Adversarial Normalization: I Can visualize Everything (ICE)

Hoyoung Choi*
Hanyang University
choiho00@hanyang.ac.kr

Seungwan Jin*
Hanyang University
seungwanjin@hanyang.ac.kr

Kyungsik Han
Hanyang University
kyungsikhan@hanyang.ac.kr

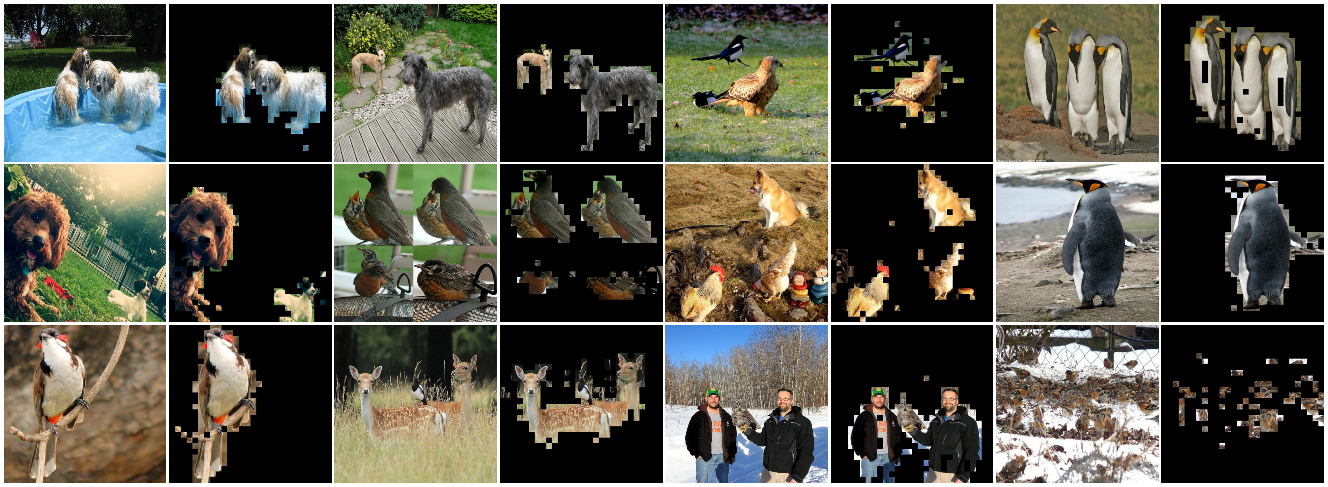

Figure 1. Semantic segmentation maps highlighted by image classes and background predictions for each patch in the images. These maps show that ICE visualizes class-specific explainability of DeiT-S [33], leading to unsupervised foreground and background segmentation.

Abstract

*Vision transformers use [CLS] tokens to predict image classes. Their explainability visualization has been studied using relevant information from [CLS] tokens or focusing on attention scores during self-attention. Such visualization, however, is challenging because of the dependence of the structure of a vision transformer on skip connections and attention operators, the instability of nonlinearities in the learning process, and the limited reflection of self-attention scores on relevance. We argue that the output vectors for each input patch token in a vision transformer retain the image information of each patch location, which can facilitate the prediction of an image class. In this paper, we propose **ICE** (Adversarial Normalization: **I** **C**an visualize **E**verything), a novel method that enables a model to directly predict a class for each patch in an image; thus, advancing the effective visualization of the explainability of a vision transformer. Our method distinguishes background from foreground regions by predicting background classes for patches that do not determine image classes. We used the DeiT-S model, the most representative model employed in studies, on the explainability visualization of vision transformers. On the ImageNet-Segmentation dataset, ICE outperformed all explainability visualization methods for four cases depending on the model size. We also conducted quantitative and qualitative analyses on the tasks of weakly-supervised object localization and unsupervised object discovery. On the CUB-200-2011 and PASCALVOC07/12 datasets, ICE achieved comparable performance to the state-of-the-art methods. We incorporated ICE into the encoder of DeiT-S and improved efficiency by 44.01% on the ImageNet dataset over that achieved by the original DeiT-S model. We showed performance on the accuracy and efficiency comparable to EViT, the state-of-the-art pruning model, demonstrating the effectiveness of ICE. The code is available at https://github.com/Hanyang-HCC-Lab/ICE.*

1. Introduction

The emergence of vision transformers in the field of computer vision has driven improvements in model performance [2, 5]. Unlike a CNN model, a vision transformer learns the association between image patches and

[*] Both authors contributed equally to this research

classifies images using a [CLS] token. A CNN model and a vision transformer have structural differences that lead to variances in explainability visualization approaches. A representative approach is GradCAM, which demonstrates the explainability of CNN models by reflecting the importance of pixel levels using the feature maps and gradients of the models. However, it is somewhat difficult to effectively apply GradCAM to vision transformers because the structural characteristics of vision transformers pose several challenges, such as skip connections, dependency on attention operators, and unstable learning due to non-linearities.

To overcome these challenges, previous research mainly used attention score information between [CLS] tokens and other patches to discriminate patches with a significant impact on learning and visualizing the explainability of vision transformers [2, 35]. Later studies have evaluated the degree to which each attention head contributes to performance [36] or integrated the relevance and attention scores in layers through the proposal of a relevance propagation rule [3]. Most recently, the optimization of relevance maps has improved the explainability of a vision transformer by assigning a lower relevance to the background region of an image, whereas high relevance is placed on the foreground region [4]. Despite the advantage of this optimization, challenges to explainability visualization for vision transformers remain given their structural characteristics [3, 4].

