Exploring artefacts of Vision Transformer feature maps

Florian Weidner
Philipps-University Marburg, Germany
Department of Mathematics and Computer Science, Deep Learning Group
February 09, 2025

***Abstract*—Vision Transformers (ViTs) became a powerfull architecture for various computer vision tasks, learning visual representations of images. Object discovery methods like LOST [9] use these representaions, enabling a self-supervised training process. Recent studies have discovered artifacts in the feature maps of various ViTs. With these high-norm tokens observed in the background of the image, methods like LOST using the feature maps, perform really bad. This paper summerizes the discovery of artifacts of Darcet et al. and Yang et al., both researching the behaviour of artifact appearnces. Darcet et al. propose to add register tokens to remove the high-norm tokens. Wang et al. also successfully applied the register tokens to Vision Mamba. Yang et al. propose a denoising approach built ontop of ViTs to remove the artifacts. The paper summerizes the findings and propsed solutions.**

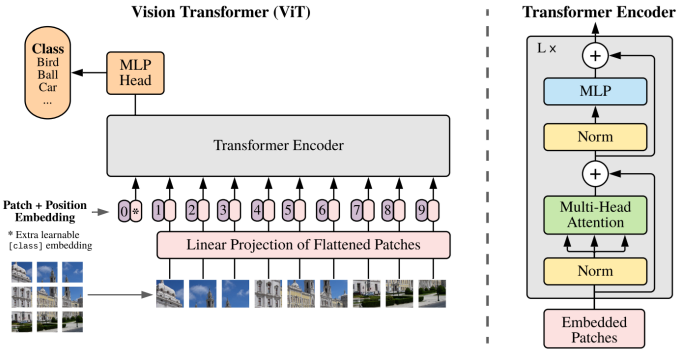


Fig. 1. Overview of a ViT architecture. Image obtained from [5]

I. Introduction

Transformers [12] using multi-head self-attention mechanisms have become the model of choice for Natural Language Processing (NLP) tasks. The approach to pre-train the model on large text data and then finetune on a smaller task-specific dataset has been very successful. The self-attention mechanism allows the modal to learn global contexts and long-range dependencies. With the efficiency and scalability of transformers, it became possible to train very large and performant models with many parametes. [12] [5] [8]

Vision Transformers (ViTs), introduced by [5], use the transformer architecture for computer vision tasks. They split the input images into patches and feed it throug a transformer encoder. They also became the state of the art architecture, achieving high prediction performance. They can learn rich visual representaions of images, that can be used for various compter vision tasks like classification, segmentation, object discovery and many more. [5] [8]

Recently studies have observed artifacts in the feature maps of ViTs, the output of the encoder after each layer. Artifacts have been discoverd for many models, mostly at semantically low background patches of the feature maps. [4] [14] The authors of [4] were the first to observe these artifacts. They describe them as high-norm tokens in the feature maps, that are part of the ViT architectur. They appear mostly in large models, after a third of the training process and lead to bad representations of the feature maps, unable to use them for clustering or object discovery. The authors propose to use additional tokens called registers, to remove the artifacts. They should enable to store global information there, that the artifacts, that stored that global informations before, won't appear anymore.

[13] found out that the high norm tokens also appear in Vision Mamba, a model using State Space Model (SSM) mechanisms instead of self-attention. The model shows even more artifacts than the ViTs, tested in [4]. Using registers in Vision Mamba, also removed artifacts and improved the performance of the model. [13]

[14] builds upon the discoveries of [4], confirming the existance of artifacts in ViTs. They also found artifacts in smaller models, and also showed that weak artifacts even appear, when using registers. They connect the artifacts with the existance of positional embeddings of the ViT architectur. Instead of register tokens, they propose a denoising approach, that can be connected to exisiting models, so that on new models have to be trained. They try to seperate the semantic information from the noisy artifacts for each image and then train a denoiser model that can generalize and remove the artifacts, created by the positional embeddings. They approach showed to be better than using registers [14]

The rest of the paper is divided into following parts. In the next section the basics about ViTs, and some concrete models, that are used in [4], are introduced. In section III the paper [4] is summerized. Afterwards the two studies [13] [14] that build upon the findings of [4] are summerized in section IV. Section V concludes the paper.

II. Vision Transformers

The Transformer architecture is a neural network model architecture, created primarily for sequence-to-sequence tasks in NLP.

