# Visualization Techniques for Improving Explainability in Visual Transformers

Ana Iulia Coporan

West University of Timisoara, Timisoara, Romania
ana.coporan01@e-uvt.ro

**Abstract.** Vision Transformers (ViTs) have emerged as a powerful alternative to convolutional neural networks (CNNs) for various computer vision tasks, demonstrating state-of-the-art performance in image classification, object detection, and segmentation. However, due to their unique self-attention mechanisms and lack of intuitive hierarchical structure, understanding the internal workings of ViTs remains a significant challenge. Visualization techniques play a crucial role in enhancing the interpretability of ViTs by providing insights into the model's decision-making process. This report reviews the current state-of-the-art visualization techniques for ViTs, categorizing them into attention-based, gradient-based, and hybrid approaches. Attention maps, class activation maps (CAM), and layer-wise relevance propagation (LRP) are explored in the context of ViTs, focusing on how these techniques can reveal the regions of an image most influential in the model's predictions. Additionally, challenges related to the reliability, scalability, and accuracy of these methods are discussed. Finally, we examine hybrid methods that combine multiple visualization strategies to improve the robustness and clarity of interpretability, paving the way for more transparent and accountable AI systems. The report concludes with an overview of ongoing research directions aimed at improving the interpretability and explainability of ViTs.

**Keywords:** ViT · XAI · CAM

## 1 Introduction

In the field of artificial intelligence(AI) applications span across various types of inputs, including text, images, audio, and more. Each type of input requires specialized techniques to process and analyze effectively. For computer vision tasks like image classification convolutional neural networks (CNNs) have been the go-to option because of their very accurate results.

Despite the dominance of CNNs in image classification, newer architectures like vision transformers (ViTs) are emerging. Originally developed for natural language processing (NLP) tasks because of their impressive performance they have been extended also to the vision domain. ViT have demonstrated performances that are comparable with those of CNN arhitectures.[2]

To expand the knowledge behind ViT it is crucial to understand their inner working procedure and examine their explainability[5]. Explainable artificial intelligence (XAI) is a domain that challanges the transparency of AI models. XAI holds the promise of making AI systems more trustworthy and accountable. Not only that but it helps in better understanding the models, uncover limitations and improve the overall performance. This endeavor ensures that AI systems are not only powerful but also transparent, reliable, and aligned with human values and expectations. Regarding ViT XAI can help in uncovering which features are most important and which image regions have impacted the most the prediction, helping to better clarify the workings of the attention mechanism. This understanding is crucial in applications like medical diagnostics, where knowing which specific image features led to a diagnosis can enhance interpretability and trust in AI-driven medical decisions.

Despite the promise of ViTs, their explainability poses significant challenges. Unlike CNNs, where convolutional layers explicitly capture hierarchical features, ViTs rely on self-attention mechanisms to integrate global context and local dependencies across image patches. This unique architecture necessitates novel approaches for interpreting how these models arrive at their predictions, highlighting which image regions and features contribute most significantly to classification outcomes.

## 2 Problem Formulation

The problem is to gain a better understanding on how the attention mechanism works in a ViT and consequently how a ViT arrives to a decision. More formally speaking, for classification tasks the flow is as follows: you give a picture to a pre-trained ViT model and the model gives you as output probabilities corresponding to its classification of the input image. Following this, two scenarios may appear. The model seems to be great at making classifications, but the users are afraid of using it because there is a lack of transparency. Let's imagine the healthcare sector, in which a model says based on a MRI scan that a patient has cancer. Doctors and medical experts can use visualization techniques as a second opinion to complement the diagnostic processes. Doctors can cross-check these regions highlighted by the visualization techniques with their knowledge to validate the prediction. The second scenario is that the model makes bad decision. By using the insights provided by the visualization techniques, developers can use the heatmaps as debugging techniques on understanding what went wrong and how to improve the models performance. The model might have learned spurious correlations or irrelevant features due to biases in the training data or it might not properly capture the semantic features of the object. By identifying such issues, researchers and engineers can refine the training data, adjust model architectures, or implement additional regularization techniques to mitigate these problems.

The question is, how did the model come to such a result, on which parts of the image did he focus his attention, which pixels were more relevant for his decision making? Vizualization techniques help uncover that by analyzing the attention through different techniques. The result of a visualization technique is a heatmap, which highlights the regions of the input picture that the model found as most important.

