

Understanding Vision Transformers through Intuition and Simple Visual Examples

M. SABRIGIRIRAJ¹, K. MANOHARAN²

¹Department of It Hindusthan College of Engineering and Technology
Coimbatore
INDIA

²Department of Ece
Sns College of Technology
Coimbatore
INDIA

Abstract: - Vision Transformers are normally used in image recognition and are considered as an alternative to Convolutional Neural Networks. Unlike traditional convolution-based models, a Vision Transformer does not rely on local filters. Instead, it segments an image into smaller patches and understands the relationships among individual patches. This technique is powerful, but it can be challenging for beginners to understand when described only with equations or programming concepts. Hence, this paper showcases the process of Vision Transformers using simple examples and visual illustrations. This paper is intended for students, teachers, and beginners who are interested in learning how transformer models are used in computer vision. The tutorial explains how an image is divided into patches, how each patch is transformed into a token, how self-attention helps different image regions share information, and how the transformer encoder helps the final classification process. The foremost purpose of this tutorial is to support readers in developing a clear understanding of Vision Transformers. After gaining this intuitive understanding, readers can more easily study the mathematical particulars and implementation techniques later.

Key-Words: - Convolutional Neural Networks, Deep Learning, Explainable Artificial Intelligence, Neural Networks, Vision Transformers, Computer Vision, Intuitive Learning

Received: March 25, 2026. Revised: April 21, 2026. Accepted: June 15, 2026. Published: September 23, 2026.

1 Introduction

Deep learning has become a significant part of current computer vision. It supports computers in understanding and analyzing visual data with high accuracy. Many years ago, Convolutional Neural Networks (CNNs) were commonly used for image recognition and image classification tasks. CNNs work well because they can automatically learn local visual features of the images. In the early layers, a CNN usually learns simple features such as edges and textures. In deeper layers, these simple features are combined to form more meaningful patterns, such as shapes and object parts. However, CNNs also have a limitation. They mostly process images locally. Because of this, relationships between far-away regions in an image are usually learned only after the information passes through several convolutional layers.

Transformer models have shown very strong performance in natural language processing. Their

core strength is their capability to learn long-range relationships through the attention mechanism. In a transformer model, each element in a sequence can interact directly with all other elements using self-attention. This helps the model capture global relationships effectively. Subsequently, transformers became successful in language tasks; researchers began applying the same idea to image data. This led to the development of Vision Transformers (ViTs). In a ViT method, an image is divided into small patches. These patches are preserved like a sequence of tokens. A self-attention mechanism is used to learn the relationships between different patches across the entire image. This provides a different way of processing visual information compared with traditional CNN-based methods. Many research papers in the Vision Transformers area mostly focus on mathematical and programming details. This makes the content more difficult for new learners. Beginners may find it difficult to understand how an image is divided into multiple patches, how attention

is found between these patches, and how the final class is predicted. To understand this process in an easier way, Figure 1 illustrates a simple overview of the ViT architecture.

This tutorial explains Vision Transformers step by step. It places more emphasis on visual understanding than on detailed formulas. Basic examples and diagrams are utilized to explain the important parts of the model. It includes image patching, patch embedding, positional encoding, self-attention, and final classification. The main aim is to help readers understand the basic working of Vision Transformers before they learn advanced mathematics or optimization methods. Therefore, this tutorial may be beneficial for students, teachers, and early-stage researchers who are learning Vision Transformers for the first time.

