Paper Reading Report – AI6121 Computer Vision

Reinelle Jan C. Bugnot
G2304329L
October 1, 2023

Chosen Computer Vision Paper:

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,
Dosovitskiy et. al, 2021

Supporting Paper:

Attention is All You Need,
Vaswani et. al, 2017

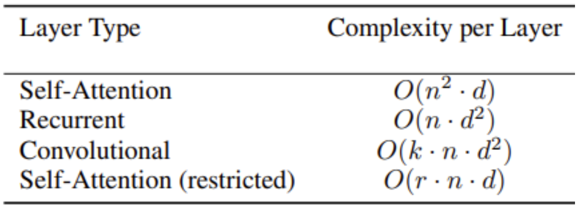
1. Introduction

Ever since the introduction of the self-attention mechanism, Transformers have been the top choice when it comes to Natural Language Processing (NLP) tasks. Self-attention-based models are highly parallelizable and require substantially fewer parameters, making them much more computationally efficient, less prone to overfitting, and easier to fine-tune for domain-specific tasks [1]. Furthermore, the key advantage of transformers over past models (like RNN, LSTM, GRU and other neural-based architectures that dominated the NLP domain prior to the introduction of Transformers) is their ability to process input sequences of *any* length without losing context, through the use of the self-attention mechanism that focuses on different parts of the input sequence, and how those parts interact with other parts of the sequence, at different times [2]. Because of these qualities, Transformers has made it possible to train language models of unprecedented size, with more than 100B parameters, paving the way for the current state-of-the-art advanced models like the *Generative Pre-trained Transformer* (GPT) and the *Bidirectional Encoder Representations from Transformers* (BERT) [1].

However, in the field of computer vision, convolutional neural networks or CNNs, remain dominant in most, if not all, computer vision tasks. While there has been an increasing collection of research work that attempts to implement self-attention-based architectures to perform computer vision tasks, very few has reliably outperformed CNNs with promising scalability [3].

The main challenge with integrating the transformer architecture with image-related tasks is that, by design, the self-attention mechanism, which is the core component of transformers, has a quadratic time complexity with respect to sequence length, i.e. $O(n^2)$, as shown in Table I and as discussed further in Part 2.1. This is usually not a problem for NLP tasks that use a relatively small number of tokens per input sequence (e.g., a 1,000-word paragraph will only have 1,000 input tokens, or a few more if sub-word units are used as tokens instead of full words). However, in computer vision, the input sequence (the image) can have a token size with orders of magnitude greater than that of NLP input sequences. For example, a relatively small 300 x 300 x 3 image can easily have up to 270,000 tokens and require a self-attention map with up to 72.9 billion parameters ($270{,}000^2$) when self-attention is applied naively.

TABLE I. Time complexity for different layer types [2].



Layer Type	Complexity per Layer
Self-Attention	$O(n^2 \cdot d)$
Recurrent	$O(n \cdot d^2)$
Convolutional	$O(k \cdot n \cdot d^2)$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$

For this reason, most of the research work that attempt to use self-attention-based architectures to perform computer vision tasks did so either by applying self-attention *locally*, using transformer blocks in conjunction with CNN layers, or by only replacing specific components of the CNN architecture while maintaining the overall structure of the network; never by only using a pure transformer [3]. The goal of Dr. Dosovitskiy, et. al. in their work, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", is to show that it is indeed possible to implement image classification by applying self-attention *globally* through the use of the basic Transformer encoder architure, while at the same time requiring significantly less computational resources to train, and outperforming state-of-the-art convolutional neural networks like ResNet.

2. The Transformer

Transformers, introduced in the paper titled "Attention is All You Need" by Vaswani et al. in 2017, are a class of neural network architectures that have revolutionized various natural language processing and machine learning tasks. A high level view of its architecture is shown in Fig. 1.