> "The key feature of transformers is the self-attention mechanism, which helps a model learn the global contexts and enables the model to acquire the long-range dependencies." [8]

It consists of an encoder, which makes the input sequence into a continuos representation and a decoder, which then generates the output sequence. The encoder is built up of n identical layers, containing following components:

- **Multi-head self-attention mechanism**: captures relationships between all tokens in the input, regardless of their distance
- **Feed-forward network**: simple two-layer MLP network with ReLU activation which is applied to each token separately
- **Add & Norm layers** using residual connections and layer normalization to stabilize the training

The result outcome of the encoder is a enriched sequence representation, which is then used by the decoder to generate the output sequence. The decoder also consits of n identical layers with:

- **Masked multi-head self-attention mechanism**: ensures a causal generation, by preventing that tokens have impact to future tokens
- **Encoder-decoder attention mechanism**: focuses on the relevant parts of the encoder's output
- **Feed-forward network**: similar to the encoder
- **Add & Norm layer**:ssimilar to the encoder

The input text is embedded and combined with a positional encoding to provide token order information. Because several attention layers can run in parallel, the architecture is significantly more parappelizable than Recurrent Neural Network (RNN) or Convolutional Neural Network (CNN) architectures, which makes it very efficient for modern hardware accelerators. That allows the Transformer to scale to very large models and datasets. [12]

Dosovitskiy et al. introduced the idea of using the stated transformer architecture for computer vision. A lot of research tried to combine self-attention mechanisms with CNN architectures, not achieving a effectively scalable method for modern hardware accelerators. [5] proposed to apply a standard Transformer directly to images, that are split into fixed-size patches. Each patch is flattened into a vector and passed through a linear projection layer to form an embedding as input for the Transformer. These embeddings are used as tokens in a NLP scenario. Positional embeddings are added to retain spatial information since they process images as sequences, unlike CNNs which inherently capture spatial hierarchies. For classification tasks, an extra learnable [class] embeding is added in front of the embedded input. At the output of the encoder, the final representation of this token is used for classification. Instead of using encoder and decoder like in NLP tasks, ViTs only use the encoder since the goal is to find a better representation but an autoregressive prediciton. Additional Layer Normalization is used before the multi-head attention layer. [8]

In figure 1 you can see the architecture of a ViT including the split image patches, their embeddings combined with positional embeddings, the encoder, and the class embedding used for the classificaion prediction. ViTs have much less image-specific inductive bias than CNNs, because other than CNNs, with the global self-attention mechanism spatial relationships needs to learned from scratch, but long-range dependencies across the entire image can be captured. As Transformers, ViTs are normally pre-trained on large datasets and then fine-tuned to more specific tasks. After pre-training, the prediction head is removed and a zero-initialized feedforeward layer, where the size is the number of classes, is added.

Like Transformers, ViTs are also very parallelizable, which makes them very efficient. But [5] found out that without large-scale pre-training, ViTs often underperform. So ViTs requires sidnificant computational resources. But when pre-trained on large datasets, ViTs outperforms CNNs on image classification tasks. The architecture performs well for transfer learning, where the pre-trained model can be fine-tuned already with limited labeled data [5]. [5] stated that further scaling of ViTs would likely lead to improved performance. Also self-supervised pre-training, where no labeled data is needed, can be improved. They found out that with mimicking the masked language modeling task used in BERT, the model performs still better than CNNs but a bit worse that with supervised pre-training of a ViT. By now different architectures and training-tricks of ViTs have been proposed to further improve ViTs including self-supervised learning. The architectur also got adapted for image recognition, object detection, image segmentation, pose estimation, and 3D reconstruction tasks. [8]

The classical ViT architecture has been adopted and improved by many others. One approach is to include CNN structures, which bring locality through the convolution kernels, into ViTs to improve the data efficiency. DeiT [11] for example uses a CNN as a teacher to train a ViT. It utilizes knowledge distillation of the CNN to add the inductive bias to a vision transformer. It allows to train a ViT without the need of large-scale pre-training the model. [8] Another approach is to diversify the features of ViTs. DeepVit [15] found out that the attention collapses in deeper layers, wich leads to lower performance. By adding a learnable transformation matrix after the attention layer, the model is stimulated to generate new attention maps also in the deeper layer, increasing the performance. [8] Also the heavy computation costs are reseached. Many also try to improve the self-supervised learning, that the pre-training with the need of large datasets can be simplified. [8]

In the following sections the concepts of different ViT architectures are introduced. These models are used in the paper [4] that will be summerized afterwards.

