Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-023-08012-3

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Compact Image Transformer Based on Convolutional Variational Autoencoder with Augmented Attention Backbone for Target Recognition in Infrared Images

Billel Nebili[1] · Atmane Khellal[1] · Abdelkrim Nemra[1] · Said Yacine Boulahia[2] · Laurent Mascarilla[3]

Received: 4 September 2022 / Accepted: 30 May 2023

Abstract
Recently, Vision Transformer (ViT) has become a relevant alternative to convolutional neural networks (CNN) for image classification tasks. However, we believe that ViT needs pre-training on large-size datasets, making it unsuitable for certain scientific fields such as infrared imaging where the amount of training data is limited. In this direction, we proposed a Compact image Transformer based on convolutional variational Autoencoder with Augmented attention backbone (referred to AA-CiT) for target recognition in infrared images, which can learn efficiently from scratch even with small-size datasets. This is performed by three main adaptations of the original ViT architecture, in which we introduced convolutions in its different parts to fully benefit from the properties of both paradigms: attention and convolution. First, we proposed an improvement in the tokenization step by introducing a new module based on a local convolutional variational autoencoder. Second, convolutional features are incorporated in ViT's encoder, which allows us to introduce some inductive bias of CNN in the proposed transformer. We finally took profit of a new sequence pooling technique on the top of ViT's encoder to make our model compact and more accurate. These modifications allow us to overcome the difficulties of ViT training and also eliminate the need for Class token and the heavy reliance on positional embeddings. We validated our approach by carrying out extensive experiments on FLIR-SEEK dataset. Globally, we achieved a 3% improvement in overall classification accuracy compared to conventional ViT while relying on fewer parameters (14% of ViT's parameters).

Keywords Augmented attention · Autoencoder · Infrared images · Target recognition · Sequence pooling · Vision Transformer

1 Introduction

Transformer is a deep learning model, firstly presented in Attention is All You Need [1]. It uses only the attention mechanism and neither recurrent nor convolutional network. Due to its success in natural language processing [2–4], many attempts have been made to explore its potential in computer vision tasks. The first work that applied transformer in computer vision is Vision Transformer (ViT) [5], in which the standard transformer has been applied directly to images with the necessary modifications. The authors of ViT have concluded that the transformer does not work well without being trained on large datasets.

The "data-hungry transformer" paradigm has dismissed the transformer in many problems, either caused by lack of data or limited computational resources [6, 7]. Until today, it is nearly impossible to create super large datasets like JFT-300M [8] and ImageNet [9] in fields such as medicine and infrared imaging. For example, it is difficult to col-

✉ Billel Nebili
billelnebili@gmail.com ; BILLEL.NEBILI@emp.mdn.dz

Atmane Khellal
khe.atmane@gmail.com

Abdelkrim Nemra
karim_nemra@yahoo.fr

Said Yacine Boulahia
boulahia.yacinesaid@gmail.com

Laurent Mascarilla
lmascari@univ-lr.fr

1 UER SAI, Ecole Militaire Polytechnique, Bordj El Bahri, 16111 Algiers, Algeria

2 UER RSI, Ecole Militaire Polytechnique, Bordj El Bahri, 16111 Algiers, Algeria

3 Laboratoire MIA, Univ. La Rochelle, 17042 La Rochelle Cedex, France

Published online: 16 June 2023

lect millions of patient data for a rare disease [10]. In infrared imaging research (our work), datasets with millions of images do not exist, and creating them is difficult [11]. RGB cameras are widely used in people's daily lives and are embedded in the majority of electronic devices such as cell phones and computers. Moreover, the explosion of social media has made it possible to exploit and collect millions of images stored on these platforms. On the other hand, a small community needs to use infrared cameras due to their specific applications (military [12], industrial [13], medical [14], etc.) and their price, which are expensive when compared to RGB cameras. Hence, the amount of infrared data flowing through the net is not sufficient to create large datasets similar to the visible spectrum. Several works [15, 16] have referred to transfer learning; however, it has been shown that in some cases, pre-trained models do not perform well on other datasets [17]. Therefore, it is crucial to improve the architecture of ViT to reduce its reliance on data.

Before the advent of ViT, convolutional neural networks (CNN) were the default choice for computer vision tasks [18, 19]. CNN architectures impose two inductive biases [20]: locality and translation equivariance. The emergence of attention mechanisms [1] has shown that CNN are unable to capture long interactions, preventing them from capturing the global context of an image [21]. Researchers have addressed the limitations of CNN in two ways. First, they have designed fully attentional architectures without convolution [5, 22–24]. The second idea is to design hybrid architectures, containing the attention mechanism and CNN [25–30]. The review of these works shows that fully attentional architectures require convolutions. CNN impair the ability to capture long-range dependencies, but they enable the network to capture local information with a local receptive field.

Therefore, we are faced with two problems of ViT that prevent us from applying it for target recognition in infrared images; the need for data for its training and the lack of some characteristics offered by CNN. In this direction, we have proposed a hybrid architecture (Fig. 1), in which we have introduced convolution in ViT. The proposed architecture is a **C**ompact **i**mage **T**ransformer based on convolutional variational **A**utoencoder with **A**ugmented attention backbone (**AA-CiT**). Three main adaptations are introduced in ViT to overcome the above problems. First, we developed a module of tokenization based on a local convolutional variational autoencoder [32] instead of the hard patching and linear embedding layers in ViT. This module introduces some inductive biases imposed by CNN. Second, feature maps inside the transformer backbone have been augmented with convolutional feature maps to improve the feature richness and overcome the difficulties in training ViT. Finally, a new technique of sequence pooling (SeqPool) [31] has been

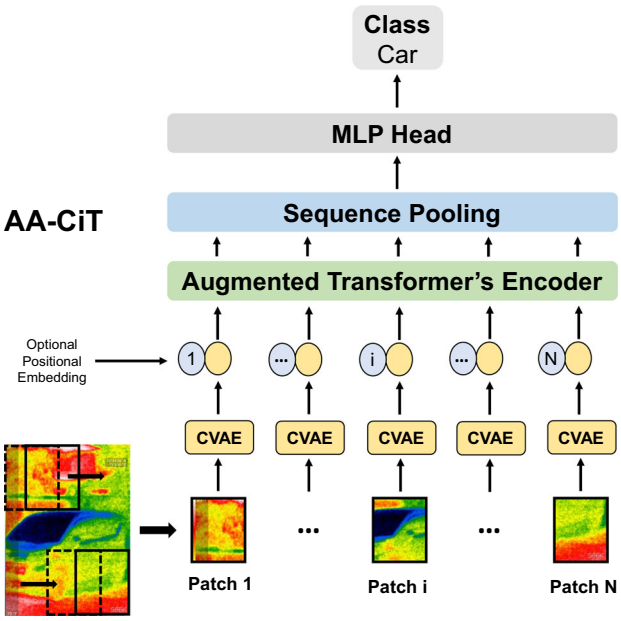

Fig. 1 Overview of AA-CiT: CVAE is the encoder part of a convolutional variational autoencoder. Figure adapted from Fig. 1 in [31]

inserted at the top of the transformer's encoder, to make our model compact and more efficient.

These three additions make it possible to train our model from scratch with very small-size datasets (FLIR-SEEK dataset [33] $\sim 5K$ images). Moreover, they resulted in a significant increase in performance, with an improvement of 3% in overall classification accuracy compared to ViT on FLIR-SEEK dataset.

The rest of this paper is organized as follows. In Sect. 2, we will describe the different parts of ViT and the recent works that have improved its architecture, especially architectures designed for small-size datasets. We then present the main contributions of our work compared with existing models. The components of our compact transformer (AA-CiT) are further discussed in Sect. 3, followed by Sect. 4, which presents the experimental details and the obtained results. Several variants of our model were tested on FLIR-SEEK dataset and then compared to state-of-the-art classification methods, including CNN-based models and ViT-based models. In Sect. 5, we provide the main findings of this work, examine its challenges and offer insights on future prospects.