## 3 Theoretical Background

**Vision Transformers**[2] work by getting 2D images converted into 1D sequences of sequentially arranged patches. The patches get through the multi head self attention mechanism, which learns the relationships between them. The architecture of ViT can be broken down into multiple steps, an overview being shown in Figure 1. The first step prepares the input image for the following steps, transforming it into a sequence of patch tokens. Each patch is flattened into a one-dimensional vector. The next step is about patch embedding. Each flattened patch vector is linearly projected to a lower-dimensional space using a trainable linear layer, producing patch embeddings. Positional embeddings are added to each patch embedding in order to capture the order of the patches. Another important ViT step is the classification head, for which a class token gets added to the sequence of patch embeddings. This token aggregates information from all patches during the self-attention operations and is ultimately used for classification. The next part are the transformer encoder layers. These layers include multi head self attention, a normalization layer, a feed forward network in the form of the multi-layer perceptron(MLP) and residual connections. Of importance to our study is the self attention mechanism. The self attention of each token denotes how much attention a token should pay to itself and to others. For each attention head, three vectors are computed: query, key and values. Through these, the self attention matrices are computed. The attention outputs from each attention head are concatenated along the feature dimension and then linearly projected to produce the final output.

ViTs use a self-attention mechanism as a core part of their architecture. This mechanism allows the model to weigh the importance of different parts of the input image when making predictions. The self-attention mechanism computes attention scores, which indicate how much focus each part of the image receives. These attention scores provide insight into how the model processes information from different parts of the image, which is why attention is widely used for explainability. There are two methods which have been proposed for mixing the attention scores across layers, attention rollout and attention flow. Attention rollout is a technique used to aggregate attention maps across multiple layers of a transformer. It provides a cumulative view of how information is propagated and aggregated throughout the layers. Although intuitive and easy appliable, it does not give a class specific explanation. Attention flow also aims to propagate attention scores through the layers of the transformer, and the idea behind this mechanism is to compute the maximum flow problem through a graph. The latter method is very slow, which usually leads to it being left out from comparisons with other methods.

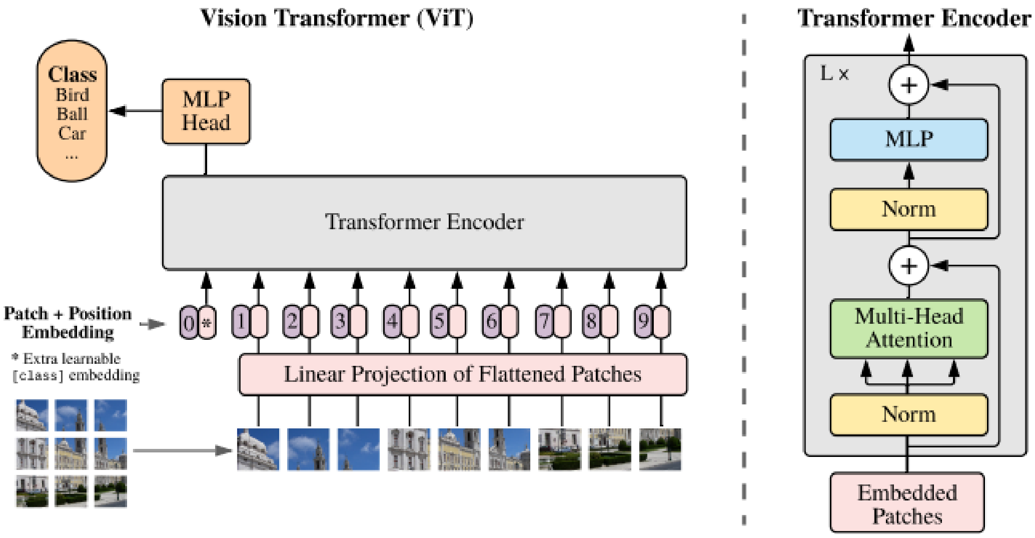

**Fig. 1.** A schematic representation of the ViT architecture[2]

To understand the inner work of a ViT one can choose one of the several existing approaches: visualization techniques, attention based methods, pruning-based methods, inherently explainable models. A more comprehensive review of ViT explainability is given by [5]. They do a thoroughly organization of the existing methods, taking into considerations factors like motivation, structure and application scenarios.

The work of this project focuses on the visual side of explainability. For visualizing attention there is also a variety of methods that can be chosen, from flow maps, parallel coordination plots and heatmaps[7]. Heatmaps are the most popular among them, they visually represent the intensity of attention strengths through color coding. Heatmaps provide an intuitive way to understand and analyze complex data patterns.

## 4 Methodology

The main visual methods for ViT explainability that have been explored in this research can be broken down into 2 different classes: (1) gradient-based and (2) attribution propagation methods.

