ConvShareViT: A Vision Transformer-like Architecture for Free-Space Optical Accelerators

Riad Ibadulla, Thomas M. Chen, *Senior Member, IEEE*, Constantino Carlos Reyes-Aldasoro, *Senior Member, IEEE*

***Abstract*—This paper introduces ConvShareViT, a novel deep learning architecture that adapts the Vision Transformer (ViTs) architecture to the 4f free-space optical system. ConvShareViT replaces linear layers in multi-head self-attention (MHSA) and Multilayer Perceptrons (MLPs) with a depthwise convolutional layer with shared weights across input channels. The effectiveness of the attention mechanism was analysed systematically in 12 experiments with different Models. Experimental results demonstrated that configurations with valid-padded shared convolutions successfully learned attention, achieving comparable quantitative attention scores to those obtained with standard ViTs. However, same-padded convolutions showed limitations in attention learning and operated like regular CNNs rather than transformer models. In terms of speed, ConvShareViT can theoretically achieve up to 3.04 times faster inference than GPU-based systems. This potential acceleration makes ConvShareViT an attractive candidate for future optical deep learning applications.**

***Index Terms*—Optical transformers, Neural Networks, Vision transformers, Convolutional transformers Free-space optics**

I. Introduction

Computer vision has advanced significantly with the development of the deep learning approaches [1]–[3]. Convolutional Neural Networks (CNNs) have been the standard approach for tasks like image classification [4], object detection [5], [6], image segmentation [7]–[9], visual recognition [10] due to their ability to efficiently capture spatial hierarchies in images [1]. CNNs were challenged by the invention of Vision (or Visual) Transformers (ViT) [11]–[15], which use the encoder of the transformer and its self-attention mechanisms [16]–[18]. Unlike the CNNs that treat an image as a whole, ViTs divide the images into patches, treating each patch as a sequence token. This approach has shown outstanding success [19], when models pre-trained on large datasets like ImageNet are fine-tuned on smaller, task-specific datasets. However, CNNs still stay the first choice for small datasets when trained from scratch, due to their simplicity and sometimes better performance over transformers for specific datasets [20], [21].

Deep Learning is well-known to be data hungry [22] and the amount of training data can have an impact on the results [23]–[25]. In addition to the large data requirements, as deep learning models grow in complexity, advanced hardware accelerators [26] are increasingly needed to efficiently manage their computational demands. Conventional electronic accelerators struggle with limitations in power consumption and processing speed as model sizes continue to expand [27]. Optical computing [28], [29], particularly the 4f system, offers a promising solution due to its ability to process data rapidly and with high energy efficiency . Optical systems take advantage of light's parallel processing capabilities. A 4f system, first described by Weaver and Goodman in 1966 [30], is an optical setup consisting of a laser, modulators, lenses, and a camera or photodetector, with each component separated by one focal length of the lens. The entire arrangement spans four focal lengths, hence the name '4f system' as illustrated in (Figure 1). This system, also known as a 4f correlator, is commonly used to perform the convolution operation in optics.

The 4f system is particularly effective for accelerating CNNs through its efficient handling of convolution operations. While various optical setups have been developed for specific neural networks [31]–[39], the 4f system's ability to accelerate CNNs could, in theory, be generalised across different network architectures, enabling a single device to support multiple model types. This creates the potential for a standardised optical approach to training neural networks. Since it is expected that the free space optics [40] and photonics markets [41] will grow in the near future, the integration with Deep Learning techniques is evidently important.

With ViTs gaining popularity in computer vision [42], and pattern recognition [43] using convolutional layers within the multi-head self-attention (MHSA) [44] layers on the same 4f system could simplify the hardware landscape. This eliminates the need for multiple specialised processors, streamlining both training and inference and making optical computing more practical and scalable for complex neural network tasks.

In this paper, we propose the Convolutional Shared Vision Transformers (ConvShareViT), a hybrid model that incorporates convolutional operations within the Vision Transformer architecture, enhancing its adaptability to optical systems like the 4f. ConvShareViT This model strictly uses convolutional operations in the MHSA layers and replaces traditional MLPs with convolution-based structures, which ensures a compatibility with the 4f system. We systematically analysed different architecture configurations strategies such as channel/kernel/mix tiling to optimise these operations and efficiently manage the computational load. Moreover, we analysed different methods of using convolutions within the ViT's MHSA layers.

ConvShareViT preserves the patch-based processing fundamental to the Vision Transformer philosophy. It is worth mentioning that there are other hybrid models such as CvT [45], which alter the intrinsic structure of Vision Transformers by incorporating pooling and downsizing mechanisms. , our ConvShareViT strictly preserves the patch-based processing

All authors are with School of Science and Technology, City St. George's, University of London, EC1V 0HB, London, UK

Constantino Carlos Reyes-Aldasoro is also with the Integrated Pathology Unit, Institute of Cancer Research, Sutton, UK

Manuscript received November 5, 2024; revised November 5, 2024.

fundamental to the Vision Transformer philosophy. In ConvShareViT This ensures that each image patch is processed independently through convolution-enhanced self-attention layers. , respecting the original Transformer design while In addition, ConvShareViT improves the model's ability to generalise from limited data by leveraging the locality and invariance properties typical of CNNs. While our work explores convolutional strategies similar to CvT, one of our objectives is also to demonstrate that MHSA can be implemented within a 4f optical system. This capability is essential not only for image classification, but also for enabling a wider range of transformer-based applications in optical computing. Therefore, we prioritise the replication and extension of the original ViT architecture, rather than adopting structural changes characteristic of CvT. Unlike prior work that adapts hardware to accommodate VITs, our approach adapts ViTs to align with the properties of the optical accelerators, bringing it to a new direction in the field of optical transformers.

The contributions introduced in this paper are significant in two key areas: firstly, we demonstrate that convolutional operations can be seamlessly integrated into Vision Transformers without compromising their essential patch-processing methodology. Secondly, we demonstrated the capability of ConvShareViT to learn attention through a systematic evaluation and visualisation of our models' performance with different methods of incorporating convolution operations into the MHSA and MLP of the ViT.

II. Related work

Deep Learning in computer vision has advanced significantly with the development of CNNs, and the introduction of Vision Transformers has elevated the field to new heights [46]. While both approaches are much more efficient than fully connected neural networks, the need to accelerate Deep Learning performance has become increasingly critical. Various techniques have been employed to speed up deep learning, such as using shallower networks [47], pruning redundant weights [48], or adopting lower quantisation levels [49]. Moreover, hardware accelerators, such as application-specific integrated circuits (ASICs) [50], have been used to greatly enhance training and inference speeds, often outperforming conventional CPUs/GPUs.

However, as Moore's law slows down [51] and the limits of electronic hardware become clearer, the need for alternative methods, such as optical accelerators, is growing more important. Optical computing, with its ability to process many tasks in parallel and at high speed, provides a promising way to overcome the challenges of traditional hardware and improve the performance of deep learning systems.

There are two main approaches to optical neural networks: the free-space method, and the silicon photonics approach. The free-space method uses spatial light modulators (SLMs) to propagate light through media such as air or vacuum, without the need for physical waveguides [31], [33], [35], [37]–[39]. The silicon photonics approach uses Mach–Zehnder interferometers (MZIs) [52], [53], which offer faster processing speeds, with clock rates reaching several GHz. However, it offers inferior parallelism compared to the free-space system.

The reader is referred to [54] for a comprehensive review of different methods of implementing optical neural networks. The review presents interesting comparisons of free-space techniques with silicon photonics methods. the latter being described as waveguide optical interconnection in their work.

Although direct optical matrix-vector multiplication approaches, such as demonstrated by Anderson *et al.* [55], implement Transformer architectures using free-space optics, our ConvShareViT model contributes uniquely by showing how existing 4f optical systems—already widespread for convolution operations—can be directly used to implement Vision Transformers. Our findings are significant because they indicate not only a path for a broad adoption of optical Transformers within existing devices but also deepen our theoretical understanding by proving that shared depthwise convolution is capable of learning attention. Therefore, our approach complements specialised matrix vector multiplication accelerators by expanding the versatility of convolution-based optical computing.

1) Free-space optical image classification: In this study, we focus on active 4f free-space optical accelerators, which play a crucial role in high-speed image processing and classification tasks. The 4f system is an optical setup used primarily for performing convolutions, a core operation in computer vision tasks such as image classification. The 4f system takes its name from the optical configuration, in which a light beam travels through four focal lengths between lenses and spatial modulators to perform a Fourier transform and its inverse as illustrated in Fig. 1. This system's ability to perform optical convolution exploits the Fourier transform, where the convolution operation in the spatial domain becomes a simple pointwise multiplication in the frequency domain.

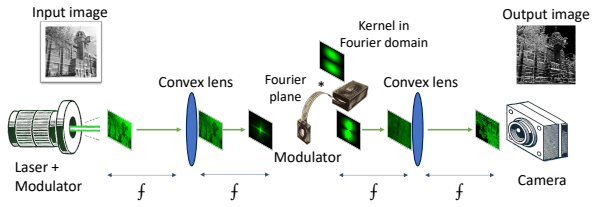


Fig. 1. Schematic illustration of a 4f optical system executing a convolution operation. The system consists of an input plane (laser source), a convex lens, and a Fourier plane (containing a modulator or phase mask), followed by another convex lens and a camera, each positioned one focal length away from the lenses. As the light passes through the first lens, it undergoes a 2D Fourier transform at the Fourier plane, where it is multiplied by the kernel in the frequency domain. The light then travels through the second lens, which transforms it back to the spatial domain, and the camera captures the output.

The process of the convolution operation in the 4f system with the simulated results is shown in Fig. 1 and performed as follows: First, light is directed onto the light modulator, where it is modulated by the input image. When the beam then passes through a convex lens, the resulting wavefront, after travelling one focal distance, represents the Fourier Transform of the image. Another light modulator, positioned in the Fourier plane, is used to perform element-wise multiplication with the image in the frequency domain. The modified beam then passes through a second lens, which performs the inverse

Fourier Transform, converting the image back into the spatial domain. The final output, after another focal distance, is the convolved image, which is captured by a camera.

2) Parallelism in 4f system: The primary advantage of 4f free-space optical neural networks is their ability to enable massive parallelism, allowing high-resolution inferences and the execution of multiple inferences simultaneously. This allows the 4f system to effectively process high-resolution inputs and kernels without compromising the frame rate. Since the 4f system efficiently processes high-resolution inputs, it is possible to parallelise the inference through the 4f system by tiling the inputs into a large block of inputs and performing the convolution of several inputs with the same kernel. This method is known as input tiling or batch tiling [36], as the inputs of the entire mini-batch can be tiled to fit the entire input tensor (n-dimensional matrix) of the CNN.

Since the convolution operation is commutative, the order of the operands does not matter. This means that tiling can be applied to the kernels instead of inputs, given that the input is a single 2D feature map. This approach has been used by Chang *et al.* [32] and also described by Li *et al.* [56] as kernel tiling, where the kernels are zero-padded to a size of $(M+N-1)\times(M+N-1)$, with $M\times M$ being the input resolution and $N\times N$ representing the kernel resolution. After padding, these kernels are arranged into a single large tiled kernel block. The input must also be padded to match the dimensions of this kernel block to enable optical convolution. The resulting output consists of multiple tiled sections, each corresponding to the convolution of the input feature map with one of the kernels from the tiled array.