2 Related Works

The main issue with ViT is the need to pre-train on large datasets like JFT-300M, which is not accessible by the large community. Our work is motivated by many recent works which have been realized to improve ViT and dispel the data-hungry paradigm for transformers. In this section, we

will present the main works addressing the above problem. First, we provide a brief presentation of ViT and its different parts. Next, we discuss two transformer-based architectures, designed especially for small datasets. Then, the contribution of other works in the architecture of ViT is described. Finally, we conclude by analyzing related works and highlighting our contributions.

2.1 Vision Transformer

Vision Transformer [5] is a new type of neural architecture applied to target recognition tasks in images. It uses the attention mechanism to encode the input data as powerful features. It allows for establishing long connections between different parts of the input image. ViT is composed of several parts (Fig. 2). We will briefly describe each part and its role.

Image tokenization This part consists of two steps. (1) Image patching: ViT divides the input image into sequences of non-overlapping patches. Let $X \in R^{H\times W\times D}$ be the input image and (P, P) the patch size. Thus, $N = H \times W/P^2$ will be the number of resulting patches. Patches are then flattened to form a 2D sequence, $X_p \in R^{N\times(P^2.D)}$. In addition, there are other alternatives to form patches from CNN feature maps.

(2) Learnable embeddings: The embedding layer helps to grab a learned vector representation for each patch, in which the flattened patches are linearly projected into a lower-dimensional space. The resulting vectors are called "tokens."

Classification token Class token is originally introduced by BERT [2]. It is an extra learnable parameter that is attached to the sequence of patch embeddings. The resulting vector (patch embeddings + Class token) serves as input to the first transformer layer. Class token gathers information from all patches using self-attention. It is processed in the same way as the patch tokens. Typically, only the state of Class token after transformer's encoder is used as input to the classification layer.

Positional embedding Unlike CNN, the attention mechanism has no idea about the patches' position in the input sequence. Therefore, a positional embeddings vector is added to the patch embeddings vector before it is processed by the transformer's encoder. This vector can be learned instead of using hard-coded vectors.

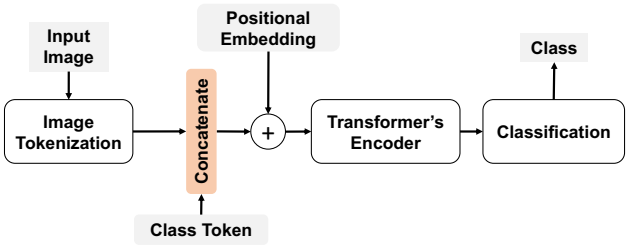

Fig. 2 The pipeline of Vision Transformer architecture

Transformer's encoder The transformer's encoder is structured as a multilayer stack of identical layers, where each layer consists of two sub-layers: a self-attention layer and a feed-forward neural network. The role of self-attention is to maintain the interdependence of patches in the sequence representation. In addition, each of these two sub-layers is surrounded by a residual connection and followed by a normalization layer.

Classification This part is simply composed of a multilayer perceptron (MLP) head, which takes the Class token vector after transformer's encoder as input and yields classification scores. Generally, it is implemented with a small feed-forward neural network.

2.2 Transformers for Small Datasets

In this subsection, we focus on ViT-based architectures that are specifically designed to learn from scratch, even with small datasets.

Compact Convolutional Transformer (CCT) [31] framework has provided state-of-the-art results on small datasets. CCT authors tried to answer the question: "Can vision transformers be trained from scratch on small datasets?". To achieve this, they proposed an architecture in which they have eliminated the need for Class token, learnable and positional embeddings by using convolution and a new sequence pooling (SeqPool) technique. Convolution is introduced in the tokenization step, using a small stride, to obtain efficient tokenization and retain local spatial relationships. This convolution block is simply composed of a succession of sub-blocks: 2D convolution, ReLU activation and max pooling. The SeqPool strategy eliminates the need for Class token and allows for efficient processing of the resulting encoder information.

SL-ViT [34] can be trained from scratch with small-size datasets. They proposed two add-on modules: Shifted Patch Tokenization (SPT) and Locality Self-Attention (LSA), which can be easily applied to different architectures based on ViT. SPT was founded to rectify the tokenization problem in ViT, by using spatial relations between adjacent pixels and thus bringing a locality inductive bias. LSA has addressed the problem of "poor attention mechanism" by applying two new blocks, diagonal masking and learnable temperature scaling.

2.3 Other Concurrent Works

In addition to the works mentioned earlier, other studies have been conducted to enhance ViT. Convolutional Vision Transformer (ConViT) [20] has tried to gain the advantages of both paradigms; CNN and transformer, by introducing a new method to achieve a soft inductive bias in ViT. Its

idea is to replace self-attention layers with gated positional self-attention layers. The latter can perform as a convolutional layer by adjusting a gating parameter. Similar to ConViT, Convolutional vision Transformer (CvT) [35] has tried to derive some desirable properties from convolutions, such as shift and scale. It introduced convolutions at two levels. Firstly, the tokenization process is performed by a convolutional layer. Secondly, the standard linear projection is replaced by a convolutional projection to achieve more modulation of the local context in the attention mechanism. Touvron et al. [36] introduced Data-efficient image Transformers (DeiT), which is an image transformer that can be trained without a super large dataset. DeiT contains a new distillation procedure based on distillation token. This latter is used similarly to Class token. It interacts with Class token and patch tokens in the transformer via the self-attention layers. Yuan et al. [37] created Convolution-enhanced image Transformer (CeiT) which achieved competitive results against DeiT in ImageNet. They brought three modifications to the original ViT. Firstly, an image-to-tokens module is used for tokenization, where patches are extracted from low-level feature maps. Secondly, a locally enhanced feed-forward layer is introduced in each encoder instead of a feed-forward layer. Finally, a layer-wise class token attention is attached to the top of the transformer, which takes as input Class tokens from different layers, not only from the last layer. Tokens-to-token ViT (T2T-ViT) [38] has shown two main limitations of ViT: (1) Patch tokenization is performed in a hard manner, preventing the capturing of local images structures such as lines and edges, and (2) the redundancy in the attention backbone results in limited and weak features. To overcome these limitations, they proposed progressive tokenization instead of the simple tokenization of ViT, in which several tokens are aggregated into one token, iteratively. Additionally, they have adopted a deep-narrow structure [39] to enrich features in the attention backbone. Class-attention in image Transformers (CaiT) [40] proposed adding a new layer (named LayerScale) after each residual block allowing to train deeper transformers. CaiT's architecture consists of two stages. (1) Self-attention layers like ViT, but without Class token. The idea is to let the attention focus on the relations between patches and not try to summarize the useful information for the linear classifier. (2) Class attention layers, where Class token is inserted. This part is entirely dedicated to summarizing the information to be provided to the linear classifier. Wang *et al.* [41] have presented Pyramid Vision Transformer (PVT), which is a multistage transformer without convolutions, designed similarly to multi-scales in CNN. Transformer-iN-Transformer (TNT) [42] used two blocks, an outer transformer to model links between patches and an inner transformer to process relations among sub-patches.

2.4 Analysis and Contributions

All works mentioned above (except SL-ViT and CCT) have used ImageNet or larger datasets to train from scratch. SL-ViT and CCT are trained with CIFAR ($\sim 60K$ images). The main specificity of our AA-CiT is its ability to learn efficiently from scratch, even on very small-size datasets such as FLIR-SEEK dataset ($\sim 5K$ images), which represents a real challenge.

By reviewing the different architectures based on ViT, we can summarize the problems of ViT in three points. First, ViT tokenization is based on non-overlapping patches, i.e., ViT tokenizes using only a few pixels, producing tokens with a small receptive field. As a result, the relationships between adjacent pixels are not sufficiently embedded, which induces poor tokenization lacking of locality inductive bias imposed by CNN. We can consider this problem as the main cause of the "data-hungry transformer" paradigm. Second, the attention backbone of ViT produces redundant and poor features because it cannot focus locally on important visual tokens. Third, Class token is attached to the patch tokens before the first layer. It passes through transformer layers and is then used for class prediction. This design forces the self-attention mechanism to spread information between Class token and patch tokens. The classification input only comes from Class token, and the only variable inputs for the model are the patch tokens.