### 4.1 Gradient-based methods

Gradient-based methods for interpreting deep learning models work by calculating the gradients of the model's output with respect to the input features. The logic is based on the idea that these gradients indicate how much each input feature contributes to the model's prediction. By visualizing these gradients, these methods highlight important regions or aspects of the input that most influence the model's decision. The result of such method is usually called a saliency map. Saliency maps highlight the regions in the input image that have the most significant impact on the model's prediction. Vanilla Saliency computes the absolute value of the gradients of the output with respect to the input. The easiest implementation is that of computing the absolute value of the gradient of the loss and taking the maximum gradient value across the color channels with respect to the input image. Another popular technique is that of using saliency maps. It shows which parts of the image have the most significant impact on the output. The saliency map is computed by taking the gradient of the output with respect to the input image, which indicates how much a small change in each pixel would affect the output.

**Class activation map** (CAM)[9] proposes to compute a linear combination between the weights and feature maps from the last layer of the model. Attention guided CAM is selectively aggregating gradients propagated from the classification output to each self-attention layer[6]. The method gathers contributions of image features from various locations in the input image. Additionally, these gradients are refined using normalized self-attention scores, which represent pairwise patch correlations. These scores enhance the gradients with patch-level context information identified by the self-attention mechanism.

$$
\begin{aligned}
\mathbf{A}_{\text{raw}} &= \text{generate\_cam\_attn}(\mathbf{I}, c) \\
\mathbf{A}_{\text{interp}} &= \text{Interpolate}(\mathbf{A}_{\text{raw}}, \text{scale factor} = \frac{H}{h}) \\
\mathbf{A} &= \frac{\mathbf{A}_{\text{interp}} - \min(\mathbf{A}_{\text{interp}})}{\max(\mathbf{A}_{\text{interp}}) - \min(\mathbf{A}_{\text{interp}})} \\
\mathbf{A}_{\text{binary}} &= \begin{cases} 1 & \text{if } \mathbf{A} > T \\ 0 & \text{otherwise} \end{cases} \\
\mathbf{V} &= \mathbf{I} \odot \mathbf{A}
\end{aligned}
$$

The equation 4.1 explains the steps behind the attention guided CAM method. This method generates a class-specific attention map from a Vision Transformer to visualize the regions of an input image that contribute most to a model's prediction. The process begins by feeding the input image **I** and the target class index $c$ into a function `generate_cam_attn`, which extracts raw attention scores ($\mathbf{A}_{\text{raw}}$) over a low-resolution feature grid. These raw scores are then upsampled ($\mathbf{A}_{\text{interp}}$) using bilinear interpolation to match the dimensions of the input image. The resulting attention map is normalized to the range $[0, 1]$, producing **A**, which highlights the relative importance of image regions. Optionally, Otsu's thresholding can be applied to binarize the attention map ($\mathbf{A}_{\text{binary}}$), isolating the most critical regions. Finally, the normalized or thresholded map is overlaid onto the input image, generating a visualization ($\mathbf{V}$) that illustrates the contribution of each region to the model's decision. This method aids in interpreting model behavior and evaluating its focus on relevant features.

**I SAW** (SAW)[8] is a method that is derived from the attention rollout mechanism. Instead of just combining all the attention maps together, which may result in not such specific results, they are weighting the self attention maps and are aggregating them based on their weights. The attention is being weighted at a specific class, which makes their method to be class specific. Let $\mathbf{A}^{(l)} \in R^{N\times N}$ denote the self-attention map from layer $l$, where $N$ is the number of tokens, and $\alpha^{(l)}$ represent the weight assigned to the attention map from layer $l$. The class-specific attention map, $\mathbf{A}_{\text{SAW}}^{(c)}$, is computed as:

$$\mathbf{A}_{\text{SAW}}^{(c)} = \sum_{l=1}^{L} \alpha^{(l)} \mathbf{A}_c^{(l)}$$

Here, $L$ is the total number of layers, and $\mathbf{A}_c^{(l)}$ is the self-attention map at layer $l$ weighted for the specific class $c$. To ensure the resulting attention map is normalized to a range of $[0, 1]$, a min-max normalization is applied:

$$\mathbf{A}_{\text{SAW}}^{(c)} = \frac{\mathbf{A}_{\text{SAW}}^{(c)} - \min(\mathbf{A}_{\text{SAW}}^{(c)})}{\max(\mathbf{A}_{\text{SAW}}^{(c)}) - \min(\mathbf{A}_{\text{SAW}}^{(c)})}$$

This approach ensures that the contribution of different layers is appropriately weighted and aggregated to generate a more specific and class-sensitive attention map.