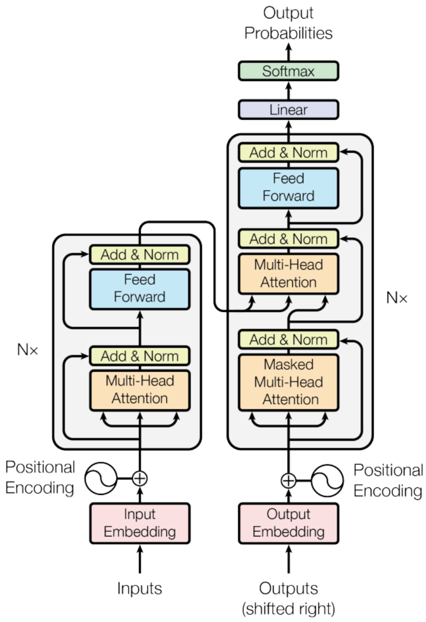


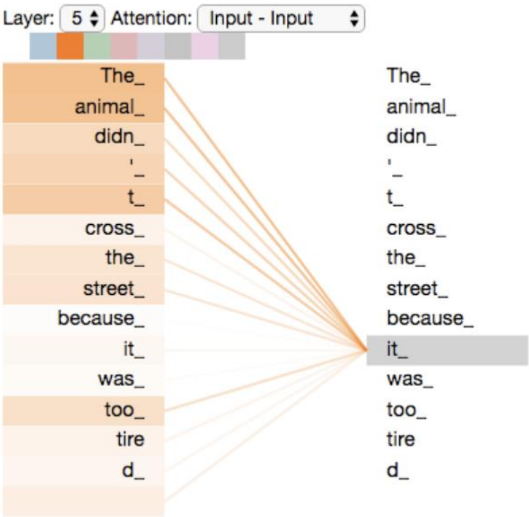
Fig. 1. The Transformer model architecture showing the encoder (left block) and decoder components (right block) [2]

Since its introduction, transformers have served as the foundation for many state-of-the-art models in NLP; including BERT, GPT, and more. Fundamentally, they are designed to process sequential data, such as text data, without the need for recurrent or convolutional layers [2]. They achieve this by relying heavily on a mechanism called *self-attention*.

The self-attention mechanism is a key innovation introduced in the paper that allows the model to capture relationships between different elements in a given sequence by weighing the importance of each element in the sequence with respect to other elements [2]. Say for instance, you want to translate the following sentence:

"The animal didn't cross the street because it was too tired."

What does the word "*it*" in this particular sentence refer to? Is it referring to the street or the animal? For us humans, this may be a trivial question to answer. But for an algorithm, this can be considered a complex task to perform. However, through the self-attention mechanism, the transformer model is able to estimate the relative weight of each word with respect to all the other words in the sentence, allowing the model to associate the word "it" with "animal" in the context of our given sentence [4].



Layer: 5 Attention: Input - Input

Fig. 2. Sample output of the 5th encoder in a 5-encoder stack self-attention block given the word "it" as an input. We can see that the attention mechanism is associating our input word with the phrase "The Animal" [4].

2.1. The Self-Attention Mechanism

A transformer *transforms* a given input sequence by passing each element through an encoder (or a stack of encoders) and a decoder (or a stack of decoders) block, in parallel [2]. Each encoder block contains a self-attention block and a feed forward neural network. In this report, I shall only focus on the *transformer encoder* block as this was the component used by Dosovitskiy et al. in their Vision Transformer image classification model.

As is the case with general NLP applications, the first step in the encoding process is to turn each input word into a vector using an embedding layer which converts our text data into a vector that represents our word in the vector space while retaining its contextual information. We then compile these individual word embedding vectors into a matrix *X*, where each row *i* represents the embedding of each element *i* in the input sequence. Then, we create three sets of vectors for each element in the input sequence; namely, Key (K), Query (Q), and Value (V). These sets are derived by multiplying matrix *X* with the corresponding trainable weight matrices W^Q, W^K, and W^V [2].

$$K = XW_k \tag{1}$$

$$Q = XW_Q \tag{2}$$

$$V = XW_V \tag{3}$$

$$attention\ factor = \frac{softmax(QK)}{\sqrt{d_k}} \tag{4}$$

$$Z = \ Attention(Q, K, V) = (attention\ factor)(V) \tag{5}$$

Afterwards, we perform a matrix multiplication between K and Q, divide the result by the square-root of the dimensionality of K, $\sqrt{d_k}$, and then apply a softmax function to normalize the output and generate weight values between 0 and 1 [2].