A. DINO and DINOv2

DINO is a self-supervised learning framework using a student-teacher network. The teacher is dynamically built from past iterations of the student network. It uses self-destillation with no labels. For each image, two high-resolution global views and several low-resolution local views are generated. The teacher only processes the global views using an Exponential Moving Average of the student's weights. The student processes global and local views and the cross-entropy loss is used to calculate the similarity between the student and teacher. It also uses mechanisms to avoid trivial solutions. [2] It is shown that the the attention maps contain explicit information about the semantic layout of an image. [4]

DINOv2 further improves the idea of DINO, enhancing scalable, efficiency and genalization of self-supervised learning in computer vision. Following techniques are used to improve the model:

- an automatic pipeline to build a dedicated, diverse, and curated image dataset
- bigger ViT model with one bilion parameters
- distilling the model into a series of smaller models [6]

The improvments enable dense prediciton tasks. On the other hand, artifacts in the attention maps of DINOv2 are observed. [4]

B. OpenCLIP

OpenCLIP is a open source implementation of Contrastive Language-Image Pre-training (CLIP) [7] from OpenAI, which uses language-image pre-training to enable zero-shot transfer to a wide range of tasks. CLIP tries to predict the caption of an image. It tries to maximize the similarity between correct pairs and minimizing the similarity for incorrect paris. The pre-trained model, that outputs text from an input image, can be used to extract information from the output text for various specific tasks. The models are competetive with compared supervised models that are specifically trained for the task. [7] OpenCLIP has trained several models of different sizes with different data sources. [3]

C. DeiT-III

DeiT-III focuses on supervised training of ViTs, trying to create a new baseline for supervised ViT models. A new data augementation approach, inspired by self-supervised learning techniques is used before training. Also they use Simple Random Cropping insted of Random Resize Cropping. The image resolution has been lowered. With a change from 224 × 224 to 126 x126, 70% fewer tokens are used. It turned out to prevent overfitting for the larger models and achieves better performance. Additionally they adopted the binary cross entropy loss, which improved the training of large ViTs. [10]

D. LOST

LOST is a self-supervised object discovery algorithm, that can be used to detect objects in an image without the need of any labled data. To find an object in an image it uses a ViT model like DINO and feeds the image through the ViT.

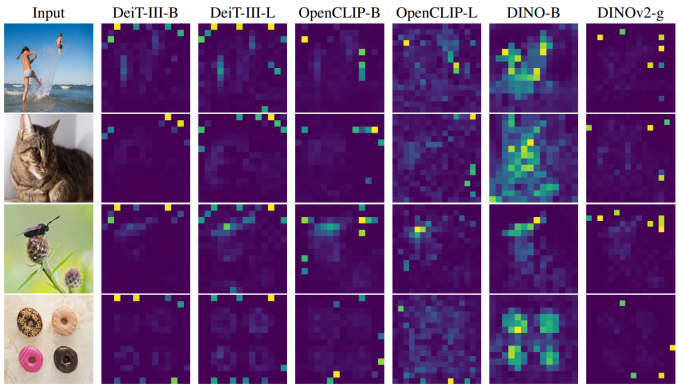


Fig. 2. Illustration of artifacts observed in the attention maps of modern vision transformers. Image obtained from [4]

It is assumed that every image has at least one object. Instead of looking at the class token for classificaion results, the attention maps of the last layer are used to compute similarites between differnt patches. The patch with the smallest number of positive correlation with other pathces is used as the *seed*. They state that is likely hit an object because

> "patches within objects correlate more with each other than with background patches and vice versa, and ... an individual object covers less area than the background. Consequently, a patch with little correlation in the image has higher chances to belong to an object." [9]

After selecting the *seed*, additional patches correlating with the seed are searched, because they are also likely to belong to the same object. After that a bounding box is calculated by comparing the seed features with all the image features. The fact that the method detects objects on a single image, without the need of exploring the image collection, makes it very scalable. After that a class-agnostic detection model is trained with the generated bounding boxes. Here also more objects in one image can be detectd. It also turns out the trained model is more accurate than finding boxes of the correlations. This method provides pseudo-boxes without a category of the object. To also detect a semantic category in a self-supervised way, the usage of K-means clustering is presented. The detected objects are cropped and resized and then fed throug a DINO pre-trained transformer. The class tokens is then extracted and than clustered with the K-means algorithm. That gives pseudo-labels that are matched with the ground truth labels at evaluation time using the Hungarian algorithm. [9]

III. Vision Transformers need registers: A summary

In this chapter we summerize the paper [4]. The paper discoverd artifacts and proposes to use additional register tokens for ViTs to remove these artifacts.