Another approach by Li *et al.* [31] achieved parallelism in CNNs using the 4f system, which is channel tiling. Similar to kernel tiling, this method involves padding of both kernels and channels. The padded channels and kernels are then tiled into respective blocks and convolved. The result of the convolution produces a block of outputs with dimensions $(2\sqrt{N_c}-1)\times(2\sqrt{N_c}-1)$, where N_c is the number of input channels. All outputs, except for the one in the centre, are considered invalid and discarded. The valid output represents the sum of the convolution of each input channel with its corresponding kernel. As this method provides both the convolution and the summation of the results, it can be utilised in the channel summation process.

Unfortunately, this method computes only a single output channel. The third and most efficient method, as described by Li *et al.*, is mixed tiling, which allows the entire convolutional layer to be performed in a single inference through the 4f system. Mixed tiling combines both kernel tiling and channel tiling, providing full parallelism for the entire convolutional layer. This method ensures that the convolution of all input channels with their respective kernels is completed, and the results are summed across the output channels. However, it requires a significant amount of spatial space within the 4f system, often making it impractical to execute the entire process in a single inference.

In this method, inputs are padded as before and tiled horizontally. Likewise, the kernels are also padded to the same dimensions and tiled along both the x and y axes, with each row corresponding to an output channel. Similar to channel tiling, the output block contains invalid regions due to unnecessary convolutions, while the valid outputs are located in the centre of each row of the output block.

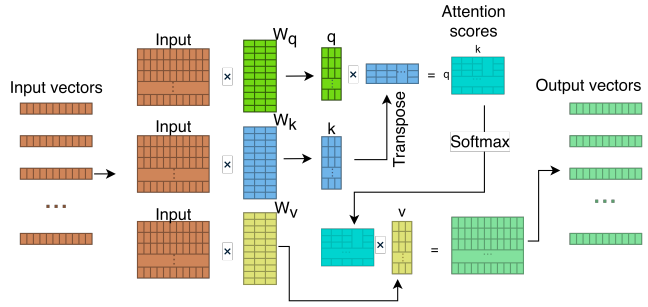


Fig. 2. Self-Attention mechanism. Inputs are mapped to Query, Key, and Value vectors. Attention scores, calculated from Query-Key multiplications, achieving dependencies between tokens. These scores are then used to weight the Value matrix, amplifying relevant information.

3) Vision Transformers: ViTs represent a paradigm shift in the field of computer vision, achieving state-of-the-art results by using transformer-based architectures, initially designed for natural language processing (NLP) [13], [57]. Unlike traditional CNNs, which rely on convolutional operations to extract hierarchical features from images, ViTs apply the transformer's attention mechanism to capture global dependencies between different regions of an image. This is illustrated in the top half of Fig. 3.

At the core of ViTs is the self-attention mechanism, a component originally introduced in the transformer models for NLP. The attention mechanism allows the network to weigh the relationships between different parts of the input data, enabling it to capture contextual information across the entire image. To process an image, the ViT model divides it into fixed-size patches, which are treated similarly to tokens in NLP models (Fig. 3). Each patch is then flattened into a vector, and positional embeddings are added to retain spatial information, ensuring that the ViT is aware of the position of the patches in the original image. This helps to maintain the spatial structure which could be lost after the tokenisation.

The tokens are passed through consecutive MHSA and MLP layers and repeated N times. The depth of the transformer is a crucial hyper-parameter, which depends on the task.

The pipeline for the self-attention mechanism is shown in Fig. 2, where the tokens are part of the matrix X. The input matrix is multiplied by matrices W_q, W_k, and W_v produce new sets of Q (Query), K (Key) and V (Value) matrices. The multiplication of the input matrix by W_q, W_k, and W_v is simply achieved using a linear layer without a bias term. It is important to note that each row of the input matrix (token) is mapped to the Q, K, and V independently without any interaction with the adjacent tokens. The interaction between the tokens occurs later when calculating the attention scores through matrix multiplication $QK^\top$.

The self-attention of each head is computed as follows:

$$Attention(Q,K,V) = softmax(\frac{QK^\top}{\sqrt{d}})V, \qquad (1)$$

where d is the dimensionality of the QKV vectors, included to scale the results of attention scores and improve numerical stability in the softmax calculation. Q, K, and V represent query, key, and value matrices respectively, which are mapped from the input embeddings by linear projections.

Multi-Heads Self-Attention (MHSA) layer processes multiple self-attention mechanisms in parallel, each focusing on different aspects of the patches.

4) Use of convolution operations in transformers: One of the ViT variations, which used a similar shape of the network as the CNNs, is the Pyramid Vision Transformer (PVT) [58]. PVT reshapes feature maps into a matrix form after each transformer block, creating a pyramid-like architecture where the feature map size progressively reduces. This hierarchical structure similar to the CNNs, unlike the standard ViT, is better suited for visual tasks such as classification. Notably, PVT-Large delivers comparable top-1 ImageNet accuracy (a 0.1% decrease) to ViT-Base/16 [11] while significantly reducing computational complexity (9.8 GFLOPs vs. 17.6 GFLOPs) and the number of parameters (61.4M vs. 86.6M).

While PVT mimics CNN-like feature extraction through its cone-shaped architecture, it does not incorporate convolutional layers. However, another model known as the Convolutional Vision Transformer (CvT) [45] model introduces convolution operations into the ViT pipeline, where overlapping convolutions are used for token embedding. This step captures local spatial information while simultaneously reducing the sequence length and increasing token feature dimensions, similar to CNN architectures. The convolutional token embedding stage also spatially down-samples the feature maps while increasing the number of channels, which enhances the model's hierarchical representation capabilities.

In each Convolutional Transformer Block, CvT replaces the standard matrix multiplication for query, key, and value embeddings with depthwise separable convolutions. After each transformer block, the 2D feature map is reconstructed and passed through another convolutional token embedding layer, ensuring a cone-shaped architecture similar to CNNs used in classification tasks.

Unlike standard ViTs, which process non-overlapping patches and rely solely on the attention mechanism for communication between patches, CvT re-tokenizes the feature map after each transformer block, allowing for integrated feature representations. Additionally, due to the use of overlapping patches, CvT eliminates the need for positional encoding.

In the work of Ding *et al.* [59], the authors revisit large kernel design in convolutional networks, proposing RepLKNet, a CNN architecture with kernel sizes as large as 31×31. This work is particularly notable because, although it does not employ the attention mechanisms seen in ViTs, it effectively closes the performance gap between CNNs and ViTs. According to the authors, the reason for the strong performance in ViTs is their high effective receptive field (ERF), which can be replicated in CNNs by leveraging large depthwise convolutions. This allows the model to capture both local and global information, similar to what attention mechanisms achieve in ViTs. Ding *et al.* demonstrate that the large kernels of RepLKNet deliver competitive results on ImageNet, achieving 84.8% top-1 accuracy, which is on par with the Swin Transformer [60] but with lower latency. Their approach shows that CNNs, through the strategic use of large kernels, can match or even surpass ViT performance without relying on attention mechanisms.

III. Proposed method

Regular vision Transformers can be divided into four main components: tokenisation, multi-head self-attention, multilayer perceptron, and classifier head. Multi-head self-attention, on the other hand, can be split into three main stages: QKV projection, attention score calculation, and weighted sum of values. In this section, we will describe how each task is transformed to use only convolution operations.

1) Shared depthwise convolution: It is important to note initially that the linear layer is the main component of all layers in the transformer's encoder, as it can be seen from Fig. 3 and as part of the MHSA from Fig. 2. Each output node of the linear layer is the weighted sum of the input nodes as illustrated in Fig. 4 (a). Since the ConvShareViT model deals with patches as matrices rather than vectors, each patch can be convolved with the weight matrix of the same resolution, with valid padding to achieve the output node, as shown in Fig. 4 (b). If the same padding is used, the middle pixel of the output will be the valid region. This highlights a key architectural constraint for ConvShareViT: using valid padding results in a single output per token (matching a linear projection), whereas same padding produces a feature map and can behave like a convolutional feature extractor rather than a token-wise linear mapping.As shown in Fig. 4 (c), when applied optically, kernel tiling allows the simultaneous computation of all output pixels.

In a typical convolutional layer, the kernels are 3D, with the number of kernels matching the number of output channels and the depth of each kernel equal to the number of input channels. Alternatively, this can be viewed as having a set of 2D kernels, where each pair of input and output channels has its own 2D kernel. The results of these 2D convolutions are then summed across the input channels to produce the final output, as shown in Fig. 5 (a).

In our case, each input channel corresponds to a separate patch that needs to be processed independently, without any interaction between patches. Therefore, summing across channels should be avoided. To achieve this, we use depthwise convolution, where the number of convolution groups is equal to the number of input channels (See Fig. 5 (b)).

However, when a tensor passes through a linear layer, the last dimensions are all mapped into new vectors, meaning the same layer is applied to all dimensions, or in other words, the weights are shared across input channels. Thus, if we emulate a linear layer using convolution, the kernels must be repeated for each input channel during depthwise convolution. This approach, shown in Fig. 5 (c), is what we refer to as shared depthwise convolution. If kernels are not shared, the layer no longer emulates a linear projection applied to all tokens, and it instead introduces channel-specific feature extraction behaviour similar to CNNs. This mimics a linear layer because each input channel (i.e., patch) is convolved with

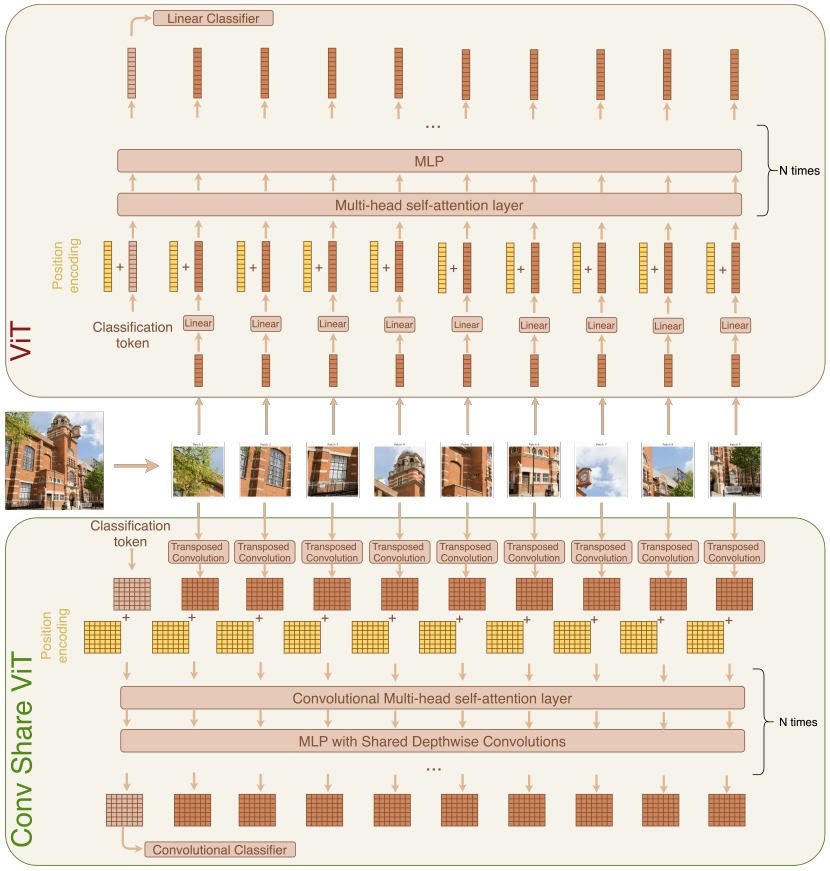


Fig. 3. Comparison of the regular ViT (top of figure) and ConvShareViT (bottom of the figure) pipelines. ViTs vectorise patches of the image and apply a linear layer to map them into higher dimensional embeddings, while ConvShareViT keeps the patches in 2D format and uses Transpose Convolution to increase the dimensionality. ConvShareViT uses MHSA and MLPs using Shared Depthwise Convolutional layers.

an identical kernel, resulting in the same linear transformation being applied independently across spatial locations, just as in a linear layer where the same weight matrix is applied to each input vector. Avoiding channel mixing and repeating kernels across all inputs preserve the behaviour of the linear transformation within the convolutional framework.