In this report, I will call this intermediary output the *attention factor.* This factor, shown in Eq. 4, represents the weight that each element in the sequence contributes to the calculation of the attention value at the current position (word being processed). The idea behind the softmax operation is to amplify the words that the model thinks are relevant to the current position, and attenuate the ones that are irrelevant. For example, in Fig. 3, the input sentence "*He later went to report Malaysia for one year*" is passed into a BERT encoder unit to generate a heatmap that illustrates the contextual relationship of each word with each other. We can see that words that are deemed contextually associated produce higher weight values in their respective cells, visualized in a dark pink color, while words that are contextually unrelated have low weight values, represented in pale pink.

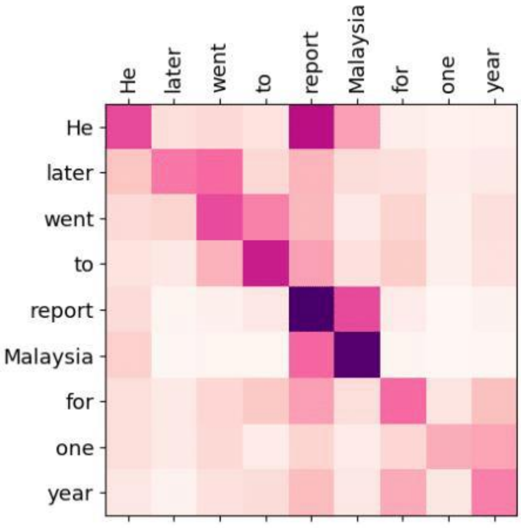


Fig. 3. Attention matrix visualization – weights in a BERT Encoding Unit [5]

Finally, we multiply the attention factor matrix to the value matrix V to compute the aggregated self-attention value matrix Z of this layer [2], where each row i in Z represents the attention vector for word i in our input sequence. This aggregated value essentially bakes the "context" provided by other words in the sentence into the current word being processed. The attention equation shown in Eq. 5 is sometimes also referred to as the *Scaled Dot-Product Attention*.

2.2 The Multi-Headed Self-Attention

In the paper by Vaswani et. al., the self-attention block is further augmented with a mechanism known as the "multi-headed" self-attention, shown in Fig 4. The idea behind this is instead of relying on a single attention mechanism, the model employs multiple parallel attention "heads" (in the paper, Vaswani et. al. used 8 parallel attention layers), wherein each of these attention heads learns different relationships and provides unique perspectives on the input sequence [2]. This improves the performance of the attention layer in two important ways:

First, it expands the ability of the model to focus on different positions within the sequence. Depending on multiple variations involved in the initialization and training process, the calculated attention value for a given word (Eq. 5) can be dominated by other certain unrelated words or phrases or even by the word itself [4]. By computing multiple attention heads, the transformer model has multiple opportunities to capture the correct contextual relationships, thus becoming more robust to variations and ambiguities in the input.

Second, since each of our *Q, K, V* matrices are randomly initialized independently across all the attention heads, the training process then yields several *Z* matrices (Eq. 5), which gives the transformer multiple *representation subspaces* [4]. For example, one head might focus on syntactic relationships while another might attend to semantic meanings. Through this, the model is able to capture more diverse relationships within the data.

