability of the structural comparison performed by SSIM.

Furthermore, the metric does not check the correlation between luminance and chrominances as the "Luminance and chrominance inconsistencies" (red labels), blue labeled **Mean chrominance** and **Sepia filter** achieve high scores alongside CIC results. The "Unsatisfactory saturation" (gray labels) desaturations and the "Non-semantic colors" (magenta labels) artifacts have lower medians but wider whiskers than color bleeding, showing that SSIM is unreliable for discriminating which hues and saturations are acceptable as colorizations. Because of the variability of the results across different groups of alterations, PSNR on CIELAB should be preferred to SSIM.

LPIPS. For this analysis, we use results of LPIPS trained on AlexNet (Figure 17) and on VGG (Figure 18). As with SSIM, some alterations are clearly penalized — with higher scores — compared to the others: the **RGB filters** in "Monochromatic colorizations" (blue labels), which indeed simulate bad colorizations. However, "Geometric transformations" (green labels) **Image translated** and **flips** are also grouped with those defects, meaning LPIPS does not achieve a correct discrimination of the worst colorizations, regardless of the network used for feature extraction.

Interestingly, the poor results for **Image translated** and **H/V flips** suggest that LPIPS does not perform better than SSIM in assessing global consistency. Indeed, these three transformations preserve the same semantics and luminance-chrominance correlations as the ground truth. Those results indicate that classical convolutional neural networks for classification may be more sensitive to the spatial position of objects rather than to their semantics. Another hypothesis is that LPIPS, which learns to weight latent space features from human perceptual judgments on classical image distortions (e.g., noise, blur, compression artifacts), may fail to identify which components of the latent representation are relevant to colorization quality assessment, as colorization artifacts are absent from its training distribution.

Although the order of the groups is identical to that of SSIM, LPIPS gave similar variations for

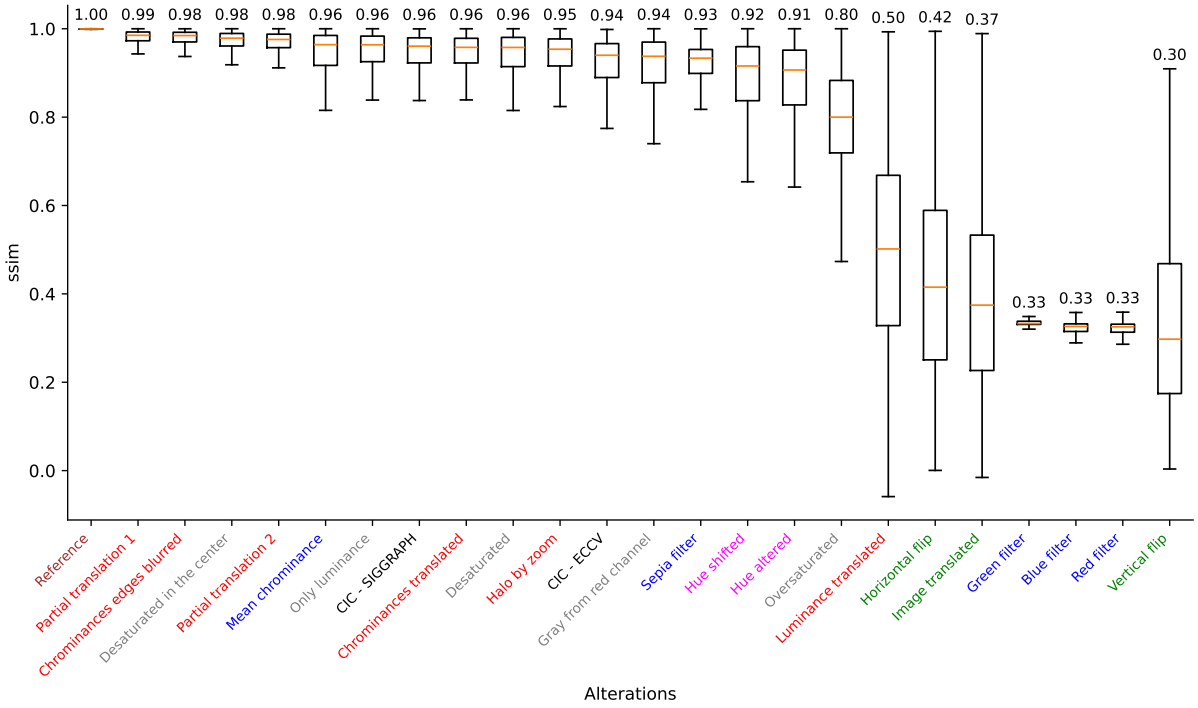


Figure 16: SSIM evaluation on the Unsplash dataset. Higher is better.

most alterations, whereas the SSIM results show strong variability for the worst groups. The order is similar to the results obtained with PSNR: color-bleeding and halos of "Luminance and chrominance inconsistencies" (red labels) are less penalized than hue modifications. Furthermore, "Non-semantic colors" (magenta labels) hue changes and all **RGB filters** show the greatest fluctuations.

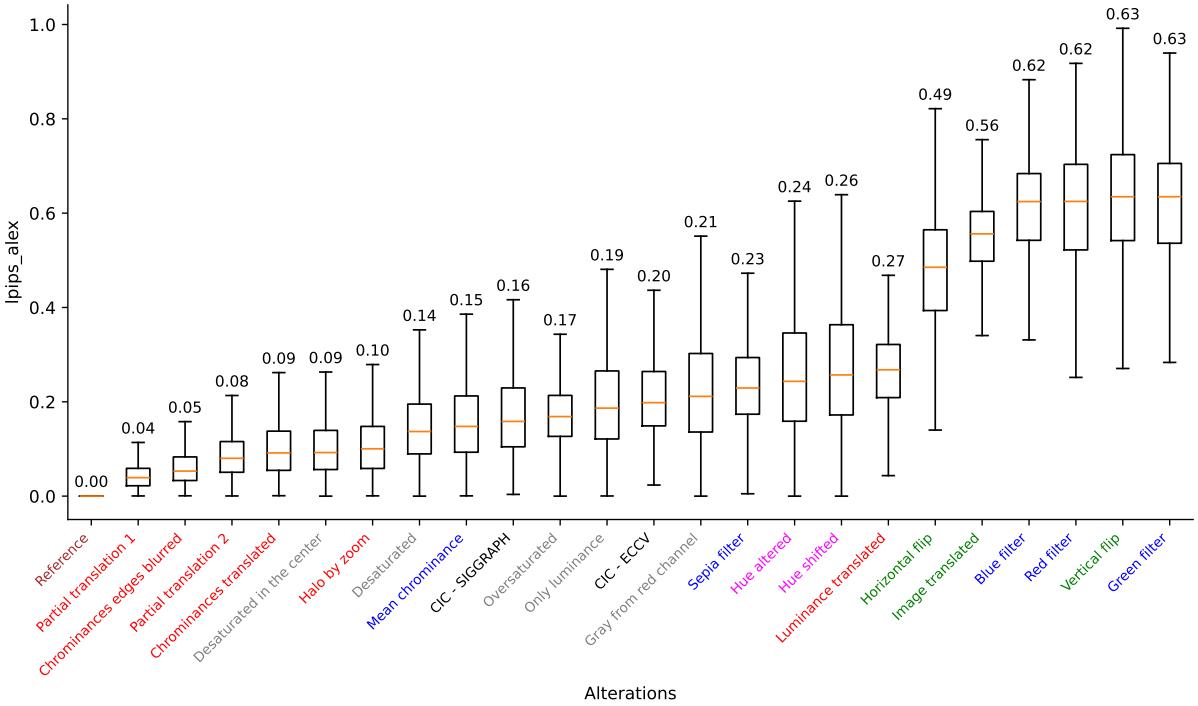


Figure 17: LPIPS with AlexNet evaluation on the Unsplash dataset. Lower is better.

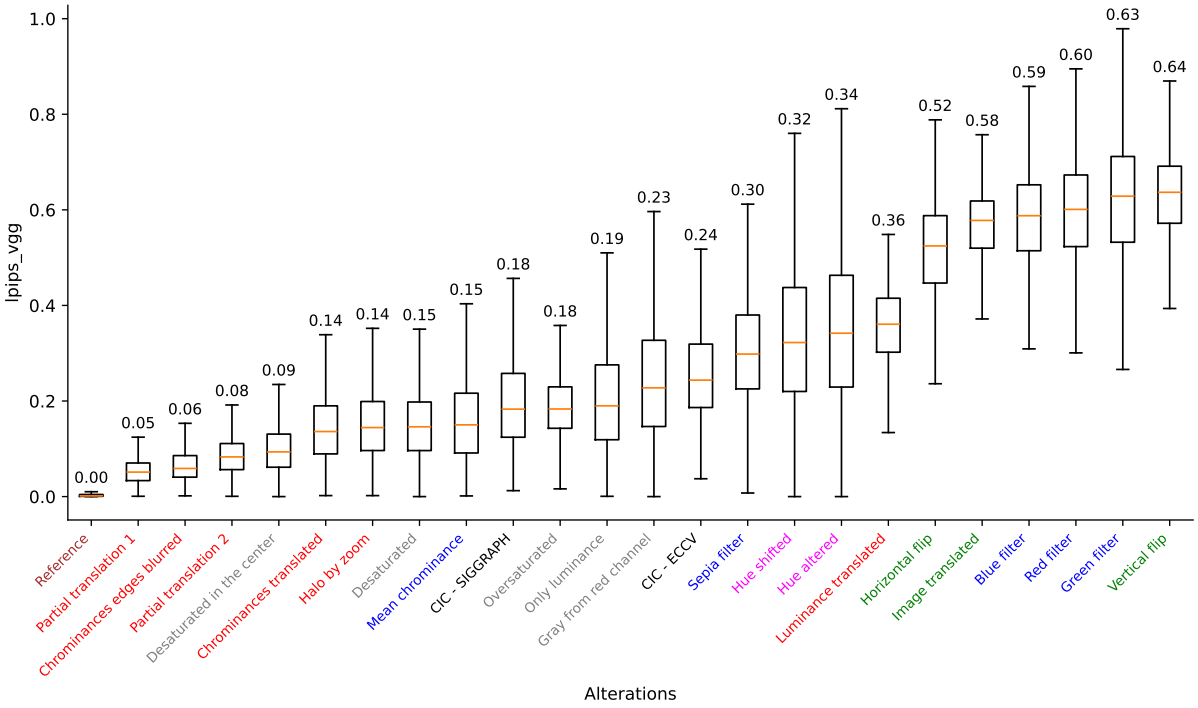


Figure 18: LPIPS with VGG evaluation on the Unsplash dataset. Lower is better.

