IPOL IMAGE PROCESSING ON LINE

Published in Image Processing On Line on 2024–05–22.
Submitted on 2024–04–10, accepted on 2024–05–03.
ISSN 2105–1232 © 2024 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.2024.539

A Brief Analysis of iColoriT for Interactive Image Colorization

Rosana García[1], Gregory Randall[1], Lara Raad[1,2]

[1]IIE, Facultad de Ingeniería, Universidad de la República, Uruguay
[2]LIGM, ESIEE Paris, Université Gustave Eiffel, France

Communicated by Pablo Musé *Demo edited by* Lara Raad and Rosana García

Abstract

This paper briefly describes and analyzes iColoriT, a hybrid colorization method based on a Vision Transformer that propagates user hints to relevant regions of a grayscale image while using color priors learned from a large image dataset. This approach gives users more control over color inference and shows a quick way to achieve results.

Source Code

The source code and documentation for this algorithm are available from the web page of this article[1]. Usage instructions are included in the `README` file of the archive. The authors' original method implementation is available here[2].
This is an MLBriefs article. The source code has not been reviewed!

Keywords: colorization; interactive; ViT

1 Introduction

Image colorization involves recovering a color image from a grayscale image. This process is attracting a lot of attention in the image-editing community as a way of restoring or colorizing old films or grayscale images. While transforming a color image into a grayscale one is a standard procedure, the reverse operation is a highly ill-posed problem, as no information about the color to be added is known. Color priors must, therefore, be taken into account. In the literature, three prior types lead to different colorization methods. Several surveys [7, 1, 3] provide a detailed overview of these approaches. In this work, we will focus on iColoriT: Towards Propagating Local Hint to the Right Region in Interactive Colorization by Leveraging Vision Transformer [14], one of the few hybrid approaches combining both color priors learned from a large dataset of images and color hints indicated by users. This method provides greater control over the colorized output while exploiting the color priors of a specific database.

[1]https://doi.org/10.5201/ipol.2024.539
[2]https://github.com/pmh9960/iColoriT

2 iColoriT

iColoriT colorizes grayscale images by training a Vision Transformer [5] capable of propagating the user color hints to relevant regions of the grayscale image. iColoriT comprises three modules: a transformer encoder in charge of propagating the color hints, a pixel shuffling module to upscale the result, and a local stabilizing layer responsible for smoothing the transitions between patches, which can be observed after the colorized patches have been reshaped in the upscaling stage, as shown in Figure 1. The inputs to the colorization network are a color image X_{rgb} of H rows, W columns, and 3 channels as well as a list of (x, y) positions indicating the location of the color hints. These two inputs are then transformed into a 3-channel image of H rows and W columns that will be fed to the colorization network. The transformation process is shown in Figure 2 and is as follows. The color image X_{rgb} is first resized to $H' \times W' \times 3$ pixels, where $H' = 224$ and $W' = 224$, and converted from RGB to CIELab, where the luminance L is taken as the grayscale image to colorize. Then, the hint positions in the input list are normalized to indicate the position relative to an image of size 224×224. These positions are then converted into a two-channel image of $224 \times 224 \times 2$ pixels, representing the chrominance channels a_h and b_h of the hints at the specified positions. For each hint, the chrominance channels (a_h, b_h) are obtained by averaging the corresponding (a, b) values of the input color image X_{rgb} at the hint locations. Note that color hints are generally larger than one pixel and that the hint size is specified as a parameter of the method. Finally, the input X to the network is the concatenation of the luminance L of the input color image and the chrominance channels a_h and b_h of the hint image, $X = L \oplus (a_h, b_h)$.

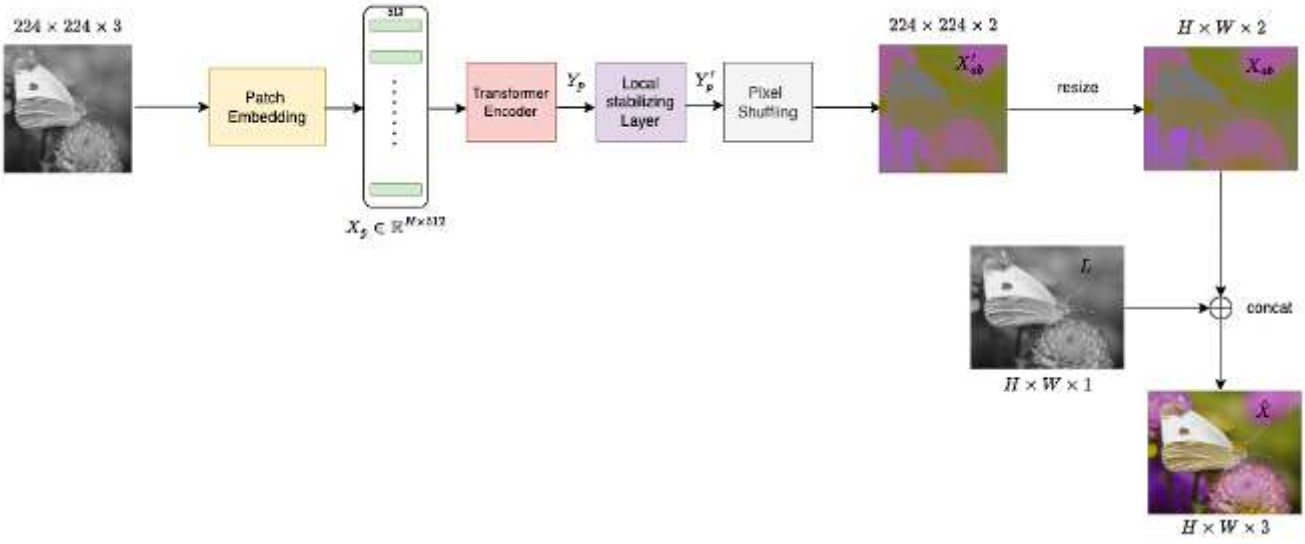


Figure 1: Overview of iColoriT. A patch embedding layer transforms the input X into X_p, a sequence of N tokens. X_p is first passed to the encoder transform, which outputs Y_p, which is then passed to the local stabilization layer, which outputs Y'_p, which is finally passed to the pixel shuffling layer, which outputs X'_{ab}. The output X'_{ab} is then resized to an image of size $H \times W \times 2$, yielding X_{ab}, which is concatenated with the luminance L of the ground truth color image X_{rgb}.

Transformer encoder. The input X is first embedded into X_p, a sequence of N tokens of dimension $d = P \times P \times C$, where $N = H'W'/P^2$ is the number of tokens, P is the size of the patch side and $C = 2$. This sequence is obtained by passing X in a patch embedding layer composed of one convolutional layer of d kernels of size $P \times P$ and a stride of size P. The resulting sequence of tokens X_p is the input to the encoder transformer, and its output is Y_p, of the same shape as X_p. The different stages of the transformer encoder are as follows. First, a sinusoidal positional encoding E_p [5] is added to