2) Multi-head self attention: As it was mentioned before, unlike Vision Transformers, ConvShareViT uses a tensor instead of a matrix as the input. In Vision Transformers, the input is the matrix, where each row vector is the embedding of the input patch. In ConvShareViT, the embedding is represented as the 2D matrix of a 3D tensor, where the depth of the tensor corresponds to the number of patches (tokens). When passed to the different heads of the multi-head self-attention layer, the matrix is split into equal patches, and each patch is fit into its corresponding head. The output of each head is then located at the correct location, where the patch was initially taken from (See Fig. 6).

When the patches are inputted into the head of the self-attention layer, they go through the QKV projection. In the original Transformer, this stage was performed by passing the inputs through the linear layer.

This can be replicated by using our shared depthwise convolutional layer, with the kernel resolution equal to the input resolution and the number of kernels (output channels) equal to the number of nodes required at the output of the mimicked linear layer. The spatial resolution of the output of the shared depthwise convolution will be 1×1 with a number of channels equal to the total number of nodes of the mimicked linear layer, which can be reshaped into the regular matrices. The input tensor is passed through three distinct shared depthwise convolutions in order to achieve Query, Key and Value tensors.

The next stage is the attention score calculation. In the original Vision Transformer, this is achieved by matrix multiplication of the Q with the transposed K matrix. In other words, the dot product of all vectors Q and K. In ConvShareViT, tokens are preserved as 2D matrices rather than flattened vectors, and the attention score between tokens i and j is therefore computed as $s_{i,j} = \sum_{u,v} Q_i(u,v)K_j(u,v)$, which is equivalent to the standard dot product after flattening.

For two matrices of identical resolution, this dot product is equivalent to a valid 2D cross-correlation between Q_i and K_j. This corresponds to sliding the key kernel over the query feature map and retaining only the valid region, resulting in one scalar per token pair.

As in deep-learning frameworks, this operation corresponds to cross-correlation rather than mathematical convolution; however, we use the term convolution throughout in keeping with the optics literature. Although these operations differ in

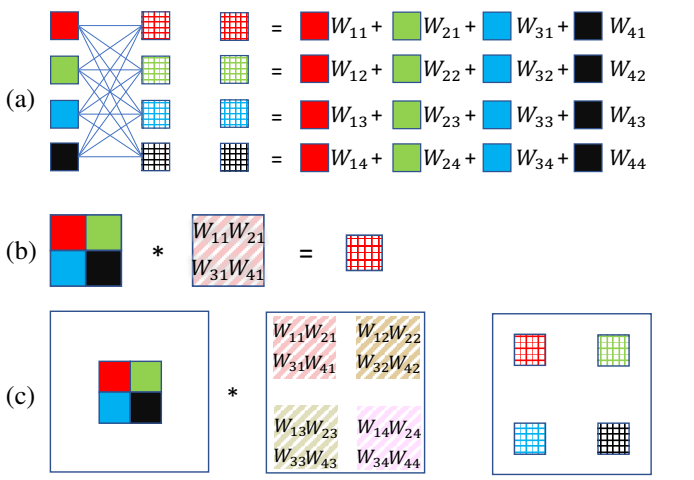


Fig. 4. Implementation of the linear layer using convolution and tiled convolution for 4f system. (a) A simple linear layer of one input vector with four values represented by four boxes with solid colours. The output vector, boxes with different colours in grid layout, is the vector-matrix multiplication of the input vector with the weight matrix with values $W_{11}, W_{21}, W_{31}, W_{41}, W_{21}, \ldots$. Each output node has its own set of weights. (b) Illustration of a case where the input is in 2D matrix format, which is then convolved with the kernel of equal size and valid padding i.e., the white region around the kernel. Weights have diagonal coloured lines. The output is similar to one output pixel of the linear layer. (c) Kernel tiling is used to tile all weights of the linear layer in the kernel block. The input is padded to the required resolution. The output archives all linear layer output nodes, with the requirement of reshaping (removes zeros in invalid regions).

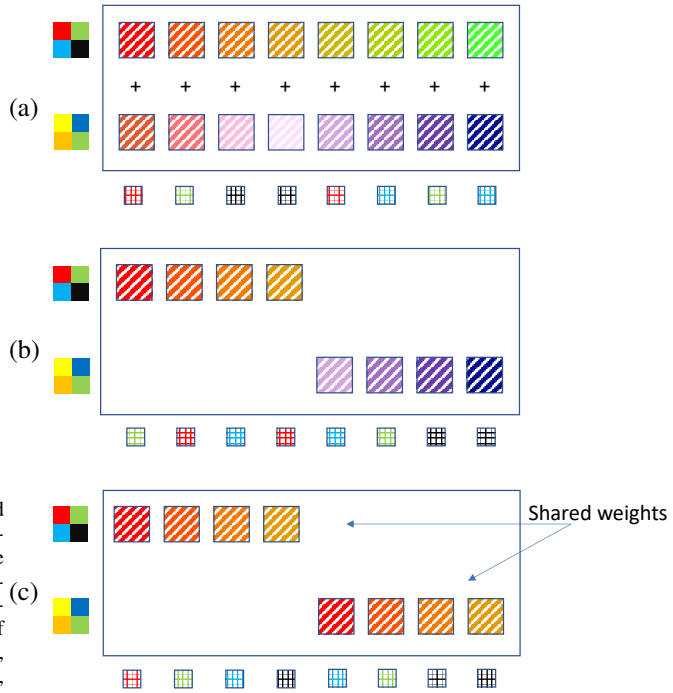


Fig. 5. Shared depthwise convolutional layer, copies the weights across all input channels. As with Fig. 4 , the input values are represented with solid colours, the weights with diagonal lines and the output with grids. (a) Regular convolutional layer, with the groups=1. The number of 2D kernels is equal to the number of input channels $\times$ the number of output channels. (b) Depthwise convolution, where the number of groups is equal to the number of input channels. In this case, each output channel gets only one 2D kernel, meaning no channel summation happens. (c) In the shared depthwise convolutional layer, unlike the regular depthwise convolutional layer, the weights are shared across input channels, making it ideal for the emulation of the Linear Layer. If kernels are the same resolution as inputs, the valid convolution yields 1 pixel for each output channel, which can be reshaped into the initial resolution.

kernel orientation and indexing conventions, they are all linear transformations and, under the reshaping scheme used in ConvShareViT, they produce equivalent dot-product interactions between tokens.

The resulting matrix can be scaled, as in the original Vision Transformer, by the square root of the token dimension – in our case, simply the width or height of the token matrix, and normalised with softmax to obtain the attention weights A.

The last step is the weighted sum of the values using the attention weights $Y = AV$, which we implement as a point-wise (1×1) convolutional layer by treating the attention weights as the layer weights. The outputs are then located in the correct location of the main larger patches.

To do this using the convolution, it first needs to look at the general formula for the 2D convolutional layer without bias:

$$Y_{k,p,q} = \sum_{c=1}^{C_{in}} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} X_{c,p+i,q+j} \times W_{k,c,i,j}, \tag{2}$$

where, $X_{c,p+i,q+j}$ refers to the input feature map with the dimension Channels, Height, Width ($[C_{in}, H_{in}, W_{in}]$), and the output Y with the dimensions $[C_{out}, H_{out}, W_{out}]$. W represents the set of kernels with the dimension $[C_{out}, C_{in}, H, W]$.

When the convolution is 1D, the spatial dimension is reduced to one, the convolution is simplified to:

$$Y_{k,p} = \sum_{c=1}^{C_{in}} \sum_{i=-a}^{a} X_{c,p+i} \times W_{k,c,i}. \tag{3}$$

When the convolutional layer uses 1x1 kernels, the convolution effectively becomes a point-wise matrix multiplication across channels, identical to dense layer operations. For 1D convolution, the formula with 1x1 kernels becomes:

$$Y_{k,p} = \sum_{c=1}^{C_{in}} X_{c,p} \times W_{k,c}, \tag{4}$$

which is equivalent to $W \times X$. This leads to the conclusion that matrix multiplication can be treated as the convolutional layer, with the left term being a weight matrix.

3) Multi Layer Perceptron: In a vanilla transformer encoder, each multi-head self-attention layer is followed by an MLP layer. This MLP usually consists of two linear layers; the first one maps the embedding vectors into the high-dimensional space, and the second one maps it back to the original dimension. One of the hyperparameters of the MLP is the MLP ratio, which indicates how many times the dimension is scaled, meaning the ratio of the hidden layer to the input or output layer. Since the original transformer uses a linear layer, our method leverages the same concept used for the QKV projection in the MHSA layer.

First, a depthwise convolutional layer with a kernel size equal to the input size is used to map the input into the higher dimension. The output of this layer's resolution is 1×1 and the number of channels equals to (number of tokens $*$ mlp_ratio $*$

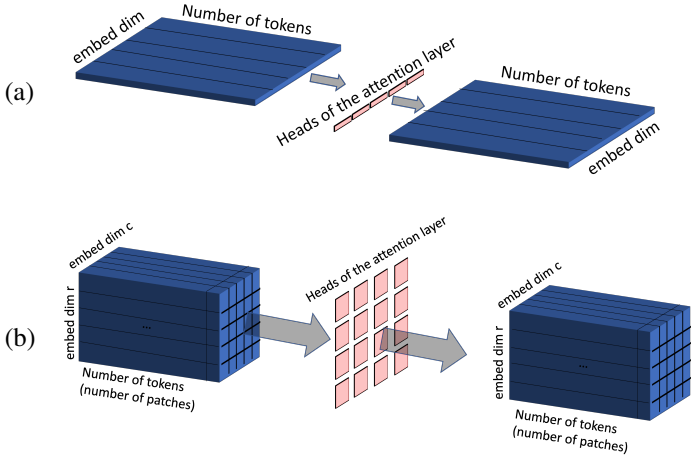


Fig. 6. Visual comparison of input split in regular multi-head attention and our method when the inputs are two-dimensional. (a) In regular multi-head attention, the input vectors are split into equal-sized vectors, each assigned to a dedicated head of attention, followed by the concatenation of the outputs. (b) In our method, the process can be viewed as patchification, where the two-dimensional input is divided into smaller patches that fit into the heads of convolutional attention layers. The outputs are then merged back into their corresponding locations.

