Real-Time User-guided Adaptive Colorization with Vision Transformer

Gwanghan Lee[*1], Saebyeol Shin[*2], Taeyoung Na[3], Simon S. Woo[†1]
[1]Department of Artificial Intelligence, Sungkyunkwan University, South Korea
[2] College of Computing and Informatics, Sungkyunkwan University, South Korea
[3] SK Telecom, South Korea
{ican0016, toquf930}@g.skku.edu, taeyoung.na@sk.com, swoo@g.skku.edu

Abstract

Recently, the vision transformer (ViT) has achieved remarkable performance in computer vision tasks and has been actively utilized in colorization. Vision transformer uses multi-head self attention to effectively propagate user hints to distant relevant areas in the image. However, despite the success of vision transformers in colorizing the image, heavy underlying ViT architecture and the large computational cost hinder active real-time user interaction for colorization applications. Several research removed redundant image patches to reduce the computational cost of ViT in image classification tasks. However, the existing efficient ViT methods cause severe performance degradation in colorization task since it completely removes the redundant patches. Thus, we propose a novel efficient ViT architecture for real-time interactive colorization, AdaColViT determines which redundant image patches and layers to reduce in the ViT. Unlike existing methods, our novel pruning method alleviates performance drop and flexibly allocates computational resources of input samples, effectively achieving actual acceleration. In addition, we demonstrate through extensive experiments on ImageNet-ctest10k, Oxford 102flowers, and CUB-200 datasets that our method outperforms the baseline methods.

1. Introduction

Despite the difficulty of colorization due to the requirement of a semantic understanding of the scenery and natural colors that dwell in the wild, various user-guided image colorization methods have shown remarkable results in restoring grayscale photographs as well as black and white films. Among the user-guided colorization, the point-interactive colorization methods [12, 27, 36] help users with user-guided hints to assist in colorizing an image, while minimizing interaction with users. In particular, [36] proposed a colorization method with U-net architecture trained on ImageNet [3] and training with synthetically generated user hints through 2-D Gaussian sampling. However, prior works suffer from partial colorization, where the unclear boundary of images is not colored successfully. Furthermore, failure in consistent colorization comes from the difficulty of propagating hints to large and distant semantic regions. In order to tackle this problem, [33] leverages the architecture of vision transformers (ViT), allowing the model to learn to propagate the user hints to other distant and similar regions with self-attention. Despite the exceptional performance of ViT in colorization applications, transformer-based models contain redundant computations resulting in slow inference speed. This problem limits users' active interactions on a variety of real-time colorization applications.

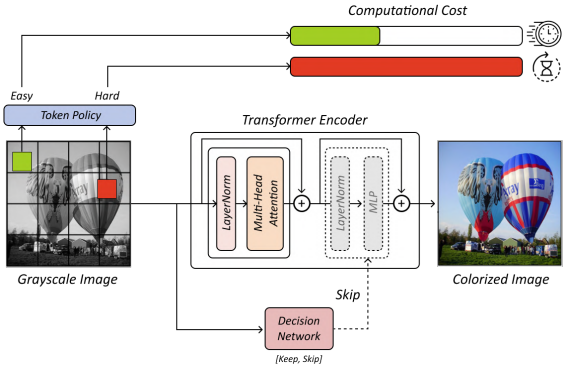

Figure 1. The overview of AdaColViT , which adaptively reduces the computational cost of less informative patches and layers of vision transformer based on the difficulty of input image.

In order to reduce the computational cost of the vision transformers, efficient vision transformers are utilized for colorization. Existing efficient ViT [18, 20, 28] uses a small number of informative patches to reduce computational cost. It uses more patches for complex images with

[*]Equal contribution
[†]Corresponding author

cluttered backgrounds or ambiguous objects and less tokens for simple images with plain backgrounds or clear objects on image classification tasks. Comparing the (b), (c), and (d) diagrams in Figure 2, we can see that the ViT-tiny model is sufficient to restore the plain background. Therefore, the existing Efficient ViT research that removes the unimportant image patches can be applied to the colorization tasks. However, existing efficient ViT causes a serious performance drop when applied to the colorization task since it completely removes the image patches of ViT. To address this issue, we propose a novel flexible end-to-end framework AdaColViT, the real-time interactive colorization ViT that softly removes the image patches according to the input samples. Unlike simply removing entire image patches or ViT blocks, our proposed approach leverages a decision network to effectively identify redundant image patches and layers within the transformer. This allows for precise removal of solely redundant components within the network. It is important to note that previous pruning methods' complete removal of image patches and attention heads contrasts with our method's soft removal approach. As shown in Figure 1, Our proposed framework effectively utilizes a decision network to determine which redundant image patches and layers to reduce in the transformer. In particular, we adapt Gumbel-Softmax trick [17] to enable backpropagation in the training process since the binary decisions from decision network are non-differentiable. In addition, we conduct extensive experiments on ImageNet-ctest10k to validate the effectiveness of AdaColViT and demonstrate that our framework outperforms the baseline methodology. Moreover, our visualization result illustrates whether computational resources are effectively allocated based on the easy and hard samples.