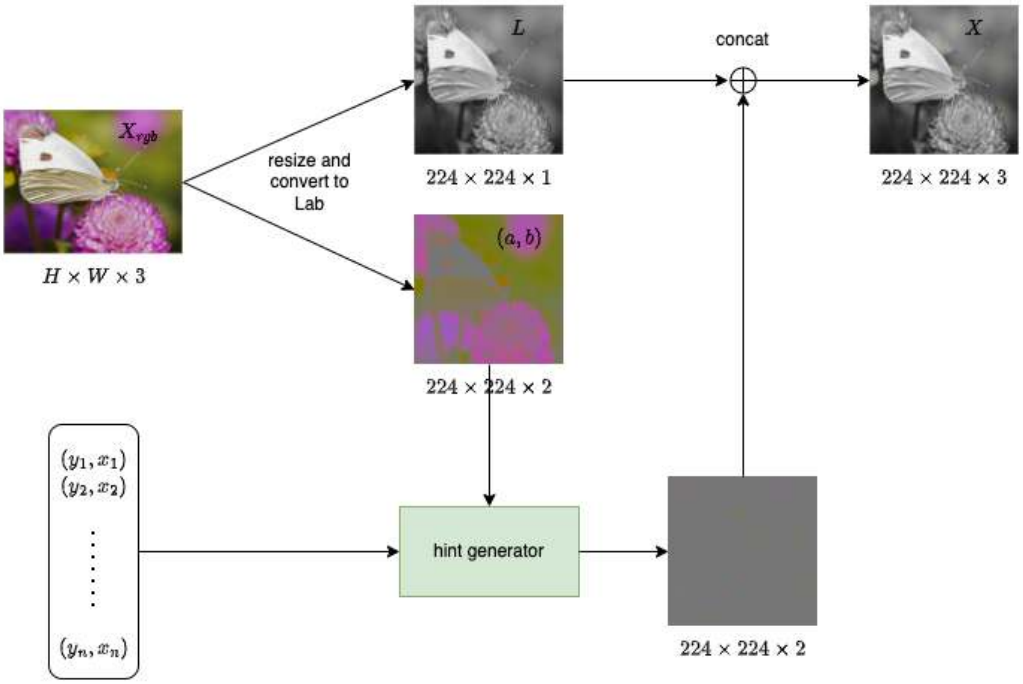


Figure 2: Process of converting the color input image and the list of position hints into the three-channel input image. The ground truth image X_{rgb} is resized to $224 \times 224 \times 3$ and transformed into the CIELab color space. A list of coordinates $\{(y_1, x_1), \ldots, (y_n, x_n)\}$, with n the number of hints provided, and the resized chrominance channels (a, b) are passed to the hint generator module, resulting in a 2-channel hint mask. The hint generator rescales the coordinates relative to an image of size 224×224. The resized luminance L is concatenated with the hint mask, resulting in the 3-channel input X.

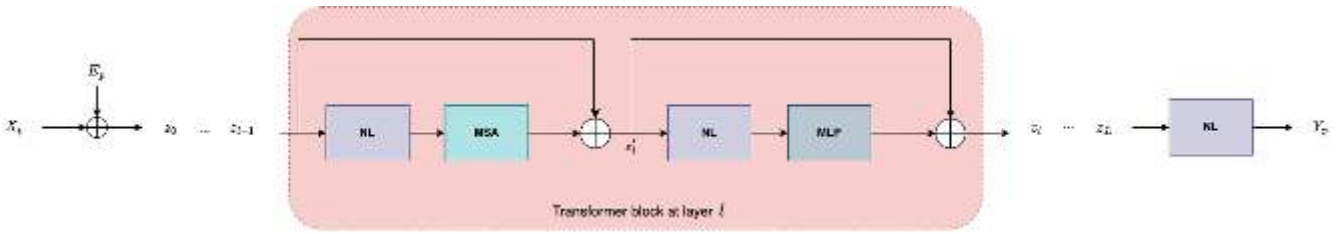


Figure 3: Encoder transformer block at layer l. The intermediate result $z_l^{'}$ is the concatenation of the output of layer l-1 and the same signal processed by a normalization layer NL and a Multi-headed self-attention layer MSA. A second step normalizes $z_l^{'}$ and passes it through an MLP that comprises one hidden layer to obtain z_l.

the input X_p yielding

$$z_0 = X_p + E_p.$$

Then follows a cascade of L multi-headed transformer blocks (see Figure 3) computed as

$$z_l' = MSA(NL(z_{l-1})) + z_{l-1},$$

and

$$z_l = MLP(NL(z_l')) + z_l'.$$

The last stage yields the output of the transformer

$$Y_p = NL(z_L).$$

Here $MSA(.)$ denotes the multi-headed self attention layer [12] and $NL(.)$ denotes the normalization layer [2]. The MSA layer is computed as,

$$Attention(Q, K, V) = softmax(QK^T/\sqrt{d} + B)V,$$

followed by a normalization layer. $Q, K, V \in \mathbb{R}^{N\times d}$ are the query, key, and value matrices, and $B \in \mathbb{R}^{N\times N}$ is the relative positional bias added in addition to the positional encoding to contemplate the fact that self-attention does not use position-related information. The NL is the LayerNorm function of `torch.nn` module. The MLP block comprises one hidden layer of $4 \times d$ dimensions and a GELU activation layer.

Pixel Shuffling. The pixel shuffling layer reshapes an image of size $H' \times W' \times (P^2 \times C)$ into an image of size $(H' \times P) \times (W' \times P) \times C$. The output Y_p of the transformer can be viewed as an image of size $(H'/P) \times (W'/P) \times (P^2 \times 2)$. Then, by passing Y_p through the pixel shuffling stage, every channel in Y_p is resized to a patch of size $P \times P \times 2$, yielding an image of size $H' \times W' \times 2$.

Local Stabilizing Layer. In [14], the authors point out that, for a patch side size P greater than 8, the resulting images, after applying pixel shuffling, have visible artifacts along the image patch boundaries. To avoid this, a stabilization layer is used before the pixel shuffling stage.

This stabilization layer consists of a convolution layer with d filters of size 3×3 and a stride of size 1. The input to the stabilizing layer is Y_p of size $H'/P \times W'/P \times d$, and the output is Y'_p of the same size. Y'_p is then passed to the pixel shuffling stage, which produces the chrominance image X'_{ab} of size $H' \times W' \times 2$, which is finally resized to an image of size $H \times W \times 2$ and labeled X_{ab}.

Predicted color image. The predicted color image $\hat{X}$ is the result of concatenating the input luminance channel $L \in \mathbb{R}^{H\times W\times 1}$ and the two predicted chrominance channels $X_{ab} \in \mathbb{R}^{H\times W\times 2}$ obtained after the pixel shuffling step and resizing step.

3 Training

To train the iColoriT network, the Huber loss between the predicted image $\hat{X}$ and the ground truth color image X in the CIELab color space is used. The network was trained on the ImageNet 2012 train split images [11]. The image set consists of 1281167 images. During training, the images were resized to 224×224. The patch size was set to $P = 16$; hence $N = H'/P \times W'/P = 196$ and $d = 512$. The number of transformer blocks was set to $L = 12$, and the number of heads per transformer block was set to $H_T = 12$. In training, the hints were generated by randomly selecting the position and number of hints and assigning the average color from the ground truth color image to each hint. Specifically, the number of hints was randomly chosen between 0 and 128 for each batch of each epoch. The AdamW [9] optimizer was used with a cosine annealing learning rate schedule [8].

4 Experiments

The purpose of our experiments was to verify some of the performance results presented by the authors in [14] by comparing the Base and Small models, the number and location of hints, and how much the size of the hint affects the results. After finding that the Base model and a hint size of 2 were the best choices, we conducted several experiments with different datasets. We colorized images from ImageNet1k [11], a subset with 1000 labels from the original ImageNet dataset [4], from CUB [13] and Oxford 102 flowers [10] datasets (mentioned by the authors in the original paper). We also did some experiments with artistic images. These results are discussed in Section 5.

4.1 Performance Experiments

Small Model vs. Base Model. The authors provided three types of models: Tiny (the smallest), Small (the middle), and Base (the largest). We tested Base vs. Small. The paper claims that with 10 hints, the performance of the three models was similar. In our tests, the qualitative performance of Base seems to be better than Small, as shown in Figure 4 up to 30 hints. After this experiment, we decided to use the Base model for the remaining experiments.