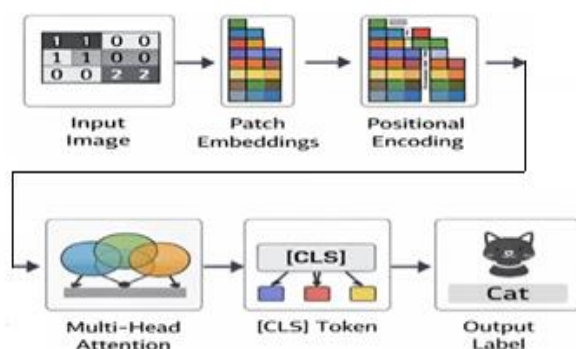


Fig. 1: Vision Transformer (like a 1-filter CNN)
Source: Created by the authors

1.1 Related Work

Several survey and review papers have explained transformer-based models in computer vision. Some tutorial-style works in machine learning and deep learning also show that difficult ideas can be explained using examples and simple reasoning. They do not always need to begin with formal mathematical derivations. The tutorial approach in [1] supports this type of example-based explanation. The present tutorial follows a similar approach. It mostly emphasizes helping beginners understand the basic concept of Vision Transformers through simple explanations with relevant examples.

In-depth survey papers on Vision Transformers [2] typically describes model architectures, uses, applications, and recent research progress. The content available in these papers is beneficial for researchers and professionals. Nevertheless, they always focus more on technical details than on beginner-level teaching. Similarly, research studies that compare different ViT architectures and report their performance on various tasks [3], [4] mainly discuss model types, training methods, and

benchmark results. These studies are useful for research purposes. However, they are not mainly written to help new learners to understand how ViTs work.

Similarly, other review papers such as [5], which discusses transformer-based methods in computer vision, and [6], which focuses on self-supervised learning for ViTs, generally follow a more technical style and often assume that readers already have some background knowledge in that area. Some research studies focus on lightweight ViT designs and optimization methods for practical deployment [7], [8], while others examine interpretability and explainability in ViT-based image classification [9], [10]. These works provide valuable insights into the performance, behavior, and practical use of ViT architecture models [11], [12], [13]. However, most of them are written from an advanced technical perspective. Overall, the current literature addresses ViT models, model variants, efficiency enhancements, applications, and explainability approaches, with more focus given to mathematical formulations, architectural modifications, and implementation specifics. On the other hand, this tutorial paper is designed as an introductory and teaching-oriented resource that can support classroom learning as well as independent study.

1. From an Image to a Sequence: Thinking Differently

To understand how a ViT processes an image, consider a very small grayscale image of size 4×4 , shown in Figure 2. A CNN analyzes this image by sliding a window over it; the reader already has some background in deep learning. Some works focus on making transformer models more efficient [7], [8]. They apply small filters over neighboring pixels, and it gradually builds local features such as edges and textures. But a ViT follows a fundamentally different strategy. As an alternative to scanning the image with filters, it divides the image into small fixed-size blocks called patches. Each of the patches is treated as an independent visual unit. As revealed in Figure 2, the 4×4 image is separated into four patches, each of size 2×2 . Once divided, the image is not observed as a grid of pixels. Instead, the ViT models consider it as a sequence of patches such as [Patch 1, Patch 2, Patch 3, Patch 4]. This is similar to the way a sentence is treated as a sequence of words. The key concept is that an image can be interpreted as an ordered sequence of visual elements, permitting transformer models to be applied to vision tasks. After converting the image into a sequence of patches, the subsequent step is to represent each patch numerically so that it can be processed by the transformer.

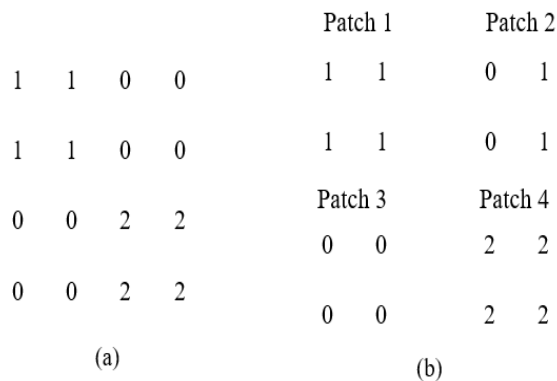


Fig. 2: (a) A grayscale image of size 4×4 pixels and (b) Image divided into 4 pieces of 2×2 patches