FID. Since the following evaluations compare sets containing a single image, we had to account for the fact that a division by zero occurs. To prevent NaN matrices, we assume that each set consists of two identical images. This results in zero matrices, reducing Equation (12) to

$$\mathrm{FID}(\hat{\mathcal{F}}, \mathcal{F}) = \|\hat{\mu} - \mu\|_2^2 . \tag{45}$$

Nonetheless, FID yields interesting results when a large number of features is used. In this section, we present the results obtained with 2048 features in Figure 19.

The FID measure presents a better perceptual evaluation than previously analyzed metrics. Indeed, it is visible in Figure 19, where the scores given to "Luminance and chrominance inconsistencies" (red labels) **Luminance translated** and **Chrominance translated** are equivalent. This shows that the features extracted from the deepest Inception layers exhibit similar sensitivity to luminance and chrominance displacements, suggesting a more balanced perceptual representation than shallow features. Also, **Horizontal flip** and **Image translated** of "Geometric transformations" (green labels) have the first and sixth best scores respectively and "Unsatisfactory saturation" (gray labels) **Oversaturated** alteration has scores similar to all desaturation defects. All those results confirm that the deepest Inception layers give a better perceptual representation than the features used in LPIPS.

Colorfulness measure. Since the three colorfulness variants yield similar results, we focus our analysis on the third variant in Figure 20. Colorfulness measures penalize desaturation of "Unsatisfactory saturation" (gray labels) as they have low scores, between "Not colorful" and "Slightly colorful" (as defined in Section 2.2) according to Table 2 of the metric in [8], while **Oversaturated** and "Monochromatic colorizations" (blue labels) **RGB filters** obtain the best results, placing them between "Quite colorful" and "Extremely colorful" according to the previously mentioned table. Most of the "Luminance and chrominance inconsistencies" (red labels) and "Geometric transformations" (green labels) results share the same order of magnitude. Because of these phenomena, the

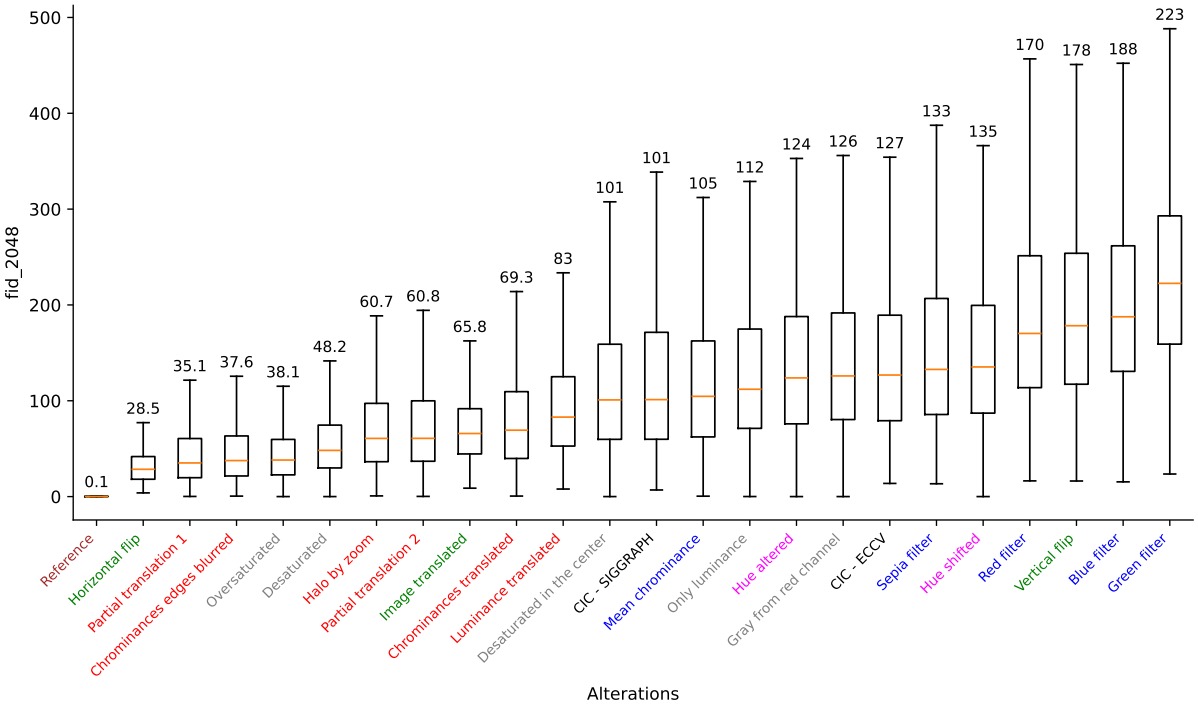


Figure 19: FID with 2048 features evaluation on the Unsplash dataset. Lower is better.

colorfulness metric is unable to detect if objects are correctly colored or present color bleeding and halos. This metric can only detect if an image is globally saturated or not.

A few defects have limited variations across the image dataset: "Unsatisfactory colorizations" (black labels), "Monochromatic colorizations" (blue labels) **Sepia filter** and **Mean chrominance** and "Unsatisfactory saturation" (gray labels) **Desaturated**. Since **Gray from red channel** and **Only luminance** do not have any colors at all, they achieve a constant null score. When saturation or hue variety is reduced, the metric scores become more tightly clustered within a narrower range, suggesting that the metric is more consistent in capturing low-saturation alterations.

This metric could be used to assess whether the colorization has the desired saturation but seems poorly suited to detecting other defects.

BRISQUE and NIQE. These metrics assign similar scores to alterations that preserve luminance. This is because only the grayscale versions of the images are processed. This is verified by the score of the "Monochromatic colorizations" (blue labels) **RGB filters** that sum all the RGB channels into one, which alter the result of the RGB to grayscale transformation. It is particularly visible with the **Blue filter** in Figure 21 and in Figure 22. Although some differences can be observed among the other alterations, they are not substantial, as the box plots show similar distributions with considerable variability and overlapping whiskers. To conclude, those two metrics are unusable in the colorization evaluation context.

However, both metrics were used in [51], together with PSNR, SSIM, FID and FVD. Since the aim of that paper is to compare colorizations and not restoration tasks as [49], this use is inappropriate, since BRISQUE and NIQE only deal with the luminance.

Concerning the extension of BRISQUE and NIQE to color images, there is no substantial improvement in detecting defects. As shown in Figure 23 and in Figure 24, the "Monochromatic colorizations" (blue labels) **RGB filters** are now well dissociated from the rest of the alterations. However, the other alterations still obtain similar scores, making them difficult to distinguish.

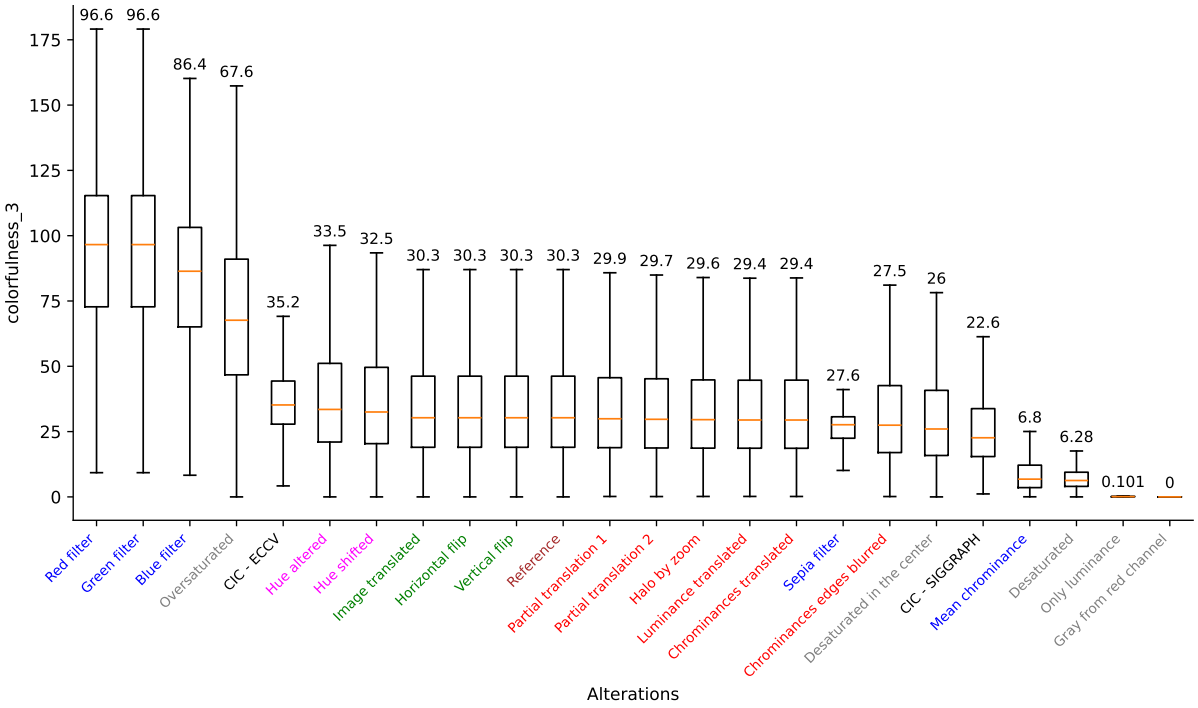


Figure 20: Colorfulness measure third version evaluation on the Unsplash dataset. Higher is better.