We note that the output embedding vectors for each input patch token in a vision transformer retain the image information of each patch location, and these vectors can help predict image classes. Based on this motivation, in this paper, we propose **ICE** (**I** **C**an visualize **E**verything), a novel method that uses the output embedding vectors of a vision transformer for each patch token, except for [CLS] tokens, in visualizing explainability. ICE initially assumes that the class of all patches is a background and gradually learns the direction in which the class of each patch in an image is predicted. With this approach, we propose a loss function for adversarial normalization that combines background and classification losses for each patch token. ICE predicts a class for each patch in a foreground region of an image where the object of the class is likely to exist and classifies other regions as a background.

To evaluate the explainability visualization performance of ICE, we mainly used DeiT-S [33], pre-trained with ImageNet [25], the most representatively adopted model in previous studies. On the ImageNet-Segmentation [14] dataset, ICE (with DeiT-S) achieved improvements of 4.05% and 3.94% in pixel-wise accuracy and mean intersection over union (mean IoU), respectively, compared with state-of-the-art methods. To verify the scalability and robustness of ICE, we additionally considered ViT AugReg (AR) [31] and evaluated ICE for four cases depending on the model size (i.e., Small and Tiny). Through qualitative analyses, we showed that ICE was good at predicting not only the class of a single object but also the same class of multiple objects in an image. We found that other methods failed to segment objects, especially in multi-object conditions.

We further evaluated the foreground and background separation performance of ICE on unsupervised semantic segmentation, weakly supervised object localization, and unsupervised object discovery tasks using the PascalVOC07/12 [11] validation sets and the CUB-200-2011 [37] dataset. As a result, ICE (with DeiT-S) achieved comparable and superior performance compared to the existing self-supervised learning-based methods (i.e., DINO [7], DINO-based LOST [8], and DINO-based TokenCut [9]). We found that ICE could distinish between background and foreground regions despite the presence of multiple objects of different sizes and classes that were not learned in the images of PascalVOC07/12 (Figure 1).

Regarding our experiment in efficiency on inference, we incorporated ICE into the encoder of DeiT-S and achieved an improved efficiency of 44.01% on ImageNet [25] compared with the original DeiT-S, while maintaining comparable accuracy. ICE also achieved accuracy and efficiency comparable to that of EViT, the state-of-the-art pruning method [20].

Our contributions are as follows.

- We propose ICE that can be employed to vision transformers based on the notion of *patch-wise classification* and *adversarial normalization*. DeiT-S models with ICE and ICE-f improve class-specific explainability visualization performance (Section 4.2).
- We show that ICE significantly improves foreground and background separations over the original DeiT-S and. Even without segmentation or object location labels, ICE achieves comparable or superior performance than existing self-supervised learning methods (Sections 4.3 and 4.4).
- We demonstrate that ICE is effective in background patch selection by showing comparable efficiency and accuracy of DeiT-S that incorporates the ICE's capability to its encoder, to EViT (Section 4.5).

With the experimental results as grounding, we discuss the scalability of our methodology in terms of improving the efficiency of a vision transformer-based model.

2. Related Work

The visualization methodologies applicable to vision transformers can be divided into two: (a) gradient or attribute propagation-based visualization, which is applied primarily to CNN-based models, and (b) visualization methods that consider transformer structure.

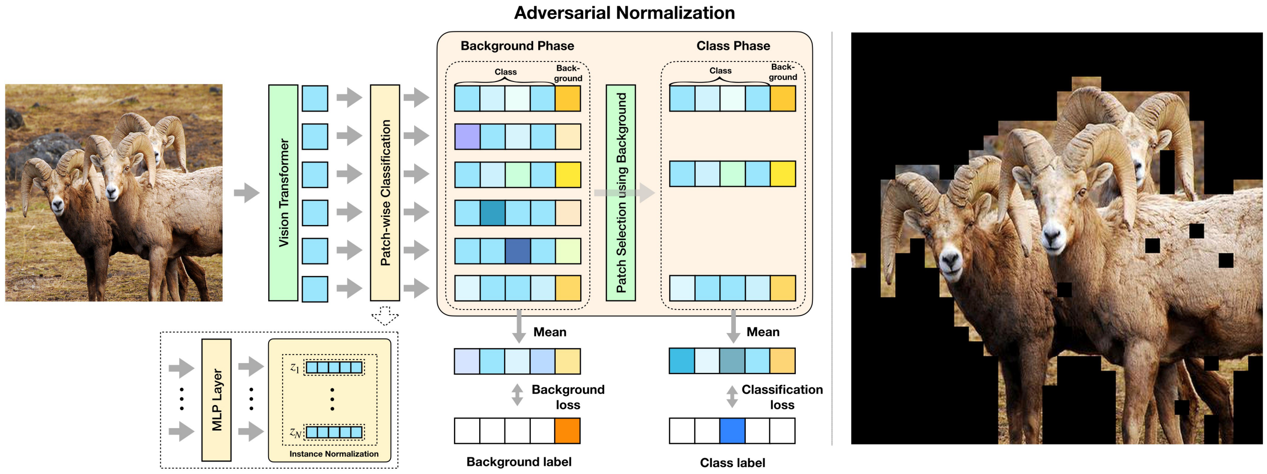

Figure 2. The overview of ICE. First, ICE performs *patch-wise classification* using an output vector for each patch token of a vision transformer. We set an additional background label, and ICE predicts the total number of [classes + one background class] in patch-wise classification. Next, ICE is trained to distinguish the background and the class region of an image through *adversarial normalization*. The figure on the right shows an explainability visualization result of ICE predicting for the class 'ram, tup' of ImageNet. ICE has visualization characteristics that highlight overall explainable regions with high relevance to a class label as much as possible.