Source: Created by the authors

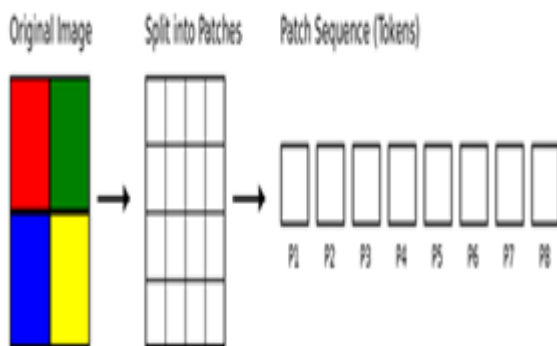


Fig. 3: Vision Transformer patching for a larger image. The input image is separated into fixed-size patches, which are then transformed into a sequence of patch tokens and managed by the transformer encoder

Source: Created by the authors

2.1 Example with a Larger Image

While the 4×4 grayscale image example supports illustrating the patching concept evidently, Vision Transformers are naturally applied to larger images such as 224×224 pixels. For example, an image of size 224×224 can be divided into patches of size 16×16 , resulting in 196 patches. Each patch is then transformed into an embedding vector and processed by the transformer encoder. Figure 3 illustrates how a larger image is divided into multiple patches and converted into a sequence of tokens. This illustration demonstrates that the same concept used for small images also applies to real-world images.

3 Patch Embedding: Giving Each Patch a Numerical Identity

Computers cannot operate directly on image patches in their raw form. So, each patch is converted into a

numerical representation. Consider Patch 1 from Figure 2. This 2×2 patch is first flattened into a single vector: $[1, 1, 1, 1]$. This vector is then converted into a shorter, fixed-length numerical representation by means of a learnable linear transformation. For example, the compressed patch might be mapped to $[0.8, 0.6]$. This resulting vector is commonly referred to as a patch embedding. It serves as a compact summary of the visual information contained in the patch. Applying the same embedding process to all patches produces the following sequence of vectors:

Patch 1 $\rightarrow [0.8, 0.6]$
Patch 2 $\rightarrow [0.1, 0.2]$
Patch 3 $\rightarrow [0.0, 0.1]$
Patch 4 $\rightarrow [0.9, 0.7]$

At this stage, note that the entire image is converted into a sequence of numerical vectors, each representing one patch. These patch embeddings capture the visual content of each region. But they do not contain any information about the original spatial location of the patches within the image. This constraint motivates the practice of positional encoding, which is discussed in the succeeding section.

4 Positional Encoding: Recalling Where Patches Came From

At this stage, one potential issue arises. That is, after the image has been converted into a sequence of patch embeddings, the transformer handles it as a sequential list such as [Patch 1, Patch 2, Patch 3, Patch 4]. Nevertheless, a transformer itself has no inherent concept of spatial order. To the model, this sequence would look identical even if the patches were shuffled. In images, the position of each patch is most important because the meaning of visual patterns depends on where they appear in the image. To handle this, positional information is included to each patch embedding. This positional information tells the model where each patch came from in the original image grid. For example:

Patch 1 embedding: $[0.8, 0.6] + \text{positional information}$

Patch 2 embedding: $[0.1, 0.2] + \text{positional information}$

Positional encoding supports the model in distinguishing between patches that may look similar but originate from different locations in the image. In simple terms, this encoding tells the model whether a patch came from the top-left, center, or bottom-right part of the image.

By combining each patch with its positional information, each patch embedding represents both appearance and location. With this enriched representation, the ViT is prepared to learn how different regions of the image interact and relate to one another. This process is achieved through attention mechanisms, which are elaborated in the next section. In practice, positional encodings can be fixed or learned, but their role is always to inject spatial structure into the transformer.

5 Learning Relationships Using Attention

Once patch content and positional information are combined, the ViT must learn how different regions of the image relate to each other. This is accomplished through the attention mechanism, which allows patches to exchange information and form a global understanding of the image.