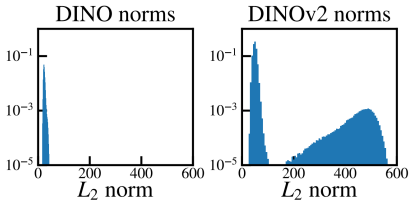
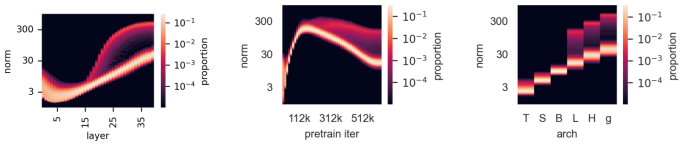


Fig. 3. Comparison of local feature norms for DINO ViT-B/16 and DINOv2. Image obtained from [4]



(a) Norms along layers. (b) Norms along iterations. (c) Norms across model size.

Fig. 4. Illustration of several properties of outlier tokens in the 40-layer DINOv2 ViT-g model. Image obtained from [4]

A. Artifacts in Vision Transformers

After introducing to ViTs like we did in this paper, the models they found the artifacts are introduced. The DINO algorithm is a self-supervised learning method, that learns rich representaions of visual data without the need of manual annotations. [2] DINO is shown to produce models, that contain semantically consistent information in the last attention layer. Object discovery algorihtms like LOST [9], built on top of DINO, are using these attention maps, that often contains semantically interpretable information, used to detect objects without supervision. DINOv2 [6] is a improved followup focusing on dense predition tasks, which are tasks, where detailed outputs are required to provide fine-grained localized informations, like semantic segementation or depth estimation. Despite good performance on these dense tasks, the authors observed that DINOv2 is incompatible with LOST [4]. The different behaviour of DINO and DINOv2 can be observed in the artifacts in the last attention maps. In figure 2 you can see the different models and their artifacts on the last attention layer. While DINO shows no peak outlier values focusing the main object in the image, DINOv2 shows a lot of artifacts on the background of the images. This qualitatively observation can be also made for the label-supervised model DeiT-III and the text-supervised model OpenCLIP. Shown in figure 2, you can observe similar artifacts in the background. To explain why and where the artifacts of ViTs in attention maps appear, the paper focuses on DINOv2.

Artifact patches show higher norm of their token embedding at the output of the model than other patches. In figure 3 you can see the distribution of the local feature norms over a small dataset. While for DINO, the norm stays under 100 for all patches, DINOv2 shows a lot of patches with a norm higher than 150. This cutoff value can vary across different models. They define artifacts as

> ”tokens with norm higher than 150 will be considered as “high-norm” tokens” [4]

The authors found different conditions, when the artifacts appear in the training process of DINOv2. Figure 4 shows the following conditions:

- artifacts start appearing around layer 15 to 40.
- artifacts start appearing after on thrid of training.
- artifacts only appear in the three largest model versions

Another discovery is that the high-norm tokens appear where patch information is redundant. The authors tested the cosine similarity between high-norm tokens and their four neighbors, directly after the image is emebdded. They observed, that the high norm patches appear where their cosine similarity to the neighbors is high. Compared to the observations, that shows that artifacts appear mostly in the background of images, high-norm pathes seem to have redundant information, that the model can ignore, to achive similar scores at the output.

To further understand the outlier tokens, two linear models were trained, to check the embeddings for different information. Both models were trained on the patch embeddings, the embeddings of the images (see figure 1). The result performance is compared between using high-norm tokens and normal tokens. The first task was position prediction. The model should predict the position of a patch token in the image and measure the accuracy. They observed that high-norm tokens have much lower accuracy than the other tokens and suggested that they contain less information about the position in the image. The second task was pixel reconstruction. The model should predict the pixel value of an image from the patch embeddings and mesaure the accuracy of this model. Also here the high-norm tokens have lower accuracy than the other tokens. The authors concluded that the high-norm tokens contain less information to reconstruct the image than the others. The authors also found out that the high-norm tokens hold more global information by training a logistic regression model. The model predicts the image class by the patch embedding of a random token. I turned out that the high-norm tokens have a much higher accuracy than the other tokens. This suggests that the high-norm tokens contain more global information about the image than the other tokens.