### 4.2 Attribution propagation

Attribution propagation methods are based on the Deep Taylor Decomposition(DTD). DTD propagates relevance scores backward from the output layer to the input layer, and the one such method is called the Layerwise Relevance Propagation (LRP), on which most of the research for ViT explainability is based. LRP can be adapted to propagate relevance through the self-attention mechanisms. Gradient based methods explain the contribution of the image features elicited through multiple layers, while LRP based methods capture the contribution of the independent pixels to the classification output[6]. A class specific LRP based visualization technique has been proposed that uses LRP based relevance to calculate scores for each attention head in every layer of the model[1]. These scores are then integrated throughout the attention graph by combining relevance and gradient information, iteratively eliminating negative contributions. The class-specific relevance score for attention head $h$ in layer $l$, $\mathbf{R}_c^{(l,h)}$, is computed as:

$$\mathbf{R}_c^{(l,h)} = \alpha^{(l,h)} \cdot \left(\mathbf{A}_h^{(l)} \odot \nabla \mathbf{A}_h^{(l)}\right)$$

Where $\odot$ denotes element-wise multiplication of relevance and gradient information.

The final class-specific relevance map, $\mathbf{R}_c$, is obtained by aggregating the relevance scores from all layers and heads:

$$\mathbf{R}_c = \sum_{l=1}^{L} \sum_{h=1}^{H_l} \mathbf{R}_c^{(l,h)}$$

Where $L$ is the total number of layers, and $H_l$ is the total number of attention heads in layer $l$.

### 4.3 Evaluation metrics

Key evaluation metrics are segmentation metrics such as Pixel Accuracy, F1 Score, and Jaccard Index. These metrics are essential for evaluating the quality of segmentation results, providing insights into how accurately and precisely the models can identify and localize relevant features in input images. The heatmaps produced were processed using OTSU thresholding to create binary masks. These resulting masks were then compared with ground truth masks to evaluate performance. Pixel Accuracy quantifies the overall correctness of the segmentation predictions. The F1 Score offers a balanced measure of precision and recall across different classes, highlighting the trade-off between these two aspects. The Jaccard Index (or Intersection over Union) measures the overlap between predicted segments and the ground truth, offering a comprehensive view of segmentation performance.

## 5 Experiments

### 5.1 Dataset

The Pascal Visual Object Classes (VOC) dataset has been used to test the potential of the visualization methods for segmentation tasks [3]. The Pascal VOC 2012 dataset is a well known benchmark in visual object category recognition and detection, containing annotated images for tasks such as classification, detection, segmentation, and action recognition.

### 5.2 Model

In the visual classification experiments, a pretrained ViT-based model has been used. The model adopts a BERT-like architecture. The images used are of size 224x224, the size of the patches are of 16, and they are followed by flattening and linear transformations to generate a sequence of vectors. A classification token is prepended to the sequence and serves as the input for classification tasks.