In this paper, we present a solution to these problems. We developed a module of tokenization based on a local convolutional variational autoencoder, applied to overlapping patches. This architecture increases the locality (the image's adjacent pixels are related to each other) and translation equivariance via weight sharing. (The autoencoder is shared between all patches.) Moreover, patches are embedded using the local convolutional variational autoencoder instead of the linear embeddings in the original ViT. Then, we explored attention augmented CNN [29] inside the attention backbone, to improve the feature richness and to focus more locally on the content of the patch tokens. In addition, we have introduced a SeqPool layer after the transformer's encoder. It allows to weight patches' information according to their quality. The SeqPool layer eliminates the need for Class token, allowing transformer attention blocks to focus on the relationships between patches instead of summarizing the relevant information in Class token embedding.

To sum up, the main contributions of this paper are:

- Proposing a hybrid architecture based on transformer (**AA-CiT**), which can be trained from scratch and achieves satisfactory results on small infrared dataset. Moreover, we demonstrated that our AA-CiT is less dependent on positional embeddings than ViT.

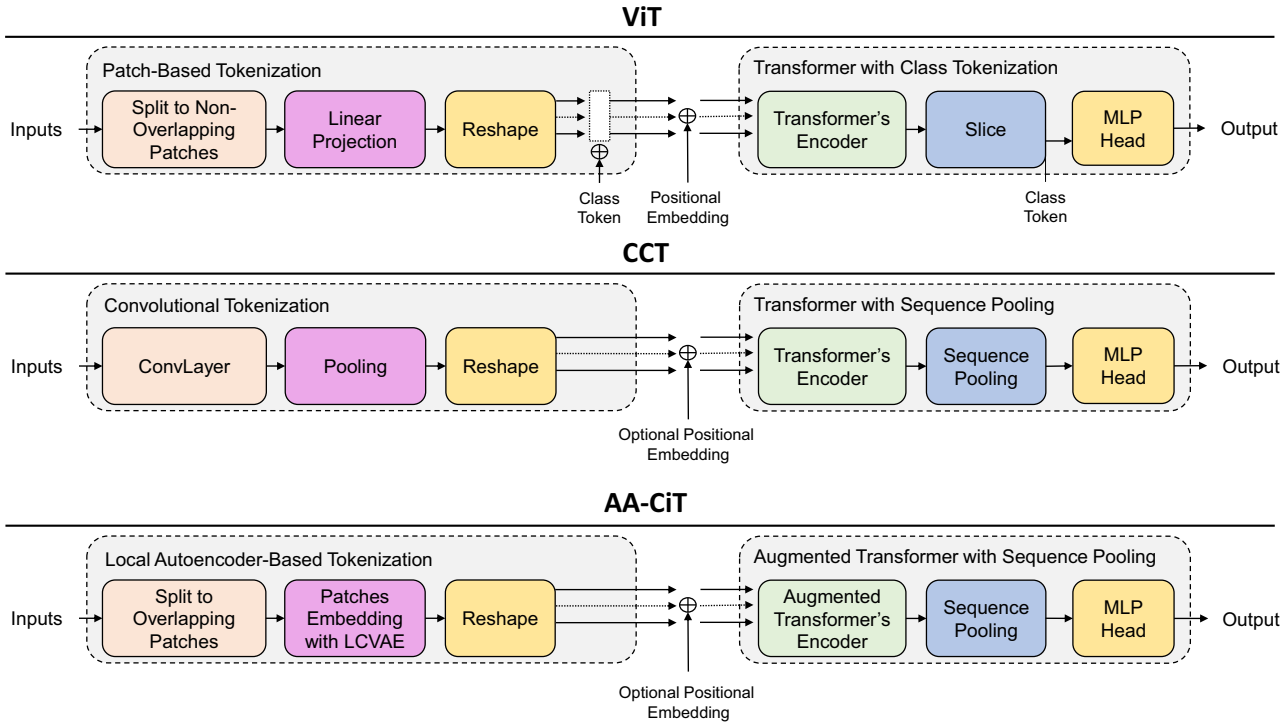

Fig. 3 Comparing ViT (top) to CCT (middle) and AA-CiT (bottom): LCVAE is the encoder part of a local convolutional variational autoencoder. Figure adapted from Fig. 2 in [31]

- Developing a tokenization step based on a local convolutional variational autoencoder instead of the hard tokenization used by ViT. This alteration improves the locality and translation equivariance inductive bias of ViT.
- Finding an efficient attention backbone for vision transformers by concatenating convolutional feature maps and attentional feature maps. The features richness is improved and redundancy is reduced.
- Introducing the SeqPool technique on the top of the transformer's encoder, which eliminates the need for Class token and makes our model more compact and accurate.

3 Proposed Method

The design of our AA-CiT is based on ViT and CCT. Figure 3 shows a detailed modular-level comparison of the three models.

ViT misses the inductive biases imposed by CNN due to its hard split, which requires more training data. In our work, we have used a local autoencoder applied to overlapping patches. This architecture preserves local spatial relationships and ensures weight sharing by using a common autoencoder between different patches. In addition, patches are well embedded using a convolutional variational autoencoder, instead of a linear projection in the original ViT. Second, we have concatenated convolutional feature maps with attentional ones to increase the richness of learned features. Lastly, our AA-CiT is supported by introducing the SeqPool technique to weight patches' information according to their quality, eliminating the need for Class token and leading to a more compact model. In the following subsections, we will further describe the components of our AA-CiT.

3.1 Local Autoencoder-Based Tokenization

To overcome the limitation of hard tokenization in ViT, we introduce some inductive bias with a tokenization block based on a local autoencoder. This block consists of two steps: (1) local patch generation and (2) patch embedding.

Local patch generation The decomposition of an image into small overlapping patches is useful and practical and allows patch tokens to have a relatively large receptive field. Let $X \in R^{H\times W\times C}$ be the input image of our transformer. X is decomposed into a sequence of patches $X_p \in R^{N\times(P\times P\times C)}$ by using a sliding window with a stride of s, where (P, P) is the resolution of each image patch, and $N = (1 + (W - P)/s) . (1 + (H - P)/s)$ is the resulting number of patches.

Patch embedding Instead of using a simple linear projection for the generated patches, we have used an unsupervised learning method to perform the patches' embedding. For each generated patch, we applied a convolutional variational