Number and location of hints. One of the challenges with this type of method is where to put the color hints and how many of them to use. How important is this decision? Can we get the same results with the same amount of hints but in different locations? To answer these questions, we performed the following experiment. For a fixed number of hints, we generated two different results: one where the hints were manually selected and one where the locations of the hints were randomly generated by sampling from a uniform distribution. We did this for different numbers of hints. As the number of manually selected hints increases, we keep the previously chosen hints and add new ones. In the case of random hint positions, each result is obtained by randomly generating all hints each time, which does not guarantee that the hint position of one experiment will be included in the set of hint positions of the next one. As shown in Figure 5 and 6, location has a notorious effect on the results, but it diminishes as the number of hints grows. For example, in Figure 5, where we used an example taken from the ImageNet dataset [11], we can see that the colorization is much more accurate when the hint positions are manually defined than in the randomly generated hint positions case, for the five hints experiment. Once the number of used hints is greater than 20, the results are very similar. The same observation applies to the experiment depicted in Figure 6, where random generation cannot achieve accurate colors with less than 20 hints.

Hints size. As in the original paper [14], we analyze the effect of the hint size on the colorization results. Since hints are generally larger than a single pixel and the colors of all pixels contained in the hint are used to determine the unique color of the entire hint, it seems relevant to analyze the influence of this parameter on the colorization performance. We tested different square hint sizes: 1×1, 2×2, 4×4, 7×7, and 8×8. Note that the hint is a square in the resized 224×224 image, and that the shape of the hint relative to the original image has the same aspect ratio and is not necessarily square. As shown in Figure 7, sizes 2×2 and 4×4 give the best results. Consequently, we set the square hint size to 2×2. Using a larger hint size will result in inaccurate coloring, partly due to how hint colors are generated. The process of determining a hint color is as follows: first, the input color image, resized to 224×244, is down-sampled by the hint size. The chrominance channels of the resulting image are then multiplied by a binary mask with ones at the hint locations and zeros otherwise. These masked chrominances are then up-sampled back to 224×244 using nearest neighbor interpolation, resulting in the color hints that are later passed to the network. Regarding the demo accompanying this article, we decided not to include the hint size as a parameter. This is because not all sizes work for the algorithm since the size of the hint down-samples 224×244 images, and thus, only hint sizes that are an integer divisor of 224 are accepted.

4.2 Natural Image Datasets

As we already mentioned, the ImageNet 2012 train split [11] was the training dataset used for iColoriT. ImageNet ctest10k [6] validation split was used as a standard benchmark for evaluating colorization models. The ImageNet ctest10K is a subset of the ImageNet 2012 validation split. Besides, the authors tested two other image datasets, CUB [13] and Oxford 102 flowers [10]. We tested the performance of iColoriT on Imagenet1K [11] (as we explained before, this is a 1000 labels

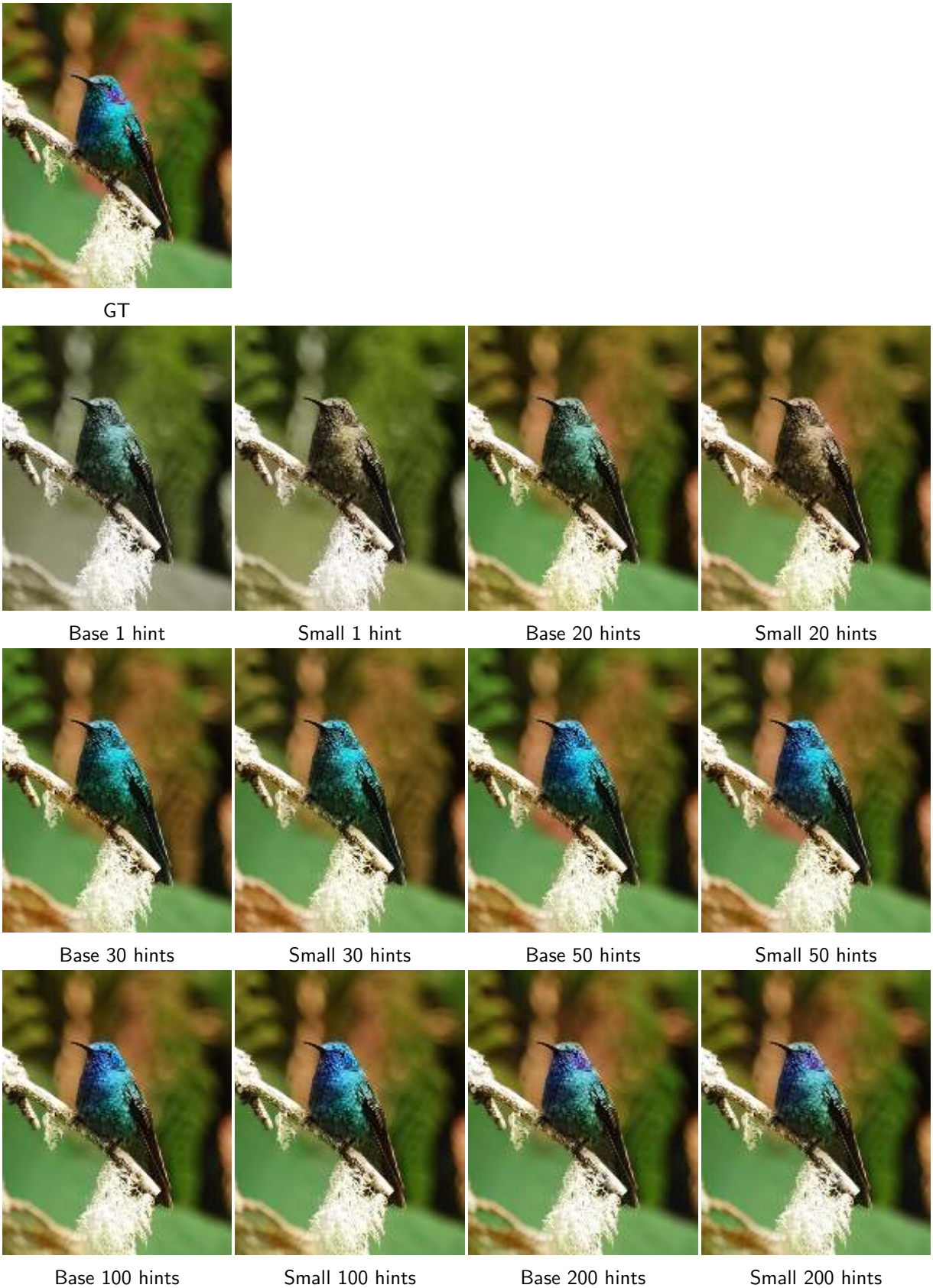


Figure 4: Comparison between Base and Small models for varying hints. The first row shows the ground truth (GT) image. The number of hints varies from 1 to 200. Note how the results become (subjectively) similar after 30 hints.