2.1. Explainability in Computer Vision

The gradient-based method normally used in CNN uses gradients calculated for each layer through backpropagation. Initially, studies proposed explainability visualization that uses input multiplied by gradient within a model learning process for image classification tasks [29]. Later research advocated visualization that adopts the average value of gradients [30, 32]. However, these methods are class-agnostic in visualizing explainability, regardless of the predicted class. Among gradient-based methods, a representative class-specific approach is GradCAM [26]. GradCAM uses a weighted gradient feature map that considers the gradients and input features of a network layer. However, GradCAM has not been effectively applied to explainability visualization for vision transformers because of the structural nature of the transformers, which classify image classes using [CLS] tokens [3].

The attribute-based method, another methodology used in CNN, visualizes a model's explainability by exploiting the contribution decomposition of previous layers from prediction to input. A typical approach in this respect is Layer-wise Relevance Propagation (LRP) [36], which is based on a method of propagating the relevance score obtained from a predicted class to an input image. Other attribute-based methods include RAP [22], AGF [15], DeepLIFT [28], and DeepSHAP [21]; however, all these have class-agnostic characteristics. Methods that are attribute-based and have class-specific characteristics include Contrastive-LRP (CLRP] [13]) and Softmax-Gradient-LRP (SGLRP [17]), whose applicability is constrained by the fact that they visualize LRP propagation results for a class, which are contrasted with the results of all other classes to highlight differences between classes.

Various visualization methods have been successfully applied to CNN. However, they still have unoptimized properties for the structural characteristics of vision transformers that utilize [CLS] tokens in prediction. They also deal with discrete tokens of input data. In this work, we evaluated the performance of explainability visualization by comparing it with GradCAM, a class-specific method and one of the most effective CNN-oriented approaches to visualizing the explicability of vision transformers. Since our method directly visualizes class-specific explainability without additional contrasting stages, we did not compare the performance of our method with those that are attribute-based and have class-specific characteristics (e.g., CLRP, SGLRP).

2.2. Explainability for Vision Transformers

Existing studies on explainability visualization for vision transformers focused on attention scores [1, 5, 7]. However, using information from raw attention poses limitations to the complete use of the structural characteristics of vision transformers, which include multiple learning modules [24]. Given the nature of a transformer layer, information is continuously mixed according to layer, thus causing difficulties in effectively applying explainability visualization that relies only on attention scores to vision transformers [1]. The Rollout [1] method involves quantifying information on radio waves from the input layer to the prediction layer. It assumes that attentions are linearly combined into subsequent contexts. However, this often tends to highlight unrelated tokens.

Partial LRP [11] visualizes explainability by reflecting relevance scores, meaning that individual attention heads of a vision transformer-based encoder contribute to the overall performance of a model. However, the relevance score of each attention head does not reflect the propagation from prediction to model input, suggesting that relevance scores are insufficiently reflected in a model's explainability. Chefer et al. (2021) proposed several methods (e.g., relevance propagation rule, integration of propagation information, relevance, and attention scores) and solved some issues due to dependence on non-positive values and skip connections propagated in the learning process caused by the structural characteristics of vision transformers [3]. Subsequently, RobustViT was developed to visualize explainability by assigning low relevance to the background region in the image and optimizing the relevance map to assign high relevance to the foreground region [4]. However, our quantitative and qualitative analyses showed that RobustViT does not adequately highlight the foreground region that determines the classes of images.

Although many methodologies have been proposed to effectively visualize explainability using information from the prediction layer of a vision transformer to the input layer, previous studies faced challenges because of the structural characteristics of such vision transformers. In this paper, we propose ICE, a novel method that visualizes the explainability of a vision transformer by directly predicting classes of foreground and background regions for each image patch. Section 4 recounts our comparison of ICE's quantitative performance with that of other explainability visualization methodologies and performs the qualitative analysis involving visualization examples.

3. Method

In this section, we propose the fine-tuning process of ICE that separates background patches by comparing the background and class probability of each patch token, and a loss function for adversarial normalization. Figure 2 illustrates the overview of ICE, and Algorithm 1 presents pseudocode implementation. We introduce a background label that is less relevant to an image class. We intend to have all patch tokens receive gradients of a background label continuously during training. The model learned the background classification from the foreground of ICE by performing adversarial normalization which handles both background and classification losses.

3.1. Patch-wise Classification

To normalize the prediction probability distributions obtained from each patch token that passed MLP, we conducted instance normalization. By making the probability of patches into one vector, we can get cross-entropy loss that all patches were affected by gradients. We constructed our model by adding a lightweight MLP to the structure of a standard vision transformer [5] model without considering a [CLS] token. Vision transformers use linear projection and patch embedding for input images to embed $k \times k$ image grids into patch $Z \in \mathbb{R}^{k^2 \times d}$ $(k^2 = N)$.