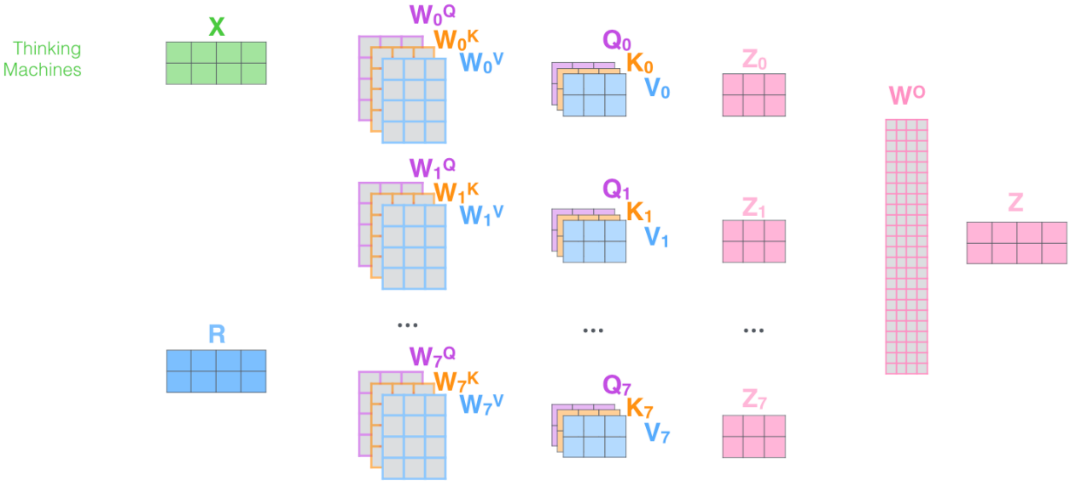


Fig. 4. Illustrating the Multi-Headed Self-Attention Mechanism. Each individual attention head yields an attention value matrix Z_i, which are concatenated and multiplied to a learned matrix W^O to generate the aggregated multi-headed self-attention value matrix *Z* [4].

3. The Vision Transformer

The fundamental innovation behind the Vision Transformer (ViT) revolves around the idea that images can be processed as sequences of tokens rather than grids of pixels. In traditional CNNs, input images are analyzed as overlapping tiles via a sliding convolutional filter, which are then processed hierarchically through a series of convolutional and pooling layers. In contrast, ViT treats the image as a collection of *non-overlapping* patches, which are treated as the input sequence to a standard Transformer encoder unit.

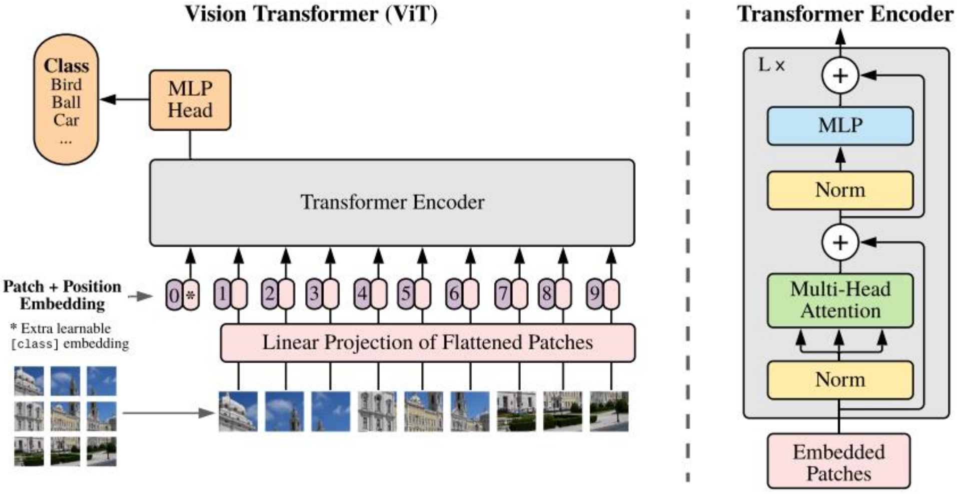


Fig. 5. The Vision Transformer architecture (left), and the Transfomer encoder unit derived from the Fig. 1 (right)[3].

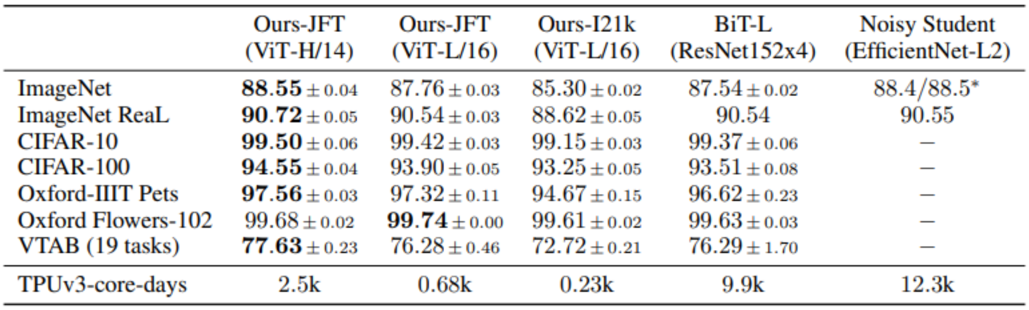
By defining the input tokens to the transformer as non-overlapping image patches rather than individual pixels, we are therefore able to reduce the dimension of the attention map from $(H\ x\ W)^2$ to $\left(n_{ph}\ x\ n_{pw}\right)^2$ given $n_{ph} \ll H$ and $n_{pw} \ll W$; where H and W are the height and width of the image, and n_{ph} and n_{pl} are the number of patches in the corresponding axes. By doing so, the model is able to handle images of varying sizes without requiring extensive architectural changes [3].