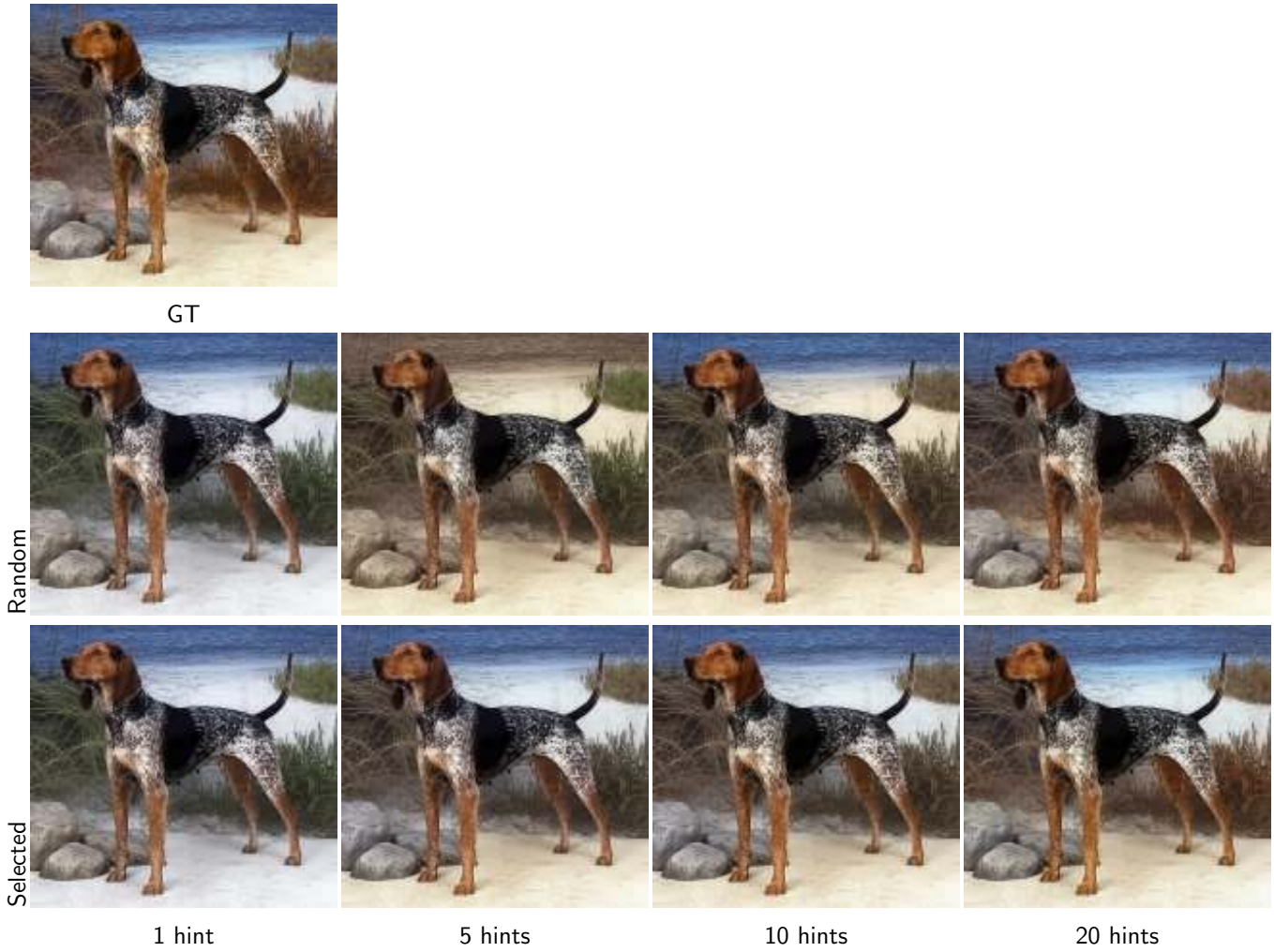


Figure 5: Results for different numbers of hints and hint locations on a dog image from the ImageNet dataset. First row: Ground truth (GT) input image. First to fourth columns: 1, 5, 10, and 20 hints. Second row: Random hint location. Third row: manually selected hint location.

subset from original ImageNet), CUB, and the Oxford 102 flowers. The following experiments used the Base model and a square-shaped hint of size 2×2.

Imagenet. First, we tested the performance on images from Imagenet1K, assuming that this is the best scenario. The results are shown in Figures 8 and 9. We can observe that for images with straightforward semantic content (as the two first rows in Figure 8), iColoriT doesn't need too many hints to achieve a good result; between 5 and 10 hints will be enough. When the complexity of the semantic content increases, the method needs more hints to show color details, as seen in the flowers in the third and fourth rows of Figure 8 or in both examples of Figure 9. In the case of the flowers example, more than 50 hints are needed for a satisfactory colorization result, although finer details are improved with even more hints. For the concert example in Figure 9, we need at least 50 hints to color the girl's shirt and up to 200 hints to avoid the greenish color on the background guitarist. For the piano example in Figure 9, we need up to 100 hints to start colorizing the little girl's pants. It is important to note that the method can colorize with zero hints in a fully automatic mode. For these experiments, hints were generated randomly, but as the number of hints increased, we kept the previously generated hints and added new ones. For example, to generate 50 hints, we kept the first 10 hints and randomly generated 40 new ones.

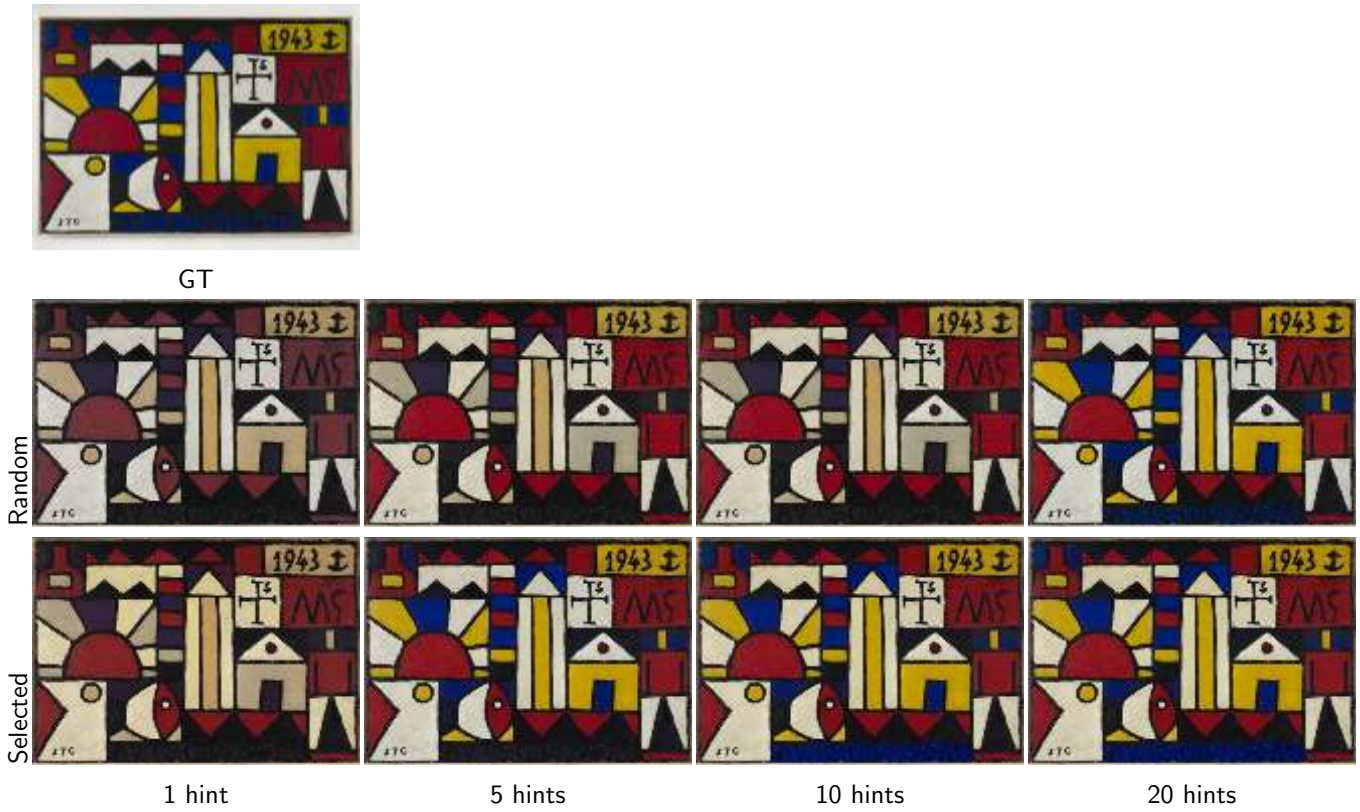


Figure 6: Results for an image from Torres García's painting with different numbers of hints and random hint generation vs. selected by user location. First row: Ground truth (GT) input image. Second row: Random location. Third row: selected location. In the last two rows, first to fourth columns: 1, 5, 10, and 20 hints.

Figure 7: Influence of the hint size. From left to right: the ground truth input (GT) and different colorization results with varying hint sizes: 1×1, 2×2, 4×4, 7×7, and 8×8. The same number of hints are used in each image.

CUB. Figure 10 shows the results for the Caltech-UCSD Birds-200-2011 (CUB) [13] dataset, a public bird's image dataset mainly used for classification tasks. The method produces satisfactory colorization above 50 hints in both examples. The observations are similar to the ones for the Imagenet images.