The main contributions of our work are summarized as follows:

- We propose AdaColViT, a flexible real-time user interactive colorization model, which input-adaptively allocates computational cost based on the easy and hard samples.
- We propose novel pruning method with a trainable decision network. Our decision network determines which redundant image patches and layers of the transformer to prune or retain to achieve efficiency and real-time colorization needs without significant performance drop.
- Through extensive quantitative experiments and qualitative analysis, we demonstrate that our model outperforms the existing point-interactive colorization with vision transformer with improved inference speed.

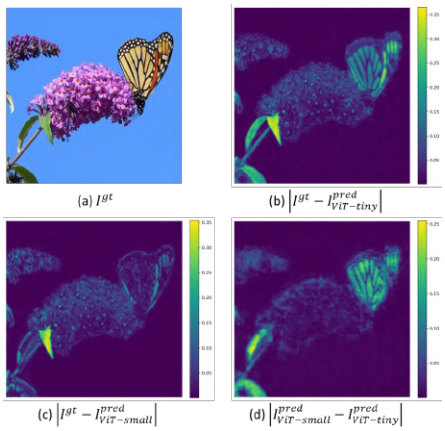

Figure 2. Figures (b), (c), and (d) show the channel-wise mean of the absolute difference of I^{gt}, $I^{pred}_{ViT-tiny}$ and $I^{pred}_{ViT-small}$ with RGB 3 channels, respectively.

2. Related Work

Interactive Colorization. Learning-based colorization methods do not require user interaction to generate adequate color images, while interactive methods require user-provided conditions to produce specified colored images. Reference-based colorization is one of the most popular interactive methods, which uses single reference images to provide overall color information [1, 5, 34]. However, since the colorized image is highly dependent on a reference image, it is challenging for the user to modify particular regions in the colorized image.

Moreover, the point-interactive colorization model [21, 30, 36] enables users to provide precise $2 \times 2 \sim 7 \times 7$ color hints on particular input image regions to cover small regions of the full image, raising the importance of minimal user effort. Previous works detected simple patterns with image filters that determine the propagation portion of each hint, which is propagated within the region by optimization methods [12, 27]. In contrast to the previous convolution-based technique in image synthesis, recent prior works utilized transformers [7, 9, 10, 29] to automatically colorize images. [10] proposed Colorization Transformer (ColTran) based on Axial Transformer [6] self-attention to unconditionally generate coarse low-resolution grayscale image and use color and spatial upsampler to produce high resolution colorized image. Also, hybrid transformer architectures are also proposed in colorization. [9] used transformer-based encoder and color memory decoder to obtain contextual semantics and color diversity, [7] uses BERT-style hybrid transformer that utilizes input masked color tokens to restore the masked tokens via training on grayscale image. Also, [33] uses the Vision Transformer as a backbone and effectively upsampling the image through the local stabilizing layer. However, despite the superior performance

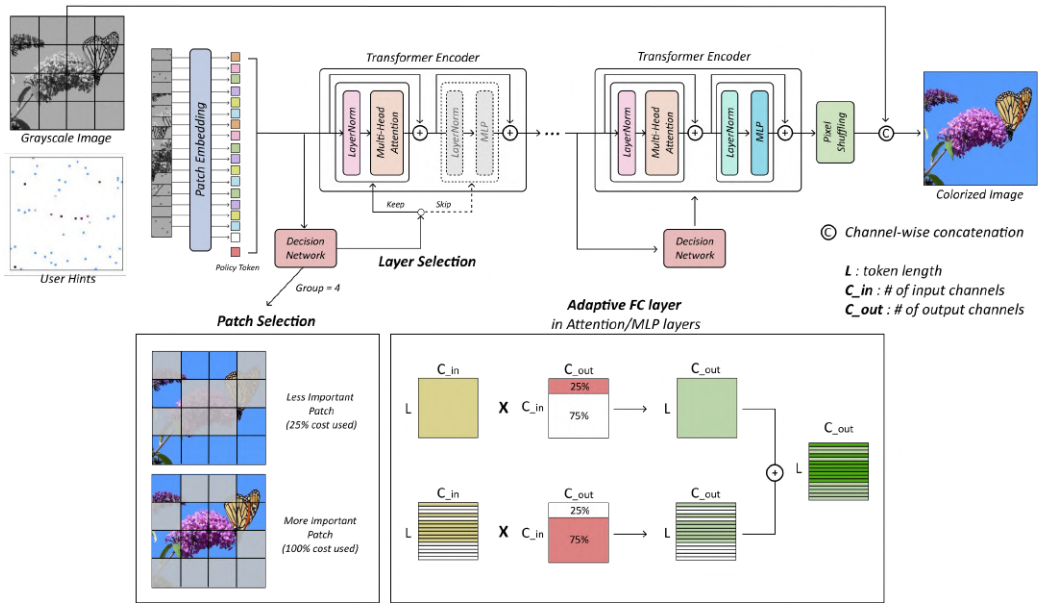

Figure 3. Overview of our proposed colorization pipeline. The main idea of our work is to use a decision network that uses a policy token in making a binary decision to dynamically skip or retain the attention layer in ViT. And, the output of transformer blocks are upsampled via pixel shuffling.

of transformer-based colorization methods, deeply stacked transformer layers are computationally expensive to use in user-interactive applications.