These image patches are then linearly embedded into lower-dimensional vectors, similar to the word embedding step that produces matrix X in Part 2.1. Since transformers do not contain recurrence nor convolutions, they lack the capacity to encode positional information of the input tokens and are therefore permutation invariant [2]. Hence, as it is done in NLP applications, a positional embedding is appended to each linearly encoded vector prior to input into the transformer model, in order to encode the spatial information of the patches, ensuring that the model understands the position of each token relative to other tokens within the image. Additionally, an extra learnable classifier *cls* embedding is added to the input. All of these (the linear embeddings of each 16 x 16 patch, the extra learnable classifier embedding, and their corresponding positional embedding vectors) are passed through a standard Transformer encoder unit as discussed in Part 2. The output corresponding to the added learnable *cls* embedding is then used to perform classification via a standard MLP classifer head [3].

4. Results Discussion

In the paper, the two largest models, ViT-H/14 and ViT-L/16, both pre-trained on the JFT-300M dataset, are compared to state-of-the-art CNNs—as shown in Table II, including Big Transfer (BiT), which employs supervised transfer learning with large ResNets, and Noisy Student, a large EfficientNet trained using semi-supervised learning on ImageNet and JFT-300M without labels [3]. At the time of this study's publication, Noisy Student held the state-of-the-art position on ImageNet, while BiT-L on the other datasets utilized in the paper [3]. All models were trained in TPUv3 hardware, and the number of TPUv3-core-days that it took to train each model were recorded.

TABLE II. Comparison of model performance against popular image classification benchmarks. Reported here are the mean and standard deviation of the accuracies, averaged over three fine-tuning runs [3].



	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

We can see from the table that Vision Transformer models pre-trained on the JFT-300M dataset outperforms ResNet-based baseline models on all datasets; while, at the same time, requiring significantly less computational resources (TPUv3-core-days) to pre-train. A secondary ViT-L/16 model was also trained on a much smaller public ImageNet-21k dataset, and is shown to also perform relatively well while requiring up to 97% less computational resources compared to state-of-the-art counter parts [3].

Fig. 6 shows the comparison of the performance between the BiT and ViT models (measured using the ImageNet Top1 Accuracy metric) across different pre-training datasets of varying sizes. We see that the ViT-Large models underperform compared to the base models on the small datasets like ImageNet, and roughly equivalent performance on ImageNet-21k. However, when pre-trained on larger datasets like JFT-300M, the ViT clearly outperforms the base model [3].

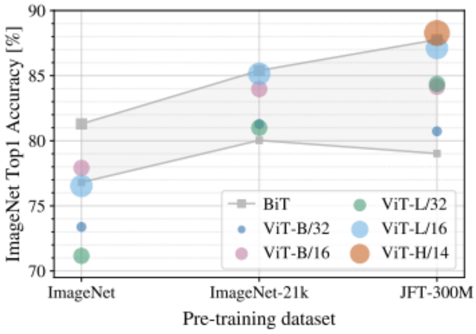


Fig. 6. BiT (ResNet) vs ViT on different pre-training datasets [3].

Further exploring how the size of the dataset relates to model performance, the authors trained the models on various random subsets of the JFT dataset—9M, 30M, 90M, and the full JFT-300M. Additional regularization was not added on smaller subsets in order to assess the intrinsic model properties (and not the effect of regularization) [3]. Fig. 7 shows that ViT models overfit more than ResNets on smaller datasets. Data shows that ResNets perform better with smaller pre-training datasets but plateau sooner than ViT; which then outperforms the former with larger pre-training. The authors conclude that on smaller datasets, convolutional inductive biases play a key role in CNN model performance, which ViT models lack. However, with large enough data, learning relevant patterns directly outweights inductive biases, wherein ViT excels [3].