Oxford 102 flowers dataset. The Oxford 102 flowers dataset [10] consists of 102 different categories of flowers common in the UK. Images from this dataset have large scale, pose, and light variations. For testing, we use this dataset following the description of the original paper. Flower images usually have a lot of detail at different scales, and it is of great interest to check the algorithm performance for this type of image. The results are shown in Figure 11. The number of hints needed to get good results in both examples lies between 50 and 100.

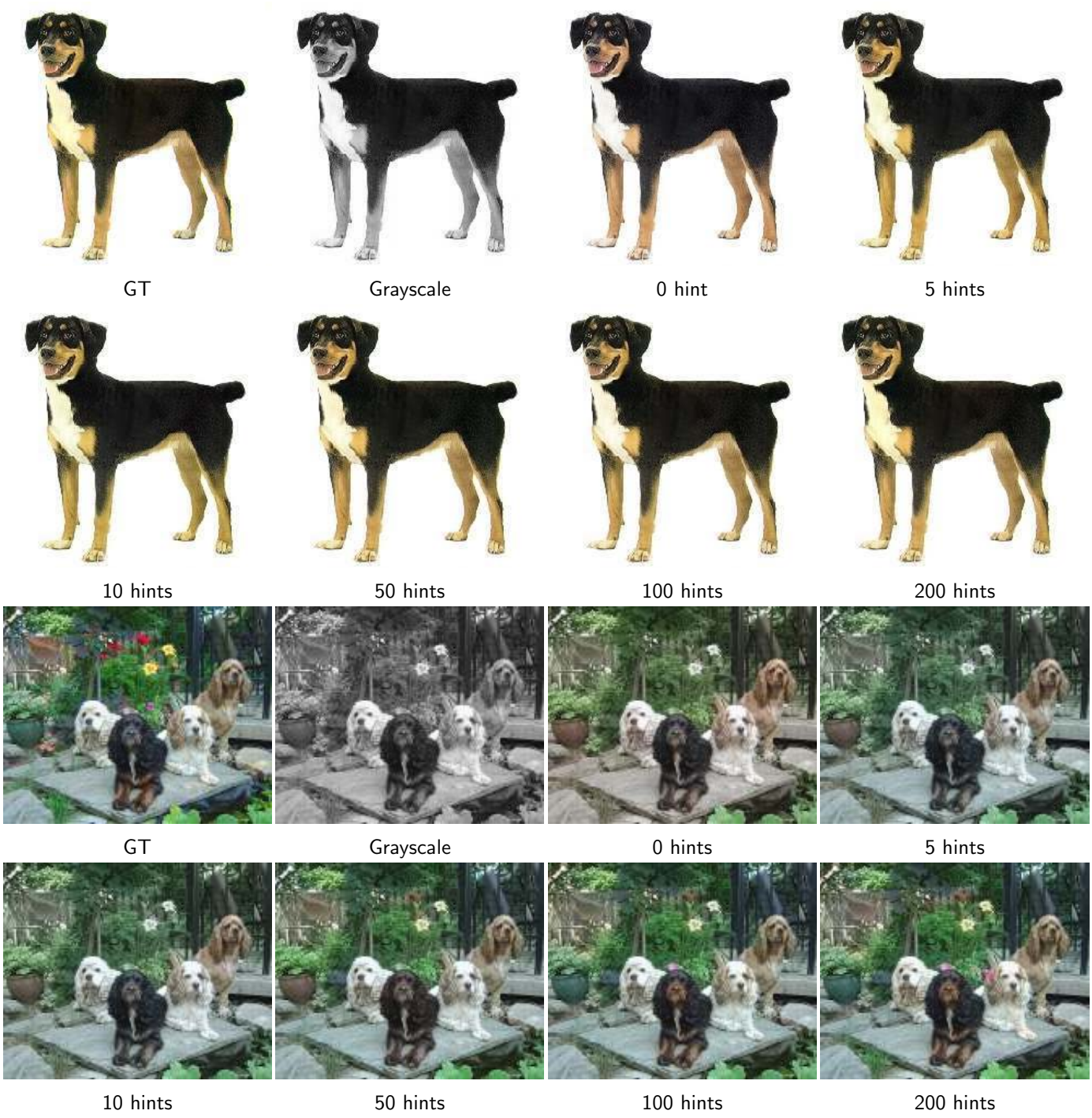


Figure 8: Experiment with the Imagenet1K dataset. The first (rows 1 and 2) shows a very simple semantic image. The second (rows 3 and 4) shows a more complex scene. For each example, several colorization results are shown for an increasing number of hints (0, 5, 10, 50, 100 and 200).

Priors. The goal of these experiments was to evaluate the trade-off between the color hints provided by the user and the colors learned by the model from the training dataset, as well as their impact on the colorization results. To accomplish this, we used the demo[3] provided by the authors, which allows users to select different hint colors for a given grayscale input (as opposed to the demo we provide with this paper, which uses the ground truth color of the image as the color hint). In Figure 12 we try different hint colors to colorize the same input image. In the left example, we use three hints