Algorithm 1 ICE PyTorch pseudocode

```
Input: Mini batch of images
  # H: cross-entropy loss,    # $\alpha$: background scale
  # B: ground truth for background phase Y: ground truth for class phase
  for $x$ in data loader:
    $logits$ = model($x$)[:,1:]
    $logits$ = Instance Normalization(MLP($logits$))
    $masks$ = argmax($logits$, dim=2)
    $class$ = $logits$.maskedfill($masks$==$(c+1)$, dim=2)
    $\hat{Y}$ = sum($class$, dim=1)/sum($masks$!=$(c+1)$, dim=1)
    $\hat{B}$ = $logits$.mean(dim=1) $\times$ $\alpha$
    $L_{class}$ = H($Y$, $\hat{Y}$)
    $L_{bg}$ = H($B$, $\hat{B}$)
    $L_{total}$ = $L_{class} \times \lambda_{class}$ + $L_{bg} \times$ (1 - $\lambda_{class}$)
```

$$Z = [z_1; z_2; ...; z_N] \tag{1}$$

The number of classes is c, and a background class is added, having a total of $c + 1$ classes. Then the d dimension patch is transformed to $c + 1$ dimension through MLP, making each of Z predict $c + 1$ classes. Since the sum of the class probabilities predicted by each patch is different, we used *Instance Normalization* [34] to normalize the probabilities predicted by each patch token.

$$Z = InstanceNormalization(MLP(Z)) \tag{2}$$

where $Z \in \mathbb{R}^{N \times (c+1)}$.

3.2. Adversarial Normalization

We introduce adversarial normalization which distinguishes between a class and a background by normalizing class prediction probability of each patch and reflecting background probability adversarially. Adversarial normalization normalizes prediction probability of each patch token, making all patches have prediction results, and at the same time, reflects the probability of a background, leading unnecessary patch tokens being trained as a background. Adversarial normalization consists of two phases. All patches were considered to be a background (background phase) and some patches became class related patches by comparing the probabilities between a class and a background (class phase).

Background phase. Using the average of Z, we can get one vector that reflects all patch tokens. We multiplied the average of Z by the scale α, which helps to maintain a degree of loss, given that there are generally more patch tokens in a background region than those in the objects. Since we experimentally found that background loss becomes significantly small, we used α in order to continuously propagate the probability that all patch tokens can be background and sufficiently operate the background loss function. We

Table 1. Segmentation performance on the ImageNet-Segmentation [14] dataset.

Model	GradCAM [26]		rollout [1]		Partial LRP [11]		Chefer et al. (2021) [3]		RobustViT [4]		ICE-f (Ours)		ICE (Ours)	
	Pixel-wise accuracy	Mean IoU	Pixel-wise accuracy	Mean IoU	Pixel-wise accuracy	Mean IoU	Pixel-wise accuracy	Mean IoU	Pixel-wise accuracy	Mean IoU	Pixel-wise accuracy	Mean IoU	Pixel-wise accuracy	Mean IoU
DeiT-S	64.33	41.54	66.84	47.85	65.76	43.37	79.30	60.60	80.80	64.00	81.09	62.12	**84.85**	**67.94**
DeiT-Tiny	70.96	48.26	70.71	52.50	67.15	44.42	79.53	60.64	80.73	63.67	82.37	65.09	**84.28**	**67.58**
AR-S [31]	67.67	41.17	68.12	48.90	72.90	51.85	80.85	63.60	83.30	67.70	82.54	66.67	**83.57**	**68.37**
AR-Tiny [31]	72.98	47.12	73.45	55.56	75.18	53.31	78.31	59.24	79.35	61.83	78.02	61.28	**82.35**	**65.91**

experimentally found optimal α=0.5. A background label is a one-hot encoded vector with only the last class as the true value. Background loss continuously propagates gradients to reflect a possibility that all patch tokens are background. We can get background loss by calculating cross entropy between $\hat{B} \in \mathbb{R}^{1\times(c+1)}$ and a background label B.

$$\hat{B} = \frac{1}{N}\sum_{i=1}^{N} z_i \times \alpha, \quad z_i \in \mathbb{R}^{1\times(c+1)} \tag{3}$$

$$L_{bg} = CrossEntropy(B, \hat{B}) \tag{4}$$

Class phase. We selected z_i associated with the class by constructing a binary decision mask $\hat{D}_i \in \{0, 1\}$. The decision mask of the patch that predicted the background the highest becomes 0; otherwise, 1. Since we randomly initialized the parameters of the MLP layers, the initial binary decision mask was also determined randomly. Most decision masks were 0 at the beginning of training, but the number of class patch tokens gradually increased as training continued.

$$\hat{D}_i = \begin{cases} 0, & \text{if } c+1 \ = argmax(z_i), \quad z_i \in \mathbb{R}^{1\times(c+1)} \\ 1, & \text{otherwise} \end{cases} \tag{5}$$

We assumed that non-background patch tokens are associated with classes. We thus averaged these Z and make one vector $\hat{Y}$. Since $\hat{Y}$ has the $c+1$ dimension and one-hot encoded class label Y has the c dimension, we changed Y to have $c+1$ dimension by adding the background class 0 at the end of Y in the one-hot form. Except for the less associated patch tokens, the average of the remaining patch tokens was calculated as one vector, $\hat{Y} \in \mathbb{R}^{1\times(c+1)}$, and L_{class} calculated through cross entropy between $Y \in \mathbb{R}^{1\times(c+1)}$, and $\hat{Y} \in \mathbb{R}^{1\times(c+1)}$.