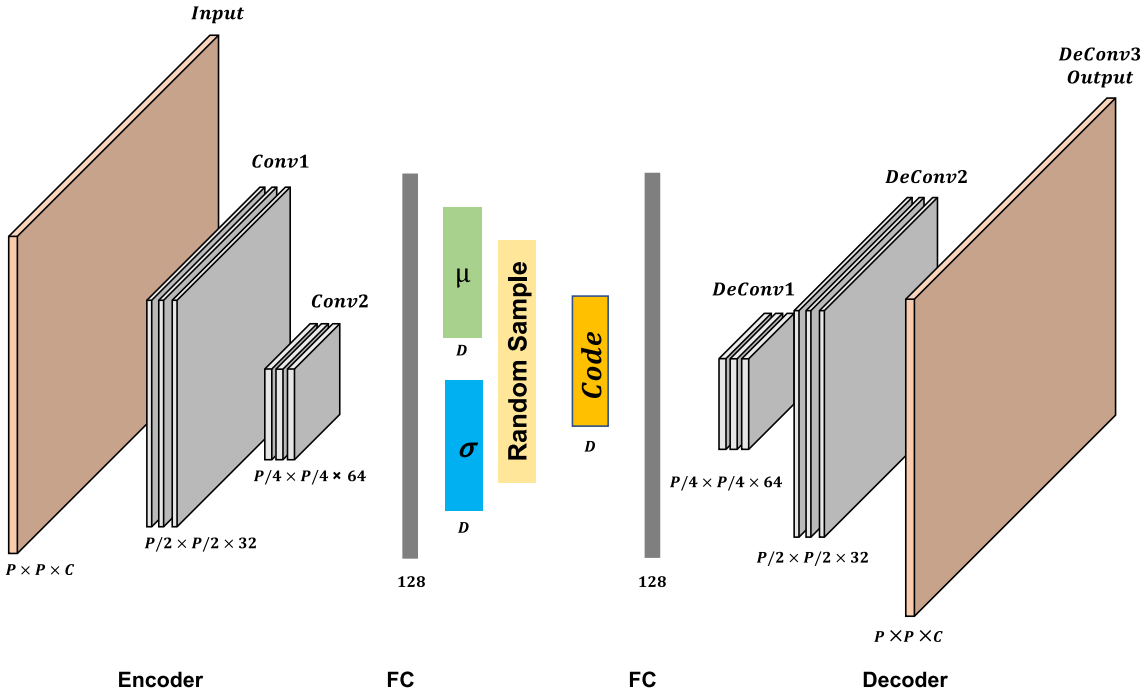

Fig. 4 The structure of the used variational autoencoder. ($P \times P \times C$) are the input and output dimensions. The latent vector *code* has a length of D. The encoder consists of two convolutions blocks with kernel size of 3 and stride 2 followed by a ReLU activation. Then, a fully connected (FC) layer with a size of 128 follows. Next, the mean μ and standard deviation σ of the hidden vector are generated by two fully connected layers. Each pair of μ and σ defines a Gaussian distribution from which the *code* is sampled. Similar to the encoder, the decoder is built by transposed Conv blocks with the same kernel and step followed by a ReLU activation, except for the output convolution block where a Sigmoid is used. The decoder learns to reconstruct the input from the code

autoencoder (the encoder part) to extract features. The architecture of the used autoencoder is shown in Fig. 4. The output code of the encoder part will be the patch embedding, and let D be its length. So for the whole input image, we will obtain a sequence of N patch tokens $X_{pe} \in R^{N \times D}$, which represents the input sequence for the transformer's encoder. The training of the autoencoder and evaluation metrics is further discussed in Appendix A.

AA-CiT's tokenization block utilizes overlapping patches to increase the number of pixels included during tokenization, resulting in a larger receptive field and enhanced locality. Additionally, the shared autoencoder employed across the entire image ensures that the model maintains translation equivariance. By leveraging an unsupervised convolutional autoencoder instead of a simple linear projection, AA-CiT's tokenization block achieves efficient encoding of patches. This method enables the model to capture complex image features and maintain local spatial relationships, ultimately enhancing its ability to accurately represent visual data.

3.2 Augmented Transformer's Encoder

Our transformer's encoder is similar to the one used in ViT, except that the self-attention layer is augmented by CNN. The integration of convolutional features into the encoder allows to focus locally on the patch tokens and to maintain more spatial information. Therefore, we have explored the architecture proposed by Bello *et al.* [29] to enrich the attentional features. We propose to augment the attention mechanism with convolutional operators by concatenating the attentional feature maps with a set of feature maps generated by CNN.

Figure 5 shows an illustration of the augmented attention backbone inside the transformer's encoder. The proposed method concatenates the feature maps from the attention mechanism with the convolutional feature maps in each self-attention layer of the transformer's encoder. Let $X_{pe} \in R^{N \times D}$ be the input feature maps of the first layer inside the proposed transformer's encoder, representing the sequence of tokens derived from the input image.

As shown in Fig. 5, X_{pe} will be processed through two paths. The first path consists of the multi-head attention mechanism block, similar to the one used in the original ViT architecture. Let $F_a \in R^{N \times D/2}$ be its output. In parallel with the first pathway, X_{pe} is further processed through a standard convolutional block with $D/2$ filters in the second pathway, letting $F_c \in R^{N \times D/2}$ be its output. Notice that a reshaping process is applied before and after the convolution block. Finally, the two feature maps F_a and F_c are concatenated, along the spa-

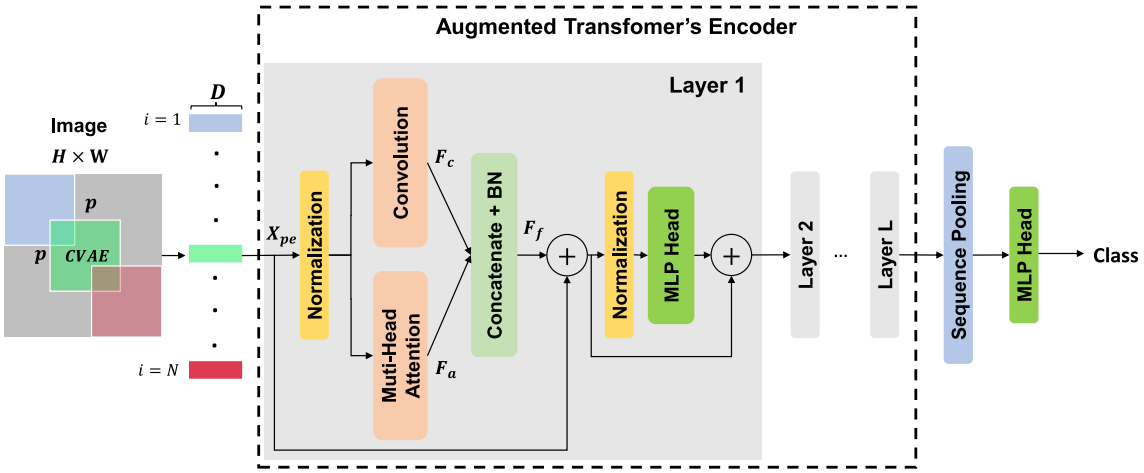

Fig. 5 Illustration of the augmented attention backbone inside the transformer's encoder. **BN:** Batch normalization

tial dimension. The resulting augmented maps are followed by batch normalization [43] to obtain the final feature maps $F_f \in R^{N\times D}$. The latter is passed through an MLP head as in the ViT baseline architecture. The same process as described for the first layer will be applied inside the remaining layers ($l = 2, ..., L$), where the input feature maps for layer l will be the output feature maps of layer $l - 1$.

The proposed augmented transformer's encoder combines the strengths of CNN and the attention mechanism. CNN are well suited for extracting low-level features from images because they are designed to preserve spatial information by using local receptive fields. In contrast, the attention mechanism can capture global relationships between the input features, which can help to identify more complex patterns and relationships between different parts of the image. By combining these two mechanisms, the resulting feature representation can capture both local and global features, as well as their relationships, which can lead to better classification accuracy.

3.3 SeqPool

As mentioned before, the use of Class token as the main source of classification information leads to a limited ViT. To eliminate Class token, we have introduced the SeqPool technique after the encoder part. This technique aims to map the sequential output of the encoder to a singular class index. We can modelize SeqPool as follows.

Let $X_L \in R^{N\times D}$ be the last layer output of the transformer's encoder. X_L is fed to a linear layer g with size $D \times 1$. Then, softmax activation is applied to $g(X_L) \in R^{N\times 1}$, so that:

$$Y_L = \text{softmax}(g(X_L))^T \in R^{1\times N} \tag{1}$$

Next, we compute the vector Z:

$$Z = Y_L X_L = \text{softmax}(g(X_L))^T \cdot X_L \in R^{1\times D} \tag{2}$$

Before feeding Z to the MLP head classifier, we pool the first dimension to obtain $Z \in R^D$.

By removing Class token and incorporating a SeqPool layer after the encoder, the efficiency of our model can be improved. By doing so, the encoder block can focus on establishing relationships between patches instead of summarizing relevant information in the Class token embedding. The entire output sequence now contains relevant information from all parts of the input image. To prevent any potential bias caused by specific patches, the sequence is pooling, which enables patches to be assigned weights based on the relevance of their informative content. Moreover, these weights are learnable, so they can be adapted to the difference in entropy between the embedded patches.

4 Experiments and Results

In this section, we conducted several experiments on FLIR-SEEK dataset to evaluate the effectiveness of the proposed AA-CiT architecture. Section 4.1 presents the used dataset and describes the implementation details. Section 4.2 shows the evaluation of several variants of our model by training them from scratch on FLIR-SEEK dataset. Section 4.3 presents a performance comparison of our AA-CiT with some common CNN and some architectures based on ViT. Finally, Section 4.4 shows the results of some ablation studies to demonstrate the effect of certain model components.