Making these obervations the authors make following hypsthesis:

> ”Large, sufficiently trained models learn to recognize redundant tokens, and to use them as places to store, process and retrieve global information.” [4]

B. Registers for Vision Transformers

To address the behaviour, the use of registers is proposed. Since the high-norm patches are overtaking local patch information, even they are mostly not important, it possibly decreases the performance on dense prediction tasks. The called registers are additional tokens after the patch embeddings of the images with a learnable value. They work similar to the [class] token, used for classificaion tasks. They are used durning training and inference and the outputs of them are discarded afterwards. In figure 5 you can see the register

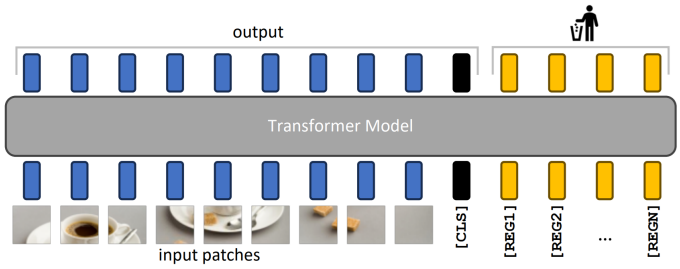


Fig. 5. Illustration of the proposed remediation and resulting model. Image obtained from [4]

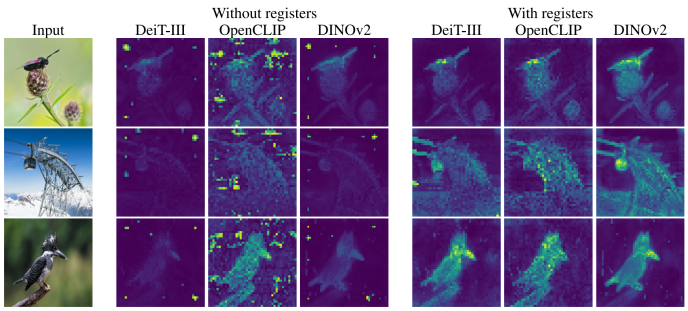


Fig. 6. Three examples of attention maps with and without register tokens. Image obtained from [4]

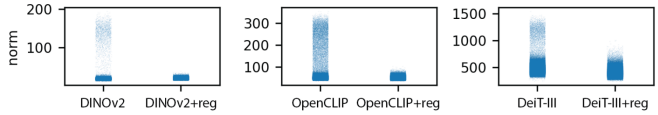


Fig. 7. : Effect of register tokens on the distribution of output norms. Image obtained from [4]

tokens additionally used after the embedding of the image. A complexity analysis show that adding registers increase the FLOPs by up to 6% for 16 registers. With four registers, that are more commonly used, the increase is below 2%.

The idea of adding additional tokens as memory to a transformer model is from [1]. The study adds trainable memory to transformer for NLP tasks. Many studies before have tried memory augementation in neural networks, to improve the performance of the models. For Transformers the paper use general purpose [mem] tokens that can be used as placeholders by the model, to store global information or copy also local representaions. They are proposing three different architectures using memory tokens. The first one is just concatenate the tokens to the input, and process them together in one encoder by layers with the same parameters. This is the approach that is adapted for the register tokens of the ViT. The second architecture of [1] is to use a sepereate memory control layer and the third architecture further restricting the processing by first updating the attention of the memory and then update the attention maps of the sequence. The evaluation showed that the basic memory architectur outperforms baseline transformers. The other architectures had not so clear results, sometime increasing, sometimes decreasing performance of baseline transformers.