5.1 Self-Attention: Patches Talking to Each Other

For the four-patch example shown in Figure 2, self-attention allows each patch to weigh the relevance of the remaining three patches before updating its representation. This allows patches to interact and influence each other, rather than being processed in isolation. For example, assume that Patch 1 represents one part of an object, and Patch 4 represents another part of the same object. Through self-attention, Patch 1 may allocate higher importance to Patch 4, efficiently reasoning something like “Patch 4 appears associated with me, so I should pay more attention to it”. At the same time, Patch 1 may assign very low importance to Patch 2 if it primarily contains background information. Since this process is applied to all patches in parallel, the model can directly capture relationships across the entire image. As a result, the ViT develops a global view of the image, where distant but related regions can directly influence each other. A model may find it difficult to understand the full image if it only looks at small local regions. This often happens in convolution-based methods. Self-attention works differently. It permits each image patch to attend to other patches in the same image. This supports the model in understanding how different slices of the image are connected. A simple form of self-attention example is shown below:

Attention (Q, K, V) = $\text{SoftMax}(\text{QK}^T / \sqrt{d})$ V

Here, Q refers to queries, K refers to keys, V refers to values, and d refers to the embedding dimension. In simple terms, this formula helps the model measure

how strongly one patch is related to another patch. Patches that are more useful for the task receive more attention. The model uses information from these important patches to improve its understanding of the image. This tutorial does not discuss the complete mathematics of attention. The formula is included only to help readers connect the visual explanation with the formal notation.

5.2 Multi-Head Attention: Looking in Multiple Ways

Vision Transformers practice the self-attention mechanism through several attention heads at the same time. This process is known as multi-head attention. Each attention head available in the architecture can focus on the image from a different perspective. For example, one head may give more attention to nearby patches, while another head may focus on patches that are far apart. This helps the model learn different types of relationships within the same image. After this, the results from all attention heads are joined together. This gives the model a better understanding of the image. This supports the model in creating a richer and more convenient representation of the image. Because of this parallel attention process, Vision Transformers can learn dissimilar visual patterns at the same time.

5.3 Feed-Forward Network: Refining Patch Information

After the attention step, each patch representation is passed through a small feed-forward neural network. This network operates independently on each patch and does not mix information across patches. The role of this stage is to refine the features learned during attention by strengthening informative patterns and suppressing less useful signals. This step can be thought of as polishing the information extracted through attention, and preparing each patch representation for the next transformer layer. By alternating attention and feed-forward refinement, the ViT progressively builds more meaningful and better discriminative image representations.

6 The Classification Token: Summarizing the Image

ViTs present a special vector known as the classification token (often called the CLS token). Unlike patch embeddings, this token does not correspond to any specific image patch. Instead, it acts as a learnable container that gathers information from across the entire image. During the attention process, the classification token participates just like other

tokens. It attends to all patch embeddings, and each patch contributes relevant information to it.

Through repeated attention layers, the classification token gradually accumulates a global summary of the image content. By the end of the transformer, this token serves as a compact representation of the entire image. It is then passed to a simple classifier, which uses this summary to produce the final prediction, such as “cat” or “car”. It is observed that the classification token functions like a summary word added to the patch sequence, whose sole purpose is to collect information useful for prediction.