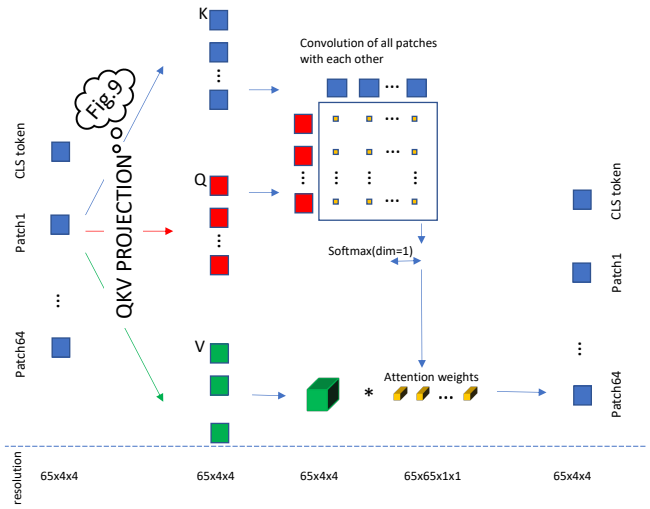


Fig. 7. Multi-head self-attention layer using only convolutional layers. The patches of the original tokens are passed through the three shared depthwise convolutional layers, which are reshaped into the original format. The resulting Query and Key matrices then convolved with each other to produce the attention score matrix, which is then softmaxed and used as the weights of the convolutional layer for the values tensor. Parallelism for the QKV projection is shown in Fig. 9, while mixed tiling is used to the remaining components. The parallelisation strategy can be adjusted based on the device's and the input resolution.

$H * W$). This is then reshaped into the $H \times W$ with the channels number of (number of tokens$*$mlp_ratio), increasing the number of tokens by a factor of MLP ratio.

Similarly, the output resolution of the second shared depthwise convolutional layer is 1×1 and number of channels (number of tokens $* H * W$), which can be reshaped to the input's original shape $H \times W \times$ number of tokens.

4) Theoretical parallelism in the 4f system: The motivation for integrating convolution operations within attention layers is

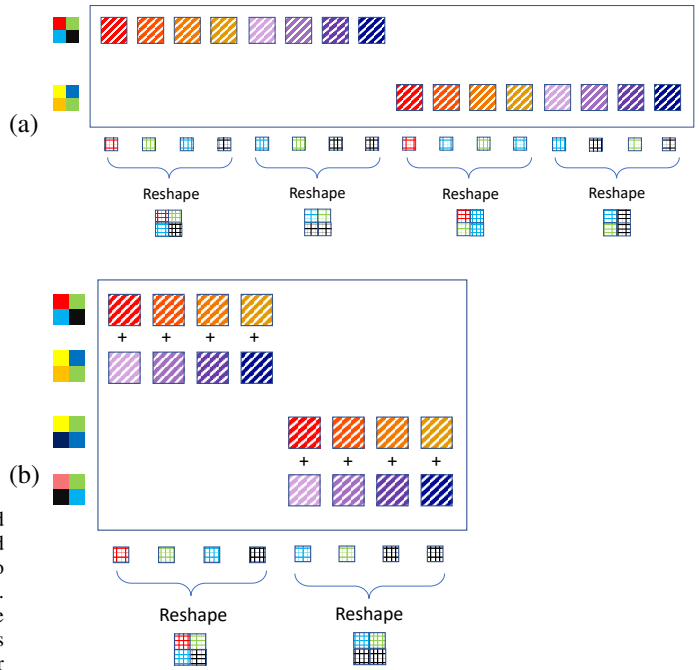


Fig. 8. Shared depthwise convolutional layer with the valid convolution and reshape of the output for full emulation of the Linear layer using convolution. (a) Shared depthwise convolutional layer from one matrix into two matrices. In this case, two matrices have been mapped into 4, where each has been mapped into corresponding two outputs. The technique can be used from few to many mapping. (b) Shared depthwise convolutional layer from two matrices into one. In this case, four matrices have been mapped into two, each group of two into one corresponding output matrix. The technique can be used from many to fewer matrices.

not only to explore the possibility of integrating convolutions into the MHSA but also to exploit the 4f system's capability to perform these tasks more rapidly and efficiently than conventional electronic components. A key benefit of free-space optics is its ability to carry out high-resolution operations without incurring latency.

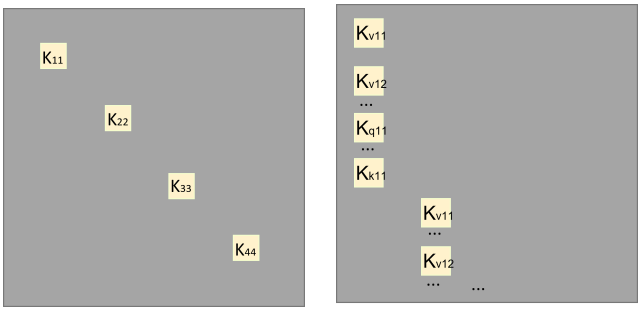
In contrast to CNNs, which rely on convolutional layers, Transformers use linear layers and are typically more efficient on GPUs, where entire tensors are loaded for optimised and rapid computation. However, in an optical setup, the standard input tiling method described earlier becomes inadequate. Parallelisation in this context can be achieved using mixed tiling following the approach of Li *et al.* [31].

While mixed tiling is generally used for standard convolution operations, here it must be adapted for depth-wise convolutional layers. This adaptation is accomplished by zeroing out all kernels except the one associated with the output channel, as shown in Fig. 9 (a).

In QKV projection, the kernels for the Q, K, and V output pixels can be tiled within a single mixed tiling block and then separated post-output, as shown in Fig. 9 (b). In most cases, the number of kernels will likely exceed the resolution capacity of the 4f system, necessitating multiple inferences.

IV. Experiments

In this work we performed 4 experiments with ViT and 12 experiments with our method, ConvShareViT. Several models



(a) (b)

Fig. 9. Mixed tiling with depthwise convolutional layers and its application in QKV projection for convolutional attention layers. (a) Basic mixed tiling of kernels, with all kernels except those corresponding to the output channel set to zero to prevent summation. (b) Illustration of how shared depthwise convolutions can be applied in the QKV projection.

intentionally test constraints like same padding or non-shared depthwise kernels to identify when convolutional layers emulate the linear projections required for attention, and when they instead act like feature extractors.

Initially, four standard ViTs were trained to establish baseline performance, shown in the first part of Table I. These included combinations of ViTs with and without trainable positional encoders, as well as models using either multi-head (12 heads) or single-head attention mechanisms. The image patch size was fixed at 4×4 from the original image resolution of 32×32, resulting in 64 patches (65 when including the classification token). Tokens were embedded into a 192-dimensional space, similar to Lee *et al.* [61]. Each model comprised 9 transformer blocks with an MLP ratio of 2. Training was conducted for 310 epochs, with the first 10 epochs allocated for warmup [62]. The Adam optimiser [63] was used, starting with a learning rate of 5×10^{-4} and using a Cosine Annealing Scheduler [62].

For the ConvShareVit models, the main twelve experiments consisting of different methods of application for MHSA, MLP, Shared depthwise convolution and valid convolutions are summarised in the second part of Table I. Each model represents a distinct combination of these methods, enabling a systematic analysis of their effects on model performance. Additional details, such as embedding dimensions, the absence of bias and the use of the trainable positional encoder in certain models, are also provided to offer a comprehensive view of the configurations tested. All models were trained using the Adam optimiser, with a learning rate of 5×10^{-4} for Models 1–5 and 8×10^{-4} for Models 6–12.

It can be seen that Models 2, 3, and 4 seem identical. The difference lies in the QKV projection, which is listed below:

- Model 2 uses a depthwise convolutional layer with same padding and a kernel size equivalent to the patch resolution (4 in this case). The number of input/output channels is the same, corresponding to the number of patches.
- Model 3 applies four consecutive depthwise convolutional layers to expand the number of channels by a factor of 7, before reducing it back to the original number and repeats this process.
- Model 4 uses two consecutive depthwise convolutional layers to expand the number of channels by a factor of 14 before reducing them back to the original count.

A similar pattern is observed with Models 5 and 6, where both models use valid depthwise convolution, reshaping the 1×1 outputs to match the original resolution. However, Model 5 includes an additional depthwise convolutional layer, similar to that in Model 2, before the valid convolution is applied

With the ConvShareViTs, the objective was to maintain as much consistency as possible with the original ViTs trained in this study, preserving key aspects such as the MLP ratio, the number of layers, and the patch size of 4×4. The primary difference lay in the embedding dimension, as the ConvShare-ViTs required maintaining a square shape for compatibility, as shown in Fig. 6. Most models, except for Models 10 and 12, had embedding dimensions of 16×16, resulting in a total of 256 dimensions—higher than the original ViTs' embedding dimensions. This increase was essential to enable the extraction of 16 attention heads, each corresponding to a 4×4 token. In contrast, Models 10 and 12 utilised an embedding dimension of 13×13, which yields 169 dimensions, lower than in our implementations of ViTs. As these models were designed with a single attention head, the tokens did not require further division into patches (Fig. 6(b)), when passed into the convolutional attention layers.

Throughout this study, the CIFAR-100 dataset was employed for all experiments. Data augmentation techniques were applied uniformly across all models, including PyTorch's built-in "CIFAR10" auto-augmentation, random cropping with a padding of 3, random horizontal flipping, and standardisation using the dataset's mean and standard deviation. Train and validation accuracy was measured by the number of correct class predictions over the total number of predictions. Attention scores were assessed in the following way. The objects of interest (apples, rocket) were delineated to create a hand-drawn mask. A ratio between the scores inside the mask and outside the mask was calculated for each attention map of each model. For the case of the rocket, it was noticed that attention may focus on the rocket itself, or the rocket and the blast, so both were considered.