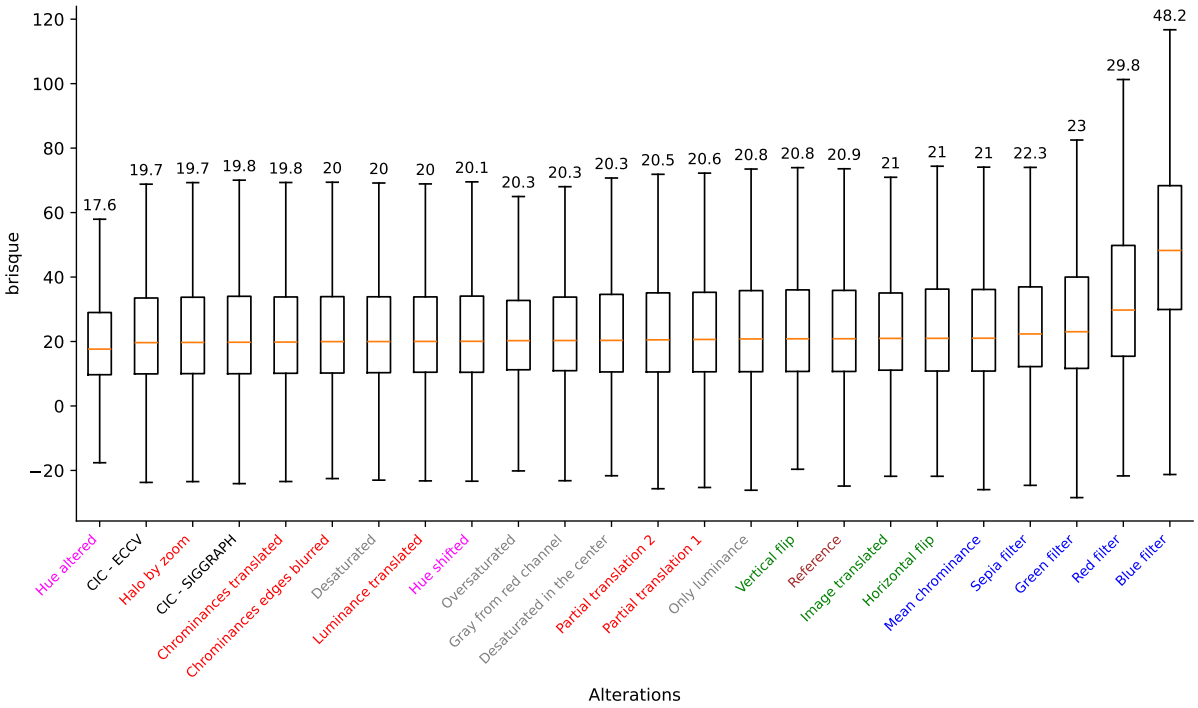


Figure 21: BRISQUE evaluation on the Unsplash dataset. Lower is better.

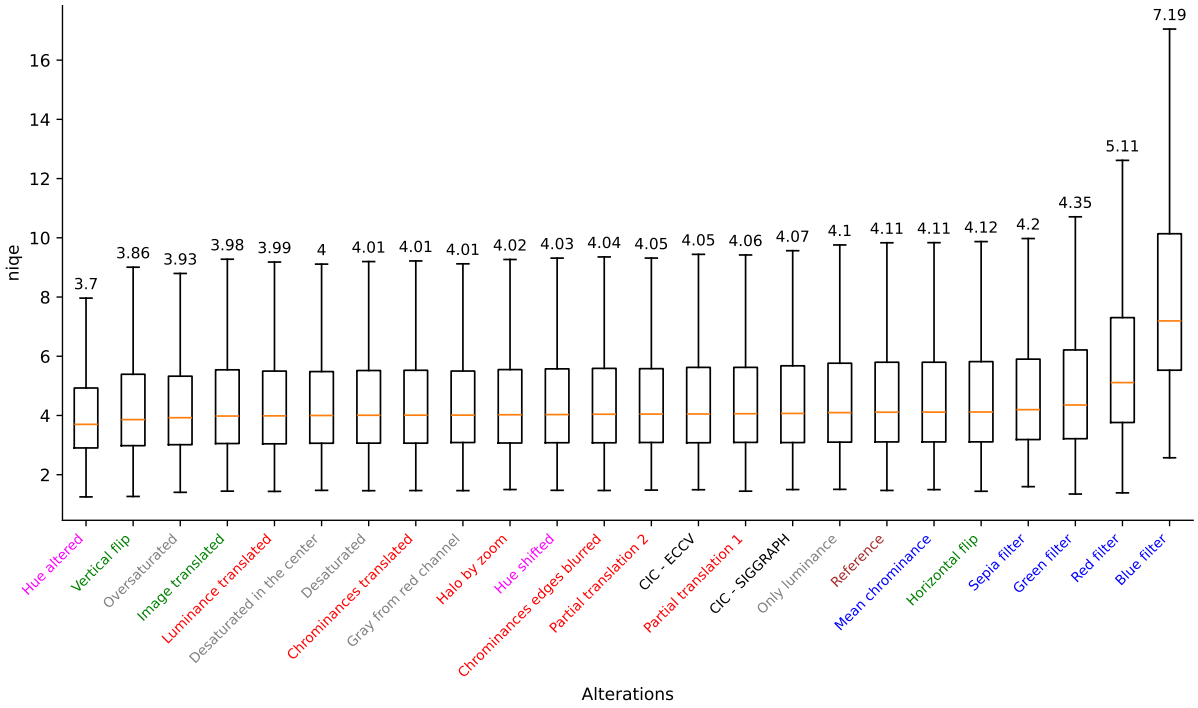


Figure 22: NIQE evaluation on the Unsplash dataset. Lower is better.

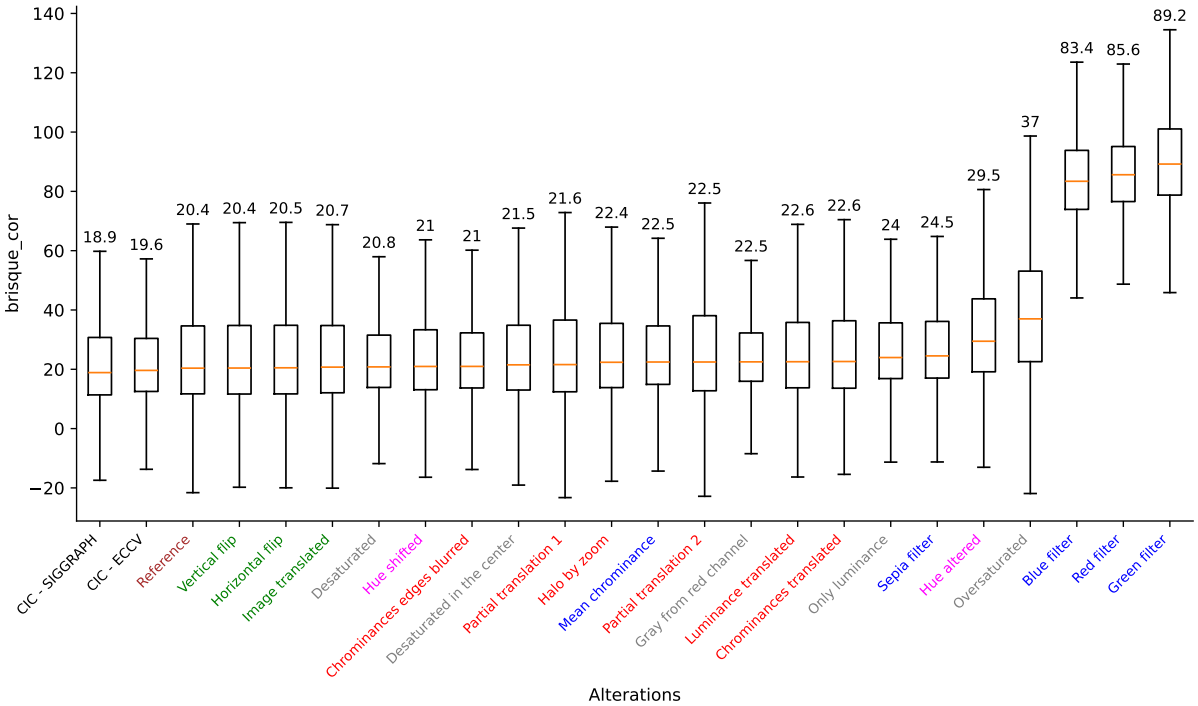


Figure 23: "BRISQUE - Correl" evaluation on the Unsplash dataset. Lower is better.