7 Putting Everything Together: A Simple Walkthrough

The full processing flow of a ViT can be described in these simple steps:

1. Begin with an input image.
2. Split the image into small patches of fixed size.
3. Change each patch into a numerical embedding.
4. Add positional information so that the model knows where each patch is located.
5. Allow patches to interact using attention mechanisms
6. Collect global information through the classification token
7. Predict the final image label

8 Training Vision Transformers: Learning from Data

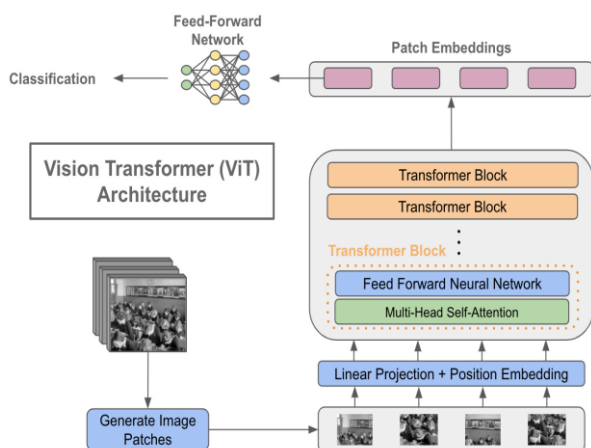


Fig. 4: A typical full-fledged Vision Transformer
Source: Created by the authors

Training a ViT involves teaching the model to adjust its internal representations so that it can

correctly predict image labels. Rather than introducing a separate processing stage, training wraps around the entire pipeline as described in previous sections 2–7. During training, an input image is first passed through the full ViT pipeline. The various stages include dividing the image into patches, converting them into embeddings, enriching them with positional information, processing them through attention layers, and summarizing them with the classification token. The output produced by the classification token is sent to a classifier, which produces a predicted label for the image. This estimate is compared with the ground-truth label using a loss function that measures how far the prediction is from the correct answer. Based on this error, the model updates its parameters through backpropagation. During training, gradients flow backward through the classifier, the classification token, the attention layers, the feed-forward networks, and the patch embedding layers. With this training method, the parts of the Transformer are learned together. During training, the model improves the patch representations and adjusts the attention weights. As a result, image regions that are more beneficial for the task can have a stronger role in the final prediction. The classification token collects information from all patches. It then uses this combined information to produce the output. As training continues, the ViT becomes better at finding visual patterns and learning relationships between image regions. A typical ViT architecture is shown in Figure 4.

In practice, ViTs usually need large datasets. This is because they do not use the same built-in image assumptions as CNNs, such as locality and translation invariance. To handle this issue, ViTs are often pretrained on a large image dataset and then fine-tuned on a smaller dataset for a specific task. The exact training procedure may differ across different applications, but the fundamental idea remains the same. The model makes a prediction, compares it with the correct label, and updates its parameters many times to reduce the error.

9 Limitations and Practical Challenges

ViTs are useful for visual recognition, but they also have some limits. These limits should be considered before using them in practical work. CNNs have some image-related assumptions built into their design. For example, they focus on local regions and can handle small shifts in an image more naturally. This helps CNNs learn from smaller datasets. ViTs do not have these assumptions in the same strong way. As a result, they typically need large data to learn visual patterns. If a ViT is trained from the beginning on a small

dataset, it may not work well on new images. In many applications, this problem is handled by pretraining the model on a large image dataset and then fine-tuning it for a smaller task-specific dataset.

Another important challenge is the computational cost of self-attention. In self-attention, every image patch interacts with every other patch. When the image resolution increases, the number of patches also increases. This requires more memory and more computation. Training a Vision Transformer can take more time. It may also need stronger computer hardware. Some newer ViT designs and efficient attention methods try to reduce this problem. Still, computational cost is an important limitation.

The output performance of a Vision Transformer depends on several design choices, such as patch size, embedding size, and the training strategy. Even small changes in these settings can affect the final result. Therefore, careful tuning is usually required. Attention maps can help show which parts of an image may have influenced the prediction. However, they may not provide a complete explanation of the model. They may highlight some useful image regions, but they do not fully explain how the final decision was made. Despite these limitations, research on Vision Transformers continues to cultivate. Many studies are focused on making them faster, more reliable, and more effective when trained with smaller datasets. Understanding both the strengths and boundaries of Vision Transformers is significant for their practical use. It can also support the development of better vision models in future designs.