V. Results and Discussion

The test accuracies of four variations of the regular ViT are shown in the last column of Table I. It is clear that the models ViT 3 and ViT 4 using the fixed sinusoidal positional encoder outperform those with trainable position encodings, ViT 1 and ViT 2. This observation aligns with previous findings in Transformer research [64], suggesting that fixed sinusoidal positional encodings provide more positional information. Furthermore, when the positional encoder is trainable, the single-head model slightly surpasses the twelve-headed model, although the difference is minimal. In contrast, with sinusoidal positional encoding, both models perform nearly identically. Although the test accuracy for both is 64%, the training curves in Fig. 10 illustrate that the twelve-headed ViT with sinusoidal positional encoding demonstrates higher validation performance. On the other hand, the twelve-headed ViT with trainable positional encoding, despite achieving better

Model	Trainable Pos Encoding	Number of Heads	MLP used	Shared dw Convolution	Valid Convolution	Bias	Embed Dimension	Accuracy (%)	Number of Parameters
ViT 1	✓	12	✓	-	-		192	61	2,702,500
ViT 2	✓	1	✓	-	-		192	63	2,702,500
ViT 3		12	✓	-	-		192	64	2,702,500
ViT 4		1	✓	-	-		192	64	2,702,500
Model 1	✓	1	✓				16×16	49	1,445,481
Model 2	✓	16	✓			✓	16×16	42	1,460,530
Model 3	✓	16	✓			✓	16×16	52	1,460,530
Model 4	✓	16	✓			✓	16×16	52	11,130,868
Model 5	✓	16	✓		✓	✓	16×16	48	11,552,068
Model 6	✓	16	✓		✓	✓	16×16	53	123,188,068
Model 7	✓	16	✓		✓		16×16	54	9,639,972
Model 8	✓	16		✓			16×16	25	103,167
Model 9	✓	16		✓	✓		16×16	58	3,004,452
Model 10	✓	1		✓	✓		13×13	62	2,763,015
Model 11	✓	1		✓	✓		16×16	63	5,992,740
Model 12		1		✓	✓		13×13	59	2,752,030

TABLE I
COMPARISON OF METHODS APPLIED TO DIFFERENT MODELS DURING CONVSHAREVIT DEVELOPMENT WITH OUR IMPLEMENTATION OF REGULAR VITS AND THEIR PERFORMANCE IN TERMS OF ACCURACY AND NUMBER OF PARAMETERS AS AN INDICATION OF COMPLEXITY. THIS TABLE OUTLINES THE PRIMARY EXPERIMENTS CONDUCTED AND THE METHODS APPLIED TO EACH MODEL. EACH ROW REPRESENTS A DISTINCT MODEL AND INDICATES THE PRESENCE OF SPECIFIC METHODS WITH A CHECKMARK. THE DIFFERENCES AMONG MODELS 2, 3, AND 4 LIE IN THE CONVOLUTIONAL LAYERS USED IN THE QKV PROJECTION (SEE DETAILS IN SECTION IV).

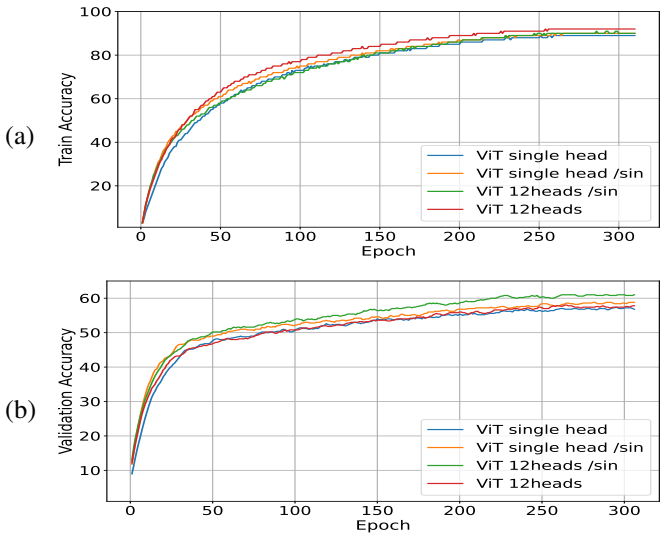


Fig. 10. Comparison of training and validation curves for four ViT models, each using different configurations of positional encoders (trainable vs. sinusoidal / ViT 1, 2 vs ViT 3, 4) and attention heads (single head vs. twelve heads/ ViT 2, 4 vs ViT 1, 3).(a) Training accuracy per epoch. (b) Validation accuracy per epoch.

training performance, shows poor validation accuracy, which is reflected in its test accuracy of 61%. This suggests that the twelve-headed model might have been over-parameterised for a relatively simple dataset like CIFAR-100, especially when combined with a trainable positional encoder, potentially leading to overfitting.

These results can be directly compared with those from prior studies. However, it is important to note that our results are not directly comparable to the results of the original ViT achieved by Dosovitskiy *et al.* [11], nor to state-of-the-art benchmarks for CIFAR-100 classification, as our architecture is novel and does not benefit from pre-trained ViT weights. Dosovitskiy *et al.* themselves emphasised that ViTs require pre-training on large-scale datasets such as ImageNet to achieve strong performance on smaller datasets like CIFAR-100. In contrast, our model is trained entirely from scratch on CIFAR-100, and is thus best compared with studies that similarly train transformer-based models from scratch on this dataset. The selection of the twelve-headed, nine-layer ViT with 4×4 patches was inspired by the work of Lee *et al.* [61]. In their study, they also trained ViTs from scratch on CIFAR-100, which is somewhat unusual, as ViTs are typically pre-trained on larger datasets before fine-tuning. Lee *et al.*achieved a test accuracy of 60.01% without augmentation and 73.81% using several augmentation techniques, including CutMix, Mixup, and AutoAugment. Additionally, they employed methods such as label smoothing, stochastic depth, and random erasing.

In contrast, our approach only used AutoAugment, yet it outperformed Lee's model without augmentation. Although our results are lower than their fully-augmented model, our method still proves effective. The exclusion of CutMix and Mixup was a deliberate choice to maintain training efficiency, as multiple experiments were required to validate our approach. Another study by Zhu *et al.* [65] trained a smaller ViT on CIFAR-100 with six layers and eight heads, achieving a test accuracy of 54.31%.

Our novel method, ConvShareViT, also yielded specific performance characteristics. Notably, Models 10, 11, and 12, which all employed a single attention head, outperformed the multi-headed models, as shown in Table I. Model 8, the only model to use the same padding in the QKV projection, achieved a poor test accuracy of 25%. This suggests that using valid convolution and reshaping the outputs (effectively mimicking an MLP) is crucial for performance. Interestingly, models up to model 4 also used the same padding but retained the original output shapes, as in Fig. 5 (b). These models had more trainable parameters due to not sharing weights across input channels, which may explain their relatively high performance. Despite their strong results, these models did not

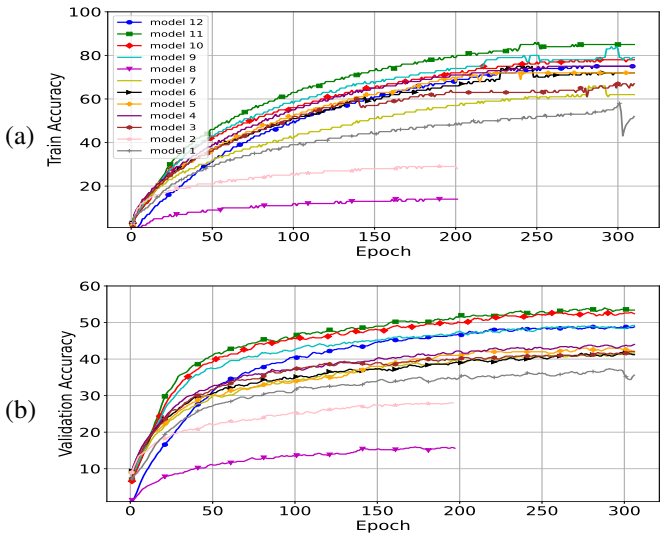


Fig. 11. Training curves comparison for the Training set and Validation set of CIFAR-100 with different ConvShareVit models (a) Training curve for train set of CIFAR-100, with the best model being model 11. (b) The training curve for the validation set of CIFAR-100, with the best model being model 11. Models with validation accuracy lower than 30% after epoch 200 were terminated early due to poor performance.

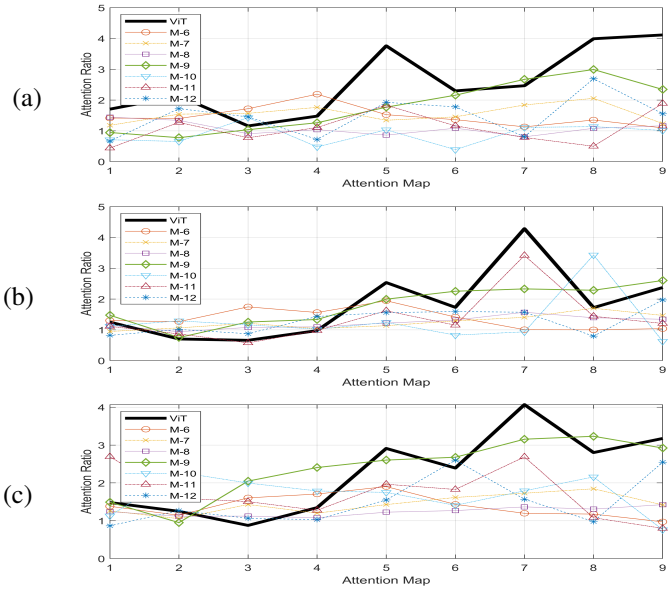


Fig. 12. Quantitive evaluation of the attention scores inside and outside a hand-drawn mask delineating the object(s) of interest. (a) Apples. (b) Rocket. (c) Rocket and blast. In general attention ratios increase with higher attention maps where ViTs perform better than most models. It should be noticed that the actual region of interest (rocket v. rocket and blast) can provide very different results.

employ traditional attention mechanisms, as revealed by later visualisation analyses.

Fig. 11 shows the training process of all models which employed the convolution operation within the MHSA layers. All models were trained for 310 epochs, with 10 epochs reserved for warmup, except for model 2, which was halted early due to poor performance and no improvement in the loss. Model 11 led both in training and validation curves, consistent with its test accuracy in Table I. While model 9 appeared second in training accuracy, model 10 outperformed it in validation accuracy. This is also evident from the test accuracies in Table I, where models 9 and 10 scored 58% and 62%, respectively, suggesting that model 9 may suffer from slight overfitting. It is also worth noting that model 10, which uses a single head and a smaller embedding dimension of 13, performed better than the multi-headed model 9. This reinforces the hypothesis that single-head attention may suffice for classifying CIFAR-100, where the dataset complexity does not require multiple attention subspaces.

To assess the robustness of our models, we evaluated the accuracy of all ConvShareViT variants alongside baseline ViTs under additive Gaussian noise. The results in Figure 13, demonstrate an immediate drop in accuracy when even a small amount of noise is introduced in all models, which may be attributable to the low resolution of the input images. ConvShareViT models, particularly Models 9-12, maintain stronger robustness than the other models, although slightly lower compared to the standard ViTs under noisy conditions. This suggests that if ConvShareViT was deployed in environments with unpredictable optical distortions or noise, additional robustness measures measures may be required.

Figure 12 presents a quantitative assessment of the attention. Figures 14 and 15 provide visual comparisons of average attention scores per layer for both the standard ViT and

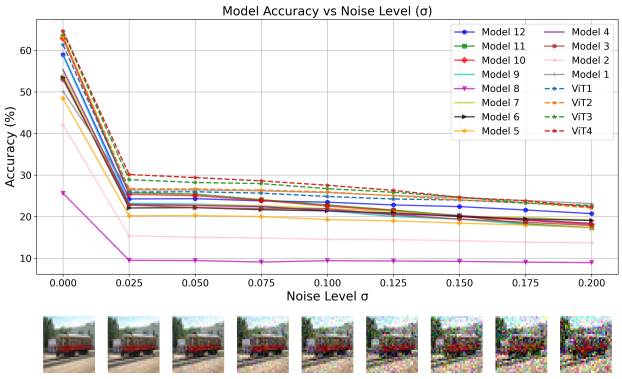


Fig. 13. Quantitative evaluation of the model robustness. The input image was added increasing levels of Gaussian noise (σ). The models' accuracy dropped considerably with the minimum level of noise and then decreased slightly as noise increased. Whilst Model 11 achieved the highest overall accuracy, Model 12 demonstrated the strongest robustness to additive Gaussian noise.

the ConvShareViT models for selected images of "apples" and a "rocket" from CIFAR-100. Notably, only models from Model 8 onwards implement shared depthwise convolution. However, Model 8 does not apply valid convolution with output reshaping, making the visualisation of attention scores only meaningful from Model 9 onwards.