C. Evaluation of the proposed architecture

In the last part of the paper they validate their architecture by training ViTs with register tokens and compare them quantitativly and qualitativly to the models without token registers. They are evaluating for DeiT-III, OpenCLIP and DINOv2 architectures, therfore including label-supervised, text-supervised and self-supervised learning approaches. In figure 6 you can see three example images including attention maps with and without the use of register tokens. Qualitativly, for all three models, the artifacts in the attention maps are gone. They mesaured quantitativly the effect by calculating the norm of the attention maps at the output of the model. In figure 7 you can see the distribution of the output norms for the three models. For all three models, training it with register tokens removes high-norm tokens, that were present without the token registers. Instead the attention maps of the register tokens have higher norm than the patch and the class tokens. The register tokens are adapting the behaviour of the outlier patches of the model without registers. Visualizations are also showing that the attention maps of the register tokens look similar to the attention maps of the class tokens, all showing a larger support area. The attention maps of the patch tokens are more localized. Since the class token carries global informations, it suggests that the register tokens are also used to store global information. Comparing the performance of the models with and without register tokens, linear probing on ImageNet classification, ADE20k Segmentation, and NYUd monocular depth estimation datasets was used. The results show no lose in performance, when additoinally using register tokens. Also for zero-shot classification on ImageNet with OpenCLIP, the performance is not affected by using register tokens. They also found out that one register is enough to remove the high-norm tokens in the attention maps. For DINOv2 and DeiT-III, adding register tokens significantly improves the discovery performance and for OpenCLIP, the performance is slighty worse with registers. The authors concluded that their proposal isolates the behaviour of the model using memory for global information. It was shown that ViTs naturally using patches to store global information. With creating registers exactly for that purpose, collateral side-effects, like bad performance of LOST with DINOv2 can be avoided.

IV. Buildup studies

The following two studies build up on the findings of [4] and discover additionl infomation about artifacts in ViTs.

A. [13] applies the idea of registers to a SSM

The authors applied the use of register tokens to the Vision Mamba model, after also discovering outlier tokens in the background, achieving higher performance than without registers.

Vision Mamba [16] is a model architecture using bidirectional State Space Models (SSMs). The VIM Blocks, that are inspired by SSM can maintain long-range dependencies

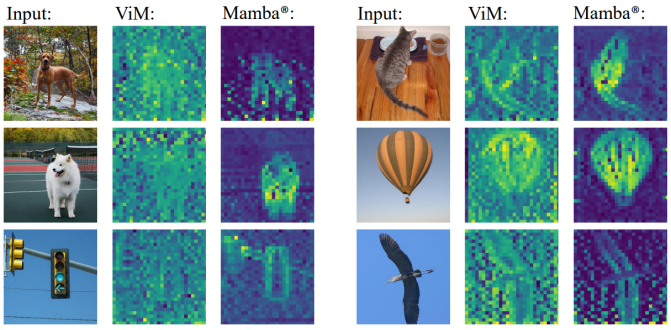


Fig. 8. Feature maps of vanilla Vision Mamba [16] and Mamba® using registers. Image obtained from [13]

in the model, similar like the attention mechanism for ViTs. Otherwise, it uses a feedforeward network, positonal encoding and normalization. Images are also decomposed into patches and then used as input to the Vision Mamba encoder. The big advantage compared to the quadratic complexity of self-attention mechanism is that its computational complexity is only linear. Therefore the training process and the memory usage is way lower than using ViTs and CNNs. The architectur outperforms ViTs like DeiT [11] on some tasks, showing the potential of using SSM in computer vision. [16] [13]

The authors found the same artifacts in the feature maps of Vision Mamba than Darcet et al. found artifacts in the attention maps of various ViTs. They even exists considerably more severe in the Vision Mamba, starting already in small models sizes. In figure 8 you can see the artifacts, which are spread all over the image, but also mainly appearing in background regions of the images. The feature maps of the Vision Mamba, that are used for the analysis, are the ℓ_2 distances between the global and local outputs. The artifacts appear to have a high normalization. Similar graphes like figure 3 are presented. It is also shown that the artifacts contain global information. Building upon the architecture from [4] using register tokens, they insert the tokens evenly between the token sequence of the image. Since the tokens are not agnostic to their position in the Vision Mamba, having the registers near the whole sequence of input tokens. Another difference is that they concatenate the register tokens at the end, to use them for the final prediction. Doing that, they observed significant improvements. They also observed that the different registers highlighting different objects or semantic elements within a picture. Since Vision Mamba architecture has no multi-head mechanism like the attention mechanism in ViTs, it offers a lot of information that can be used for interpreting the result of the model. The proposed Mamba® architectur outperforms all prior Mamba variants for image classification and semantic segmentation. [13]