$$\hat{Y} = \frac{\sum_{i=1}^{N} \hat{D}_i z_i}{\sum_{i=1}^{N} \hat{D}_i}, \quad \hat{D} \in \mathbb{R}^{N\times 1}, \quad z_i \in \mathbb{R}^{1\times(c+1)} \tag{6}$$

$$L_{class} = CrossEntropy(Y, \hat{Y}) \tag{7}$$

We can get the final loss L_{total} by adding L_{class} and L_{bg} with the class weight, λ_{class}.

$$L_{total} = L_{class} \times \lambda_{class} + L_{bg} \times (1 - \lambda_{class}) \tag{8}$$

Table 2. Extensive segmentation performance evaluation using the ImageNet-Segmentation dataset. RobustViT+GradCAM and RobustViT+ICE refer to the models where each visualization method is applied with the fine-tuned model through RobustViT.

Model	RobustViT		RobustViT + GradCAM		RobustViT + ICE (Ours)	
	Pixel-wise accuracy	Mean IoU	Pixel-wise accuracy	Mean IoU	Pixel-wise accuracy	Mean IoU
DeiT-S	80.80	64.00	69.01	44.39	**85.48**	**71.35**
DeiT-Tiny	80.73	63.67	75.07	51.42	**84.42**	**67.67**
AR-S	**83.30**	**67.70**	74.35	52.60	82.14	66.80
AR-Tiny	77.57	59.42	73.55	47.92	**82.56**	**66.16**

4. Experiments

4.1. Experimental Setting

Datasets. To train ICE, we utilized the ImageNet [25] (ILSVRC) 2012 training dataset. We did not use any additional data related to segmentation maps or object locations. To evaluate the explainability visualization performance of ICE, we used four datasets, ImageNet-Segmentation [14], CUB-200-2011 [37], and PascalVOC07/12 [10, 11].

Implementation details. For training strategies and optimization methods of ICE, we employed the pre-trained DeiT-S and set hyperparameters as follows: background scale α=0.5, class weight λ_{class}=0.975, learning rate=$1e^{-5}$, and batch size=256. We followed other hyperparameters specified in the official DeiT repository [1]. We set a class weight λ_{class}=0.99 in order to train ICE-f, a method that freezes the parameters of DeiT-S and trains only two MLP layers. Our intention in considering ICE-f was to verify that output vectors for each input patch token in a vision transformer retain the image information of each patch location, which can facilitate the prediction of an image class. The other hyperparameters were the same as those applied to ICE. To visualize the explainability of other methods, we used the official repository of Chefer et al. (2021) [3] [2], Chefer et al. (2022) [4] [3]. We ran our experiments on the machine equipped with two NVIDIA RTX3090 GPUs.

4.2. Explainability on ImageNet

To verify the effectiveness of the *class-specific* explainability visualization of ICE, we measured quantitative performance and analyzed visualization examples compared

[1] https://github.com/facebookresearch/DeiT

[2] https://github.com/hila-chefer/Transformer-Explainability

[3] https://github.com/hila-chefer/RobustViT

with existing explainability visualization methodologies on ImageNet-Segmentation.

Quantitative analysis. Table 1 shows that ICE outperformed all explainability visualization methods. To verify the scalability and robustness of ICE, we considered ViT AugReg (AR) [31] that has been used in the previous study [4]. We evaluated the segmentation performance for the ImageNet-Segmentation dataset for four cases depending on the model size (i.e., Small and Tiny). The explainability visualization of ICE-f showed comparable segmentation performance to RobustViT.

We also evaluated the performance improvement of explainability visualization using ICE on the models fine-tuned by RobustViT, the state-of-the-art visualization method. Table 2 demonstrates that ICE can improve explainability visualization even when applied to models fine-tuned by RobustViT. We demonstrate that the output embedding vectors for each patch token in a vision transformer-based model sufficiently preserve the information in the original image and can be effectively used to visualize the explainability of the model.

Qualitative analysis. Figure 3 shows samples visualizing the explainability of the top-1 prediction for the image. The first row shows the results of the detection of a single object, and the second row shows the results of multiple objects with various sizes. We can see that other methods tend to focus on only a small portion of the image or unmatched areas in the examples of multiple object detection. On the other hand, ICE adequately distinguishes foreground and background regions. We found that ICE can highlight object areas, regardless of object size.

Figure 4 illustrates a case where two different classes of objects exist in one image. The first and second rows show the explainability visualization results for top-1 predictions with each class. Rollout and Partial LRP methods show class-agnostic characteristics that highlight the same region regardless of the predicted class of the model, while GradCAM, RobustViT, and ICE show class-specific characteristics. Overall, ICE shows the result of highlighting the area of the objects in the image. ICE-f also shows comparable results with existing visualization methods. However, when it was necessary to distinguish fine-grained characteristics (e.g., tusker, african elephant, indian elephant), ICE still showed a tendency to predict patches into one class, which also occurred in other models.

4.3. Discovering Semantic Layouts

To verify the effectiveness of ICE on background patch selection from images that contain new classes, we evaluated the performance of background and foreground separation on PascalVOC12, which contain classes that were not learned in our DeiT-S model with ICE and ICE-f.

Table 3. The Jaccard similarity between the ground truth and predicted foreground on the PascalVOC12 validation set [10]. Only ImageNet labels are used for the training of ICE and ICE-f.