Model 9 provides a consistent good performance, especially from the fifth layer upwards and the attention scores start to concentrate primarily on the objects in the image. For the apples, it is only outperformed by ViT, but for the rocket it provides equivalent results for both the rocket alone and the rocket and the blast. It is interesting to note that despite Model 10 performing well on training, some of its attention scores did not focus on the object with the last layer focusing on the background. This occurred only for the

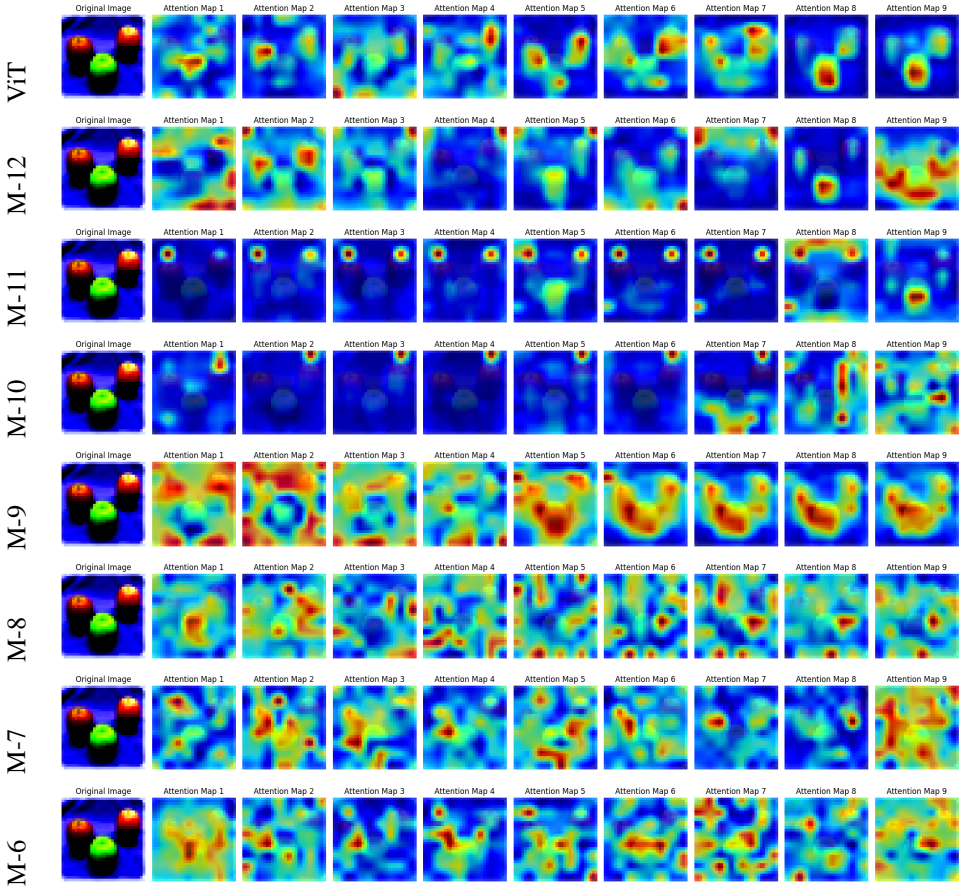


Fig. 14. Visualisation of the average attention scores projected onto the original input image of Apples from the test set of CIFAR-100. This figure compares the performance of the last seven models with the regular ViT (Vision Transformer with 12 heads). The vertical axis corresponds to the models and the horizontal to the attention layers. The ViT model achieved good attention scores in the final layers using a standard attention mechanism. Models 9, 11, and 12 also achieved attention scores similar to the original ViT. In contrast, Model 10's attention scores look incorrect as it is focusing on the background instead, as evidenced by other visualisations. Model 8 did not converge, while Models 7 and 6 did not employ the Shared DW convolutional methods without emulating the linear layer, causing the models to not learn the attention scores in the same manner as the ViT.

rocket class and only in the final layer; previous layer correctly attended to the object. Residual connections likely mitigated the impact, allowing accurate classification despite this. The top-performing model, model 11, shows good learning, with noticeable attention scores throughout the layers. Overall, Model 11 is most suitable when maximising classification accuracy is the primary objective, whereas Model 12 offers a trade-off when lower computational cost and memory usage are prioritised. But Model 11 also demonstrates bias in its learnable positional encoding across all layers, which produces high attention scores near the corners.

To mitigate the issue of biased positional encoding, Model 12 was developed with fixed sinusoidal positional encoding. This change led to more balanced attention scores, though there was a slight drop in test accuracy to 59%. This performance decline is likely attributable to Model 12's reduced size, as it uses an embedding dimension of 13×13.

These results suggest that while the models can achieve high accuracy, the use of same padding in shared depthwise convolution yields suboptimal results (Model 8). This behaviour is expected from the architectural constraints described in previous sections: same padding produces spatial feature maps rather than a single output per token, weakening the intended equivalence to a linear projection. Models that used same padding but without weight sharing across the input channels performed better than Model 8, likely because they operate as regular feature extractors as in CNNs. This suggests that the suboptimal performance of Model 8 arises from the fact that, while behaving like a standard CNN due to the use of same padding, the weight sharing reduced number of trainable parameters, therefore led to underfitting. The visualisations indicate that models only effectively learn attention when shared depthwise convolution is combined with valid padding and output reshaping—essentially when convolution functions similarly to a standard linear layer.

This finding implies that the use of convolution in self-attention layers is only beneficial when it can replicate linear layers. In cases where this is not achieved, the convolution behaves more like traditional convolutional neural networks, introducing additional complexity without improving model performance. Although models 7 and earlier converged, it is difficult to categorise them as transformers. Nonetheless, the results confirm that convolution can be used to replicate linear layers by employing shared depthwise convolutional layers, and can be effectively integrated into the 4f system. Based on these observations, the explored design space can

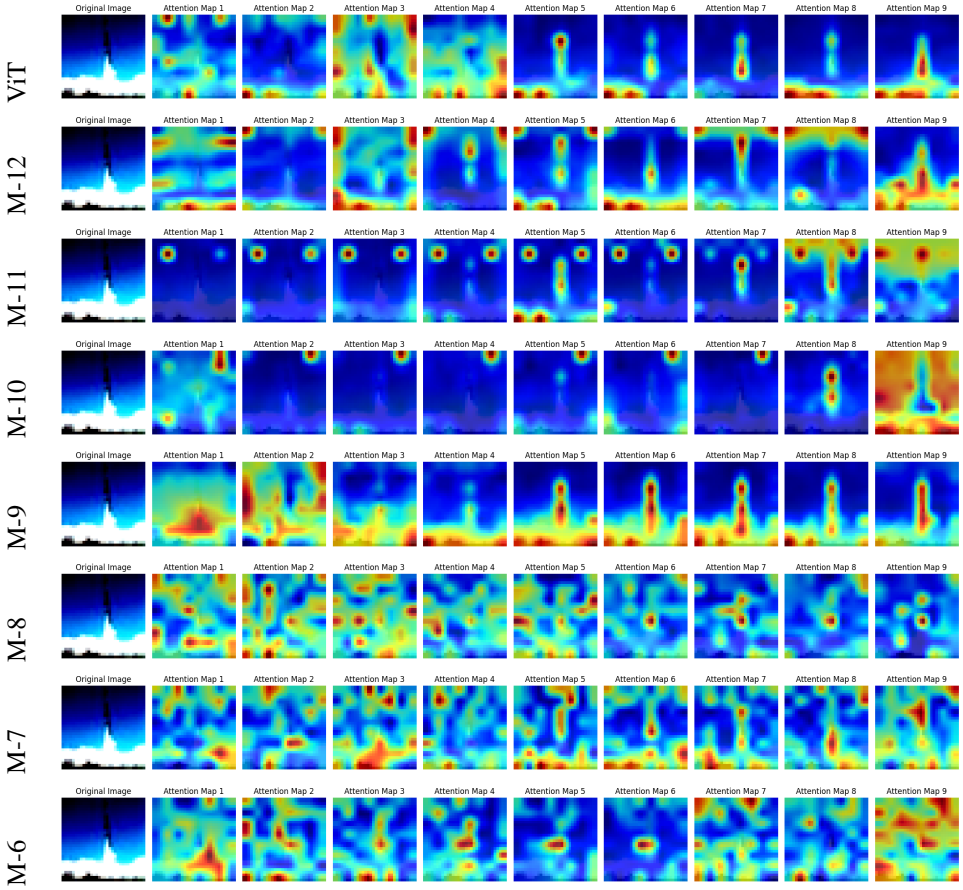


Fig. 15. Visualisation of the average attention scores projected onto the original input image of a rocket from the test set of CIFAR-100. This figure compares the performance of the last seven models with the regular ViT (Vision Transformer with 12 heads). The vertical axis corresponds to the models and the horizontal to the attention layers. The ViT model achieved good attention scores in the final layers using a standard attention mechanism. Models 9, 11, and 12 also achieved attention scores similar to the original ViT. In contrast, Model 10's attention scores focused on the background instead, which still managed to achieve a good performance. Model 8 did not converge, while Models 7 and 6 did not employ the Shared DW convolutional methods without emulating the linear layer, causing the models to not learn the attention scores in the same manner as the ViT.

be interpreted in three categories: (i) architectures that closely emulate transformer linear projections through shared depthwise convolution with valid padding (Models 9-12), (ii) hybrid models that combine token-wise processing with local feature extraction (Models 5-7), and (iii) configurations that behave primarily as convolutional feature extractors despite using attention terminology (Models 1-4 and 8).

For efficient parallel inference, as discussed earlier, the most effective setup in a 4f system network is a single-head model with a 13×13 embedding dimension, similar to models 10 and 12. This setup can be parallelised using mixed tiling. The number of possible input or output channels in mixed tiling depends on the kernel size and input size. The maximum number of input channels or output channels for the convolutions is given by $n = \left\lfloor \frac{R}{M+N-1} \right\rfloor$ (5), where M, N, R are the input, kernel and device resolutions respectively.

In shared depthwise convolution with valid padding, the input size equals the kernel size, which in this case is 13. With an estimated 4K resolution (2160 pixels per dimension), the number of input and output channels is 86. This results in 65 input channels and 65×13^2 output channels. For the three QKV (Query, Key, Value) projections, 384 inferences are needed. To compute the attention scores, 50 inferences are required, using a 4f system with mixed tiling at 4K resolution. Finally, the weighted sum attention score is computed in one inference using mixed tiled convolution.

For the convolutional MLP blocks, kernel tiling is more efficient. In this case, the required number of input channel inferences is 65 and 65×2, leading to 195 inferences per block. Across nine layers, this totals 5,670 inferences for the 4f system. On a 2 MHz device [31], [35], this process would take **2.8 ms**, compared to measured **8.5 ms** on a T4 GPU, based on 700 inference runs, with the GPU warmed up for 10 iterations beforehand.