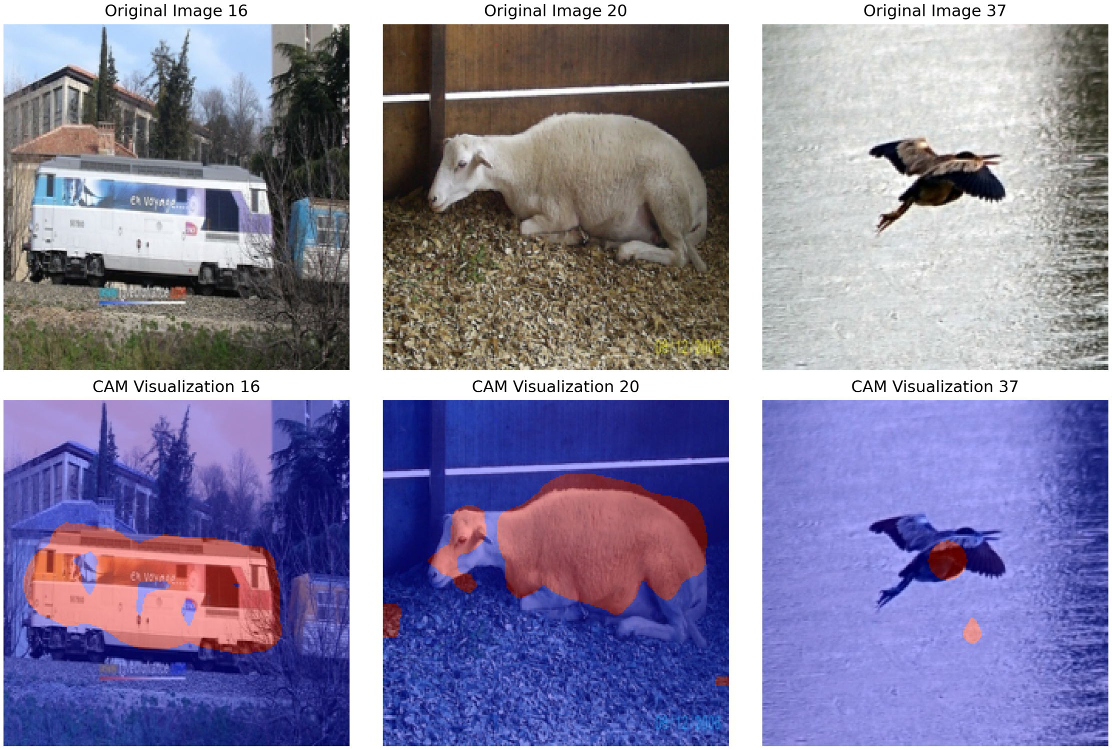

**Fig. 2.** Heatmaps of attention guided CAM visualization technique

### 5.3 Results

The method which was implemented up until now is the attention guided CAM[6]. The first part of the evaluations focus on visual comparisons of generated heatmaps, emphasizing their collective ability to offer detailed insights into the reasoning behind ViT. Our evaluation criterion for explanation quality centers on how well explanation maps align with human observations of images in visual recognition tasks. This was found effective in [8] [1] [6] and was proposed in [4]. The results can be seen in Figure 2, where you can clearly see that the attention of the model lands exactly on the objects which are recognized. The first row of the figure shows the original images while the second displays the results of the heatmap generated by the attention guided CAM technique. As for the images, the exact index from the dataset for each of them is being provided in the figure.

In addition to directly evaluating the generated heatmaps, we employed specific metrics aimed at assessing the performance of the explainability methods. The second part of the evaluation has been done using evaluation metrics. The tests were performed on a subset of Pascal VOC, where the predict probability for the main class was higher than 85%. Such measures were needed as our model was pretrained on ImageNet21k, and we had to ensure the model found the relevant parts in the tested images.

**Table 1.** Average Segmentation Metrics for attention guided CAM method. Threshold set to 0.85 accuracy minimum.

| Method | Jaccard Index (IoU) | F1 Score | Pixel Accuracy |
|---|---|---|---|
| CAM | 12.43 | 19.13 | 65.94 |

In Table 1, the performance of the CAM method is evaluated using three metrics: Jaccard Index (IoU), F1 Score, and Pixel Accuracy. The CAM method achieved a Jaccard Index of 12.43, indicating a relatively low overlap between the predicted and true regions. This suggests that the model's segmentation or classification results have limited similarity to the ground truth. The F1 Score for CAM is 19.13, reflecting a suboptimal balance between precision and recall, as it implies a significant trade-off between false positives and false negatives. Finally, the Pixel Accuracy of 65.94 shows that approximately 66% of the pixels in the prediction match the true values. While this indicates that the model correctly classifies a majority of the pixels, the accuracy is not particularly high, suggesting room for improvement in correctly identifying all relevant regions. Overall, the CAM method shows moderate performance but has substantial room for improvement, particularly in terms of the Jaccard Index and F1 Score, which are crucial for evaluating the quality of image segmentation and classification tasks.

## References

1. Chefer, H., Gur, S., Wolf, L.: Transformer interpretability beyond attention visualization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 782–791 (2021)
2. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
3. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. **88**, 303–338 (2010)
4. Fuad, K.A.A., Martin, P.E., Giot, R., Bourqui, R., Benois-Pineau, J., Zemmari, A.: Features understanding in 3d cnns for actions recognition in video. In: 2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA). pp. 1–6. IEEE (2020)
5. Kashefi, R., Barekatain, L., Sabokrou, M., Aghaeipoor, F.: Explainability of vision transformers: A comprehensive review and new perspectives. arXiv preprint arXiv:2311.06786 (2023)
6. Leem, S., Seo, H.: Attention guided cam: Visual explanations of vision transformer guided by self-attention. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 2956–2964 (2024)

7. Li, Y., Wang, J., Dai, X., Wang, L., Yeh, C.C.M., Zheng, Y., Zhang, W., Ma, K.L.: How does attention work in vision transformers? a visual analytics attempt. IEEE transactions on visualization and computer graphics **29**(6), 2888–2900 (2023)
8. Mallick, R., Benois-Pineau, J., Zemmari, A.: I saw: a self-attention weighted method for explanation of visual transformers. In: 2022 IEEE international conference on image processing (ICIP). pp. 3271–3275. IEEE (2022)
9. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2921–2929 (2016)