Adaptive Inference in Vision Transformer. Pruning methods have demonstrated considerable performance in reducing model redundancy, while enhancing inference speed. In contrast to static pruning methods, adaptive inference method has demonstrated acceleration in transformer-based models [2, 13, 18, 24, 28, 31]. A-ViT [28] and DynamicViT [20] reduced the redundancy of the model by removing redundant image patches of each input, while AdaViT removed image patches, attention heads, and blocks. However, it is not appropriate to apply the existing methods to image colorization tasks, since most of the previous research completely removes the image patches of ViT. Therefore, we propose a novel adaptive image colorization method that softly removes image patches without the performance degradation.

3. Method

In this work, we propose AdaColViT, an adaptive user-interactive colorization framework to reduce the computational cost of vision Given an input sample, AdaColViT is trained to satisfy the reconstruct error, and obtain desirable computational cost at the same time. An overview of our method is presented in Figure 3.

3.1. Preliminaries

We adopt vision transformer architecture to propagate user hints. Given a colored train image $I_c \in \mathbb{R}^{H\times W\times 3}$, we convert the colored image to grayscale image, $I_g \in \mathbb{R}^{H\times W\times 1}$, by changing RGB color space to CIELab color space and extracting perceptual lightness value L. We generate user hints $I_{hint} \in \mathbb{R}^{H\times W\times 3}$ through masking the non-hint regions with 0 for a, b channels. I_{hint} consists of ab channel and the mask channel that represents user hints where hint-regions have values of 1 and non-hint regions have values of 0. During training, we simulate user hints by determining the hint's location and its corresponding color. Hints are sampled from uniform distribution, as users may provide hints anywhere within the image. The color of user hint is selected via taking the average color values for each channel in adjacent to the hint region. Hence, the final input $X \in \mathbb{R}^{H\times W\times 4}$ is obtained by concatenating grayscale image I_g and hint input I_{hint}. This final input is divided into patches X_p that is fed into transformer encoder. The equation of obtained input X is defined as follows:

$$X = I_g \oplus I_{hint}, \tag{1}$$

where $\oplus$ is channel-wise concatenation.

The vision transformer [4, 22, 23, 32] takes sliced image patches as input and consists of self-attention layers and a feed-forward network. As patch embedding of ViT, the model gets a sequence of embedded tokens $Z \in R^{(N+1)\times C}$ as input I, where N and C denote the sequence length and

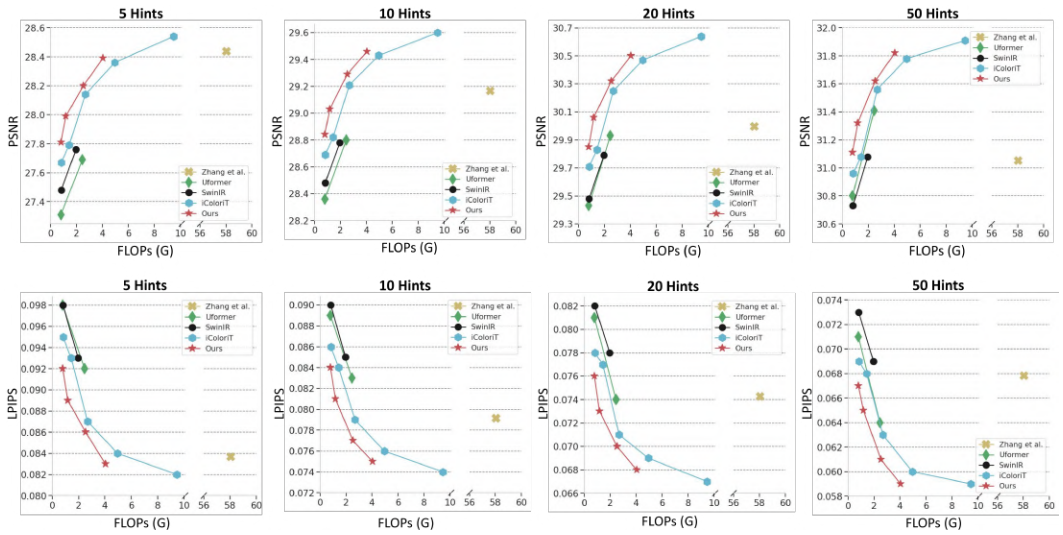

Figure 4. Experimental results of user-guided colorization methods demonstrating the performance (PSNR, LPIPS) with different FLOPs. The result shows the performance of each method according to the number of hints. In particular, the group was set to 2 in AdaColViT.