Vision Transformers are known to benefit from large-scale pretraining, and ConvShareViT is expected to follow a similar trend on larger datasets such as ImageNet. Since ConvShareViT preserves the patch-based structure of ViTs, pretrained weights could in principle be transferred, provided the convolutional layers emulate the linear projections. From an optical perspective, scaling to larger token sizes or more channels is limited by the resolution of the 4f system, which determines the level of parallelism.

The present work has a number of limitations. First, the work is based on simulations that do not consider a number of physical systems parameters such as lens misalignment

[66], ambient or electronic noise [67], environmental variations [68], weather influence [69], degradation of signal quality [70], and more factors like light source inconsistency. These could be incorporated to the experiments but we consider these to be outside the scope of the present work in which the objective is to introduce a novel architecture that adapted ViTs to optical systems. Second, we restricted the work to the CIFAR-100 dataset. Future work should consider other datasets with different characteristics; image dimensions, class imbalance, background complexity, etc. Despite these limitations, we and others consider that the future of free space optics will grow in areas like computer vision [71] and LLMs [72], [73]. In addition, since ConvShareViT relies on convolution to emulate linear projections, deviations in optical conditions (e.g., noise or drift) may gradually alter the learned attention behaviour while still performing well. In such cases, the model may no longer preserve transformer-like token interactions. As a simple safeguard for deployment, monitoring attention statistics or prediction confidence during inference could help detect such degradation. Furthermore, performance variability across multiple training runs was not evaluated due to the computational cost of architectural configurations and simulations; future work should include reporting mean and standard deviation to further assess the statistical significance of the observed differences.

VI. Conclusion

In this work we have presented a ConvShareViT architecuture, which is a Vision Transformer adapted for the 4f free-space optical accelerators to enhance neural network speed by taking advantage of the high-resolution capabilities of free-space optics. ViTs were trained from scratch on the CIFAR-100 dataset, and their performance was compared with twelve models incorporating convolutional operations into the attention mechanism. The only models that effectively learned the attention mechanism were those employing the newly developed shared depthwise convolutional layers with valid padding, reshaped to mimic linear layers, as evidenced by the average attention score visualisations.

The study successfully demonstrated the viability of training ViT models using only convolutional layers within the 4f system for acceleration. The changes in performance and increase in the inference speed were investigated with various tiling methods, including mixed and kernel tiling. These techniques broaden the potential of 4f free-space optical acceleration to transformer-based models like ViTs, moving beyond traditional CNN architectures. ConvShareViT is suitable for optical computing applications such as edge-based sensing [74] and industrial inspection [75], where rapid inference and low power consumption are critical, as well as accelerator modules in data centres, where optical processing can complement electronic computation to increase throughput [76], [77]. As a standardised building block, ConvShareViT enables attention-based models to be deployed within existing optical convolution pipelines, demonstrating that 4f convolution-based systems can support a wider class of neural networks without requiring entirely new photonic architectures.

Future work could explore approaches similar to FatNet [9], [36] to further optimise ConvShareViTs for minimal inference use by taking advantage of the high-resolution capabilities of the 4f system for additional performance improvements. Moreover, more efficient methods of running the models could be investigated, particularly to evaluate their performance on a simulator or in a real 4f free-space optics environment. Additionally, the effects of noise [78] and misalignment of the elements [32] during optical training could be explored in relation to the models used in this work. Future studies could also extend the experiments to perform ablation studies on kernel size, attention head count, transformer depth with inclusion of the stochastic depth and LayerScale [79] to further investigate the generalisability of ConvShareViT. In addition, reinforcement learning could be explored to adapt optical parameters such as masks, tiling strategies, or attention configurations in response to physical noise and system drift. In such a framework, the system state could include measurements related to optical alignment, noise levels, or recent model confidence, while actions correspond to adjusting system parameters or triggering recalibration, with a reward balancing accuracy, latency, and stability.

In conclusion, ConvShareViT demonstrates, for the first time, that full attention mechanisms in ViTs can be realised using only convolution operations compatible with existing 4f optical systems. Moreover, its core mechanism, shared depthwise convolution, which is mimicking linear layers opens the possibility for a standardised approach allowing free-space optics to support a wider range of neural networks that rely on linear projections. Furthermore, while larger ConvShareViT models successfully replicate ViT's behaviour with high accuracy, smaller variants, despite lack in attention learning, remain viable for low-power edge cases for preferred efficiency.

Dedication

This work is dedicated to the memory of Prof. Thomas M. Chen, whose guidance and mentorship played a pivotal role in shaping this research. His contributions to cybersecurity and AI were matched by his generosity as a teacher and colleague. He will be greatly missed by all who knew him.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in NIPS*, volume 25. Curran Associates, Inc., 2012.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE CVPR*, pages 770–778, June 2016. ISSN: 1063-6919.

[3] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE TNNLS*, 33(12):6999–7019, December 2022.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE CVPR*, pages 248–255, June 2009. ISSN: 1063-6919.

[5] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. YOLOv10: Real-Time End-to-End Object Detection, May 2024. arXiv:2405.14458.

[6] Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE TNNLS*, 30(11):3212–3232, November 2019.

[7] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on PAMI*, 39(12):2481–2495, December 2017.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, May 2015. arXiv:1505.04597 [cs].

[9] Riad Ibadulla, Constantino C. Reyes-Aldasoro, and Thomas M. Chen. Fat-U-Net: non-contracting U-Net for free-space optical neural networks. In *AI and Optical Data Sciences V*, volume 12903, pages 42–52. SPIE, March 2024.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE PAMI*, 37(9):1904–1916, September 2015.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021. arXiv:2010.11929 [cs].

[12] Maria Kaselimi, Athanasios Voulodimos, Ioannis Daskalopoulos, Nikolaos Doulamis, and Anastasios Doulamis. A vision transformer model for convolution-free multilabel classification of satellite imagery in deforestation monitoring. *IEEE TNNLS*, 34(7):3299–3307, July 2023.

[13] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He. A survey of visual transformers. *IEEE Trans Neural Netw Learn Syst*, 35(6):7478–7498, June 2024.

[14] Nannan Li, Yaran Chen, Weifan Li, Zixiang Ding, Dongbin Zhao, and Shuai Nie. Bvit: Broad attention-based vision transformer. *IEEE Trans Neural Netw Learn Syst*, 35(9):12772–12783, September 2024.

[15] Licheng Jiao, Dan Wang, Yidong Bai, Puhua Chen, and Fang Liu. Deep learning in visual tracking: A review. *IEEE Trans Neural Netw Learn Syst*, 34(9):5497–5516, September 2023.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NIPS*, volume 30. Curran Associates, Inc., 2017.

[17] Andrea Galassi, Marco Lippi, and Paolo Torroni. Attention in natural language processing. *IEEE TNNLS*, 32(10):4291–4308, October 2021.

[18] Omar M. Saad, Wei Chen, Fangxue Zhang, Liuqing Yang, Xu Zhou, and Yangkang Chen. Self-attention fully convolutional densenets for automatic salt segmentation. *IEEE Trans Neural Netw Learn Syst*, 34(7):3415–3428, July 2023.

[19] Arshi Parvaiz, Muhammad Anwaar Khalid, Rukhsana Zafar, Huma Ameer, Muhammad Ali, and Muhammad Moazam Fraz. Vision transformers in medical computer vision—a contemplative retrospection. *Eng. Applications of Artificial Intelligence*, 122:106126, June 2023.

[20] Lulu Gai, Mengmeng Xing, Wei Chen, Yi Zhang, and Xu Qiao. Comparing CNN-based and transformer-based models for identifying lung cancer: which is more effective? *Multimedia Tools and Applications*, 83(20):59253–59269, June 2024.

[21] Azucena Ascencio-Cabral and Constantino Carlos Reyes-Aldasoro. Comparison of Convolutional Neural Networks and Transformers for the Classification of Images of COVID-19, Pneumonia and Healthy Individuals as Observed with Computed Tomography. *MDPI: Journal of Imaging*, 8(9):237, September 2022.

[22] Gary Marcus. Deep learning: A critical appraisal, 2018.

[23] Alexandre Bailly, Corentin Blanc, Élie Francis, Thierry Guillotin, Fadi Jamal, Béchara Wakim, and Pascal Roy. Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. *Computer Methods and Programs in Biomedicine*, 213:106504, January 2022.

[24] Cefa Karabağ, Mauricio A. Ortega-Ruíz, and Constantino Carlos Reyes-Aldasoro. Impact of training data, ground truth and shape variability in the deep learning-based semantic segmentation of hela cells observed with electron microscopy. *J Imaging*, 9(3):59, March 2023.

[25] Yu-Shiang Lin, Pei-Hsin Huang, and Yung-Yaw Chen. Deep learning-based hepatocellular carcinoma histopathology image classification: Accuracy v. training dataset size. *IEEE Access*, 9:33144–33157, 2021.

[26] G. Armeniakos, G. Zervakis, D. Soudris, and J. Henkel. Hardware approximate techniques for deep neural network accelerators: A survey. *ACM Comp Surveys*, 55(4):1–36, November 2022.

[27] Biagio Peccerillo, Mirco Mannino, Andrea Mondelli, and Sandro Bartolini. A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives. *J Systems Architecture*, 129:102561, August 2022.

[28] Xiubao Sui, Qiuhao Wu, Jia Liu, Qian Chen, and Guohua Gu. A review of optical neural networks. *IEEE Access*, 8:70773–70783, 2020.

[29] Nikolay L. Kazanskiy, Muhammad A. Butt, and Svetlana N. Khonina. Optical computing: Status and perspectives. *Nanomaterials*, 12(13):2171, June 2022.

[30] C. S. Weaver and J. W. Goodman. A Technique for Optically Convolving Two Functions. *Applied Optics*, 5(7):1248–1249, July 1966. Publisher: Optica.

[31] Shurui Li, M. Miscuglio, V. Sorger, and Puneet Gupta. Channel Tiling for Improved Performance and Accuracy of Optical Neural Network Accelerators. *ArXiv*, 2020.

[32] Julie Chang, Vincent Sitzmann, Xiong Dun, Wolfgang Heidrich, and Gordon Wetzstein. Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification. *Scientific Reports*, 8(1):12324, August 2018. Number: 1 Publisher: Nature.

[33] Mario Miscuglio, Zibo Hu, Shurui Li, Jonathan K. George, Roberto Capanna, Hamed Dalir, Philippe M. Bardet, Puneet Gupta, and Volker J. Sorger. Massively parallel amplitude-only Fourier neural network. *Optica*, 7(12):1812–1819, December 2020.

[34] Shane Colburn, Yi Chu, Eli Shilzerman, and Arka Majumdar. Optical frontend for a convolutional neural network. *Applied Optics*, 58(12):3179–3186, April 2019. Publisher: Optica.

[35] Puneet Gupta and Shurui Li. 4F optical neural network acceleration: an architecture perspective. In *AI and Optical Data Sciences III*, volume 12019, pages 77–84. SPIE, March 2022.

[36] Riad Ibadulla, Thomas M. Chen, and Constantino Carlos Reyes-Aldasoro. FatNet: High-Resolution Kernels for Classification Using Fully Convolutional Optical Neural Networks. *AI*, 4(2):361–374, June 2023. Number: 2 MDPI.