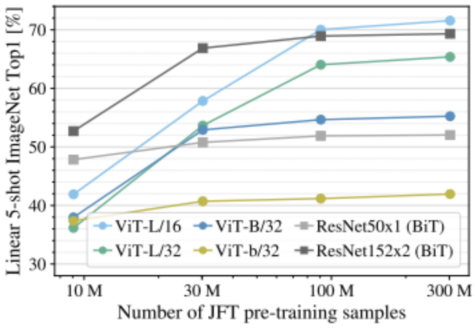


Fig. 7. ResNet vs ViT on different subsets of the JFT training dataset [3].

Finally, the authors analyzed the models' transfer performance from JFT-300M vs total pre-training compute resources allocated, across different architectures, as shown in Fig. 8. Here, we see that Vision Transformers outperform ResNets with the same computational budget across the board. ViT uses approximately 2-4 times less compute to attain similar performance as ResNet [3]. Implementing a hybrid model does improve performance on smaller model sizes, but the discrepancy vanishes for larger models, which the authors find surprising as the initial hypothesis is that the convolutional local feature processing should be able to assist ViT regardless of compute size [3].

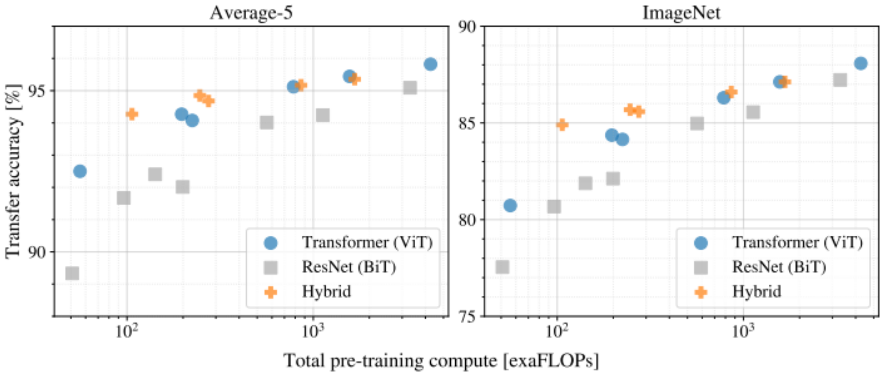


Fig. 8. Performance of the models across different pre-training compute values—exa floating point operations per second (or exaFLOPs) [3].

4.1 What does the ViT model learn?

In order to understand how ViT processes image data, it is important to analyze its internal representations. In Part 3, we saw that the input patches generated from the image are fed into a linear embedding layer that projects the 16x16 patch into a lower dimensional vector space, and its resulting embedded representations are then appended with positional embeddings. Fig. 9 shows that the model indeed learns to encode the relative position of each patch in the image. The authors used cosine similarity between the learned positional embeddings across patches [3]. High cosine similarity values emerge on similar relative area within the position embedding matrix corresponding to the patch; i.e., the top right patch (row 1, col 7) has a corresponding high cosine similarity value (yellow pixels) on the top-right area of the position embedding matrix [3].

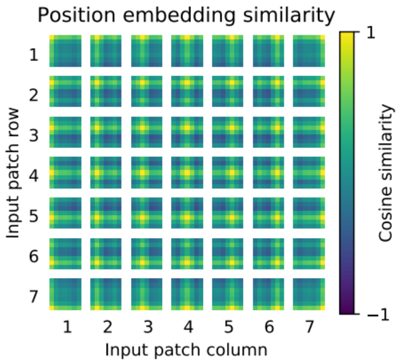


Fig. 9. Learned positional embedding for the input image patches [3].

Meanwhile, Fig. 10 (left) shows the top principal components of learned embedding filters that are applied to the raw image patches prior to the addition of the positional embeddings. What's interesting for me is how similar this is to the learned hidden layer representations that you get from Convolutional neural networks, an example of which is shown in the same figure (right) using the AlexNet architecture.