B. [14] uses denoising to remove artifacts

The paper also discovers noise artifacts in feature maps of ViTs, including DINOv2, DeIT-III, CLIP and EVA02. They state that the noise hinder feature interpretability and worsens the performance of applying additional methods form the output of ViTs like clustering. The paper focuses on dense recognition tasks, where these artifacts affect the performance of the model, unlike for simple classification. It is hypothesized that positional embeddings play a fole in the appearance of the artefacts. The authors found a correlation between the inclusion of positional embeddings and the emergence of undesirable artifacts in ViT outputs. With the maximal information coefficient, the dependency between grid features and their normalized patch coordinates are measured. The outputs of the original ViT show a higher spatial correlation than the denoised features (denoised by their solution explained later).

Comparing the ViT findings from [4], artifacts are also observed in small or base ViTs that cannot be easily identified by their high norm values. Also weak artifacts are found in DINOv2 using registers. The artifacts are shown in all layers, even using only zeros as input. Shallower layers show more low-frequency patterns that deeper layers, that show more high-frequency patterns.

The authors propose an aproach to dennoise the feature maps in two steps, without the need to retrain the models, called Denoising Vision Transformers (DVT). The first step is per-image dennoising with neual fields, trying to seperate usefull semantic information from the noisy positional artifacts. The authors propose that a feature map can be factorized in three components.

- The clean semantic feature representaion
- The artifacts, that depends on the positional embeddings
- A residual interaction term.

For each image, multiple cropped and transformed versions are used to seperate the artifacts from the clean semantic feature representation. The same objects should have similar features across different transformations and artifacts are tied to position that they remain fixed across different transfomations of the image. With the definitions and the use of coordinate networks, known as neural fields, the artifacts can be seperated from the clean representaions over multible iterations with transformed images, minimizing a regularized reconstruction loss. This method effectively removes artifacts from ViTs but needs a high computational effort.

The second step of the denoising approach trains a genearlizable lightweight denoiser enabling to denoise images in real-time applications. Also denoising the images individually can lead to feature distribution shifts due to the bias of the single images. The denoised images of the first step are used to create a dataset to train a denoiser network. A single Transformer block is used, that learns to map raw ViT features to denoised features. Additional learnable positional embeddings are added after the foreward pass, to mitigate the input-independent artifacts. The trained model can now also generalize across samples, mitigating the distribution shifts of the first step. In figure 9 you can see the effect of using DVT on several images and models. In all images you can see the artifacts, mostly occuring in the background. Also on the model using registers [4], weaker but some artefacts are visible. Especially compared to the by DVT denoised image. On all denoised images, the artifacts are nearly completely removed. The feature

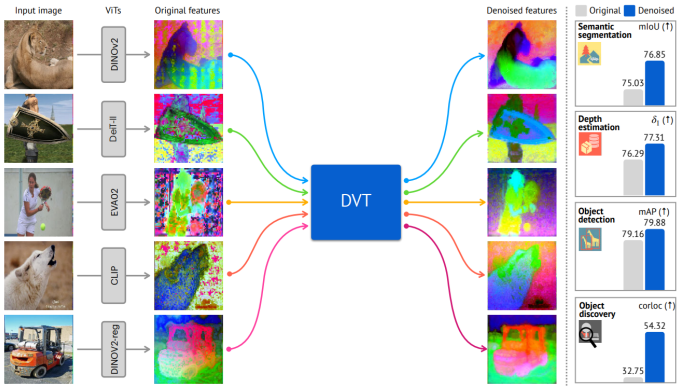


Fig. 9. Demonstration of DVT. Image obtained from [14]

maps of the denoised images show a much clearer and more interpretable objects. Also the performance improvements are visualized. [14]

The authors also stated that artifacts appear in all following task objectives that they evaluated. Here are the results of the evaluation of the different tasks.