[37] Jun Dai, Xiaowen Dong, Chong Li, and Jian-Jun He. On-chip 4F-system based on concave mirrors for optical neural networks. In *Holography, Diffractive Optics, and Applications XIII*, volume 12768, pages 292–297. SPIE, November 2023.

[38] Eloy Schultz, Joris de Nijs, Bin Shi, and Ripalta Stabile. Optical 4F Correlator for Acceleration of Convolutional Neural Networks. In *25th Annual Symp. of IEEE Photonics Benelux*, Belgium, November 2021.

[39] Jie Chen, Huarong Gu, Hongwei Zhang, Jie Zhong, and Yi Xiong. Multilayer optoelectronic hybrid convolutional neural network with an optical 4f-system recurrent structure. In *Holography, Diffractive Optics, and Appl. XIII*, volume 12768, pages 120–129. SPIE, November 2023.

[40] Isiaka A. Alimi and Paulo P. Monteiro. Revolutionizing free-space optics: A survey of enabling technologies, challenges, trends, and prospects of beyond 5g free-space optical (fso) communication systems. *Sensors*, 24(24):8036, December 2024.

[41] Yasha Yi. *Industry Trends and Future Directions*, page 149–160. Springer Nature Switzerland, June 2025.

[42] Balamurugan Palanisamy, Vikas Hassija, Arpita Chatterjee, Arpita Mandal, Debanshi Chakraborty, Amit Pandey, G. S. S. Chalapathi, and Dhruv Kumar. Transformers for vision: A survey on innovative methods for computer vision. *IEEE Access*, 13:95496–95523, 2025.

[43] Arun Kumar Sharma and Nishchal K. Verma. A novel vision transformer with selective residual in multihead self-attention for pattern recognition. *Pattern Recognition*, 172:112497, April 2026.

[44] Hongchen Tan, Xiuping Liu, Baocai Yin, and Xin Li. MHSA-Net: Multihead self-attention network for occluded person re-identification. *IEEE T Neural Netw Learn Syst*, 34(11):8210–8224, November 2023.

[45] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. CvT: Introducing Convolutions to Vision Transformers. pages 22–31, 2021.

[46] Priyanka Patel and Amit Thakkar. The upsurge of deep learning for computer vision applications. *IJECE*, 10(1):538–548, February 2020.

[47] Lei Jimmy Ba and Rich Caruana. Do Deep Nets Really Need to be Deep? In *Advances in NIPS*, volume 27, 2014. _eprint: 1312.6184.

[48] Song Han, Huizi Mao, and William J. Dally. Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. *arXiv: CVPR*, 2016.

[49] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 525–542, Cham, 2016. Springer International Publishing.

[50] Fu-Wei Chen and Yi-Yu Liu. Performance-driven dual-rail routing architecture for structured asic design style. *IEEE Trans Computer-Aided Design of Integrated Circuits and Systems*, 29(12):2046–2050, December 2010.

[51] M. Mitchell Waldrop. The chips are down for Moore's law. *Nature News*, 530(7589):144, February 2016.

[52] Tyler W. Hughes, Momchil Minkov, Yu Shi, and Shanhui Fan. Training of photonic neural networks through in situ backpropagation and gradient measurement. *Optica*, 5(7):864–871, July 2018.

[53] Yichen Shen, Nicholas C. Harris, Scott Skirlo, Mihika Prabhu, Tom Baehr-Jones, Michael Hochberg, Xin Sun, Shijie Zhao, Hugo Larochelle, Dirk Englund, and Marin Soljačić. Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11(7):441–446, July 2017. Number: 7 Publisher: Nature Publishing Group.

[54] Xiubao Sui, Qiuhao Wu, Jia Liu, Qian Chen, and Guohua Gu. A Review of Optical Neural Networks. *IEEE Access*, 8:70773–70783, 2020.

[55] Maxwell Anderson, Shi-Yuan Ma, Tianyu Wang, Logan Wright, and Peter McMahon. Optical Transformers. *Transactions on Machine Learning Research*, October 2023.

[56] Baopeng Li, Okan K. Ersoy, Caiwen Ma, Zhibin Pan, Wansha Wen, and Zongxi Song. A 4F optical diffuser system with spatial light modulators for image data augmentation. *Optics Comms.*, 488:126859, 2021.

[57] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, Zhaohui Yang, Yiman Zhang, and Dacheng Tao. A survey on vision transformer. *IEEE Trans Pattern Anal Mach Intell*, 45(1):87–110, January 2023.

[58] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. In *IEEE/CVF (ICCV)*, pages 548–558, October 2021. ISSN: 2380-7504.

[59] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling Up Your Kernels to 31×31: Revisiting Large Kernel Design in CNNs. In *2022 IEEE/CVF CVPR*, pages 11953–11965, June 2022. ISSN: 2575-7075.

[60] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. pages 10012–10022, 2021.

[61] Seunghoon Lee, Seunghyun Lee, and Byung Cheol Song. Improving Vision Transformers to Learn Small-Size Dataset From Scratch. *IEEE Access*, 10:123212–123224, 2022.

[62] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations*. arXiv, May 2017. arXiv:1608.03983 [cs, math].

[63] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

[64] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional Positional Encodings for Vision Transformers. September 2022.

[65] Haoran Zhu, Boyuan Chen, and Carter Yang. Understanding Why ViT Trains Badly on Small Datasets: An Intuitive Perspective, February 2023. arXiv:2302.03751 [cs].

[66] Tian Siheng, Huang Yongmei, Xu Yangjie, Nan Xinyuan, Wu Qiongyan, Xiang Chunsheng, and Tang Wei. Study of off-axis telescope misalignment correction method using out-of-focus spot. *Opto-Electronic Engineering*, 50(7):230040–1–230040–13, 2023.

[67] Mohammad Ali Khalighi and Murat Uysal. Survey on free space optical communication: A communication theory perspective. *IEEE Communications Surveys & Tutorials*, 16(4):2231–2258, 2014.

[68] Maha Achour. Simulating atmospheric free-space optical propagation: Ii. haze, fog, and low clouds attenuations. In Eric J. Korevaar, editor, *Optical Wireless Communications V*, volume 4873, page 1–12. SPIE, 2002. Backup Publisher: International Society for Optics and Photonics.

[69] G. Susanna, S. Di Bartolo, D. Carleo, S. Penna, S. Betti, and A. Reale. Weather influence on performance of a seamless free space optic (fso) link in a pon scenario. In *2019 21st ICTON*, page 1–5. IEEE, July 2019.

[70] T. Ismail, E. Leitgeb, and T. Plank. Free space optic and mmwave communications: Technologies, challenges and applications. *IEICE Transactions on Communications*, E99.B(6):1243–1254, 2016.

[71] Minho Choi and Arka Majumdar. Free-space optical encoder for computer vision. *npj Nanophotonics*, 2(1), September 2025.

[72] Renjie Li, Qi Xin, Wenjie Wei, Sixuan Mao, Erik Ma, Zijian Chen, Jingxing Gao, Malu Zhang, Haizhou Li, and Zhaoyu Zhang. What is next for LLMs? pushing the boundaries of next-gen AI computing hardware with photonic chips. *Nanophotonics*, 14(22):3499–3525, October 2025.

[73] Neil Na, Chih-Hao Cheng, Shou-Chen Hsu, Che-Fu Liang, Chung-Chih Lin, Nathaniel Y. Na, Andrew I. Shieh, Erik Chen, Haisheng Rong, and Richard A. Soref. Implementation of transformer-based LLMs with large-scale optoelectronic neurons on a CMOS image sensor platform. *arXiv:2511.04136*, 2025.

[74] Shiji Zhang, Xueyi Jiang, Bo Wu, Haojun Zhou, Wenguang Xu, Hailong Zhou, Zhichao Ruan, Jianji Dong, and Xinliang Zhang. Photonic edge intelligence chip for multi-modal sensing, inference and learning. *Nat. Commun.*, 16(1):10136, November 2025.

[75] Satya Pratap Singh, Than Singh Saini, and Umesh Kumar Tiwari. Real-time monitoring and quality control using photonic technologies in industrial environments. In *Advances in Optics and Optoelectronics*, pages 95–111. Springer Nature Singapore, Singapore, 2025.

[76] Hong Zhou, Hemin Zhang, Ruirong Zhang, Xichen Yuan, and Honglong Chang. AI-driven photonic noses: from conventional sensors to cloud-to-edge intelligent microsystems. *Microsyst. Nanoeng.*, 11(1):209, November 2025.

[77] Shupeng Ning, Hanqing Zhu, Chenghao Feng, Jiaqi Gu, Zhixing Jiang, Zhoufeng Ying, Jason Midkiff, Sourabh Jain, May H Hlaing, David Z Pan, and Ray T Chen. Photonic-electronic integrated circuits for high-performance computing and AI accelerators. *J. Lightwave Technol.*, 42(22):7834–7859, November 2024.

[78] M. Eren Akbiyik. Data Augmentation in Training CNNs: Injecting Noise to Images, July 2023. arXiv:2307.06855 [cs].

[79] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going Deeper With Image Transformers. pages 32–42, 2021.

Riad Ibadulla received his BEng in Computer Systems Engineering from City, University of London, and his MSc in Artificial Intelligence from the University of St Andrews. After completing his PhD titled High-Resolution Capabilities of Free-space Optical Neural Networks, which explored novel methods to optimise CNNs and ViTs for optical AI accelerators, he worked as a research assistant on the development of robust IDS by combining machine learning with formal methods. He is currently a lecturer in computer science at City St George's, University of London, and his primary areas of expertise are deep learning and computer vision.

Thomas M. Chen (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1983 and 1984, respectively, and the Ph.D. degree in electrical engineering from the University of California, Berkeley, CA, USA, in 1990., Recently he was a Professor in Cyber Security at the School of Engineering and Mathematical Sciences, City University London, London, U.K. Previously, he was a Professor in Networks at Swansea University, Swansea, U.K., and an Associate Professor at Southern Methodist University, Dallas, TX, USA.,Prof. Chen received the IEEE Communications Society Fred Ellersick Best Paper Award in 1996. He has been Editor-in-Chief for IEEE Communications Surveys (1996–1997), IEEE Communications Magazine (2006–2007), and IEEE Network (2009–2011). He served as an Associate Editor for the Journal on Security and Communication Networks and the International Journal of Digital Crime and Forensics.

Constantino Carlos Reyes-Aldasoro (Senior Member, IEEE) received the B.Sc. in electrical engineering from Universidad Nacional Autónoma de México (1993), the M.Sc. in communications and signal processing from the Imperial College of Science Technology and Medicine (1994), and the Ph.D. in computer science from Warwick University (2005). He was a Postdoctoral Research Associate with the Tumor Microcirculation Group, Medical School, University of Sheffield (2005-2011). He was with Sussex University (2011-2013), and joined the City, University of London, in 2013. His research interests include biomedical image analysis with special interest in cancer, inflammation, and microcirculation. He has worked with x-rays, computed tomography, magnetic resonance imaging, light, fluorescent, confocal, multiphoton, and electron microscopy. He was Chair of the Vision and Imaging Technical Network of the Institute of Engineering and Technology and member for Executive Committees of the British Association for Cancer Research and the Royal Microscopical Society. He was Chair of the 2014 Medical Image Understanding and Analysis Conference and the 2022 British Machine Vision Conference.