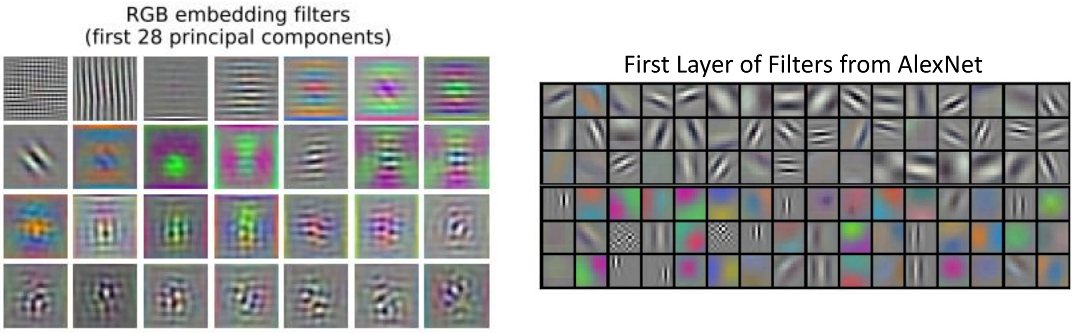

RGB embedding filters
(first 28 principal components)
First Layer of Filters from AlexNet

Fig. 10. Filters of the initial linear embedding layer of ViT-L/32 (left) [3].
The first layer of filters from AlexNet (right) [6].

By design, the self-attention mechanism should allow ViT to integrate information across the entire image, even at the lowest layer, effectively giving ViTs a global receptive field at the start. We can somehow see this effect in Fig. 10 where the learned embedding filters captured lower level features like lines and grids, as well as higher level patterns combining lines and color blobs. This in contrast with CNNs whose receptive field size at the lowest layer is very small (because local application of the convolution operation only *attends* to the area defined by the filter size), and only widens towards the deeper convolutions as further applications of convolutions extract context from the combined information extracted from lower layers. The authors further tested this by measuring the *attention distance* which is computed from the "average distance in the image space across which information is integrated based on the attention weights [3]." The results are shown in Fig. 11.

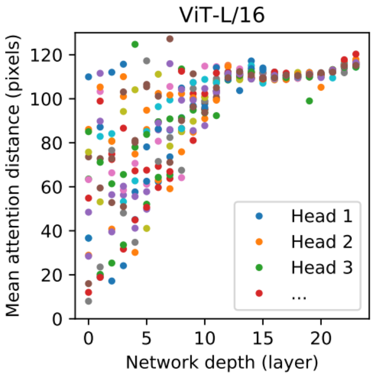


Fig. 11. Size of attended area by head and network depth [3].

From the figure, we can see that even at very low layers of the network, some heads attend to most of the image already (as indicated by data points with high mean attention distance value at lower values of network depth); thus proving the ability of the ViT model to integrate image information globally, even at the lowest layers.

Finally, the authors also calculated the attention maps from the output token to the input space using Attention Rollout by averaging the attention weights of the ViT-L/16 across all heads and then recursively multiplying the weight matrices of all layers. This results in a nice visualization of what the output layer attends to prior to classification, shown in Fig. 12 [3].