4.1 Dataset and Experimental Settings

Dataset FLIR-SEEK dataset [33] is a thermal image dataset for object classification. It consists of a total of 6892 images, of which 1092 images were captured using FLIR and 5800 images were captured using Seek Thermal. It consists of three classes: man, cat and car (Fig. 6). Table 1 gives more details of the samples' distribution on FLIR-SEEK dataset.

Implementation Details In all experiments, images were resized to 128 × 128 pixels. Unless stated otherwise, we trained all models for 500 epochs with a batch size of 64, using AdamW optimizer [44] with 1e−3 learning rate and 1e−4 weight decay. For all models, we used the following data augmentation methods: random rotation, random zoom, random horizontal flipping, width and height shift, rescaling in [0, 1], brightness shift and shear intensity.

The VIT-based architectures that contain the proposed tokenization block have been trained separately. Initially, the convolutional variational autoencoder was trained using the evidence lower bound (ELBO) as a loss function (see Appendix A). Secondly, we trained the remaining components of the VIT-based architectures in question, namely the transformer's encoder and the classification stage, using the categorical cross-entropy loss function. The encoder component of the pre-trained convolutional variational autoencoder was used as a fixed component during this training phase to extract local features from each patch.

More details will be mentioned in the following sections. This setting has been determined experimentally.

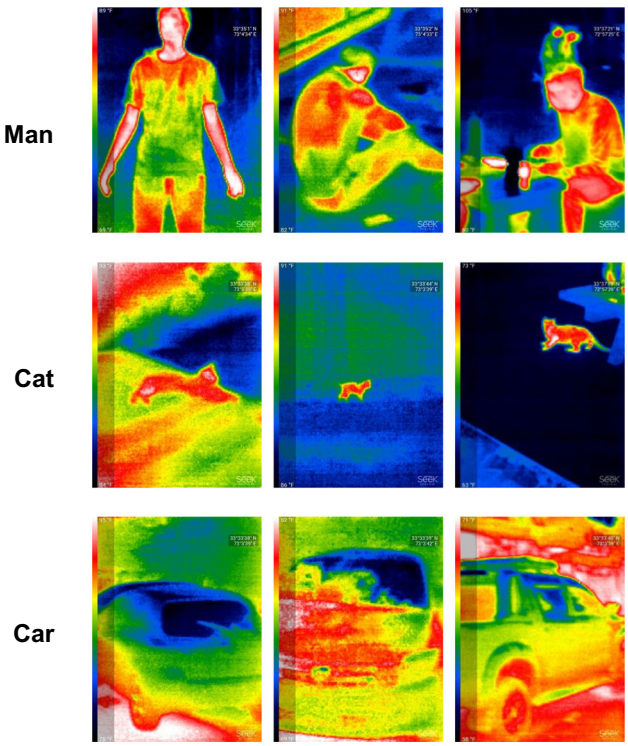

Fig. 6 Few example images of FLIR-SEEK dataset

Table 1 Number of training and test samples on FLIR-SEEK dataset

Set	Class	FLIR	SEEK	FLIR-SEEK
Training	Man	289	1782	2071
	Cat	272	1782	2054
	Car	250	1168	1418
Training Total		811	4732	5543
Test	Man	64	356	420
	Cat	70	356	426
	Car	147	356	503
Test Total		281	1068	1349

4.2 Model Variants

There are several possible design choices for our model. We instantiated four model variants by changing the patching step **s**, the number of attention heads N_h and the layers' number **L**, as summarized in Table 2. The four variants are AA-CiT-Base, AA-CiT-Medium, AA-CiT-Large and AA-CiT-Huge. The patch size for all variants is $p = 16$. The patch token resulting from the local autoencoder has a fixed length $D = 128$. We set the MLP encoder size **d** to (256,128). The classification head is an MLP with two layers of length 2048 and 1024, respectively. During training, we also used stochastic depth [45] with a value of 0.1. MLP dropout and attention dropout were both set to 0.1.

Table 3 presents the results achieved by the four model variants on FLIR-SEEK dataset. We can see that for all variants, accuracy exceeds 94%, which represents high classification performance. As expected, the largest models are the most successful, AA-CiT-Huge exceeds AA-CiT-Large

Table 2 Details of AA-CiT model variants

Model	s	N	N_h	L
AA-CiT-Base	7	289	8	12
AA-CiT-Medium	5	529	8	12
AA-CiT-Large	7	289	16	24
AA-CiT-Huge	5	529	16	24

Table 3 Accuracy comparisons between AA-CiT variants

Model	Accuracy (%)	Params (M)
AA-CiT-Base	94.74	09.7
AA-CiT-Medium	95.31	10.5
AA-CiT-Large	95.63	28.1
AA-CiT-Huge	**95.98**	29.6

Bold value signifies the best result achieved for the respective experiment

Table 4 Achieved accuracies by AA-CiT-Base on FLIR-SEEK dataset when trained with more epochs

Epochs	Accuracy (%)
500	94.74
1000	94.97
5000	**95.28**

Bold value signifies the best result achieved for the respective experiment

by 0.35% and AA-CiT-Medium outperforms AA-CiT-Base by 0.57%. In fact, as we increase the number of overlapping patches, we obtain tokens with larger receptive fields, which means a more efficient embedding of adjacent pixels' relationships.

The second parameters to be considered are L and N_h. AA-CiT-Huge and AA-CiT-Large exceed AA-CiT-Medium and AA-CiT-Base by 0.67% and 0.89%, respectively. This confirms that improving performance requires going further deeper and wider. Increasing the number of heads and layers offers the potential for our model to attend to more information and to capture multiple relationships between tokens.

Furthermore, in Table 4, we performed additional experiments to see how far AA-CiT can reach with training longer.

4.3 Performance Comparison

Our approach has been compared with state-of-the-art classification methods, including CNN-based models and ViT-based models. The first point of comparison concerns two well-known networks: ResNet [19] and MobileNet [46]. After that, we compared with ViT [5], CvT [35] and both transformers that addressed small datasets: CCT [31] and SL-ViT [34].

4.3.1 Models' Configuration

The CNN models are configured as mentioned in Sect. 4.1. To make a fair comparison, all ViT-based methods are similarly configured to AA-CiT. We set the projection dimension $D = 128$, the heads number $N_h = 8$, the number of layers $L = 12$ and the MLP encoder size $d = [256, 128]$. The MLP head for classification contains two linear layers of length 2048, 1024, respectively. ViT, CvT and SL-ViT use patches of size 8. Moreover, the tokenization step for CCT uses 3 convolutional blocks with 3×3 convolutions and 32, 64 and 128 filters, respectively. Finally, we performed slight modifications to the ViT architecture to enhance its original design, including the integration of the stochastic depth technique with a value of 0.1. In addition, the vector resulting from the concatenation of all feature vectors with Class token is used as input for the final classification stage. These modifications served as a basis for developing our final AA-CiT architecture, which is designed to enable the training of ViT from scratch, even on small datasets.

4.3.2 Classification Results

Table 5 shows accuracy comparison of different models on FLIR-SEEK dataset. All reported accuracy results are best out of 3 runs.

It was observed that pre-trained CNN had the lowest classification accuracy, which may be due to the sensitivity of these models to changes in input image size.

ViT, with the slight modifications introduced, could reach more than 92% in classification accuracy.

AA-CiT achieves much higher accuracy compared to CNN-based models, achieving a performance gain of 8.75%, 10.23%, 7.34%, 5.93% over ResNet50, ResNet101, MobileNet and MobileNetV2, respectively. These results show that our ViT-based model has effectively eliminated the gap with CNN on small-size datasets.

Our architecture can further improve performance over transformer-based models. Compared to the original ViT and SL-ViT, our AA-CiT is much smaller in the number of parameters, yet offers superior performance (+3.12%, +4.98%, respectively). CvT has smaller parameters, while it has a lower performance than AA-CiT (−3.82%). CCT and AA-CiT-Base achieve similar performances. When we keep the same parameters and only increase the number of patches, our model (AA-CiT-Medium) obtains 95.31%, which is 0.5% higher than CCT. The high performance of CCT can be explained by the fact that it contains overlapping convolutions at the tokenization step and incorporates a pooling technique, which makes it more efficient. However, our designed tokenization module is better than convolution layers as it can deeply and efficiently model and encode the local content using a local autoencoder. So, AA-CiT further reduces the performance gap of transformer-based models. Our smallest model, AA-CiT-Base, with 9M parameters outperforms most state-of-the-art classification methods.

