DACoN: DINO for Anime Paint Bucket Colorization with Any Number of Reference Images

Supplementary Material

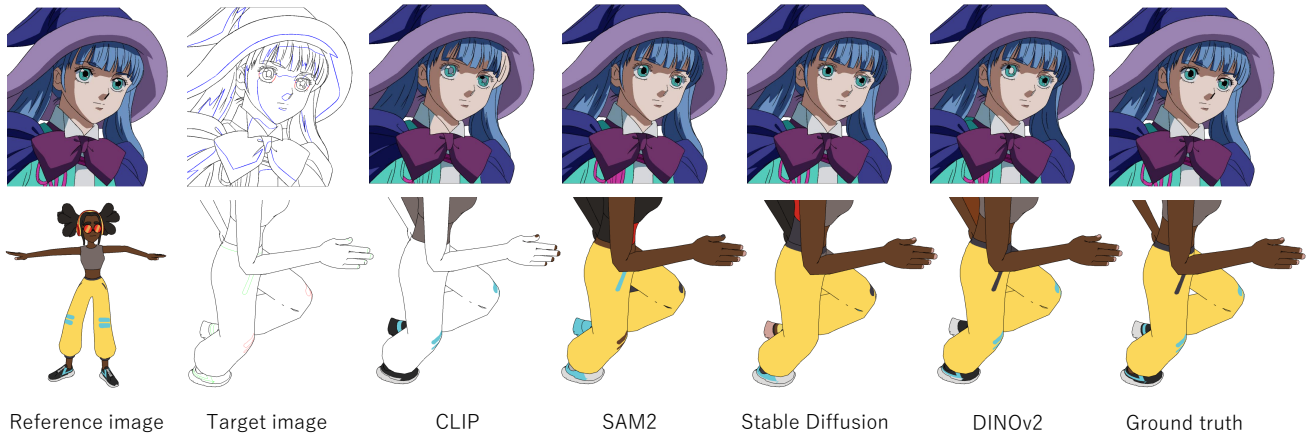

Figure 1. Visual comparison of zero-shot colorization by foundational models. The top row shows the results of consecutive frame colorization, while the bottom row presents those of keyframe colorization.

Table 1. Quantitative comparison of zero-shot colorization by foundation models. "Ours (SD)" indicates the results obtained by replacing DINOv2 in our method with Stable Diffusion during training and evaluation.

Method	Keyframe (3D rendered, 1-shot)					Consecutive frame (Hand-drawn)				
	Acc	Acc-Thresh	Pix-Acc	Pix-F-Acc	Pix-B-MIoU	Acc	Acc-Thresh	Pix-Acc	Pix-F-Acc	Pix-B-MIoU
CLIP	36.72	38.95	86.76	60.46	88.37	67.13	69.56	97.07	88.65	99.00
SAM2	34.54	36.96	85.06	54.12	95.62	82.44	85.51	98.35	94.68	99.18
Stable Diffusion	49.72	53.80	89.84	71.10		**84.12**	**87.33**	98.46	**95.56**	99.03
DINOv2	**57.49**	**61.86**	**95.35**	**87.24**	**97.45**	80.64	83.39	**98.79**	95.35	**99.78**
Ours	67.87	72.58	96.99	91.00	99.08	87.44	90.48	99.19	96.91	99.83
Ours (SD)	62.26	66.88	93.88	82.13	98.43	87.27	90.46	98.89	96.54	99.29
Ours (SD) w/o L_{dc}	60.96	65.52	93.43	80.57	98.79	86.93	90.23	98.64	95.99	99.74

1. Comparisons with Foundation Models

To evaluate the impact of DINOv2 features [6] on the proposed method, we assess zero-shot colorization performance using only DINO features—that is, without any additional training or fine-tuning. Additionally, we compare DINOv2 with other visual foundation models to explore how their feature representations contribute to automatic colorization. For this comparison, we select Stable Diffusion (SD) [10], which captures part-level semantic information similar to DINOv2 [12], as well as SAM2 [9] and CLIP [7], which are widely used foundation models.