embedding dimension, respectively. Also, the policy token $Z_{policy} \in R^{1\times C}$ is propagated as an input to the decision network and is included in Z. The input of the model can be demonstrated as follows:

$$Z = [Z_{policy}; Z_1; Z_2; ...; Z_N] + E_{pos}, \tag{2}$$

where E_{pos} represents positional encoding matrix. The single-head attention containing query, key, and value projected from the same input can be computed as below:

$$Attn(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{3}$$

Multi-head self-attention (MSA) concatenates the output from numerous single-head attentions and projects it with another parameter matrix to focus attention more efficiently on various representation subspaces:

$$head_{i,l} = Attn(Z_l W^Q_{i,l}, Z_l W^K_{i,l}, Z_l W^V_{i,l}), \tag{4}$$

$$MSA(Z_l) = Concat(head_{1,l}, ..., head_{H,l})W^O_l, \tag{5}$$

where Z_l stands for the input at the l^{th} block and $W^Q_{i,l}, W^K_{i,l}, W^V_{i,l}$, and W^O_l are the parameter matrices in the i^{th} attention head of the l^{th} transformer block. The output of the MSA is fed into FFN, a two-layer MLP, to create the output of the transformer block Z_{l+1}. Residual connections are applied to MSA and FFN as follows:

$$Z'_l = MSA(Z_l) + Z_l, Z_{l+1} = FFN(Z'_l) + Z'_l. \tag{6}$$

Using the policy token from the previous transformer block (Z^0_L) as inputs, a linear layer generates the final prediction. By rearranging a (H/P, W/P, C×P^2) feature map into the shape of (H, W, C), we use pixel shuffling, an upsampling technique, to create a full-resolution image.

3.2. Decision Network

Each of the decision network at l^{th} attention layer and FFN layer consists of linear layer with parameter W^p_l to produce usage policies for *patch selection*. Moreover, the decision network at l^{th} transformer block consists of linear layer with parameter W^b_l to produce usage policies for *layer selection*.

Giving the input Z_l and Z'_l to l^{th} attention layer and FFN layer respectively, the usage policy matrices for this block is computed as follows:

$$(m^p_l, m^b_l) = (W^p_l Z_l, W^b_l Z_l). \tag{7}$$

where m^p_l and m^b_l denote the usage policies of image patches and transformer block, respectively. m^p_l and m^b_l passed through a *sigmoid* function, indicate the probability of keeping the corresponding input patch and block of the transformer, respectively. Thus, we define M^p_l and M^b_l to make decisions by sampling from m^p_l and m^b_l. In addition, since the binary decisions are non-differentiable, we adopt a Gumbel-Softmax trick [17] to enable backpropagation.

Patch selection module. The decision network distinguishes between less and more informative patches when inputs are fed to the attention layer and the FFN layer.

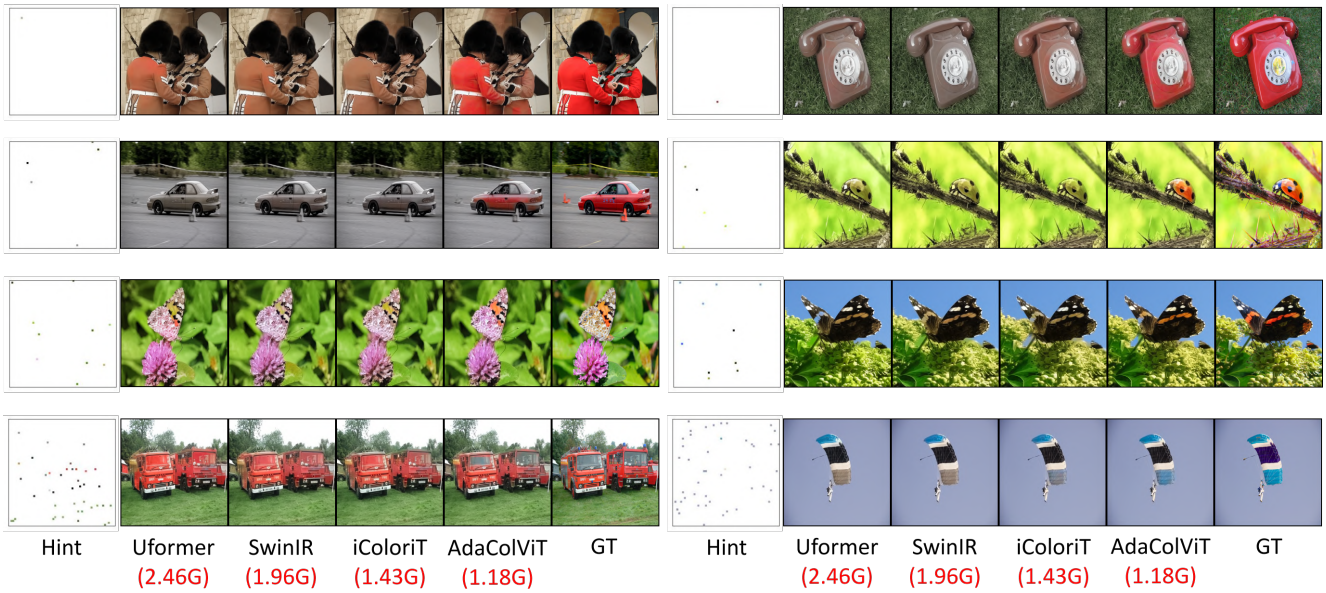

Figure 5. Qualitative visualization result of Uformer, SwinIR, iColoriT, AdaColViT and ground truth image. In each row, the first square shows the number of hints used. We show the results on 1, 5, 10, and 50 number of user hints, respectively. As shown, AdaColViT can generate images equivalent to iColoriT-T, despite the decreased GFLOPs.