Method	Threshold	Output	Jaccard similarity
DeiT-S + Raw Attention [33]	0.9	mean	23.17
	0.8	mean	17.22
	0.9	head-4	21.80
	0.8	head-4	15.90
DeiT-S-SIN (DeiT-S + Shape Distillation) [23]	0.9	mean	33.00
	0.8	mean	34.30
	0.9	head-3	29.30
	0.8	head-3	25.10
DeiT-S + DINO [7]	0.9	mean	29.60
	0.8	mean	35.20
	0.9	head-1	37.40
	0.8	head-1	40.30
DeiT-S + ICE-f (Ours)	-	-	34.85
DeiT-S + ICE (Ours)	-	-	**41.32**

Quantitative analysis. Table 3 shows ICE achieved an 18.15 and 7.02 higher result based on the Jaccard similarity index than DeiT-S-Raw-Attention and DeiT-S-SIN [23], respectively. DeiT-S-Raw-Attention is a baseline model. DeiT-S-SIN uses a shape distillation token in DeiT-S and employs Resnet50-SIN [12] learned with the SIN dataset with strong shape characteristics. However, we have achieved better distinguishing performance between foreground and background by applying ICE methodology to the same DeiT-S without additional datasets and models.

As shown in Table 3, ICE achieved superior performance compared to DINO [7], the standard self-supervision method employed in previous studies [19, 23]. Other methods except ICE showed experimental tendencies in which performance varies according to the key hyperparameters (i.e., threshold, output head type). ICE does not have such constraints, implying a possibility of more flexible and easier adaptation of such a method to vision transformers.

Qualitative analysis. Figure 5 shows samples visualizing semantic layouts on the PascalVOC12 validation set. Compared to the visualization by applying raw attention to the original DeiT-S, ICE significantly improved the performance of semantic map segmentation by separating background and foreground regions. Furthermore, we found a tendency that our method distinguishes background relatively well compared to self-supervised learning methods.

4.4. Single Object Discovery

We conducted weakly-supervised single-object localization and unsupervised single-object discovery experiments to evaluate the explainability visualization performance of ICE and compare it to other state-of-the-art methods (Table 4). We evaluated the weakly-supervised single-object localization performance on the CUB-200-2011 using the Top-1 accuracy, GT Loc, and Top-1 Loc metrics, and unsupervised single-object discovery performance on the Pascal VOC 07/12 using the CorLoc metric. On CUB-200-2011,

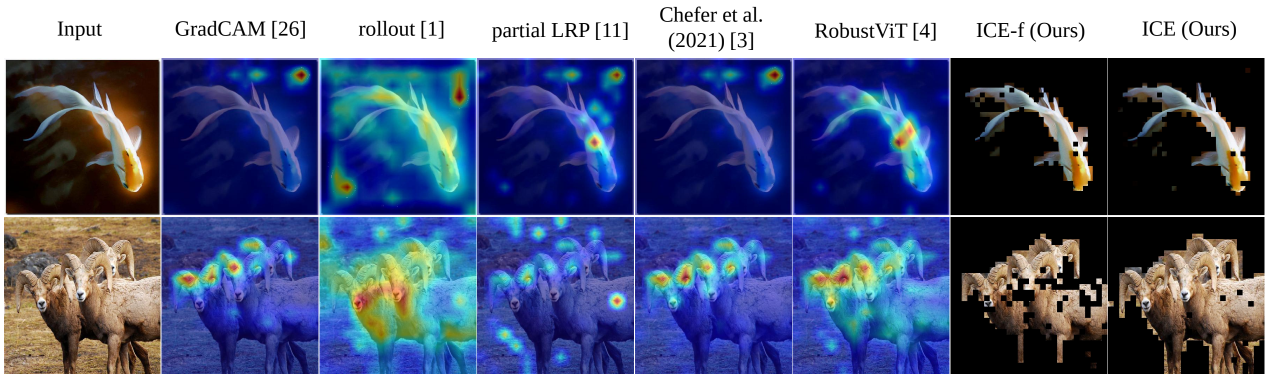

Figure 3. Results of explainability visualization of the images with the same class. The first row illustrates the result of the images that contain a single object. The second row illustrates those that contain multiple objects with various sizes.

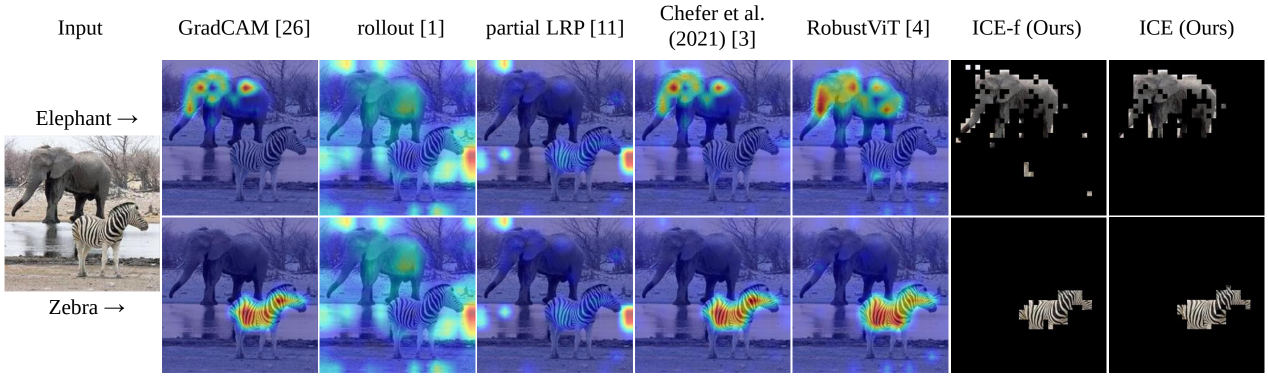

Figure 4. Results of explainability visualization for the presence of two different class objects in the image.