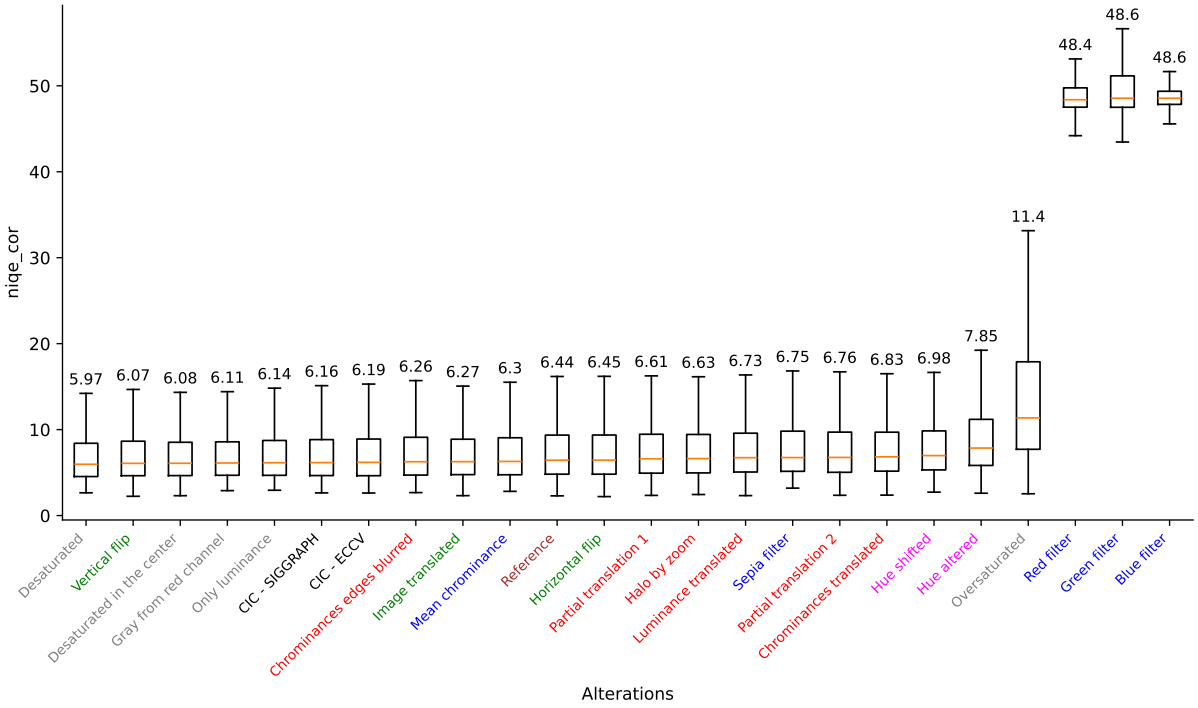


Figure 24: "NIQE - Correl" evaluation on the Unsplash dataset. Lower is better.

MANIQA. This metric assigns similar scores to most alterations, except "Non-semantic colors" (magenta labels) **Hue altered** and "Monochromatic colorizations" (blue labels) **RGB filters**, that are penalized. "Geometric transformations" (green labels) alterations rank first and obtain the highest scores. Nonetheless, critical degradations such as the "Unsatisfactory saturation" (gray labels) desaturations and "Monochromatic colorizations" (blue labels) **Mean chrominance** are evaluated as slightly better than partial color bleedings **Halo by zoom** and **Partial translations** of "Luminance and chrominance inconsistencies" (red labels). So MANIQA is less sensitive than the other metrics to the most critical colorization issues.

Computation times. To estimate the computation time of each metric, we report measurements obtained on a computer equipped with an Intel Xeon W-2295 CPU, 64 GB of RAM, and an NVIDIA RTX A6000 GPU. The script used to run this experiment is in `experiments/compute/exp5_time.py`. The image used is the image EVENT.JPG shown in Figure 2, of resolution 427 × 640. The results for each metric, ordered by increasing computation time, are shown in Table 3.

We observe in Table 3 that metrics based on simple statistical computations, such as Colorfulness and PSNR, are the fastest to compute. Their low complexity and reliance on basic matrix operations contribute to minimal execution times. In the case of "PSNR on a* and b* chrominances", the conversion from RGB to the CIELAB color space introduces a non-negligible overhead, nearly doubling the computation time compared to PSNR in RGB.

Metrics like SSIM and BRISQUE, which involve local neighborhood analysis, remain computationally efficient, with average times around 0.12 seconds. Notably, BRISQUE maintains a high processing speed despite relying on a trained SVR, thanks to the small feature vector and fast inference. NIQE, which uses a similar feature extraction process, incurs longer computation times due to its processing of multiple image patches.

For metrics based on neural networks, computation times increase significantly. LPIPS with AlexNet is relatively fast (0.15 s), but even the smallest configuration of FID (with 64 features) already exceeds 0.5 seconds. Heavier models, such as FID with 2048 features, LPIPS with VGG, and

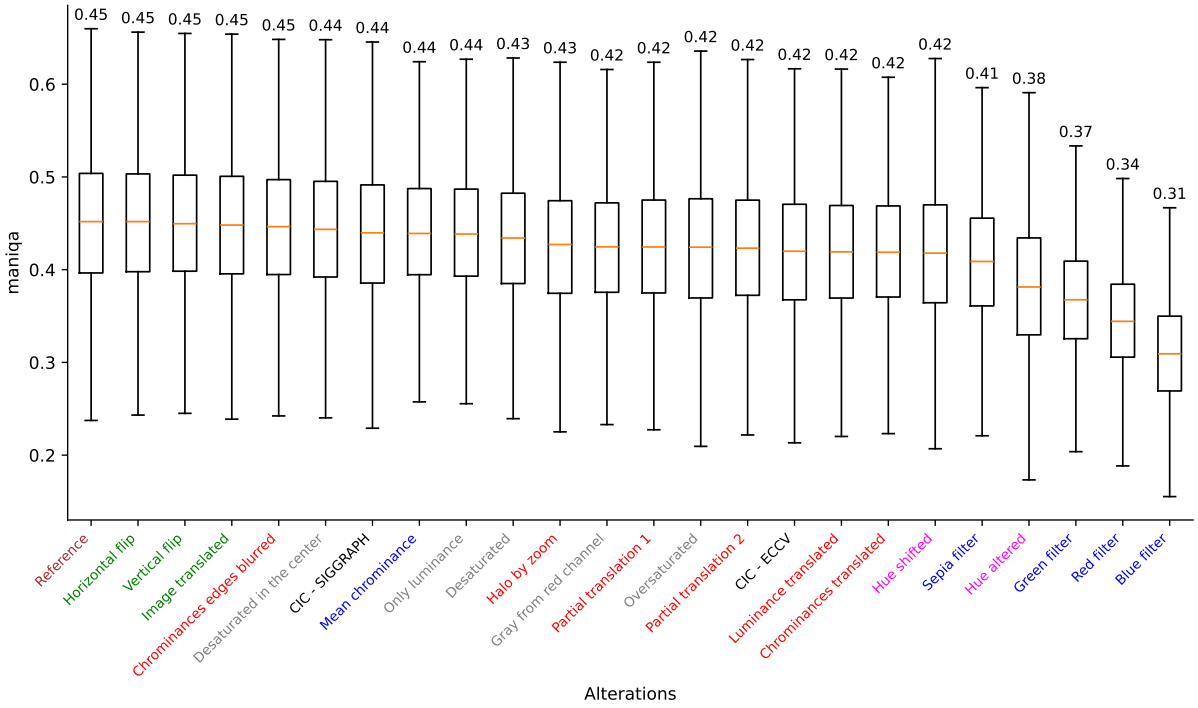


Figure 25: MANIQA evaluation on the Unsplash dataset. Higher is better.

Metric	Mean time (s)	SD (s)
Colorfulness version 1	0.03	2.6e-06
Colorfulness version 2	0.03	1.4e-06
Colorfulness version 3	0.03	5.9e-06
PSNR in RGB color space	0.05	2.3e-06
PSNR on a* and b* chrominances	0.09	3.1e-06
BRISQUE - Original Retrained	0.10	1.9e-05
BRISQUE	0.11	7.2e-06
SSIM	0.13	2.9e-05
LPIPS with an AlexNet network	0.15	2.1e-03
BRISQUE - RGB Correlation	0.29	8.0e-05
NIQE	0.40	8.6e-05
FID with 64 features	0.54	1.9e-03
NIQE - RGB Correlation	0.57	8.6e-04
FID with 192 features	0.61	1.7e-03
FID with 768 features	0.76	4.1e-03
FID with 2048 features	1.39	6.0e-03
LPIPS with a VGG network	2.07	1.2e-02
MANIQA	2.51	1.1e-02

Table 3: Computation times for the image "event" (427 × 640 pixels). For each metric, the computation time was measured 100 times.