- **Semantic segmentation**: DVT brings significant and consistent enhancements in all pre-trained ViTs across datasets including DINOv2 with registers from [4]
- **Depth estimation**: clearly enhances the performance of most pretrained ViTs
- **Object detection**: shows consistent improvements over the studied ViTs. Unlike [4], which didn't impove DINOv2 in object detection, using DVT improves the performance.
- **Object discovery**: DVT significantly improves DINOv2 in all the evaluated datasets. Also here DVT achieves better improvements than [4] using registers. Similar to the findings of [4], using DVT turned out to not only remove artifacts, but also makes the obejcts of images more distinctly visible from the feature maps. Even that was not the goal of DVT it helps methods like LOST (see section II-D). [14]

Additionally to [4], the paper gives more insights why and where artifacts in the feature maps of ViTs appear. Their approach DVT to remove the artifacts had better results that in [4] adding registers. Additionaly using DVT, you don't need to retrain the whole ViT, but you can additionally train the denoiser component and add it to you inference pipeline. The results of [14] also show that the combination of both proposals don't alwaly further improve the performance of the models. Only in the evaluation of the deph estimation and the semantic segmentation on the ADE20k dataset, the combination of DVT and using registers outperforms only using DVT. [14]

V. Conclusion

The studies [4] [13] and [14] all obeserved artifacts in different ViT-models. The high norm tokens in the feature maps of background patches, worsens the semantic representation of the feature maps and therefore the use of clustering or object discovery methods. Whereas [4] states the artifacts are just part of the ViT-architeture, [14] claims that the positoinal embeddings cause the artifacts. The solution proposals both remove artifacts and makes the obejcts of images more distinctly visible. While [4] tries to get to the bottom of the problem, by presenting a ViT architecture with register tokens, [14] builds a denoising approach built on top of ViTs. Additional studies are needed to further discover or verify the reason for appearing artifacts. These studies help to further understand the behaviour of ViTs and make them more interpretable.

References

[1] Mikhail S. Burtsev et al. *Memory Transformer*. 2021. arXiv: 2006.11527 [cs.CL]. URL: https://arxiv.org/abs/2006.11527.

[2] Mathilde Caron et al. *Emerging Properties in Self-Supervised Vision Transformers*. 2021. arXiv: 2104.14294 [cs.CV]. URL: https://arxiv.org/abs/2104.14294.

[3] Mehdi Cherti et al. "Reproducible Scaling Laws for Contrastive Language-Image Learning". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2023, 2818–2829. DOI: 10.1109/cvpr52729.2023.00276. URL: http://dx.doi.org/10.1109/CVPR52729.2023.00276.

[4] Timothée Darcet et al. *Vision Transformers Need Registers*. 2024. arXiv: 2309.16588 [cs.CV]. URL: https://arxiv.org/abs/2309.16588.

[5] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

[6] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. 2024. arXiv: 2304.07193 [cs.CV]. URL: https://arxiv.org/abs/2304.07193.

[7] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV]. URL: https://arxiv.org/abs/2103.00020.

[8] Bo-Kai Ruan, Hong-Han Shuai, and Wen-Huang Cheng. *Vision Transformers: State of the Art and Research Challenges*. 2022. arXiv: 2207.03041 [cs.CV]. URL: https://arxiv.org/abs/2207.03041.

[9] Oriane Siméoni et al. *Localizing Objects with Self-Supervised Transformers and no Labels*. 2021. arXiv: 2109.14279 [cs.CV]. URL: https://arxiv.org/abs/2109.14279.

[10] Hugo Touvron, Matthieu Cord, and Hervé Jégou. "DeiT III: Revenge of the ViT". In: *Computer Vision – ECCV 2022*. Ed. by Shai Avidan et al. Cham: Springer Nature Switzerland, 2022, pp. 516–533. ISBN: 978-3-031-20053-3.

[11] Hugo Touvron et al. *Training data-efficient image transformers & distillation through attention*. 2021. arXiv: 2012.12877 [cs.CV]. URL: https://arxiv.org/abs/2012.12877.

[12] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[13] Feng Wang et al. *Mamba-R: Vision Mamba ALSO Needs Registers*. 2024. arXiv: 2405.14858 [cs.CV]. URL: https://arxiv.org/abs/2405.14858.

[14] Jiawei Yang et al. *Denoising Vision Transformers*. 2024. arXiv: 2401.02957 [cs.CV]. URL: https://arxiv.org/abs/2401.02957.

[15] Daquan Zhou et al. *DeepViT: Towards Deeper Vision Transformer*. 2021. arXiv: 2103.11886 [cs.CV]. URL: https://arxiv.org/abs/2103.11886.

[16] Lianghui Zhu et al. *Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model*. 2024. arXiv: 2401.09417 [cs.CV]. URL: https://arxiv.org/abs/2401.09417.

that's all folks