[3] https://github.com/pmh9960/iColoriT

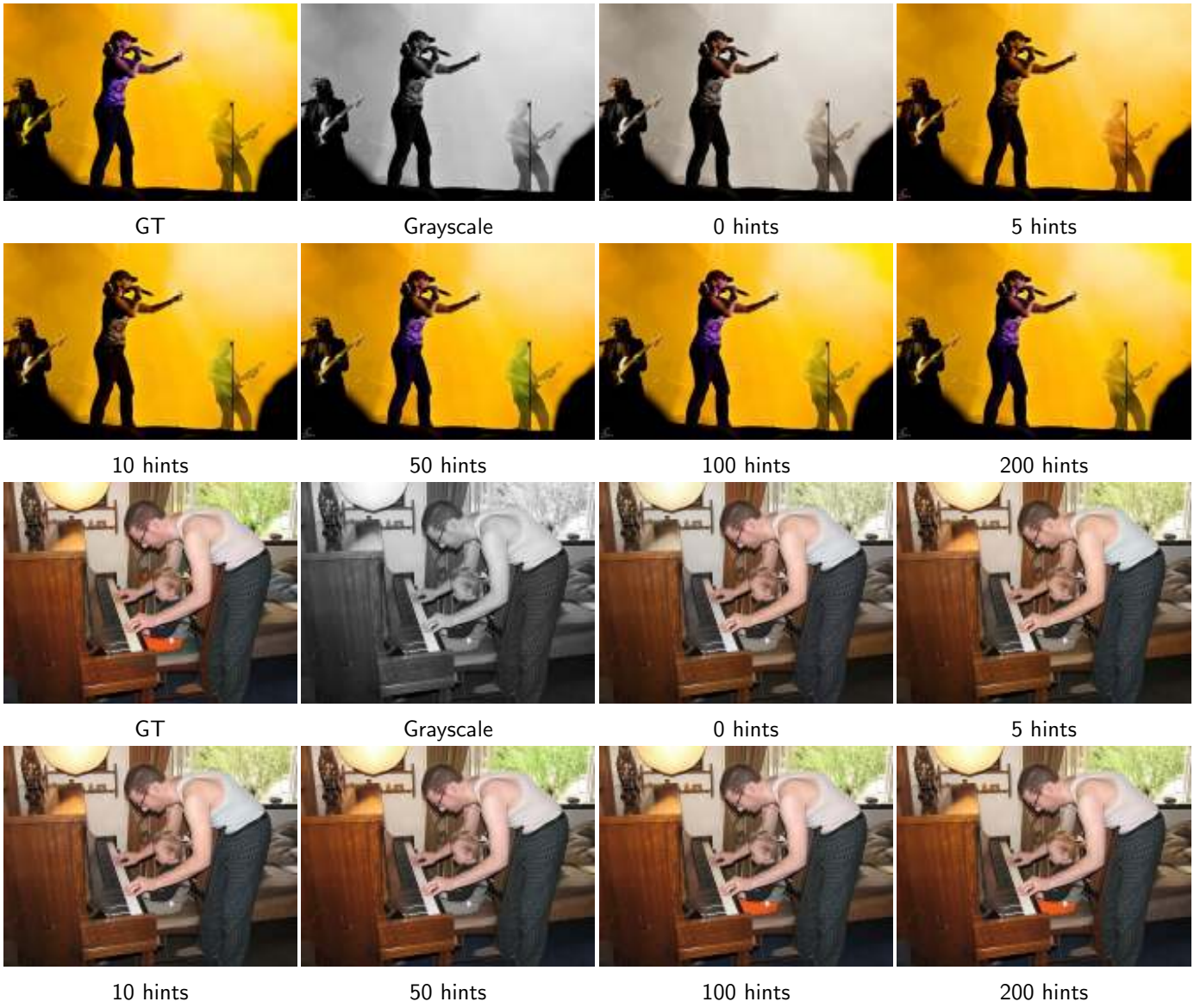


Figure 9: Imagenet1K dataset. The first (rows 1 and 2) shows a concert scene with fog and blurred people in the background. The second (rows 3 and 4) shows a scene with multiple object instances. For each example, several colorization results are shown for an increasing number of hints (0, 5, 10, 50, 100, and 200).

with "normal colors in the sense that they are coherent with the color priors of the image dataset used for training. In the middle example, we keep the same hint locations and change the color of the hint in the ear from brown to purple. There are no priors for purple dogs, so the propagation of this color hint is limited, and we can see that the other ear is not colored in purple. In the right example, we added several hints on the grass using colors such as purple and pink, and again, they are inconsistent with the color priors learned from the dataset, and thus are not propagated to the rest of the grass. The grass color changes to a less saturated brownish color without clearly accepting these peculiar colors.

4.3 Paintings

This section focuses on experiments aiming to colorize abstract paintings. This is an interesting test, as the colors and semantics of the abstract painters differ from those represented in the training set.

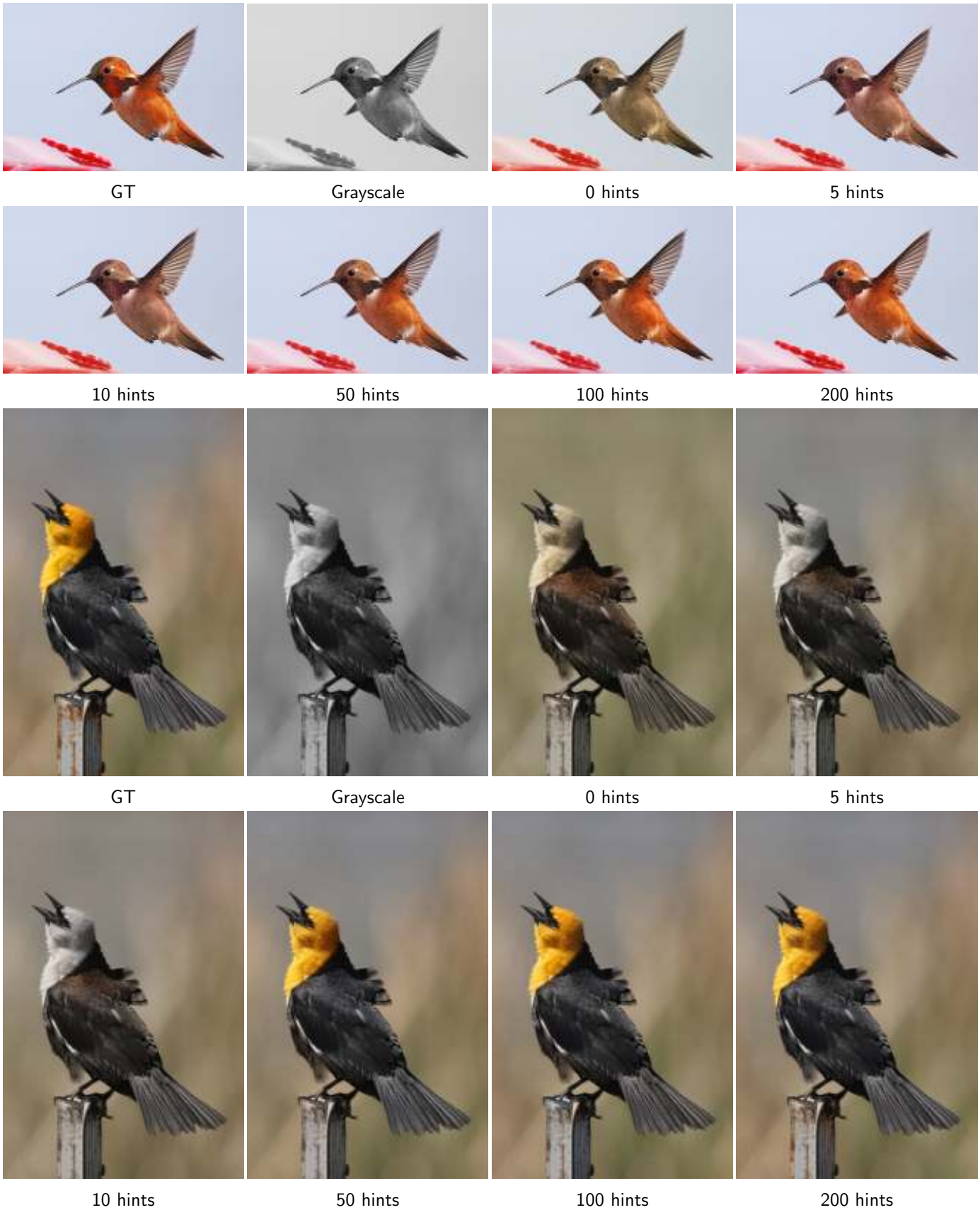


Figure 10: Experiments with CUB dataset for two bird examples. For each example, several colorization results are shown for an increasing number of hints (0, 5, 10, 50, 100 and 200).