Table 5 Performance comparison of our approach and the state-of-the-art classification methods on FLIR-SEEK dataset

Model	Accuracy (%)	Params (M)
ResNet50 [19]	86.56	23.6
ResNet101 [19]	85.08	42.7
MobileNet [46]	87.97	03.2
MobileNetV2 [47]	89.38	02.3
ViT [5]	92.19	76.4
CvT [35]	91.49	07.2
SL-ViT [34]	90.33	76.5
CCT [31]	94.81	07.2
AA-CiT-Base (ours)	94.74	09.7
AA-CiT-Medium (ours)	**95.31**	10.5

Bold value signifies the best result achieved for the respective experiment

Table 6 Ablation study results on autoencoder-based tokenization module

Model	Accuracy (%)	Params (M)
ViT	92.19	76.4
AE-ViT	93.12	85.1
CCT	94.81	07.2
AE-CCT	95.18	07.3

Table 7 Ablation study results on augmented attention backbone

Model	Accuracy (%)	Params (M)
ViT	92.19	76.4
ViT-A	93.35	76.5
CCT	94.81	07.2
CCT-A	95.23	07.3

Table 8 Ablation study results on SeqPool layer

Model	Accuracy (%)	Params (M)
ViT	92.19	76.4
ViT-SP	92.83	09.6
CCT-NSP	94.22	76.4
CCT	94.81	07.2
AA-CiT-Base-NSP	94.14	85.3
AA-CiT-Base	94.74	09.7
AA-CiT-Medium-NSP	94.42	148.9
AA-CiT-Medium	95.31	10.5

4.4 Ablation Studies

We performed various ablation experiments to identify further the effects of the proposed components of our architecture. First, we studied the impact of the proposed tokenization step. Second, we experimented with introducing convolution blocks in the transformer's encoder. Next, the effect of attaching a SeqPool layer at the top of the transformer was studied. Finally, we investigated the effect of removing the positional embeddings.

A summary table about different architectures used in this section is given in Appendix B.

4.4.1 Tokenization Based on Autoencoder

We studied the effectiveness of the proposed autoencoder-based tokenization. We implemented our proposed solution for tokenization with ViT and CCT while keeping the same configuration and parameters mentioned in Sect. 4.3.1. They are denoted as AE-ViT and AE-CCT, respectively.

From Table 6, we can find that autoencoder-based tokenization is effective. AE-ViT and AE-CCT are better than ViT and CCT by 0.93% and 0.37%, respectively. The improvement is more significant for ViT than for CCT because CCT's tokenization module is based on convolutional layers, whereas ViT uses hard tokenization.

4.4.2 Augmented Attention Backbone

To show the efficiency of the proposed attention backbone, we applied the augmented transformer's encoder to ViT and CCT. We denote ViT and CCT variants with ViT-A and CCT-A, respectively.

Table 7 shows the performance improvement when the proposed attention backbone was applied to ViT and CCT. The introduction of convolutional feature maps in the attention backbone provides an improvement of 1.16% and 0.42% for ViT and CCT, respectively, without requiring a significant increase in parameters ($\sim 100k$). These results allow us to validate that the non-augmented attention backbone has limited features. Indeed, the role of convolution is to improve the modeling of patch tokens by embedding spatial information, which has been overlooked by the attention mechanism. This is very important in the presence of little data as is the case of infrared imagery. Therefore, the resulting features include global, local and spatial information, which allows a better description of the input data, and thus, better classification of the infrared images.

4.4.3 SeqPool Layer

In this part, we aim to show that with the introduction of SeqPool, our model will be more compact and accurate. For this, three intermediate models have been tested. First, we integrated the SeqPool layer into ViT architecture, and this model is denoted as ViT-SP. Then, we removed this layer from CCT, AA-CiT-Base and AA-CiT-Medium models, and the resulting models are denoted as CCT-NSP, AA-CiT-Base-NSP and AA-CiT-Medium-NSP, respectively. The obtained results are reported in Table 8.

It can be seen that the presence of a SeqPool layer, rather than using Class token, leads to an improvement in performance for all models. Moreover, the number of parameters is significantly decreased while achieving higher classification accuracy. Therefore, the SeqPool technique yields a compact model with few parameters and high efficiency.

4.4.4 Positional Embeddings

Given that we have introduced a tokenization step based on local convolutional autoencoders, which allows us to capture local context, we examined whether positional embeddings remain necessary for AA-CiT. Furthermore, we looked at

Table 9 Ablation study results on positional embeddings

Model	Pos. Embedd.	Accuracy (%)
ViT	Learnable	92.19 (baseline)
	None	86.08 (− 6.11%)
AE-ViT	Learnable	93.12 (baseline)
	None	91.96 (− 1.16%)
CCT	Learnable	94.81 (baseline)
	None	93.23 (− 1.58%)
AE-CCT	Learnable	95.18 (baseline)
	None	93.11 (− 2.07%)
AA-CiT-Base	Learnable	94.74 (baseline)
	None	93.04 (− 1.70%)
AA-CiT-Medium	Learnable	95.31 (baseline)
	None	93.67 (− 1.64%)

ViT, AE-ViT, CCT and AE-CCT. The results are shown in Table 9.

In these experiments, we found that positional embeddings are relevant in all variants with varying degrees. Removing CCT positional embedding resulted in a 1% drop in accuracy. Also, we can see that there is a significant gap between the performance of ViT with and without positional embeddings. These results confirm that CCT is flexible in terms of removing positional embedding while ViT is indispensable to positional encoding. In the presence of autoencoder-based tokenization (AA-CiT, AE-ViT, AE-CCT), accuracy drops by 1% to 2%. In particular, AA-CiT-Base and AA-CiT-Medium rely less on positional encoding and it can be removed without greatly impacting accuracy.

5 Conclusion and Discussion

In this paper, we demonstrated that it is possible to configure ViT to enhance its performance even with small-size datasets. The idea was to incorporate CNN inside to benefit from their inductive bias and thus overcome the limitations of ViT. We proposed a new **C**ompact **i**mage **T**ransformer based on convolutional variational **A**utoencoder with **A**ugmented attention backbone (**AA-CiT**), which can be trained from scratch even with small-size datasets. Extensive experiments conducted on FLIR-SEEK dataset have shown the effectiveness of the proposed modules: Local Autoencoder-based Tokenization, Augmented attention backbone and SeqPool technique. Our model with a few parameters outperformed state-of-the-art classification methods.

This type of research is highly beneficial for target recognition in infrared images, where the amount of data is very limited. Moreover, it opens the way to developing transformer-based models for vision tasks in several scientific domains, such as medical and hyper-spectral imaging.

Challenges Although AA-CiT has improved ViT's performance in small infrared datasets, it presents some limitations and challenges that must be addressed in future work. One limitation of the proposed AA-CiT architecture is its lack of flexibility regarding patch size. Changing the patch size requires retraining the local variational autoencoder, which could be time-consuming. This could be a potential limitation in scenarios where the optimal patch size may not be known a priori or may vary depending on the dataset or the specific task at hand. It would be useful to investigate ways to make the architecture more flexible, for example, by designing a tokenization module that can adapt to different patch sizes without requiring retraining.

Another limitation is the separately trained nature of our architecture. While this training strategy has shown superior performance compared to end-to-end training in our experiments and facilitates the optimization of individual components, it may lead to suboptimal results due to the absence of end-to-end training. Furthermore, this approach may not be suitable for tasks that require crucial interaction between different parts of the model. One potential solution is to design a loss function that accounts for the architecture's nature by incorporating terms for the tokenization part (i.e., the autoencoder) and other parts (i.e., the transformer's encoder and classification part).

The development of efficient transformer models for computer vision tasks remains an open challenge, and this work represents only the initial foray into this area with ample scope for advancement. It is worth noting that there is still much work to be done in this field.