Segment pooling, as employed in our method, is applied to each model's feature map to enable segment correspondence and color propagation between the reference and target images. The feature extraction methods for each model are as follows:

- DINOv2: We use the Large model with an input size of 518×518, and features are extracted from the final encoder layer.
- SD: We use the Stable Diffusion v2-1 model with an input size of 768×768. The input text prompt is "a photo of an anime character." The feature map is extracted via the correspondence method proposed in [11] from the first layer of upsampling, with the dimensional step set to $261/1000$.
- SAM2: We use the Large model of SAM2.1 with an input size of 1024×1024, and features are extracted from the final encoder layer.
- CLIP: We select the ConvNext-Large model from OpenCLIP, as used in BasicPBC [3], with an input size of 224×224, and features are extracted from the final en-

Table 2. Comparison of inference time and memory usage during consecutive frame colorization on 3D synthetic test data. "Time" represents the average colorization time per sample, "FPS" denotes the number of frames colorized per second, "Params" indicates the model's parameter size, and "Peak Mem" refers to the peak memory usage during inference. "Ours*" corresponds to the configuration with segment pooling fixed at 512×512. All experiments were conducted on an NVIDIA GeForce RTX 4090 GPU.

Method	Time [ms]	FPS	Params [M]	Model Size [GB]	Peak Mem [GB]
BasicPBC	1454.52	0.69	26.33	0.10	2.62
Ours	264.37	3.78	339.85	1.30	6.01
Ours*	249.11	4.01	339.85	1.30	3.20
Ours w/o DINOv2	—	—	35.49	0.14	—

Table 3. Quantitative comparison of different segment pooling sizes. "Ours" uses the same size as the input image, while "Ours*" is fixed at 512×512.

Method	Keyframe (3D rendered, 1-shot)			Consecutive frame (Hand-drawn)		
	Acc	Acc-Thresh	Pix-Acc	Acc	Acc-Thresh	Pix-Acc
Ours	67.87	72.58	96.99	87.44	90.48	99.19
Ours*	67.79	72.49	96.98	87.48	90.61	99.21

coder layer.

In all models, both the reference and target images are line drawings.

The quantitative results are presented in Table 1. DINOv2 achieves the highest accuracy in keyframe colorization, while SD outperforms the others in consecutive frame colorization, suggesting that DINOv2 excels at capturing semantic, part-level information, whereas SD effectively leverages spatial details. SAM2, as shown in Figure 1, performs well in consecutive frame colorization, likely due to its training on high-resolution images, which helps capture fine-grained details.

Further experiments replacing DINOv2 with SD in the proposed method reveal that although SD performs well in zero-shot consecutive frame colorization, DINOv2 still achieves higher accuracy when integrated into our method. This result indicates that the U-Net, which learns spatial features, complements DINOv2's strength in capturing semantic details. In contrast, since both SD and the U-Net excel at capturing spatial information, their strengths overlap, thereby reducing the added benefit of using SD in this context.

In addition, ablation results show that our proposed Feature Consistency Loss improves performance not only with DINOv2 but also with other foundation models such as SD, which capture semantic information.

2. Inference Time and Memory Usage

To evaluate the implementation cost for anime production, we measure the inference time and memory usage of the proposed method and compare it with BasicPBC [4], which serves as our baseline. A comparison with AnT [2], which also utilizes segment correspondence, is not possible because the internal processing details of the Cadmium application [1] are not publicly available. For this comparison, consecutive frame colorization is performed on a test set of 2,850 samples of 3D rendered data. The measurements are taken from the moment the images are fed until the predicted color information for each segment is produced, with segment information pre-prepared and models preloaded into memory.

The results are presented in Table 2. Our proposed method is capable of colorizing approximately three samples per second, making it roughly five times faster than the baseline method. This improvement can be attributed to the absence of optical flow estimation, which is required by previous approaches in addition to feature extraction. However, since the segment region and color information are provided in advance for this dataset, the measured speed does not fully reflect the actual cost of automatic colorization. In the case of test images where all segment regions are completely enclosed by line drawings, it takes an average of 1.67 seconds to extract the segment areas and corresponding colors. As a result, the average time for complete automatic colorization is 3.12 seconds for BasicPBC and 1.9 seconds for our method. According to prior research [5], professional animators reported that automatic colorization of 20 to 30 frames within 5 to 10 minutes—accounting for manual correction time—is considered acceptable. Therefore, both methods are practically viable in terms of speed.

On the other hand, in terms of memory consumption, our method relies on a large foundation model, resulting in a significantly larger model size than the baseline. In particular, during segment pooling, the DINO feature map (with a feature dimension of 1024) is expanded to match the input image size, leading to a peak memory usage of over 6 GB during inference. This suggests that memory usage may become a bottleneck when processing high-resolution target images.

To address this issue, we apply max pooling to downscale the segment masks to a resolution of 512×512, consistent with training conditions, and evaluate the performance under this setting (see "Ours*" in Tables 2 and 3). As a result, we reduce the peak memory usage to approximately 3 GB—about half—without significantly sacrificing colorization accuracy. This indicates that the proposed method can perform robust automatic colorization under memory constraints regardless of input resolution, further enhancing its practicality.