MANIQA, which relies on multiple transformer blocks, incur the highest computational costs.

While this is not currently a major limitation, since most state-of-the-art colorization models do not operate in real time, it remains a critical consideration for the design of future evaluation metrics. Avoiding neural networks not only reduces computation time but also minimizes the need for large-scale training, as demonstrated by models like BRISQUE.

3.3 Video Results

For this analysis, we use the `dance-twirl` video from the DAVIS semi-supervised dataset [35]. This sequence consists of 90 frames featuring a dancer in the foreground. The dancer performs rotational movements while moving her arms. The camera follows her motion, leading to motion blur in several frames. Figure 26 shows the 33rd key frame of the sequence. The sequence is



Figure 26: One frame of the `dance-twirl` sequence.

challenging for colorization due to occlusions caused by the dancer passing in front of the audience. These occlusions result in people in the background disappearing and reappearing in subsequent frames. This combination of motion, blur, and occlusions makes the sequence relevant for evaluating colorization algorithms on complex images.

To observe if the studied metrics are sensitive to temporal defects, we propose three types of alterations:

1. Previous alterations such as **Desaturated**, **Sepia filter**, **Hue altered**, **Chrominances translated**, and **Image translated** are applied to a single frame of the sequence. This modification affects only one dimension of the video: the temporal dimension.

2. The same alterations are applied across the entire sequence, but restricted to a few lines located at the same positions in all the frames. This modification affects only the spatial dimension along the height of the sequence. The number of lines per frame is computed as the ceiling of the total number of lines divided by the number of frames, ensuring that the number of altered pixels is comparable to that in the first type of alteration.

3. The alterations are applied every four frames, starting from the first frame, introducing periodic temporal inconsistencies. The goal is to mimic flickering-like effects.

To generate those effects, we use the code in the following path: `experiments/compute/-exp4_video.py`.

Metrics averaged on all frames. Figure 27 shows that, for each type of alteration, the SSIM scores follow the same ranking across the three application patterns: "One frame" achieves the highest scores, followed by "Lines on all frames" and "Every four frames". However, the differences are generally small, except for "Image translated - Every four frames", which obtains a considerably lower score. This is consistent with the previous analysis of SSIM in Section 2.1, which showed that the metric is particularly sensitive to luminance alterations.

From those observations, averaging a metric across all frames is not an effective way to assess temporal consistency. Indeed, there is no significant difference between "One frame" and "Every four frames".

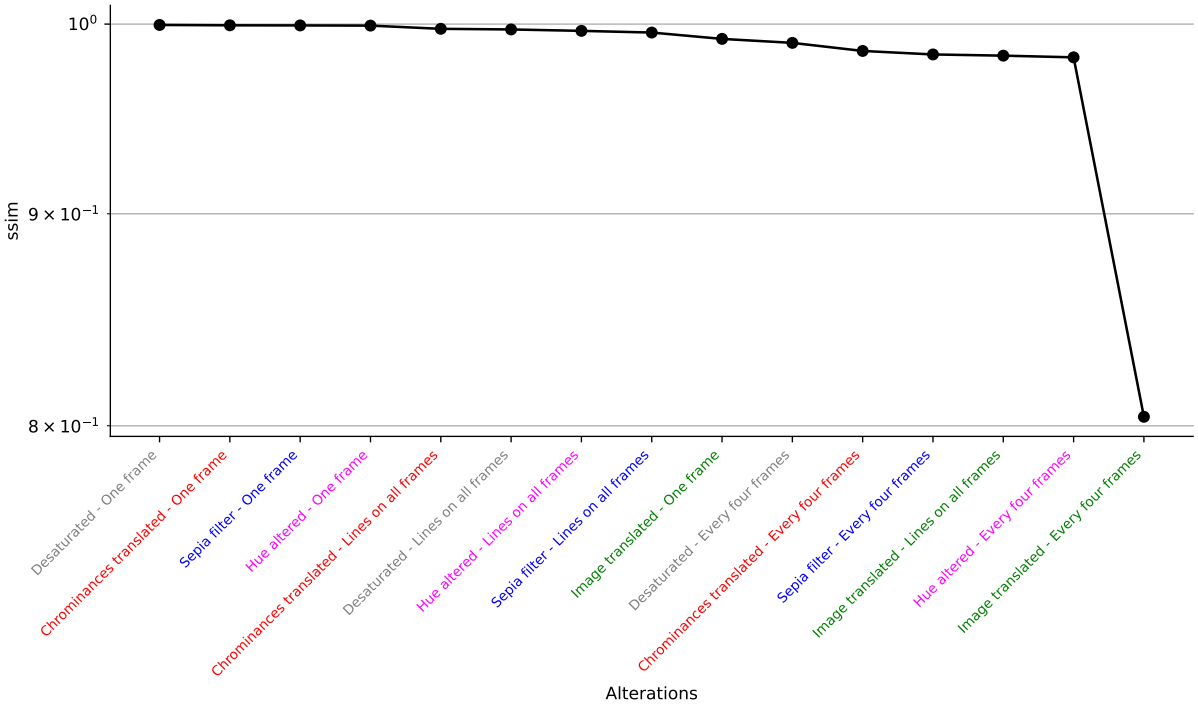


Figure 27: SSIM evaluation on the video. Higher is better.

A similar analysis applies to the Colorfulness measure in Figure 28 where only **Sepia filter - Every four frames** and **Desaturated - Every four frames** are significantly separated from the other alterations.

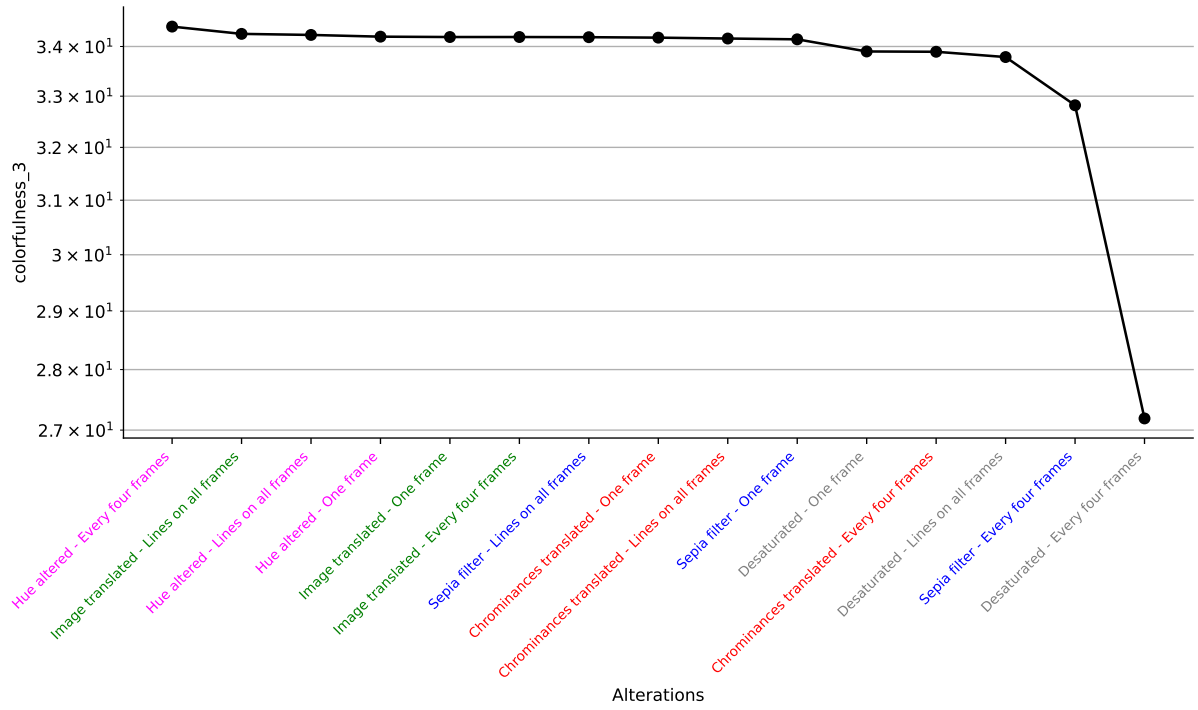


Figure 28: Colorfulness measure third version evaluation on the video. Higher is better.

For LPIPS, the results shown in Figure 29 are similar to those obtained for individual images in Figure 17. However, there is a visible difference in temporal alterations: alterations in the "One

frame" category are less detected than those in the "Lines on all frames" category. This illustrates how temporal averaging can hinder the detection of sparse temporal defects to which the human visual system is very sensitive. Indeed, while SSIM and the colorfulness measure assign comparable contributions to all pixels, LPIPS weights the contribution of individual pixels differently thanks to the w_l terms present in Equation (11). This finally gives a metric that is more sensitive to pixel defects across all frames than to defects on only some frames, since the alterations "One frame" and "Lines on all frames" share an equivalent number of altered pixels on the video sequence.