The less informative patch requires less computation cost according to the number of groups. For example, when the group is set to 4, the less informative patch only utilizes 25% of the channel to reduce the computational cost. In contrast, if the patch is considered as more informative, the other 75% of channels are computed and added with the previous output from the less informative patch, thus using 100% of the channels in total. Determining the less informative patch and more informative patch from l^{th} attention layer can be demonstrated as follows, and it is considered as the more informative patch if M_l value is 1.

$$Z_l^{**} = [z_{l,policy}; M_{l,1}^p z_1; ...; M_{l,N}^p z_N], \tag{8}$$

$$\begin{aligned} head_{i,l}^{*} &= Attn(Z_l W_{i,l}^{'Q}, Z_l W_{i,l}^{'K}, Z_l W_{i,l}^{'V}), \\ head_{i,l}^{**} &= Attn(Z_l^{**} W_{i,l}^{''Q}, Z_l^{**} W_{i,l}^{''K}, Z_l^{**} W_{i,l}^{''V}), \end{aligned} \tag{9}$$

$$\begin{aligned} MSA(Z_l) = Concat(head_{1,l}^{*}, ..., head_{H,l}^{*}) W_l^{'O} \\ + Concat(head_{1,l}^{**}, ..., head_{H,l}^{**}) W_l^{''O}. \end{aligned} \tag{10}$$

where $Z_l^{''}$ denote the more informative tokens. Also, $W_l^{'}$ and $W_l^{''}$ are masked parameters, and when the group value is 4, $W_l^{'}$ uses only 25% of the embedding dimension and $W_l^{''}$ uses 75%. At this time, the policy token is always maintained and Z_l' is formulated equal to Z_l.

Layer selection module. Using our patch selection method alone is not sufficient to reduce the redundancy of the model. Therefore, when a transformer layer is redundant, that layer can be skipped. In this paper, we dynamically skip the attention layer and the FFN layer according to the input sample. The operation according to skip can be expressed as follows:

$$\begin{aligned} Z_l^{'} &= M_{l,0}^b \cdot Attention(Z_l) + Z_l, \\ Z_{l+1}^{'} &= M_{l,1}^b \cdot FFN(Z_l^{'}) + Z_l^{'}. \end{aligned} \tag{11}$$

3.3. Loss function

Our goal is to optimize overall huber loss [8] and the sparsity loss to train a vision transformer with an ideal target computational cost and minimal performance drop at the same time. The loss function of our AdaColViT can be defined as follows:

$$\begin{aligned} \mathcal{L}_{huber} = \frac{1}{2}(I_{pred} - I_{GT})^2 \mathbb{1}_{|I_{pred} - I_{GT}|<1} \\ + (|I_{pred} - I_{GT}| - \frac{1}{2}) \mathbb{1}_{|I_{pred} - I_{GT}| \geq 1}, \end{aligned} \tag{12}$$

$$\mathcal{L}_{sparsity} = (\frac{1}{L}\sum_{l=1}^{L} M_l^p - \beta_1)^2 + (\frac{1}{L}\sum_{l=1}^{L} M_l^b - \beta_2)^2, \tag{13}$$

Dataset	Method	FLOPs (G)	PSNR@10 ↑	LPIPS@10 ↓
Oxford 102flowers	Uformer-S	2.46	24.64	0.132
	SwinIR-S	1.96	24.64	0.131
	iColoriT-T	1.43	24.67	0.130
	AdaColViT-T (Ours)	**0.78**	**24.71**	**0.128**
Cub-200	Uformer-S	2.46	29.56	0.092
	SwinIR-S	1.96	29.57	0.092
	iColoriT-T	1.43	29.60	0.090
	AdaColViT-T (Ours)	**0.78**	**29.63**	**0.089**

Table 1. Performance of the model trained with ctest10k, evaluated on the Oxford 102flowers and CUB-200 datasets, which are frequently used in existing colorization tasks.

Methods	Latency	FLOPs (G)	PSNR@10
Uformer-S	$472\text{ms}_{\pm 6\text{ms}}$	2.46	$28.79_{\pm 0.02}$
SwinIR-S	$97\text{ms}_{\pm 8\text{ms}}$	1.96	$28.78_{\pm 0.01}$
iColoriT-T	$61\text{ms}_{\pm 6\text{ms}}$	1.43	$28.81_{\pm 0.01}$
AdaColViT-T (Ours)	$\mathbf{34ms}_{\pm 8\text{ms}}$	$\mathbf{0.78}_{\pm 0.01}$	$\mathbf{28.83}_{\pm 0.01}$
AdaColViT-T-d24 (Ours)	$\mathbf{48ms}_{\pm 7\text{ms}}$	$\mathbf{1.18}_{\pm 0.01}$	$\mathbf{29.03}_{\pm 0.01}$

Table 2. Comparison between actual acceleration (Throughput and Speed up) and theoretical acceleration (GFLOPs) of Uformer, SwinIR, iColoriT, and AdaColViT. Also, the group was set to 2 in AdaColViT. We repeatedly ran our network 5 times and measured the corresponding mean and std values.

$$\mathcal{L} = \mathcal{L}_{huber} + \mathcal{L}_{sparsity}. \tag{14}$$

where L, $\mathcal{L}_{huber}$, and $\mathcal{L}_{sparsity}$ represent the number of transformer layers, huber loss, and sparsity loss. Also, the hyperparameters β_1 and β_2 are the target computation budgets with values between 0 and 1, which can adjust the remaining ratio of patches and transformer blocks, respectively. Moreover, we determined the values of β_1 and β_2 by considering model performance and computational cost.