3. Clip-Wise Colorization

To further demonstrate the practical applicability of our method, we evaluate a scenario where only the first frame of each clip is provided as a reference. As shown in Table 4, this setting significantly reduces accuracy compared

Table 4. Quantitative comparison of colorization using only the first frame of each clip or with additional reference images.

Method	Ref. data	3D rendered			Hand-drawn		
		Acc	Acc-Thresh	Pix-Acc	Acc	Acc-Thresh	Pix-Acc
Ours	first frame	69.91	73.59	97.30	65.85	69.22	93.50
Ours	first frame + 1 shot	74.59	78.67	98.20	—	—	—
Ours	first frame + 5 shot	76.81	80.82	98.38	—	—	—
Ours	first frame + max shot	77.28	81.28	98.39	—	—	—
Ours	-1 frame	84.66	88.23	99.27	87.44	90.48	99.19

Table 5. Ablation results on architecture design.

Method	Keyframe (3D rendered, 1-shot)			Consecutive frame (Hand-drawn)		
	Acc	Acc-Thresh	Pix-Acc	Acc	Acc-Thresh	Pix-Acc
Ours (MLP based)	**67.87**	**72.58**	**96.99**	**87.44**	**90.48**	**99.19**
DPT (alt. dim.red.)	66.55	71.30	96.51	86.96	90.10	99.11
Cross Atten. (alt. fusion)	47.60	50.69	66.45	85.59	88.79	98.41
Multiplex Transformer (add. agg.)	66.54	71.47	96.81	87.08	89.99	99.09

to the consecutive frame colorization setting, where the previous frame (*i.e.*, the -1st frame) is used as the reference. Nevertheless, supplementing the first frame with additional color design sheets helps mitigate this gap, particularly improving accuracy in regions not visible in the first frame and highlights the strength of our multi-reference approach. Improving performance under such limited-reference conditions remains a challenging problem, and leveraging temporal consistency across frames will be a key direction for future work.

4. Ablation for Architectures

We further investigate alternative architectures within our framework. Specifically, we evaluate the following three variants:

- **DPT-based dimensionality reduction:** Following DPT [8], we aggregate features from multiple layers of DINOv2 (*i.e.*, layers 5, 12, 18, and 24) and project them into a 128-dimensional space. The output resolution is set to 296×296.
- **Cross-attention-based feature fusion:** Instead of using simple concatenation followed by an MLP, we adopt a cross-attention mechanism to fuse CNN and DINO features. In this setup, CNN features serve as queries, and DINO features are used as keys and values. The module consists of 4 attention heads and 9 layers.
- **Multiplex Transformer integration:** We incorporate a Multiplex Transformer module—commonly used in prior work but deliberately not employed in our proposed method—to jointly process and aggregate segment features from both reference and target images for comparison purposes. This module also uses 4 attention heads and 9 layers.

As summarized in Table 5, our MLP-based design consistently outperforms these more complex alternatives, including Transformer-based modules. Given that the evaluation dataset consists of 11,345 samples, with character diversity limited to 12 identities and the domain restricted to line drawings, we suggest that lightweight architectures with fewer parameters, such as MLPs, may offer a better trade-off under such constraints compared to heavier models like CNNs or Transformers.

Figure 2. Additional qualitative comparisons of keyframe colorization results.

References

[1] Cadmium. https://cadmium.app/. 2

[2] Evan Casey, Víctor Pérez, and Zhuoru Li. The animation transformer: Visual correspondence via segment matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11323–11332, 2021. 2

[3] Yuekun Dai, Qinyue Li, Shangchen Zhou, Yihang Luo, Chongyi Li, and Chen Change Loy. Paint bucket colorization using anime character color design sheets. *arXiv preprint arXiv:2410.19424*, 2024. 1

[4] Yuekun Dai, Shangchen Zhou, Qinyue Li, Chongyi Li, and Chen Change Loy. Learning inclusion matching for animation paint bucket colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25544–25553, 2024. 2

[5] Akinobu Maejima, Hiroyuki Kubo, Seitaro Shinagawa, Takuya Funatomi, Tatsuo Yotsukura, Satoshi Nakamura, and Yasuhiro Mukaigawa. Anime character colorization using few-shot learning. In *SIGGRAPH Asia 2021 Technical Communications*, pages 1–4. 2021. 2

[6] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021. 1

[8] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 3

[9] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1

[10] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 1

[11] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:1363–1389, 2023. 1

[12] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable Diffusion complements DINO for zero-shot semantic correspondence. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. 1