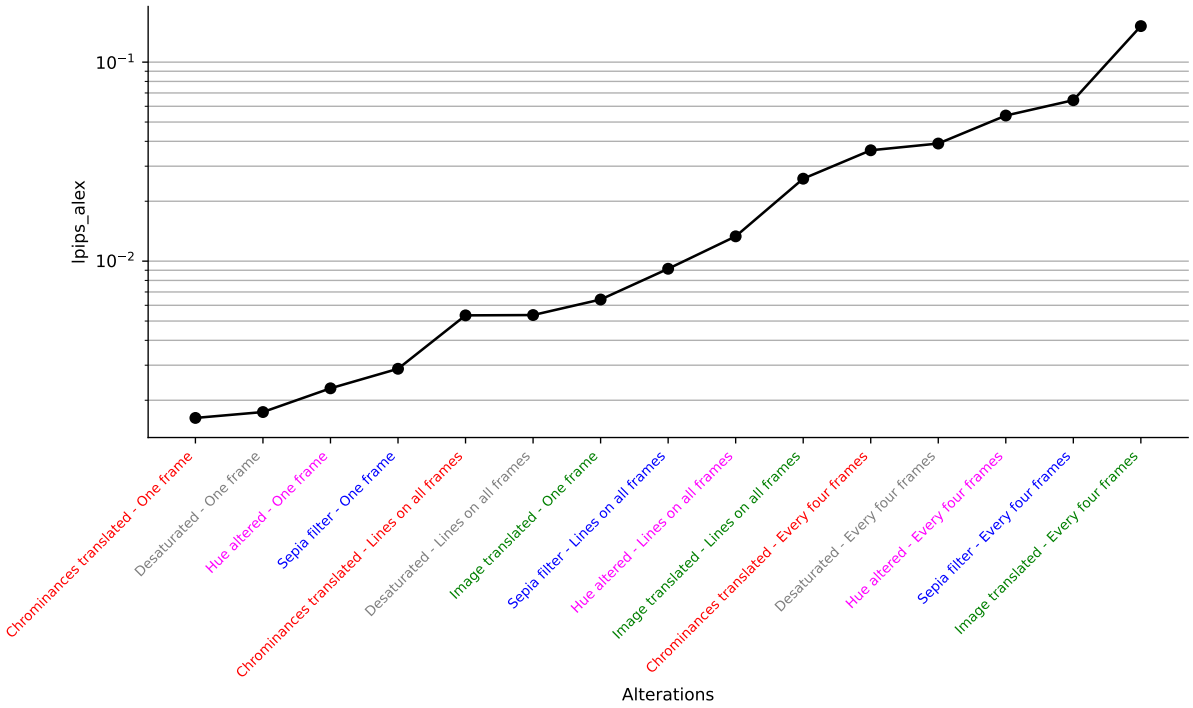


Figure 29: LPIPS with AlexNet evaluation on the video. Lower is better.

Since PSNR involves division by the MSE between altered and ground-truth images, and our two temporal alterations, "One frame" and "Every four frames", do not modify all frames, division by zero occurs, making it impossible to compare our temporal alterations.

FID. Unlike other image metrics, FID is not calculated by averaging metric scores across all frames but is designed to estimate the quality of all frames collectively. Furthermore, it does not rely solely on the mean of the extracted features, as it also uses their covariance to compute the final score. Regarding temporal alterations, FID produces a similar ranking to the previously discussed metrics (Figure 30): "One frame" ranks first, followed by "Lines on all frames" and finally "Every four frames". However, the alteration **Image translated - Lines on all frames** is the alteration with the worst score in this graph. This shows that FID is more sensitive to spatial defects than to temporal defects. Indeed, while **Image translated** applied to one image is a geometric transformation that does not alter the quality of the colorization, applying it on a few frames creates a temporal inconsistency by introducing undesired displacements. However, its application to a few lines on all frames in **Image translated - Lines on all frames** also creates semantic alterations. Also, **Chrominances translated - Lines on all frames** achieves a better score than **Chrominances translated - Every four frames**. This indicates that FID is equally sensitive to temporal inconsistencies affecting luminance and chrominance but is more sensitive to spatial displacements of luminance than to those of chrominance. Nonetheless, colorization, as we define it, only adds

chrominances and does not alter luminance, meaning that FID is sufficiently sensitive to temporal coherence to be used on videos, despite treating the video as an unordered set of images.

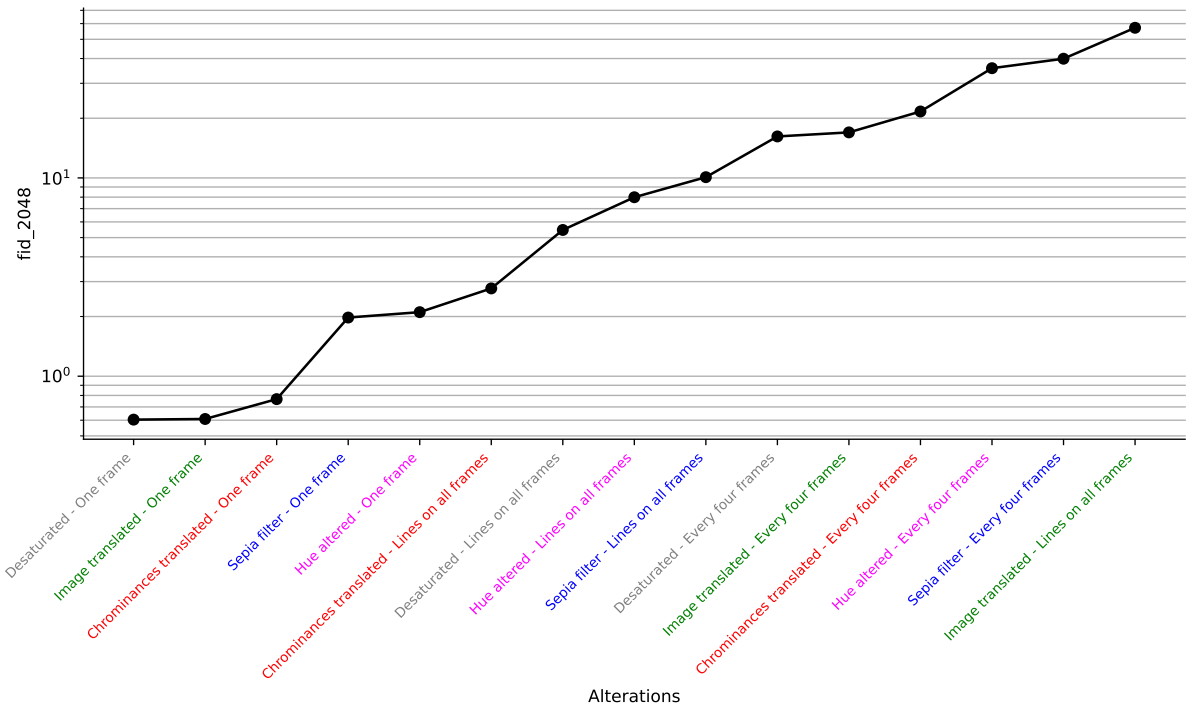


Figure 30: FID with 2048 features evaluation on the video. Lower is better.

CDC. In Figure 31, the ordering of temporal defects differs from that in the previous plots. Indeed, a small CDC index means the video is temporally consistent, and the order obtained is "Lines on all frames" first, followed by "One frame" and finally "Every four frames". While the first group of spatial alterations has similar scores, the "One frame" category exhibits an increasing trend in scores, indicating that CDC not only considers the number of altered pixels but also evaluates temporal consistency. This is visible with the **Sepia filter - One frame** that obtains a similar score to **Image translated - Every four frames**. The high score obtained by such temporal alterations shows that CDC can detect even sparse temporal alterations. However, CDC is not sensitive to color-bleeding alterations. This is confirmed by the scores of **Chrominance translated** which are higher than those of **Image translated** for the same temporal alteration. Nonetheless, CDC is highly sensitive to temporal color inconsistencies as **Hue altered**, **Desaturated** and **Sepia filter** obtain the highest scores.

4 Conclusion

In this study, we implemented a comprehensive metric evaluation framework that can be integrated into colorization pipelines to evaluate colorization results during training or assess several colorizations afterward. This framework enabled us to conduct an extensive study on how commonly used colorization metrics—PSNR, SSIM, LPIPS, FID, Colorfulness measure, BRISQUE, NIQE, MANIQA, and CDC—respond to classical artifacts produced by colorization algorithms, such as color bleeding, desaturation, or incorrect hues. Our software library generates such artifacts and applies them to a dataset of 25,000 images.