4. Experiments

Experimental settings. In training, we use ViT [4] as the transformer backbone for equitable comparison with iColoriT [33]. First, we set the image size to 224×224 and use 8 GPUs with 1024 batches. Second, we use patch size of $P = 16$ with sequence length N of 196. Moreover, we use AdamW optimizer [16] with 0.004 learning rate, a weight decay 0.05 and a cosine annealing scheduler [15] for 50 epochs.

Methods	FLOPs (G)	PSNR@10
DynamicViT	1.24	28.75
AdaViT	1.20	28.81
A-ViT	1.23	28.74
AdaColViT-T (Ours)	**1.18**	**29.03**

Table 3. Comparison with efficient ViTs: DynamicViT, AdaViT, A-ViT, and AdaColViT. Also, the group was set to 2 in AdaColViT.

FLOPs (G)	# of Groups	PSNR@10 ↑	LPIPS@10 ↓
0.74	2	**28.71**	**0.085**
0.74	4	28.66	0.086
0.75	8	28.67	0.086

Table 4. Performance according to the number of groups used in AdaColViT-T.

Baselines. We compare the performance of our AdaColViT with iColoriT [33], a recent interactive colorization method based on Vision Transformer. iColoriT used ViT-tiny, ViT-small, and the ones that scaled depth to 6 and 24 as baselines: iColoriT-tiny-d{6,24}, and iColoriT-small-d{6,24}. Moreover, we compare our model with SwinIR [14] and Uformer [25], which use Transformer-based architecture for image restoration. We trained the SwinIR and Uformer models by referring to the official implementation, and we named them SwinIR-S, SwinIR-T, Uformer-S, and Uformer-T as we adjusted the width of the model to compare performance. Moreover, with regards to the naming convention of our model, the x in our model name, AdaColViT-T-dx refres to the depth of the model (i.e. AdaColViT-T-d24 has 24 layers).

Datasets. To extensively explore model scalability, we utilize ILSVRC-2012 ImageNet dataset with 1.3M images and $1,000$ classes for training. We used 10,000 images for test set (also referred as ImageNet ctest10k). ImageNet ctest10k [11] is a subset of ImageNet that is used as a benchmark for colorization tasks. To further evaluate the performance of our model, we also selected CUB-200 dataset [26] and Oxford 102 Flower dataset [19] with $5,794$ test images of 200 classes and $1,000$ flower images of 102 classes, respectively.

Evaluation metric. To quantitatively evaluate the performance of our method, we measure and compare PSNR and learned perception image patch similarity LPIPS [35] between the ground truth and the output image. Also, to

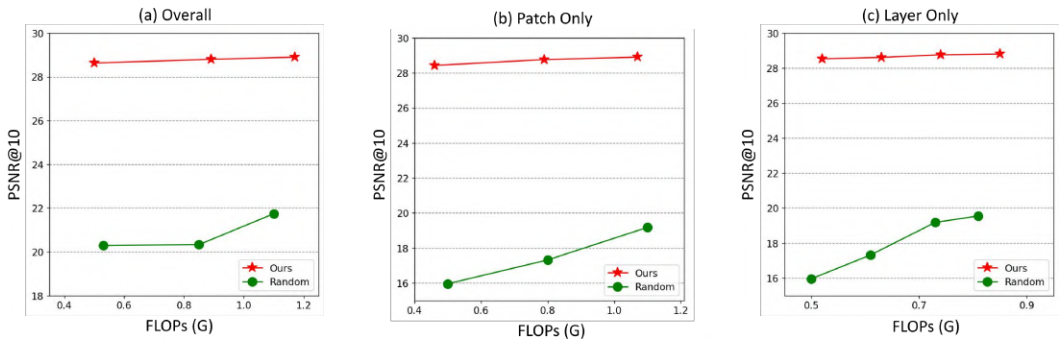

Figure 6. Comparison between 3 methods of AdaColViT to demonstrate the effect of each component. In this figure, (a) patch selection method and layer selection method; (b) patch selection method; (c) layer selection method are compared with the random selection method, respectively.

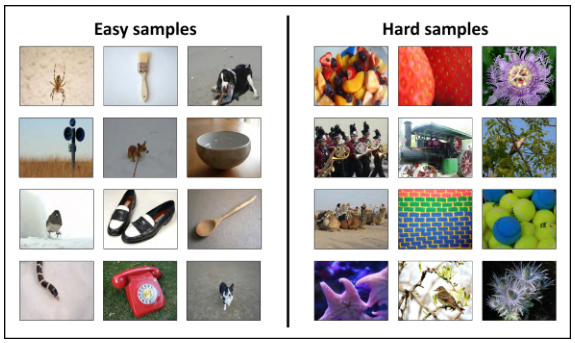

Figure 7. A set of sample images that require the least and most computation.

evaluate and compare model efficiency, we mention then number of giga floating-point operations (GFLOPs).