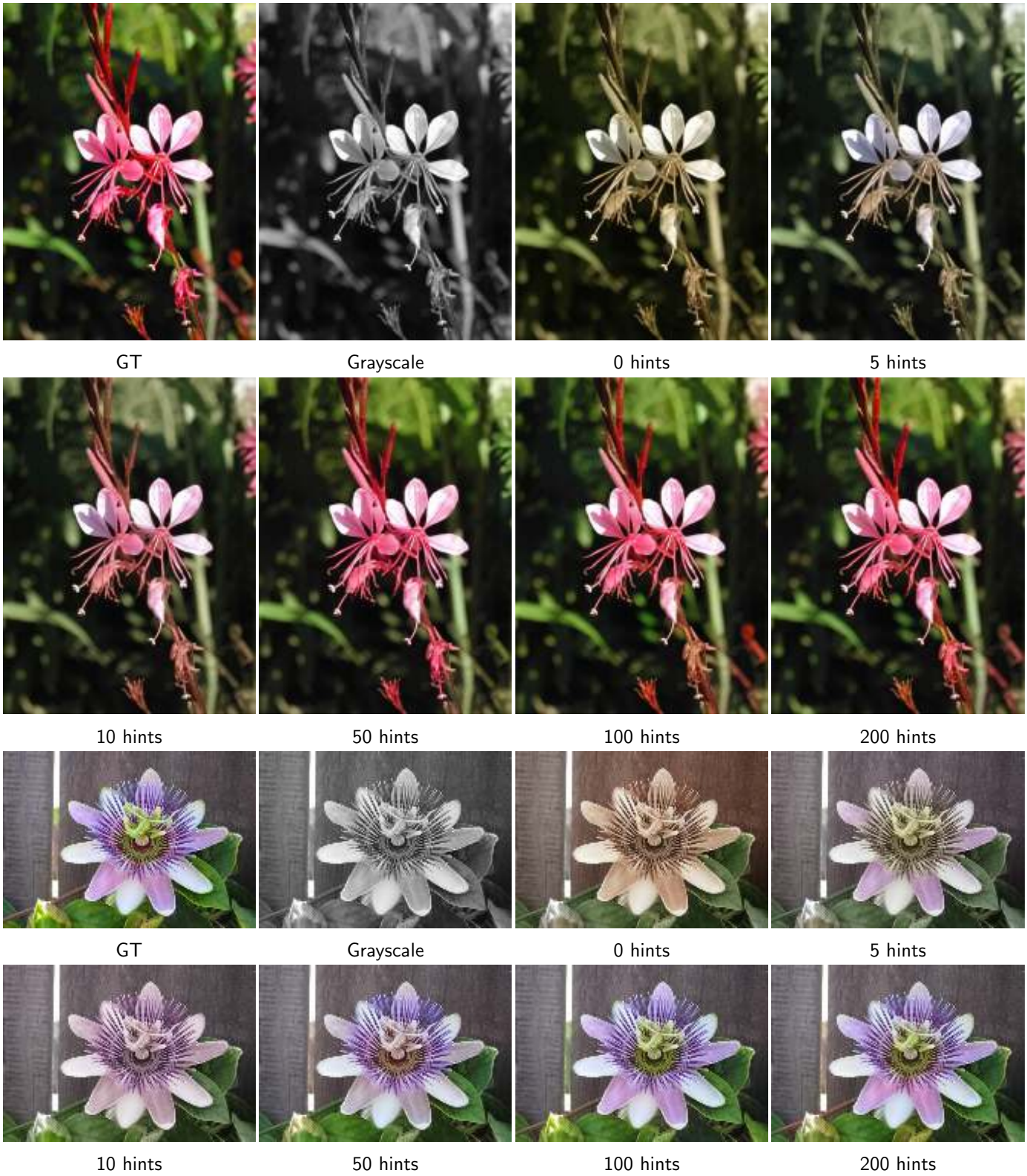


Figure 11: Experiments with Oxford 102 flowers dataset for two flower examples. For each example, several colorization results are shown for an increasing number of hints (0, 5, 10, 50, 100 and 200).

Torres García's paintings. Joaquín Torres García was a Uruguayan painter known for creating the major artistic movements of Modern Classicism and Universal Constructivism. In 1978, 74 of his best paintings were destroyed in a major fire at the Museum of Modern Art in Rio de Janeiro. The color records of those paintings are either of poor quality or didn't exist. As a part of an ongoing

Figure 12: Trade-off between color hints and color priors learned from the dataset. Each column shows the grayscale image to be colorized with its color hints at the top and the colorization result at the bottom. Left column: three "normal" color hints. Middle column: unusual purple hint on the dog's ear. Right column: unusual purple and pink hints on the grass.

project, we want to restore Torres García's paintings. Figure 6 and Figure 13 show experiments using color photographs taken from existing paintings that belong to the Joaquín Torres García Museum in Montevideo, Uruguay. In Figure 13, a Mondrian painting provides another example of coloring abstract artworks. In general, we can observe how well this method colorizes the paints. In this particular case, both Torres García's and Mondrian's paintings present fairly good results using 50 hints or more.

5 Discussion and Conclusions

The iColoriT method is a hybrid colorization method; it combines color priors learned from a large dataset and color hints indicated by users. This gives better control over the colorized output. This method generally achieves very good results with less than 20 hints, mainly if the hint's location and color are accurate. Results for images with simple semantics require very few hints, as shown in Figures 8, and 10, in contrast to images with more complex semantics, as the ones shown in Figures 9

Figure 13: Experiments with Torres García and Mondrian Paintings. Torres García paint: rows 1 and 2. Mondrian paint: rows 3 and 4. From top to down, left to right: Ground Truth (GT), Grayscale, 0, 5, 10, 50, 100 and 200 hints.)

and 11, which require twice as many hints.

Because of the interactive nature of the method, some of the typical drawbacks of image colorization methods, such as color bleeding and the misrepresentation of the variety of colors present in an image, can be quickly addressed and corrected. Figure 14 shows a color bleeding artifact (note the reddish zone in the upper right part of the yellow triangle, produced by the red triangle on top of it). As the second row of Figure 14 shows, this is corrected by adding a blue hint in this zone.

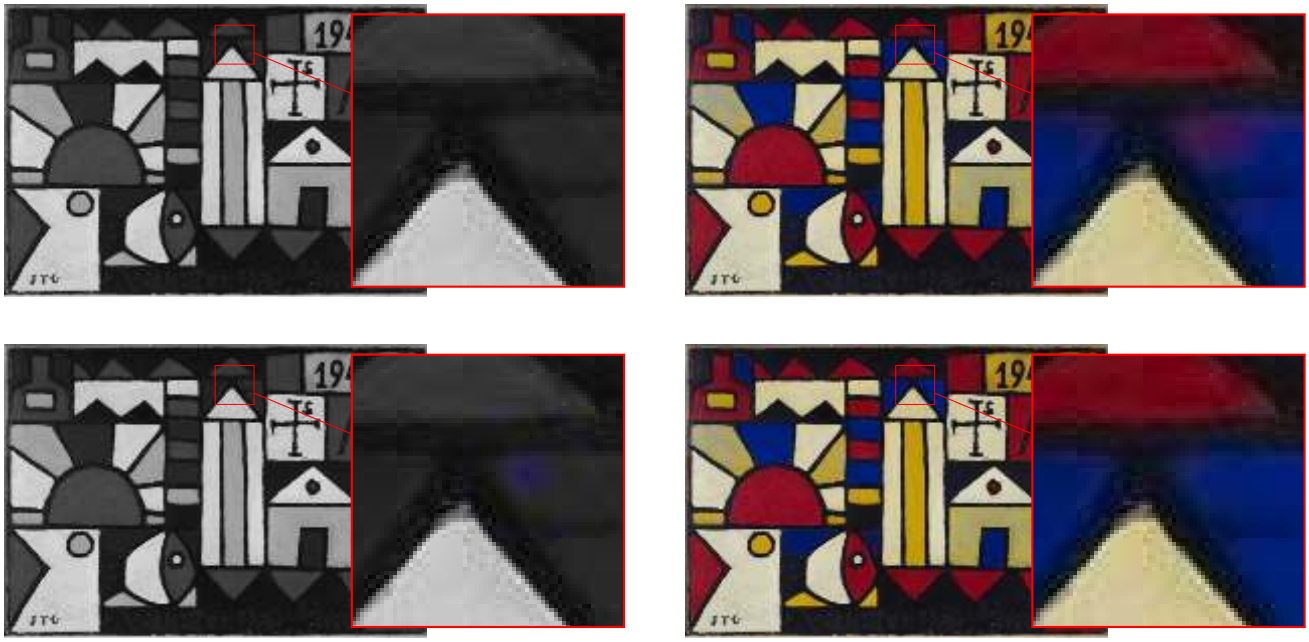


Figure 14: Color bleeding example. Top left: grayscale image to be colorized with a close-up showing no hint in the bleeding area. Top right: colorized image and close-up of the bleeding area. Bottom left: grayscale image to be colorized with a close-up showing we added a blue hint in the bleeding area. Bottom right: colorized image and close-up showing the removal of the bleeding artifact.

Figure 15 illustrates a case of a color misrepresentation. The background of the red structure must be blue on both sides, but the left part is gray. This is corrected again by adding an additional blue hint in that region, as seen in the second row of Figure 15. This is of great help for tasks requiring precise colorization, such as image restoration.