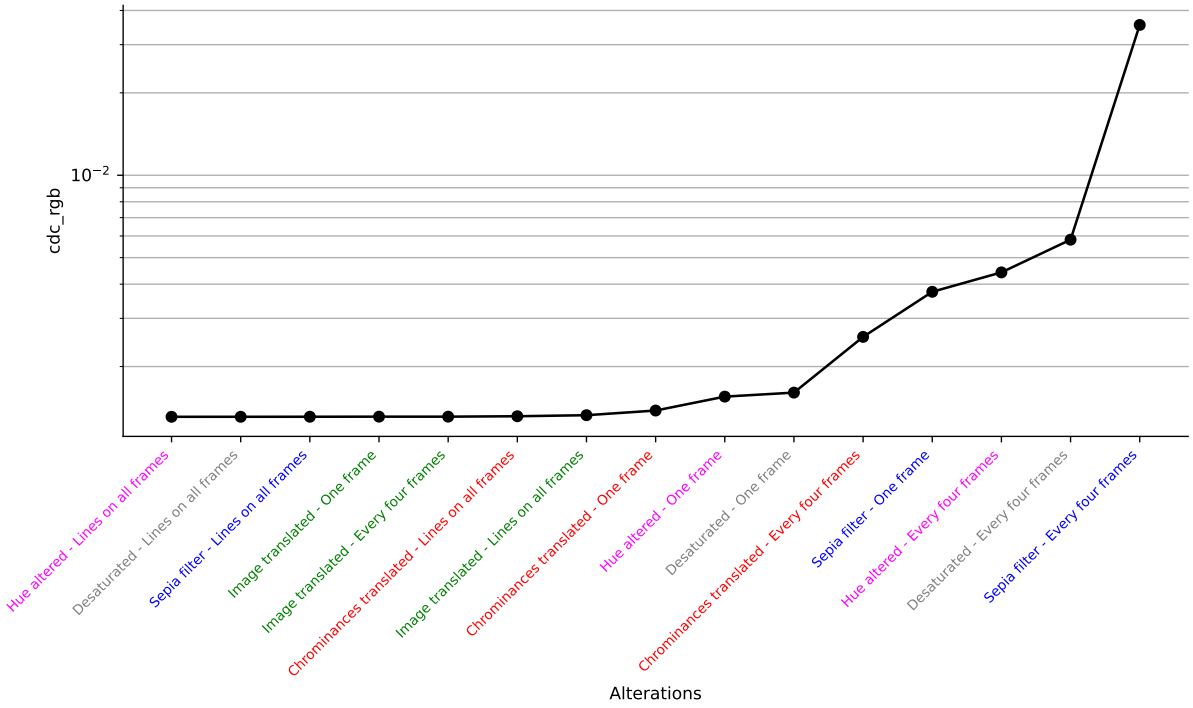


Figure 31: CDC evaluation in RGB space on the video. Lower is better.

Our findings indicate that FID is the most effective metric for determining whether the evaluated colorization exhibits color bleeding or sufficient saturation. Although we adapted BRISQUE and NIQE to operate on color images, their performance remains limited, highlighting the need for a large image dataset annotated with human perceptual scores.

Additionally, we examined these metrics in the context of colorized video evaluation, revealing that they are inadequate for detecting sparse temporal inconsistencies. The CDC score, specifically designed to identify such defects, proved to be the most sensitive metric in this setting. This highlights the importance of distinguishing between still image and video colorization, as temporal coherence becomes critical only in the latter. Moreover, historical grayscale images may exhibit different types of degradation or semantic ambiguity, which can influence metric performance and interpretation. It is worth noting that a large dataset of colorized legacy videos is not available yet. This severely hinders both the ability to develop new metrics tailored to colorization and the training of more effective neural networks for colorizing images and videos. Indeed, until such a dataset is created — in consultation with colorization professionals — it will remain difficult to develop efficient colorization methods that align with the artistic and/or historical choices of experts and whose results are quantitatively verified.

Regarding computation times, metrics relying on neural networks tend to be significantly more computationally intensive than those based on statistical computations. This can hinder their application to full-length movie colorization or their integration as loss terms during neural network training.

To accurately compare colorization algorithms, the optimal approach is to combine FID and CDC evaluations. However, none of the studied metrics effectively detect color bleeding artifacts. Future research should focus on developing a metric specifically tailored to identify and localize such defects, and on constructing an annotated colorization dataset to support its evaluation. For this latter task, good practices are provided in Section 6 of [4].

Acknowledgments

This research was funded, in whole or in part, by l'Agence Nationale de la Recherche (ANR), project ANR-21-CE23-0008. For the purpose of open access, the authors have applied a CC-BY public copyright license to any Author Accepted Manuscript (AAM) version arising from this submission.

Image Credits

 Fabien Pierre, CC-BY

 Mircea Iancu, Pixabay[9]

 DAVIS [35] authors, CC-BY

References

[1] B. Arbelot, R. Vergne, T. Hurtut, and J. Thollot, *Local Texture-Based Color Transfer and Colorization*, Computers & Graphics, 62 (2017), pp. 15–27, https://doi.org/10.1016/j.cag.2016.12.005.

[2] J. Carreira and A. Zisserman, *Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4724–4733, https://doi.org/10.1109/CVPR.2017.502.

[3] S. Chen, X. Li, X. Zhang, M. Wang, Y. Zhang, J. Han, and Y. Zhang, *Exemplar-Based Video Colorization with Long-Term Spatiotemporal Dependency*, ArXiv:2303.15081, (2023), https://doi.org/10.48550/arXiv.2303.15081.

[4] S.-Y. Chen, J.-Q. Zhang, Y.-Y. Zhao, P. L. Rosin, Y.-K. Lai, and L. Gao, *A Review of Image and Video Colorization: From Analogies to Deep Learning*, Visual Informatics, 6 (2022), pp. 51–68, https://doi.org/10.1016/j.visinf.2022.05.003.

[5] J. Davis, Y.-H. Hsieh, and H.-C. Lee, *Humans Perceive Flicker Artifacts at 500 Hz*, Scientific Reports, 5 (2015), p. 7861, https://doi.org/10.1038/srep07861.

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, *An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale*, in International Conference on Learning Representations (ICLR), 2021, https://doi.org/10.48550/arXiv.2010.11929.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative Adversarial Networks*, Communications of the ACM, 63 (2020), pp. 139–144, https://doi.org/10.1145/3422622.

[8] D. Hasler and S. Süsstrunk, *Measuring Colorfulness in Natural Images*, in IS&T/SPIE Electronic Imaging, Human Vision and Electronic Imaging VIII, vol. 5007, 2003, pp. 87–95, https://doi.org/10.1117/12.477378.

[9] https://pixabay.com/photos/people-ladies-girls-colors-flags-8168554/

[9] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan, *Deep Exemplar-Based Colorization*, ACM Transactions on Graphics, 37 (2018), pp. 47:1–47:16, https://doi.org/10.1145/3197517.3201365.

[10] 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 International Conference on Neural Information Processing Systems (NIPS), vol. 30, 2017, pp. 6629–6640, https://doi.org/10.48550/arXiv.1706.08500.

[11] M. Hofinger, E. Kobler, A. Effland, and T. Pock, *Learned Variational Video Color Propagation*, in European Conference on Computer Vision (ECCV), 2022, pp. 512–530, https://doi.org/10.1007/978-3-031-20050-2_30.

[12] J. D. Hunter, *Matplotlib: A 2D Graphics Environment*, Computing in Science & Engineering, 9 (2007), pp. 90–95, https://doi.org/10.1109/MCSE.2007.55.

[13] Q. Huynh-Thu and M. Ghanbari, *The Accuracy of PSNR in Predicting Video Quality for Different Video Scenes and Frame Rates*, Telecommunication Systems, 49 (2012), pp. 35–48, https://doi.org/10.1007/s11235-010-9351-x.

[14] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, *SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5MB Model Size*, ArXiv:1602.07360, (2016), https://doi.org/10.48550/arXiv.1602.07360.

[15] S. Iizuka and E. Simo-Serra, *DeepRemaster: Temporal Source-Reference Attention Networks for Comprehensive Video Enhancement*, ACM Transactions on Graphics, 38 (2019), pp. 176:1–176:13, https://doi.org/10.1145/3355089.3356570.

[16] X. Ji, B. Jiang, D. Luo, G. Tao, W. Chu, Z. Xie, C. Wang, and Y. Tai, *ColorFormer: Image Colorization Via Color Memory Assisted Hybrid-Attention Transformer*, in European Conference on Computer Vision (ECCV), vol. 13676, 2022, pp. 20–36, https://doi.org/10.1007/978-3-031-19787-1_2.

[17] G. Jinjin, C. Haoming, C. Haoyu, Y. Xiaoxing, J. S. Ren, and D. Chao, *PIPAL: A Large-Scale Image Quality Assessment Dataset for Perceptual Image Restoration*, in European Conference on Computer Vision (ECCV), 2020, pp. 633–651, https://doi.org/10.1007/978-3-030-58621-8_37.

[18] X. Kang, X. Lin, K. Zhang, Z. Hui, W. Xiang, J.-Y. He, X. Li, P. Ren, X. Xie, R. Timofte, Y. Yang, J. Pan, Z. Peng, Q. Zhang, J. Dong, J. Tang, J. Li, C. Lin, Q. Li, Q. Liang, R. Gang, X. Liu, S. Feng, S. Liu, H. Wang, C. Feng, F. Bai, Y. Zhang, G. Shao, X. Wang, L. Lei, S. Chen, Y. Zhang, H. Xu, Z. Liu, Z. Zhang, Y. Luo, and Z. Zuo, *NTIRE Video Colorization Challenge*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2023, pp. 1570–1581, https://doi.org/10.1109/CVPRW59228.2023.00159.

[19] E. Kim, S. Lee, J. Park, S. Choi, C. Seo, and J. Choo, *Deep Edge-Aware Interactive Colorization Against Color-Bleeding Effects*, in IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 14647–14656, https://doi.org/10.1109/ICCV48922.2021.01440.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, Communications of the ACM, 60 (2017), pp. 84–90, https://doi.org/10.1145/3065386.

[21] M. Kumar, D. Weissenborn, and N. Kalchbrenner, *Colorization Transformer*, in International Conference on Learning Representations (ICLR), 2021, https://doi.org/10.48550/arXiv.2102.04432.

[22] G. Larsson, M. Maire, and G. Shakhnarovich, *Learning Representations for Automatic Colorization*, in European Conference on Computer Vision (ECCV), vol. 9908, 2016, pp. 577–593, https://doi.org/10.1007/978-3-319-46493-0_35.