Quantitative results. In Figure 4, we provide quantitative results in ImageNet-ctest10k. We set the number of hints to 5, 10, 20, and 50 and compared the models with PSNR, LPIPS, and FLOPs, respectively. We compared our method with Uformer-S, Uformer-T, SwinIR-S, SwinIR-T, iColoriT-S-d24, iColoriT-S, iColoriT-S-d6, iColoriT-T, iColoriT-T-d6, AdaColViT-S-d24, AdaColViT-S, AdaColViT-T-d24, and AdaColViT-T. Our model showed significantly better performance with 5, 10, 20, and 50 hints than other baseline models. Also, we provide additional quantitative results in two datasets: Oxford 102flowers and CUB-200, respectively. In Table 1, our model with 10 hints showed better performance than other baseline models. In other words, AdaColViT , which removed the width and depth redundancy of model was more effective than other baselines. This demonstrates the effectiveness of our method that removes the less informative tokens and layers.

Qualitative results. We provide the visualization of the qualitative results in Figure 5. Given a test grayscale image, our goal is to reproduce a realistic colorized image that is equivalent to the ground truth. The results illustrate that the colorized output of ours is most similar to the ground truth image relative to other methods, with respect to the quality of the produced result.

Actual acceleration. Table 2 demonstrates the latency, GFLOPs, and PSNR of Uformer-S, SwinIR-S, iColoriT-T AdaColViT-T, and AdaColViT-T-d24. Latency was measured by the CPU and to provide a fair comparison, we experimented with only single thread. AdaColViT-T-d24 achieved the higest PSNR value and 47ms of latency with only 1.18 of GFLOPs, which is more than 10× faster than the comparison method Uformer-S. In particular, AdaColViT-T outperformes icoloriT-T by 1.75× faster, showing better PSNR value 28.83 with only 0.79 GFLOPs. In conclusion, our method reduced theoretical FLOPs with performance enhancement while achieving actual acceleration.

Patch removal method. Table 3 demonstrates a performance comparison between methods that reduce model redundancy through patch removal. To compare with baselines, we made slight modifications to the publicly available implementation codes. Existing patch removal methods permanently delete image patches, resulting in significant performance degradation. However, our method excels in performance compared to existing methods as we softly remove image patches.

5. Discussion

Number of groups. Table 4 demonstrates PSNR and LPIPS values according to the number of groups of AdaColViT with different FLOPs. The model with 2 groups showed the best PSNR value of 28.71 at 0.74 GFLOPs. Therefore, we conduct our experiment by setting the number of groups to 2.

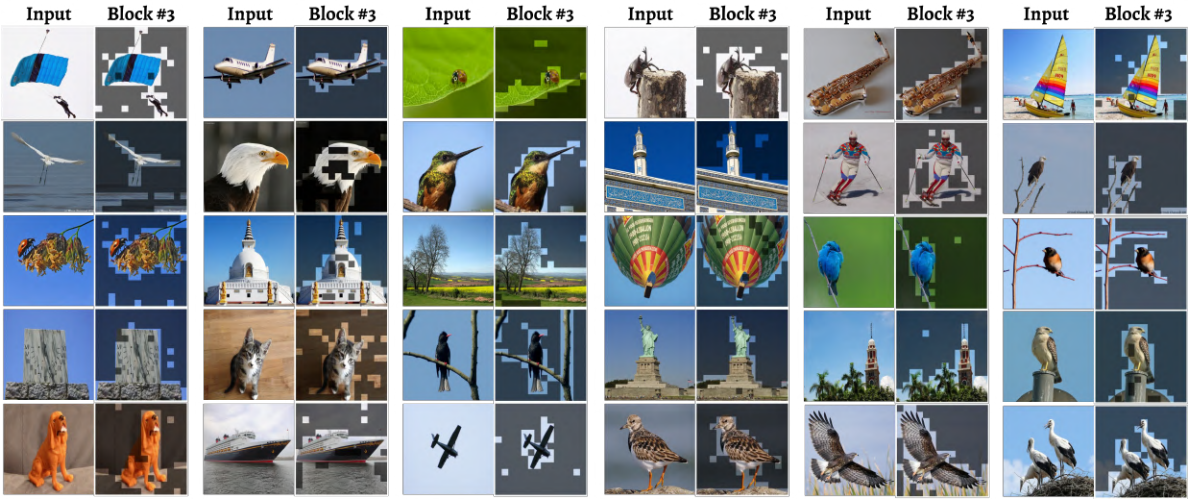

Figure 8. Visualization of tokens from the third block of our AdaColViT over the ctest10k dataset.

Effectiveness of each component. Figure 6 demonstrates that AdaColViT effectively adjusts computational budgets according to hyperparameters β_1 and β_2. In addition, performance changes between each selection methods represent the effectiveness of our method compared to random selection.