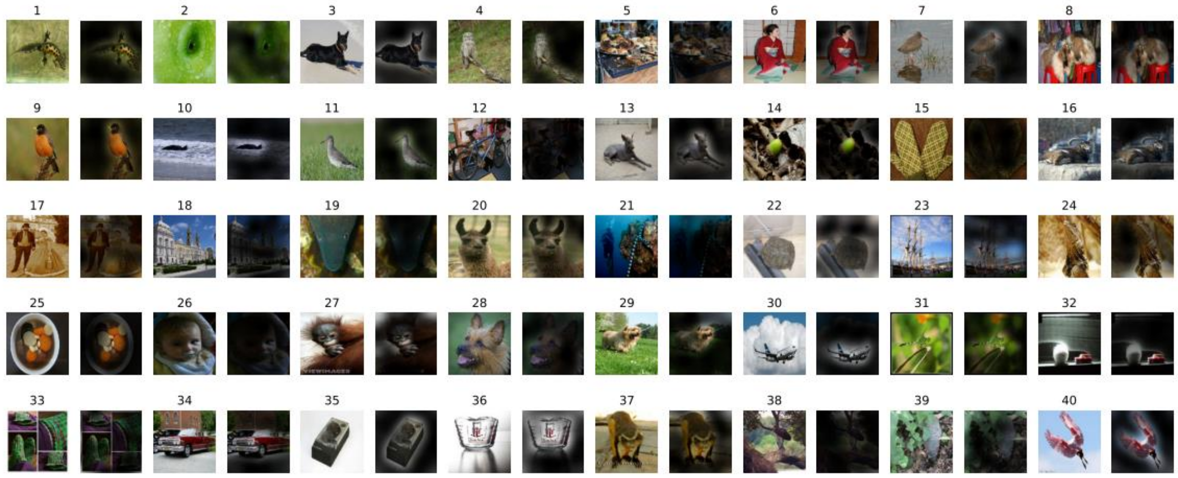


Fig. 12. Representative examples of attention from the output token to the input space [3].

5. Limitations and Future Work

The Vision Transformer (ViT) introduced by Dosovitskiy et. al. in the research study showcased in this paper is a groundbreaking architecture for computer vision tasks. Unlike previous methods that introduce image-specific biases, ViT treats an image as a sequence of patches and process it using a standard Transformer encoder, such as how Transformers are used in NLP. This straightforward yet scalable strategy, combined with pre-training on extensive datasets, has yielded impressive results as discussed in Part 4. The Vision Transformer (ViT) either matches or surpasses the state-of-the-art on numerous image classification datasets (Fig. 6, 7, and 8), all while maintaining cost-effectiveness in pre-training [3].

However, like in any technology, it has its limitations. First, in order to perform well, ViTs require a very large amount of training data that not everyone has access to in the required scale, especially when compared to traditional CNNs. The authors of the paper used the JFT-300M dataset, which is a limited-access dataset managed by Google [7]. The dominant approach to get around this is to use the model pre-trained on the large dataset, and then fine-tune it to smaller (downstream) tasks. However, second, there are still very few pre-trained ViT models available as compared to the available pre-trained CNN models, which limits the availability of transfer learning benefits for these smaller, much more specific computer vision tasks. Third, by design, ViTs process images as sequences of tokens (discussed in Part 3), which means they do not naturally capture spatial information [3]. While adding positional embeddings do help remedy this lack of spatial context, ViTs may not perform as well as CNNs in image localization tasks, given CNNs convolutional layers that are excellent at capturing these spatial relationships.

Moving forward, the authors mention the need to further study scaling ViTs for other computer vision tasks such as image detection and segmentation, as well as other training methods like self-supervised pre-training [3]. Future research may focus on making ViTs more efficient and scalable, such as developing smaller and more lightweight ViT architectures that can still deliver the same competitive performance. Furthermore, providing better accessibility by creating and sharing a wider range of pre-trained ViT models for various tasks and domains can further facilitate the development of this technology in the future.

~ *** ~

References

[1] N. Pogeant, "Transformers - the NLP revolution," Medium, https://medium.com/mlearning-ai/transformers-the-nlp-revolution-5c3b6123cfb4 (accessed Sep. 23, 2023).

[2] A. Vaswani, et. al. "Attention is all you need." NIPS 2017.

[3] A. Dosovitskiy, et. al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," ICLR 2021.

[4] J. Alammar, "The illustrated transformer," The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time., http://jalammar.github.io/illustrated-transformer/ (accessed Sep. 24, 2023).

[5] H. Wang, "Addressing Syntax-Based Semantic Complementation: Incorporating Entity and Soft Dependency Constraints into Metonymy Resolution", Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Attention-matrix-visualization-a-weights-in-BERT-Encoding-Unit-Entity-BERT-b_fig5_359215965 [accessed 24 Sep, 2023]

[6] A. Krizhevsky, et. al. "ImageNet Classification with Deep Convolutional Neural Networks," NIPS 2012.

[7] C. Sun, et. al. "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era," Google Research, ICCV 2017.

** ChatGPT, used sparingly to rephrase certain paragraphs for better grammar and more concise explanations. All ideas in the report belong to me unless otherwise indicated. Chat Reference: https://chat.openai.com/share/165501fe-d06d-424b-97e0-c26a81893c69*