[23] C. Lei and Q. Chen, *Fully Automatic Video Colorization With Self-Regularization and Diversity*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3748–3756, https://doi.org/10.1109/CVPR.2019.00387.

[24] A. Levin, D. Lischinski, and Y. Weiss, *Colorization Using Optimization*, ACM Transactions on Graphics, 23 (2004), pp. 689–694, https://doi.org/10.1145/1015706.1015780.

[25] J. Li, Q. Liang, Q. Li, R. Gang, J. Fang, C. Lin, S. Feng, and X. Liu, *RTTLC: Video Colorization with Restored Transformer and Test-Time Local Converter*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2023, pp. 1722–1730, https://doi.org/10.1109/CVPRW59228.2023.00173.

[26] S. Liu, *Two Decades of Colorization and Decolorization for Images and Videos*, ArXiv:2204.13322, (2022), https://doi.org/10.48550/arXiv.2204.13322.

[27] Y. Liu, H. Zhao, K. C. K. Chan, X. Wang, C. C. Loy, Y. Qiao, and C. Dong, *Temporally Consistent Video Colorization with Deep Feature Propagation and Self-Regularization Learning*, Computational Visual Media, 10 (2024), pp. 375–395, https://doi.org/10.1007/s41095-023-0342-8.

[28] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, *Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows*, in IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 9992–10002, https://doi.org/10.1109/ICCV48922.2021.00986.

[29] Z. Liu, P. Mu, H. Xu, and C. Bai, *Histogram-Guided Video Colorization Structure with Spatial-Temporal Connection*, in IEEE International Conference on Multimedia and Expo (ICME), 2023, pp. 1625–1630, https://doi.org/10.1109/ICME55011.2023.00280.

[30] N. Maignan, F. Pierre, and F. Sur, *Analyse de Métriques de Qualité Pour Des Images En Couleurs*, in ORASIS, 2025. https://hal.science/hal-05131221/document.

[31] S. Meyer, V. Cornillère, A. Djelouah, C. Schroers, and M. Gross, *Deep Video Color Propagation*, in British Machine Vision Conference (BMVC), 2018, https://doi.org/10.3929/ethz-b-000319608.

[32] A. Mittal, A. K. Moorthy, and A. C. Bovik, *No-Reference Image Quality Assessment in the Spatial Domain*, IEEE Transactions on Image Processing, 21 (2012), pp. 4695–4708, https://doi.org/10.1109/TIP.2012.2214050.

[33] A. Mittal, R. Soundararajan, and A. C. Bovik, *Making a "Completely Blind" Image Quality Analyzer*, IEEE Signal Processing Letters, 20 (2013), pp. 209–212, https://doi.org/10.1109/LSP.2012.2227726.

[34] K. T. Mullen and W. H. A. Beaudot, *Comparison of Color and Luminance Vision on a Global Shape Discrimination Task*, Vision Research, 42 (2002), pp. 565–575, https://doi.org/10.1016/S0042-6989(01)00305-4.

[35] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. V. Gool, *The 2017 DAVIS Challenge on Video Object Segmentation*, ArXiv:1704.00675, (2018), https://doi.org/10.48550/arXiv.1704.00675.

[36] A. R. Robertson, *The CIE 1976 Color-Difference Formulae*, Color Research & Application, 2 (1977), pp. 7–11, https://doi.org/10.1002/j.1520-6378.1977.tb00104.x.

[37] 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, 115 (2015), pp. 211–252, https://doi.org/10.1007/s11263-015-0816-y.

[38] S. Salmon and J. Malthête, *ELisabeth and Berthe Thuillier*, (2020), https://doi.org/10.7916/d8-734m-kr16.

[39] A. Salmona, L. Bouza, and J. Delon, *DeOldify: A Review and Implementation of an Automatic Colorization Method*, Image Processing On Line, 12 (2022), pp. 347–368, https://doi.org/10.5201/ipol.2022.403.

[40] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, *A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms*, IEEE Transactions on Image Processing, 15 (2006), pp. 3440–3451, https://doi.org/10.1109/TIP.2006.881959.

[41] K. Simonyan and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, ArXiv:1409.1556, (2015), https://doi.org/10.48550/arXiv.1409.1556.

[42] A. R. Smith, *Color Gamut Transform Pairs*, SIGGRAPH Computer Graphics, 12 (1978), pp. 12–19, https://doi.org/10.1145/965139.807361.

[43] J.-W. Su, H.-K. Chu, and J.-B. Huang, *Instance-Aware Image Colorization*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 7965–7974, https://doi.org/10.1109/CVPR42600.2020.00799.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, *Rethinking the Inception Architecture for Computer Vision*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818–2826, https://doi.org/10.1109/CVPR.2016.308.

[45] B. Turquety, *Le Naturel et le Mécanique : le Kinémacolor À la Conquête de Paris, Ou Charles Urban Vs Charles Pathé*, 1895. Mille Huit Cent Quatre-Vingt-Quinze. Revue de L'association Française de Recherche Sur L'histoire du Cinéma, (2013), pp. 81–105, https://doi.org/10.4000/1895.4772.

[46] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, *FVD: A New Metric for Video Generation*, in Deep Generative Models for Highly Structured Data, ICLR Workshop, 2019, https://doi.org/10.48550/arXiv.1812.01717.

[47] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, *Scikit-Image: Image Processing in Python*, PeerJ, 2 (2014), p. e453, https://doi.org/10.7717/peerj.453.

[48] P. Vitoria, L. Raad, and C. Ballester, *ChromaGAN: Adversarial Picture Colorization with Semantic Class Distribution*, in IEEE Winter Conference on Applications of Computer Vision (WACV), 2020, pp. 2434–2443, https://doi.org/10.1109/WACV45572.2020.9093389.

[49] Z. Wan, B. Zhang, D. Chen, and J. Liao, *Bringing Old Films Back to Life*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 17673–17682, https://doi.org/10.1109/CVPR52688.2022.01717.

[50] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image Quality Assessment: From Error Visibility to Structural Similarity*, IEEE Transactions on Image Processing, 13 (2004), pp. 600–612, https://doi.org/10.1109/TIP.2003.819861.

[51] R. Ward, D. Bigioi, S. Basak, J. G. Breslin, and P. Corcoran, *LatentColorization: Latent Diffusion-Based Speaker Video Colorization*, IEEE Access, 12 (2024), pp. 81105–81121, https://doi.org/10.1109/ACCESS.2024.3406249.

[52] S. Weng, J. Sun, Y. Li, S. Li, and B. Shi, *CT^2: Colorization Transformer Via Color Tokens*, in European Conference on Computer Vision (ECCV), 2022, pp. 1–16, https://doi.org/10.1007/978-3-031-20071-7_1.

[53] M. Xia, W. Hu, T.-T. Wong, and J. Wang, *Disentangled Image Colorization Via Global Anchors*, ACM Transactions on Graphics, 41 (2022), pp. 204:1–204:13, https://doi.org/10.1145/3550454.3555432.

[54] S. Yang, T. Wu, S. Shi, S. Lao, Y. Gong, M. Cao, J. Wang, and Y. Yang, *MANIQA: Multi-Dimension Attention Network for No-Reference Image Quality Assessment*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022, pp. 1190–1199, https://doi.org/10.1109/CVPRW56347.2022.00126.

[55] Y. Yang, Z. Peng, X. Du, Z. Tao, J. Tang, and J. Pan, *BiSTNet: Semantic Image Prior Guided Bidirectional Temporal Feature Fusion for Deep Exemplar-Based Video Colorization*, 2022, https://doi.org/10.48550/arXiv.2212.02268.

[56] L. Yatziv and G. Sapiro, *Fast Image and Video Colorization Using Chrominance Blending*, IEEE Transactions on Image Processing, 15 (2006), pp. 1120–1129, https://doi.org/10.1109/TIP.2005.864231.

[57] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, *Restormer: Efficient Transformer for High-Resolution Image Restoration*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 5718–5729, https://doi.org/10.1109/CVPR52688.2022.00564.

[58] Y. Zhai, Z. Tao, L. Dai, H. Wang, X. Huang, and L. Yang, *Flow-Guided Transformer for Video Colorization*, in IEEE International Conference on Image Processing (ICIP), 2023, pp. 2485–2489, https://doi.org/10.1109/ICIP49359.2023.10223177.

[59] B. Zhang, M. He, J. Liao, P. V. Sander, L. Yuan, A. Bermak, and D. Chen, *Deep Exemplar-Based Video Colorization*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 8044–8053, https://doi.org/10.1109/CVPR.2019.00824.

[60] R. Zhang, P. Isola, and A. A. Efros, *Colorful Image Colorization*, in European Conference on Computer Vision (ECCV), 2016, pp. 649–666, https://doi.org/10.1007/978-3-319-46487-9_40.

[61] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 586–595, https://doi.org/10.1109/CVPR.2018.00068.

[62] ——, *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*, ArXiv:1801.03924, (2018), https://doi.org/10.48550/arXiv.1801.03924.

[63] 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*, ACM Transactions on Graphics, 36 (2017), pp. 119:1–119:11, https://doi.org/10.1145/3072959.3073703.

[64] Y. Zhang, S. Chen, M. Wang, X. Zhang, C. Zhu, Y. Zhang, and X. Li, *Temporal Consistent Automatic Video Colorization Via Semantic Correspondence*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2023, pp. 1836–1845, https://doi.org/10.1109/CVPRW59228.2023.00182.