Allocation of computational resources. To validate the result whether we have adjusted the computational cost appropriately for difficulty of each input, we visualize example images that take the least and most computation in Figure 7. For least computed images, the images do not show diverse colors and illustrate only a single object (i.e. airplane, bird). Furthermore, background of least computed images are mostly plain background with simple pattern. For images that take most computation, multiple objects appear in a single image. For example, in the 3^{rd} row, image of people riding camels have multiple mixed objects with various colors. This makes the model more challenging resulting in taking more computation compared to the easy images.

Visualization. Figure 8 illustrates the tokens of the third block's MLP layer of AdaColViT that are adaptively pruned during inference over the ctest10k image samples. Remarkably, our AdaColViT shows high efficiency by removing redundant tokens and their corresponding computations, only focusing on relatively important regions of images. For example, airplane image at the 1^{st} and 5^{th} rows mainly retains the plane object while using fewer tokens in the sky. In addition, the images with birds on trees at the 2^{th}, 3^{th}, 4^{th}, and 5^{th} rows distinctly keep the bird and tree region while removing the simple background. The result demonstrates that adaptive tokens from AdaColViT can effectively focus on relevant objects and efficiently reduce the computational cost.

6. Conclusion

In this work, we present AdaColViT, an adaptive vision transformer for real-time interactive colorization. Our approach adaptively reduces the amount of computation of less informative patches and vision transformer layers based on the difficulty of input samples. To achieve this, we use a trainable decision network to determine more important patches and layers in the transformer architecture. Our extensive experiments demonstrate that our method significantly reduces computational costs while maintaining performance.

7. Acknowledgements

This work was supported by SKT AI Fellowship funded by SK Telecom (No. 4 Development of deep learning-based image colorization technology). This work was partly supported by Institute for Information & communication Technology Planning & evaluation (IITP) grants funded by the Korean MSIT: (No. 2022-0-01199, Graduate School of Convergence Security at Sungkyunkwan University), (No. 2019-0-00421, AI Graduate School Support Program at Sungkyunkwan University) and (No. RS-2023-00230337, Advanced and Proactive AI Platform Research and Development Against Malicious deepfakes). Also, this work was supported by Korea Internet & Security Agency(KISA) grant funded by the Korea government (PIPC) (No.RS-2023-00231200, Development of personal video information privacy protection technology capable of AI learning in an autonomous driving environment).

References

[1] Yunpeng Bai, Chao Dong, Zenghao Chai, Andong Wang, Zhengzhuo Xu, and Chun Yuan. Semantic-sparse colorization network for deep exemplar-based colorization. In *European Conference on Computer Vision*, pages 505–521. Springer, 2022. 2

[2] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021. 3

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3, 6

[5] Mingming He, Dongdong Chen, Jing Liao, Pedro V Sander, and Lu Yuan. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)*, 37(4):1–16, 2018. 2

[6] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019. 2

[7] Zhitong Huang, Nanxuan Zhao, and Jing Liao. Unicolor: A unified framework for multi-modal colorization with transformer. *arXiv preprint arXiv:2209.11223*, 2022. 2

[8] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992. 5

[9] Xiaozhong Ji, Boyuan Jiang, Donghao Luo, Guangpin Tao, Wenqing Chu, Zhifeng Xie, Chengjie Wang, and Ying Tai. Colorformer: Image colorization via color memory assisted hybrid-attention transformer. In *European Conference on Computer Vision*, pages 20–36. Springer, 2022. 2

[10] Manoj Kumar, Dirk Weissenborn, and Nal Kalchbrenner. Colorization transformer. *arXiv preprint arXiv:2102.04432*, 2021. 2

[11] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016. 6

[12] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, pages 689–694. 2004. 1, 2

[13] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 3

[14] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 6

[15] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[17] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016. 2, 4

[18] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12309–12318, 2022. 1, 3

[19] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 6

[20] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021. 1, 3

[21] Jheng-Wei Su, Hung-Kuo Chu, and Jia-Bin Huang. Instance-aware image colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7968–7977, 2020. 2

[22] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 3

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[24] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021. 3

[25] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17683–17693, 2022. 6

[26] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 6

[27] Hui Yin, Yuanhao Gong, and Guoping Qiu. Side window filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8758–8766, 2019. 1, 2

[28] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for

efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10809–10818, 2022. 1, 3

[29] Wang Yin, Peng Lu, Zhaoran Zhao, and Xujun Peng. Yes," attention is all you need", for exemplar based colorization. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2243–2251, 2021. 2

[30] Seungjoo Yoo, Hyojin Bahng, Sunghyo Chung, Junsoo Lee, Jaehyuk Chang, and Jaegul Choo. Coloring with limited data: Few-shot colorization via memory augmented networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11283–11292, 2019. 2

[31] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10819–10829, 2022. 3

[32] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021. 3

[33] Jooyeol Yun, Sanghyeon Lee, Minho Park, and Jaegul Choo. icolorit: Towards propagating local hint to the right region in interactive colorization by leveraging vision transformer. *arXiv preprint arXiv:2207.06831*, 2022. 1, 2, 6

[34] Lvmin Zhang, Yi Ji, Xin Lin, and Chunping Liu. Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan. In *2017 4th IAPR Asian conference on pattern recognition (ACPR)*, pages 506–511. IEEE, 2017. 2

[35] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[36] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017. 1, 2