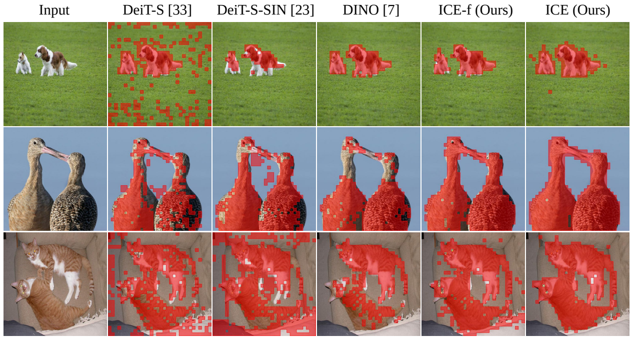

Figure 5. Sampled segmentation maps. The input image in the first row is the one presented in the previous research [23], and the input images in the second and third rows are from PascalVOC12.

ICE outperformed other methods on GT Loc and achieved comparable performance on Top-1 Loc. We note the reasonably high performance of ICE on Top-1 Loc, even though its Top-1 result was the lowest. This implies that, once an object in an image is correctly detected, ICE localizes the object well. On the Pascal VOC 07/12, ICE achieved comparable (although lower) performance on CorLoc compared to the state-of-the-art methods. Our intention here is to highlight the potential of ICE for open foreground object discovery from classification-based backbones compared to others that are based on self-supervised learning. Overall, we have verified the explainability visualization performance of ICE from various angles by showing the superiority of the class-aware single object localization performance and the foreground segmentation performance of ICE.

Table 4. Results of comparison with other methods for weakly-supervised and unsupervised single object discovery tasks.

Method	Backbone	CUB dataset [37]			PascalVOC07 [11]	PascalVOC12 [10]
		Top-1 Cls	GT Loc	Top-1 Loc	CorLoc	CorLoc
TS-CAM [6]	DeiT-S	**80.3**	87.7	71.3	-	-
LOST [8]	DINO	79.5	89.4	71.3	61.9	64.0
TokenCut [9]	DINO	79.5	91.8	**72.9**	**68.8**	**72.1**
ICE	DeiT-S	76.9	**92.5**	72.0	62.2	66.0

4.5. Efficiency Improvement

To examine another aspect of the effectiveness of background patch selection by ICE, we evaluated the accuracy and efficiency of ImageNet by applying ICE's background patch selection inside the encoder of the original DeiT-S and compared its results with DeiT (baseline) and EViT [20], a state-of-the-art model. Background patch selection is one of the key requirements in EViT, thus by comparing the performance of accuracy and efficiency between ICE and EViT, we can verify the role of ICE in visualizing the ex-

Table 5. Comparisons on ICE and EViT [20]. For fair comparisons, all models are initialized with a pre-trained DeiT-S and trained 30 epochs, and the throughput (img/s) is measured on the same machine with the same setting using a maximum batch size.

Method	Keep-rate	Top-1 accuracy (%)	# Params (M)	Throughput
DeiT-S	-	79.8	22.1	818
EViT	0.5	78.39	22.1	**1701**
ICE	0.5	**78.49**	22.4	1547
EViT	0.7	79.27	22.1	**1291**
ICE	0.7	**79.34**	22.4	1178

plainability of a vision transformer. We trained ICE in the same environment as EViT by referring to the EViT's official repository code [4] and set ICE to maintain keep rates after the 4th, 7th, and 10th layers in the pre-trained DeiT-S.

Table 5 shows our experimental results. By applying ICE to DeiT-S, the throughput performance significantly improved by 44.01% in the inference while maintaining comparable accuracy (only a decrease by 0.46% compared to the original DeiT-S). ICE showed 0.1% higher accuracy than the EViT under the same keep rate condition. This result means that the patch selection of ICE may help improve classification performance more than the patch selection of EViT. However, the throughput performance of ICE is 8.7% slower than EViT under the same keep rate condition, and this may be because ICE trains additional layers.

4.6. Ablation study

We conduct an ablation study to test the influence of the background scale (α) and class weight (λ_{class}) on the result of the pixel-wise accuracy, mean IoU, and Jaccard similarity. As illustrated in Table 6, the significant drop in overall performance without the background scale suggests that it is essential for effective learning of ICE. By adjusting with class weights, the optimal hyperparameter values found for effective learning of ICE were $\alpha = 0.5$ and $\lambda_{class} = 0.975$. These results indicate the effective role of these parameters in continuously learning the characteristics of classes and a background.

5. Discussion

ICE shows superior explainability visualization performance compared to other explainability visualization methodologies in the case that multiple objects of a single class exist in different sizes within one image, as Figure 3 shows. ICE adequately predicts the learned image classes or background classes for all patch locations in the image. Furthermore, ICE separates the background region and highlights the region that determines the class of the image as much as possible, regardless of the size of the object in the image.