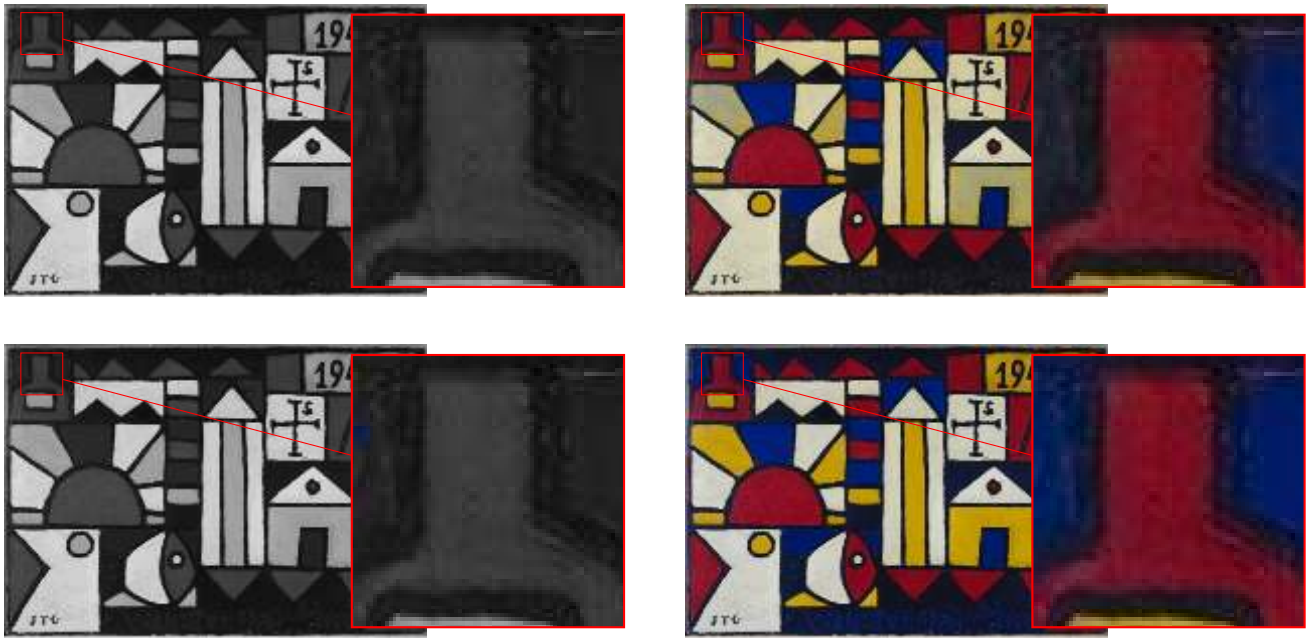


Figure 15: Color inconsistency example. Top left: grayscale image to be colorized and a close-up showing the missing color region. Top right: colorized image and close-up showing that both sides of the bottle are inconsistent, both should be blue. Bottom left: Grayscale image to be colorized and a close-up showing that we added a blue hint in the inconsistent area. Bottom right: Colorized image and close-up showing that both sides of the bottle are now consistent.

Note the method's behavior when no hints are provided. This case is equivalent to an automatic colorization method. The results obtained, even if the exact colors are not recovered, are a good starting point and show how the model uses color priors from the training dataset and color hints when the latter are provided.

It's important to note that the results for all experiments in this analysis are obtained with color hints that are almost exactly the ground truth colors (since an average of all pixels of the hint region is taken as the color of the hint). In a real-world colorization problem, it will be necessary to specify the position of the hints and the color of each hint, which can make the task more difficult and degrade the colorization results. In this paper, we provide a demo where the hint locations can be specified, but not their color, due to technical limitations of the demo system. In addition, the code provided by the authors uses ground-truth image color hints by default. This also means that some of their scripts have to be modified to receive any color hints. These two aspects are beyond the scope of this paper and will be the subject of future work.

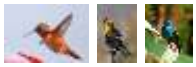
Another crucial point to consider is the significance of the priors learned by the model, which play an essential role in determining the colorization outcomes. As depicted in Figure 12, the number of hints required to achieve satisfactory results varies depending on the color coherence with the prior color. For instance, in the experiment illustrated in Figure 12, purple color isn't typically associated with elements within the dog image. Therefore, iColoriT restricts the propagation of the violet hint even when a large amount of hints of this color are provided. This aspect is especially relevant for art image restoration problems, where we navigate beyond the realm of natural images, each with distinct semantics.

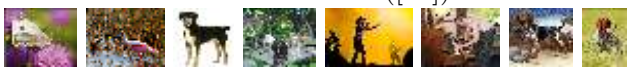
This study focuses on presenting the results obtained with iColoriT and does not attempt to reproduce the training phase. However, the results on artistic images are very impressive (Figures 6 and 13), as well as the results when no hints are provided (Figures 8 and 9 with 0 hints), i.e., automatic colorization. This suggests that it is possible to adapt this colorization model to artistic datasets. This could allow us to incorporate painting semantics (which differs from natural image semantics) and obtain better results in the field of art restoration.

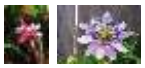
Acknowledgment

We thank the Museo Torres García in Montevideo, Uruguay, for providing us with images of paintings by Joaquín Torres García and for allowing their non-profit use in this article. We thank Luis Sosa and Ignacio Seimanas for their photographic work and for taking images of those paintings.

Image Credits

 Dataset CUB ([13]).

 Imagenet1K dataset ([11])

 Oxford 102 flowers dataset ([10]).

 Images from Museo Torres García, Montevideo, Uruguay, Taken by Ignacio Seimanas and Luis Sosa.

 Image from Museo Torres García, Montevideo, Uruguay.

 Piet Mondrian paint. Image from Public Domain Collections from picryl.com.

References

[1] S. Anwar, M. Tahir, C. Li, A. Mian, F. S. Khan, and A. W. Muzaffar, *Image Colorization: A Survey and Dataset*, ArXiv Preprint ArXiv:2008.10774, (2020), https://doi.org/10.48550/arXiv.2008.10774.

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton, *Layer Normalization*, ArXiv Preprint ArXiv:1607.06450, (2016), https://doi.org/10.48550/arXiv.1607.06450.

[3] C. Ballester, A. Bugeau, H. Carrillo, M. Clément, R. Giraud, L. Raad, and P. Vitoria, *Influence of Color Spaces for Deep Learning Image Colorization*, ArXiv Preprint ArXiv:2204.02850, (2022), https://doi.org/10.48550/arXiv.2204.02850.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, *ImageNet: A Large-Scale Hierarchical Image Database*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255, https://doi.org/10.1109/CVPR.2009.5206848.

[5] 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*, International Conference on Learning Representations (ICLR), (2020).

[6] G. Larsson, M. Maire, and G. Shakhnarovich, *Learning Representations for Automatic Colorization*, CoRR, abs/1603.06668 (2016), https://doi.org/10.48550/arXiv.1603.06668.

[7] B. Li, Y.-K. Lai, and P. L. Rosin, *A Review of Image Colourisation*, Handbook Of Pattern Recognition And Computer Vision, (2020), pp. 139–157, https://doi.org/10.1142/9789811211072_0008.

[8] I. Loshchilov and F. Hutter, *SGDR: Stochastic Gradient Descent with Warm Restarts*, ArXiv Preprint ArXiv:1608.03983, (2016), https://doi.org/10.48550/arXiv.1608.03983.

[9] ——, *Decoupled Weight Decay Regularization*, ArXiv Preprint ArXiv:1711.05101, (2017), https://doi.org/10.48550/arXiv.1711.05101.

[10] M.-E. Nilsback and A. Zisserman, *Oxford 102 Flowers*. https://www.robots.ox.ac.uk/~vgg/data/flowers/102/.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, *ImageNet Large Scale Visual Recognition Challenge*, International Journal of Computer Vision (IJCV), 115 (2015), pp. 211–252, https://doi.org/10.1007/s11263-015-0816-y.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention Is All You Need*, Advances in Neural Information Processing Systems, 30 (2017).

[13] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, *CUB-200-2011*, 2022, https://doi.org/10.22002/D1.20098.

[14] J. Yun, S. Lee, M. Park, and J. Choo, *IColoriT: Towards Propagating Local Hints to the Right Region in Interactive Colorization by Leveraging Vision Transformer*, in IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2023, pp. 1787–1796, https://doi.org/10.1109/WACV56688.2023.00183.