Data Availability Data are available in Mendeley repository, https://doi.org/10.17632/btmrycjpbj.1

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Appendix A: Variational Autoencoder Training

In our work, we used the convolutional variational autoencoder, as shown in Fig. 4. The variational autoencoder (VAE) [32] is a variant of the autoencoder. The difference between them is the method of achieving a representation of the latent attributes. While the autoencoder produces a single value from the encoder to describe each attribute of the latent state, the VAE describes an observation in the latent space using a probabilistic way.

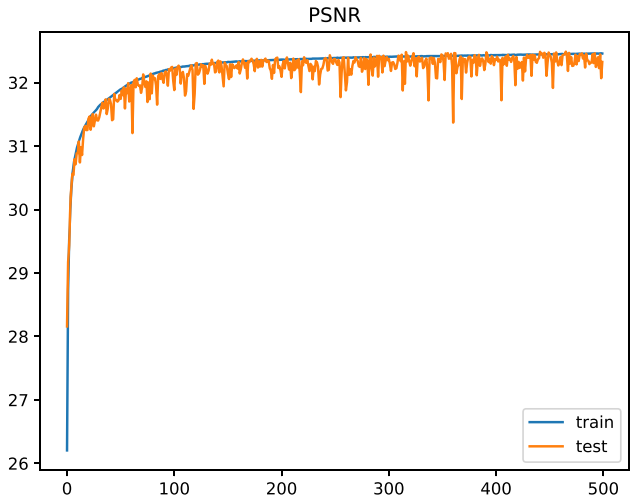

Fig. 7 Peak signal-to-noise ratio values

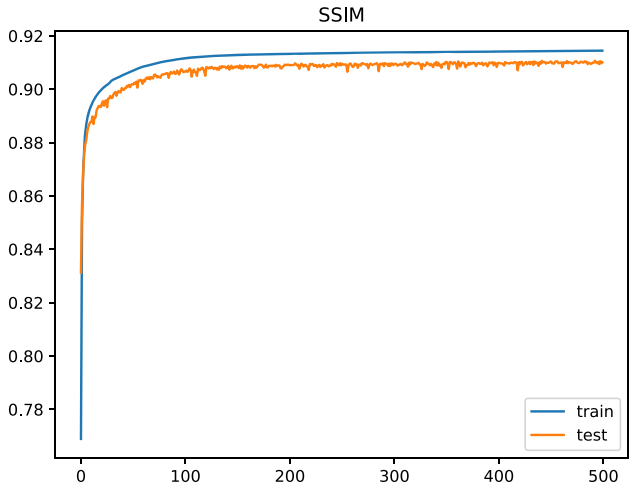

Fig. 8 Structural similarity index measure values

As is common, the learning process of a VAE is based on optimizing a loss function. VAE's loss function is the negative of the evidence lower bound (ELBO). Therefore, the optimization process of the autoencoder involves minimizing this loss function or maximizing ELBO, which is composed of two terms, namely the reconstruction loss term and the Kullback–Leibler (KL) divergence term. In our work, we have used binary cross-entropy as a reconstruction loss. The exact form of the loss function used for the training of the variational autoencoder is as follows.

Let x be the input image, and $x^{'}$ be the reconstructed image. Then, the binary cross-entropy reconstruction loss can be expressed as:

$$RL(x, x^{'}) = -\frac{1}{N}\sum x \log(x^{'}) + (1-x)\log(1-x^{'}) \quad (3)$$

where N is the total number of pixels in the image and $\sum$ is the sum over all pixels in the image.

For the KL divergence term, let $q(z\|x)$ be the learned latent distribution given the input x, and let $p(z)$ be the prior distribution (in our case, $\mathcal{N}(0, 1)$). Then, the KL divergence penalty can be expressed as:

$$D_{KL}\left(q(z\|x)\,\|\,p(z)\right) = -\frac{1}{2}\sum\left(1+\log(\sigma^2)-\mu^2-\sigma^2\right) \quad (4)$$

where $\sum$ is the sum over all elements of the latent vector z, μ is the mean of $q(z\|x)$, and σ is the standard deviation of $q(z\|x)$.

Finally, VAE's loss function $\mathcal{L}$ is given by:

$$\mathcal{L} = RL(x, x^{'}) + D_{KL}\left(q(z\|x)\,\|\,p(z)\right) \quad (5)$$

Training To train our VAE, we formed two sets: training and test, composed of patches generated from the FLIR-SEEK dataset images. As mentioned before, input images are divided into 289 overlapping patches of size 16×16. We used Adam optimizer with a 0.0005 learning rate to train the VAE before introducing it into the global architecture of AA-CiT. We trained for 500 epochs with a batch size of 128.

Evaluation metrics In addition to the loss function, two well-known image quality metrics are used: peak signal-to-noise

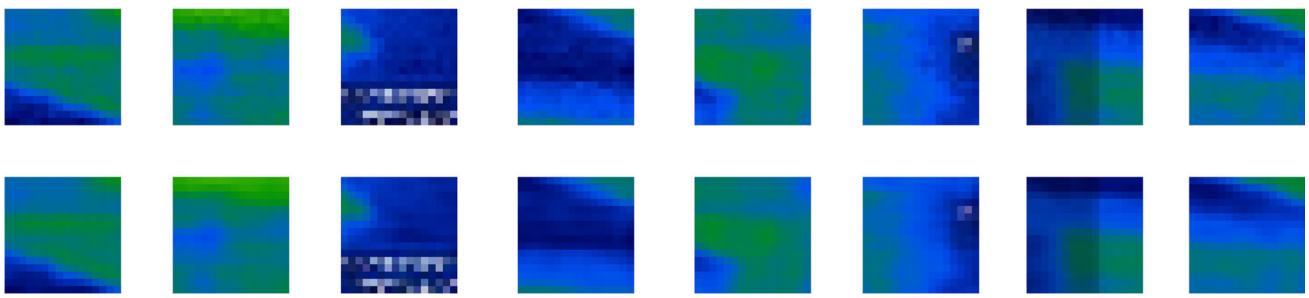

Fig. 9 Reconstruction results on some test patches of FLIR-SEEK dataset: the top row shows original input; the second row shows the reconstruction of those original inputs

Table 10 Recapitulation of ViT-based models used for ablation studies

Model	Image Tokenization	Positional Embedding	Learnable Embedding	Class Token	Augmented Transformer	Sequence Pooling
ViT	Hard	Yes	Yes	Yes	No	No
ViT-A	Hard	Yes	Yes	Yes	Yes	No
ViT-SP	Hard	Yes	Yes	No	No	Yes
CCT	Convolution	Optional	No	No	No	Yes
CCT-A	Convolution	Optional	No	No	Yes	Yes
CCT-NSP	Convolution	Optional	No	No	No	No
AE-ViT	LCVAE	Optional	No	Yes	No	No
AE-CCT	LCVAE	Optional	No	No	No	Yes
AA-CiT-NSP	LCVAE	Optional	No	No	Yes	No
AA-CiT	LCVAE	Optional	No	No	Yes	Yes

ratio (PSNR) and structural similarity index measure (SSIM) [48].

Figures 7 and 8 show the evaluation metric values. The reconstruction results on some test samples are shown in Fig. 9.

Appendix B: Models' Summary

Table 10 shows the key differences in terms of the necessity of positional embedding, presence of learnable embedding, Class token and SeqPool and Transformer structure in the backbone, between the different models used in ablation studies.