10 Conclusion

Vision Transformer models process images in a different way from traditional CNNs. While CNNs mostly inspect small local regions of an image through convolution operations, on the other hand, Vision Transformers divide the image into patches and use an attention mechanism to learn the relationships between different image regions. This paper presented the main conceptual ideas of Vision Transformers in a simple and step-by-step manner. It explained image patching, patch embedding, positional encoding, self-attention, and the classification token. The discussion focused on visual explanation and basic understanding for beginners rather than advanced mathematical details. This work can help readers to understand how a Vision Transformer works before moving on to its formal concepts. In future work, the same tutorial-based method can be extended to related areas such as multimodal learning, efficient transformer architectures, and other advanced transformer models used in computer vision.

References:

- [1] M. Sabrigiriraj, K. Manoharan. Teaching Machine Learning and Deep Learning Introduction: An Innovative Tutorial-Based Practical Approach. WSEAS Transactions on Advances in Engineering Education. Vol. 21, pp. 54–61, 2024. [10.37394/232010.2024.21.8](https://doi.org/10.37394/232010.2024.21.8)
- [2] S. Khan, M. Naseer, M. Hayat, S. Waqas Zamir, F. S. Khan and M. Shah "Transformers in vision: A survey." ACM computing surveys (CSUR) Vol. 54, no. 10s, pp. 1–41, 2022.
- [3] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, E. Xu, C. Xu, Y. Xu and D. Tao "A survey on vision transformer." IEEE transactions on pattern analysis and machine intelligence Vol. 45, no. 1, pp. 87–110, 2023.
- [4] Y. Liu, T. Zhang, K. Chen, X. Liu, Y. Deng and F. Wang, "A survey of visual transformers." IEEE transactions on neural networks and learning systems Vol. 35, no. 6, pp. 7478–7498, 2024.
- [5] Jamil, Sonain, Md Jalil Piran, and Oh-Jin Kwon. "A comprehensive survey of transformers for computer vision." drones, Vol. 7, no. 5, Art. no. 287, 2023.
- [6] A. Khan, M. Khan, S. Khan and M. A. Khan "A survey of the self-supervised learning mechanisms for vision transformers." arXiv preprint arXiv:2408.17059 (2024).
- [7] Fournier, Quentin, Gaétan Marceau Caron, and Daniel Aloise. "A practical survey on faster and lighter transformers." ACM Computing Surveys Vol. 55, no. 14s, pp. 1–40, 2023.
- [8] R. Kashefi, M. Shafiee, A. M. N. Nik and M. H. Mahoor, "Explainability of vision transformers: A comprehensive review and new perspectives." arXiv preprint arXiv:2311.06786 (2023).
- [9] Khalil, Mahmoud, Ahmad Khalil, and Alioune Ngom. "A comprehensive study of vision transformers in image classification tasks." arXiv preprint arXiv:2312.01232 (2023).
- [10] K. Al-Hammuri, M. Alhaj, M. Alsmirat and A. Al-Ayyoub, "Vision transformer architecture and applications in digital health: a tutorial and survey." Visual computing for industry, biomedicine, and art , Vol. 6, no. 1, Art. no. 14, 2023.
- [11] Y. Yang, Y. Wu, M. Wang, H. Hu, L. Lin and B. Du, "Transformers meet visual learning

- understanding: A comprehensive review." arXiv preprint arXiv:2203.12944 (2022).
- [12] Mauricio, José, Inês Domingues, and Jorge Bernardino. "Comparing vision transformers and convolutional neural networks for image classification: A literature review." *Applied Sciences* Vol. 13, no. 9, Art. no. 5521, 2023.
- [13] L. Papa, G. Fiameni, L. Rundo, C. Distanto and I. M. Ray, "A survey on efficient vision transformers: algorithms, techniques, and performance benchmarking." *IEEE transactions on pattern analysis and machine intelligence* Vol. 46, no. 12, pp. 7682–7700, 2024.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US