Table 6. Ablation study of ICE (with DeiT-S) on the ImageNet-Segmentation dataset and the PascalVOC12 validation set.

Method	Pixel-wise accuracy	Mean IoU	Jaccard similarity
ICE ($\alpha = 0.5, \lambda_{class} = 0.975$)	**84.85**	**67.94**	**41.32**
w/o background scale ($\alpha = 1.0$)	63.45	45.36	22.98
w/o class weight ($\lambda_{class} = 0.5$)	79.53	57.59	29.14
w/o background scale ($\alpha = 1.0$) w/o class weight ($\lambda_{class} = 0.5$)	66.26	47.70	23.87

We expect our methodology to be useful for visualizing the explainability of image classification models when there are multiple objects with different classes and different sizes exist in an image. For example, for tasks regarding medical image classification [27], defect classification [16], and fashion style classification [18], key objects with various sizes may exist in a target image. In these examples, it may be necessary to visualize the regions that determine image classes as much as possible when the end-user (e.g., domain expert) of the classification model needs to check the visualized explainability of the model. Since ICE visualizes the explainability of all image patches that have features of an image class, the ICE methodology may be useful and well-applicable to many domains.

6. Conclusion

In this paper, we proposed ICE, a vision transformer-based explainability visualization methodology, which applies patch-wise classification and adversarial normalization for each patch token of a vision transformer. We demonstrated the effectiveness and superiority of ICE in class-specific explainability visualization and separation of a background region from a foreground region through quantitative and qualitative analyses. We incorporated ICE into the encoder of DeiT-S, resulting in significant improvements in efficiency while maintaining accuracy comparable to the original DeiT-S and EViT. We demonstrated that the output representations for each patch token retained sufficient image information at each patch location and confirmed the robustness of ICE in the background patch selection task. ICE does not use information from prediction to input layers, thus, is rarely affected by penalties derived from the structural characteristics of a vision transformer. Based on these results, we expect that ICE can be employed by vision transformers of various structures for explainability visualization.

Acknowledgments

This research was supported by the National Research Foundation and the Institute for Information Communication Technology Planning Evaluation grant funded by the Korean government (NRF-2021M3A9E4080780, IITP-2020-0-01373, and IITP-2022-2018-0-01431).

[4] https://github.com/youweiliang/evit

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020. 3, 5

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1, 2

[3] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791, 2021. 2, 3, 4, 5

[4] Hila Chefer, Idan Schwartz, and Lior Wolf. Optimizing relevance maps of vision transformers improves robustness. *arXiv e-prints*, pages arXiv–2206, 2022. 2, 4, 5, 6

[5] 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. 1, 3, 4

[6] Gao et al. Ts-cam: Token semantic coupled attention map for weakly supervised object localization. In *CVPR*, 2021. 7

[7] Mathilde et al. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 2, 3, 6

[8] Siméoni et al. Localizing objects with self-supervised transformers and no labels. In *BMVC*, 2021. 2, 7

[9] Wang et al. Self-supervised transformers for unsupervised object discovery using normalized cut. In *CVPR*, 2022. 2, 7

[10] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015. 5, 6, 7

[11] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 2, 3, 5, 7

[12] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 6

[13] Jindong Gu, Yinchong Yang, and Volker Tresp. Understanding individual decisions of cnns via contrastive backpropagation. In *Asian Conference on Computer Vision*, pages 119–134. Springer, 2018. 3

[14] Matthieu Guillaumin, Daniel Küttel, and Vittorio Ferrari. Imagenet auto-annotation with segmentation propagation. *International Journal of Computer Vision*, 110(3):328–348, 2014. 2, 5

[15] Shir Gur, Ameen Ali, and Lior Wolf. Visualization of supervised and self-supervised neural networks via attribution guided factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11545–11554, 2021. 3

[16] Joakim Bruslund Haurum and Thomas B Moeslund. Sewer-ml: A multi-label sewer defect classification dataset and benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13456–13467, 2021. 8

[17] Brian Kenji Iwana, Ryohei Kuroki, and Seiichi Uchida. Explaining convolutional neural networks using softmax gradient layer-wise relevance propagation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4176–4185. IEEE, 2019. 3

[18] Youngseung Jeon, Seungwan Jin, and Kyungsik Han. Fancy: human-centered, deep learning-based framework for fashion style analysis. In *Proceedings of the Web Conference 2021*, pages 2367–2378, 2021. 8

[19] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022. 6

[20] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. 2, 7, 8

[21] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017. 3

[22] Woo-Jeoung Nam, Shir Gur, Jaesik Choi, Lior Wolf, and Seong-Whan Lee. Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2501–2508, 2020. 3

[23] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021. 6, 7

[24] Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C Lipton. Learning to deceive with attention-based explanations. *arXiv preprint arXiv:1909.07913*, 2019. 3

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2, 5

[26] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 3, 5

[27] Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu. Transformers in medical imaging: A survey. *arXiv preprint arXiv:2201.09873*, 2022. 8

[28] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017. 3

[29] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016. 3

[30] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 3

[31] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021. 2, 5, 6

[32] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017. 3

[33] 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. 1, 2, 6

[34] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4

[35] 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. 2

[36] E Voita, D Talbot, F Moiseev, R Sennrich, and I Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. The Association for Computational Linguistics, 2019. 2, 3

[37] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2, 5, 7