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I.: Attention is all you need. In: International Conference on Neural Information Processing Systems, vol. 30, pp. 6000–6010 (2017)
2. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186 (2019). https://doi.org/10.18653/v1/N19-1423
3. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
4. Conneau, A.; Lample, G.: Cross-lingual Language Model Pretraining. In: Wallach, H.; Larochelle, H.; Beygelzimer, A.; d' Alché-Buc, F.; Fox, E.; Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32 (2019)
5. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; Houlsby, N.: An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
6. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M.: Transformers in vision: A survey. ACM Comput. Surv. **54**(10s), 1–41 (2022). https://doi.org/10.1145/3505244
7. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; Yang, Z.; Zhang, Y.; Tao, D.: A survey on vision transformer. IEEE Trans. Patt. Anal. Mach. Intell. **45**(1), 87–110 (2023). https://doi.org/10.1109/TPAMI.2022.3152247
8. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning Era. In: International Conference on Computer Vision (ICCV), pp. 843–852 (2017). https://doi.org/10.1109/ICCV.2017.97
9. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
10. Yan, K.; Wang, X.; Lu, L.; Summers, R.M.: DeepLesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning. J. Med. Imag. **5**(3), 036501–036501 (2018). https://doi.org/10.1117/1.JMI.5.3.036501
11. (2018): FLIR thermal starter dataset. [Online]. Available: https://www.flir.com/oem/adas/adas-dataset-form/
12. Kim, S.; Song, W.-J.; Kim, S.-H.: Infrared variation optimized deep convolutional neural network for robust automatic ground target recognition. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 195–202 (2017). https://doi.org/10.1109/CVPRW.2017.30
13. Hong, F.; Song, J.; Meng, H.; Wang, R.; Fang, F.; Zhang, G.: A novel framework on intelligent detection for module defects of PV plant combining the visible and infrared images. Solar Energy **236**, 406–416 (2022). https://doi.org/10.1016/j.solener.2022.03.018
14. Abreu de Souza, M.; Krefer, A.G.; Borba, G.B.; Centeno, T.M.; Gamba, H.R.: Combining 3D models with 2D infrared images for medical applications. In: International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 2395–2398 (2015). https://doi.org/10.1109/EMBC.2015.7318876
15. Akula, A.; Sardana, H.K.: Deep CNN-based feature extractor for target recognition in thermal images. In: IEEE Region 10 Conference (TENCON), pp. 2370–2375 (2019). https://doi.org/10.1109/TENCON.2019.8929697
16. Ke, A.; Ellsworth, W.; Banerjee, O.; Ng, A.Y.; Rajpurkar, P.: CheXtransfer: performance and parameter efficiency of ImageNet models for chest X-Ray interpretation. In: Conference on Health, Inference, and Learning, pp. 116–124 (2021). https://doi.org/10.1145/3450439.3451867

17. Zhang, W.; Deng, L.; Zhang, L.; Wu, D.: A survey on negative transfer. IEEE/CAA J. Autom. Sin. **10**(2), 305–329 (2023). https://doi.org/10.1109/JAS.2022.106004
18. Simonyan, K.; Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations, San Diego, CA, USA (2015)
19. He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
20. D'Ascoli, S.; Touvron, H.; Leavitt, M.L.; Morcos, A.S.; Biroli, G.; Sagun, L.: ConViT: Improving vision transformers with soft convolutional inductive biases. In: International Conference on Machine Learning, vol. 139, pp. 2286–2296 (2021)
21. Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; Feng, J.: A^2-Nets: Double attention networks. In: Advances in Neural Information Processing Systems, 31 (2018)
22. Ramachandran, P.; Parmar, N.; Vaswani, A.; Bello, I.; Levskaya, A.; Shlens, J.: Stand-Alone Self-Attention in Vision Models **32**, 68–80 (2019)
23. Wang, H.; Zhu, Y.; Green, B.; Adam, H.; Yuille, A.; Chen, L.-C.: Axial-DeepLab: Stand-alone axial-attention for Panoptic segmentation. In: European Conference on Computer Vision, pp. 108–126 (2020). https://doi.org/10.1007/978-3-030-58548-8_7
24. Zhao, H.; Jia, J.; Koltun, V.: Exploring self-attention for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 10073–10082 (2020). https://doi.org/10.1109/CVPR42600.2020.01009
25. Meng, H.; Yuan, F.; Tian, Y.; Wei, H.: A ship detection method in complex background via mixed attention model. Arab. J. Sci. Eng. **47**(8), 9505–9525 (2022). https://doi.org/10.1007/s13369-021-06275-2
26. Boulahia, S.Y.; Benatia, M.A.; Bouzar, A.: Att2ResNet: a deep attention-based approach for melanoma skin cancer classification. Int. J. Imag. Syst. Technol. **32**(2), 476–489 (2022). https://doi.org/10.1002/ima.22687
27. Park, J.; Woo, S.; Lee, J.-Y.; Kweon, I.-S.: BAM: bottleneck attention module. In: British Machine Vision Conference, p. 147 (2018)
28. Billel, N.; Atmane, K.; Abdelkrim, N.; Laurent, M.: Augmented convolutional neural network models with relative multi-head attention for target recognition in infrared images. Unmanned Syst. (2022). https://doi.org/10.1142/S2301385023500085
29. Bello, I.; Zoph, B.; Le, Q.; Vaswani, A.; Shlens, J.: Attention augmented convolutional networks. In: International Conference on Computer Vision (ICCV), pp. 3285–3294 (2019). https://doi.org/10.1109/ICCV.2019.00338
30. Srinivas, A.; Lin, T.-Y.; Parmar, N.; Shlens, J.; Abbeel, P.; Vaswani, A.: Bottleneck transformers for visual recognition. In: Conference on Computer Vision and Pattern Recognition, pp. 16519–16529 (2021). https://doi.org/10.1109/CVPR46437.2021.01625
31. Hassani, A.; Walton, S.; Shah, N.; Abuduweili, A.; Li, J.; Shi, H.: Escaping the big data paradigm with compact transformers. arXiv preprint arXiv:2104.05704 (2021)
32. Kingma, D.P.; Welling, M.: Auto-encoding variational Bayes. In: International Conference on Learning Representations, ICLR (2014)
33. Ashfaq Qirat, Z.R. Akram Usman: Thermal Image dataset for object classification (2021). https://doi.org/10.17632/btmrycjpbj.1
34. Lee, S.H.; Lee, S.; Song, B.C.: Vision transformer for small-size datasets. arXiv preprint arXiv:2112.13492 (2021)
35. Wu, H.; Xiao, B.; Codella, N.; Liu, M.; Dai, X.; Yuan, L.; Zhang, L.: CvT: Introducing convolutions to vision transformers. In: International Conference on Computer Vision (ICCV), pp. 22–31 (2021). https://doi.org/10.1109/ICCV48922.2021.00009
36. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning, vol. 139, pp. 10347–10357 (2021)
37. Yuan, K.; Guo, S.; Liu, Z.; Zhou, A.; Yu, F.; Wu, W.: Incorporating convolution designs into visual transformers. In: International Conference on Computer Vision (ICCV), pp. 579–588 (2021). https://doi.org/10.1109/ICCV48922.2021.00062
38. Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.-H.; Tay, F.E.H.; Feng, J.; Yan, S.: Tokens-to-token ViT: training vision transformers from scratch on ImageNet. In: International Conference on Computer Vision (ICCV), pp. 558–567 (2021). https://doi.org/10.1109/ICCV48922.2021.00060
39. Zagoruyko, S.; Komodakis, N.: Wide residual networks. In: British Machine Vision Conference (BMVC), pp. 87–18712 (2016). https://doi.org/10.5244/C.30.87
40. Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; Jégou, H.: Going deeper with image transformers. In: International Conference on Computer Vision (ICCV), pp. 32–42 (2021). https://doi.org/10.1109/ICCV48922.2021.00010
41. Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L.: Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: International Conference on Computer Vision (ICCV), pp. 548–558 (2021). https://doi.org/10.1109/ICCV48922.2021.00061
42. Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; Wang, Y.: Transformer in transformer. Adv. Neural Inf. Process. Syst. **34**, 15908–15919 (2021)
43. Ioffe, S.; Szegedy, C.: Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015)
44. Loshchilov, I.; Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations, ICLR (2019)
45. Huang, G.; Sun, Y.; Liu, Z.; Sedra, D.; Weinberger, K.Q.: Deep networks with Stochastic depth. In: European Conference on Computer Vision, pp. 646–661 (2016). https://doi.org/10.1007/978-3-319-46493-0_39
46. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
47. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C.: MobileNetV2: Inverted residuals and linear bottlenecks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018). https://doi.org/10.1109/CVPR.2018.00474
48. Hore, A.; Ziou, D.: Image quality metrics: PSNR vs. SSIM. In: International Conference on Pattern Recognition, pp. 2366–2369 (2010). https://doi.org/10.1109/ICPR.2010.579