Transformers applied to Computer Vision

Jordi Segura Pons
Universitat de Barcelona
jsegurpo8@alumnes.ub.edu

Pol Riba Mosoll
Universitat de Barcelona
pribamos17@alumnes.ub.edu

Leonardo Bocchi
Universitat de Barcelona
lbochi3@alumnes.ub.edu

Abstract

Natural Language Processing experienced a major breakthrough on year 2017, when Transformers were introduced in the paper "Attention Is All You Need" [7](Pol) emerging as a viable alternative to Convolutional Neural Networks (CNNs). After their irruption to the Machine Learning field, Transformers became one of the most promising and widely used tools for data modeling in the field of NLP. When working with pairs of input tokens, transformers measure the relationship between each pair. Before Visual Transformers, the computational cost of measuring relationships between pairs of input measures in images was prohibitively high due to the large number of pixels in an image. During year 2020, Transformers were then applied to Computer Vision, by breaking down input images and transforming each of them into vectors, which then can be seen as words in a normal transformer and reduced the computational costs severely. Visual transformers can be compared to Neural Networks, as the former also depends on the optimizer and specific hyper-parameters. During this article we will go through the main characteristics that differentiate Convolutional Neural Networks from Visual Transformers, understanding most of them lie on the architectural structure. This paper contributes on the Visual Transformers understanding while studying how they improve on their predecessors.

1 Introduction

In computer vision, arrays of picture capture the visual information from images or visual inputs. Comparing pairs of input tokens in order to receive relationship measures is now implemented by Visual Transformers. With the introduction of Transformers, Natural Language Processing made a huge advance, which has been translated with a natural growth on the field and the future of the technology related to the field. As said previously, before the implementation of Visual Transformers, the computational costs to compute relationships between pairs of graphical inputs was prohibitive, especially when working with heavy inputs. The Visual Transformers changed the methodology previously used by dividing the input into groups of pixels or group-sized patches (e.g. 20×20 pixels) and embedding them in the correct position. One of the most important advantages that Visual Transformers have contributed with to Computer Vision is that to train them, it can be done in parallel, with a later merge of the outputs. Computational Neural Networks are considered as Visual Transformers field predecessors, as the Computer Field relied on them for years, as the most innovative tool. Nowadays, when working with Computer Vision problems, Visual Transformers are preferred because of some advantages CCN's lack:

- Transformers learn with a more inductive bias and have a better learning rate when working with large data sets.

- The Transformers architecture is suitable for most fields they are used, as it is a visual model based on the architecture of a transformer.
- Long range interactions are efficiently computed with powerful and consistent results.
- Computational and storage costs are reduced drastically.
- The learned representation of relationships is general and robust, most of the times improving the results from convolution architectures.
- There is no need of convolutions when using Transformers.

During this article, we will demonstrate the points argued above, by putting into practice a Visual Transformer and comparing it to a Computational Neural Network.

2 Literature Review

The aim of this article is to demonstrate that Visual Transformers can solve Computer Vision problems in a more efficient way than other tools can. This is an important subject to discuss when talking about accuracy, computational and time cost, etc., specially when working with big inputs. Before 2020, and the application of Visual Transformers to Computer Vision, it would have obvious to work with a Computational Neural Network to solve a Computer Vision Problem, but times change and the field has evolved, and this project has the aim to contrast both methods and compare results in order to see if ViT really outperforms CNN.

We will demonstrate this, by comparing the results obtained when using Visual Transformers to the ones obtained when using the other methodologies. First of all, theoretical foundation of Visual Transformers and thus Transformers must be reviewed to make sure the methodology used is correct and every reader understands the basis of the subject. A typical Vision Transformer contains five basic steps:

- The first procedure is to split the input images into smaller sized- groups of pixels, called local patches.
- A number of stacked transformer blocks as Layer Normalization (LN), multi-head self-attention (MSA), skip-connection layer, multi-layer perception (MLP), feed-forward network (FNN), etc.

Figure 1 shows the typical structure of a Vision Transformers, also dividing the encoder in the different steps:

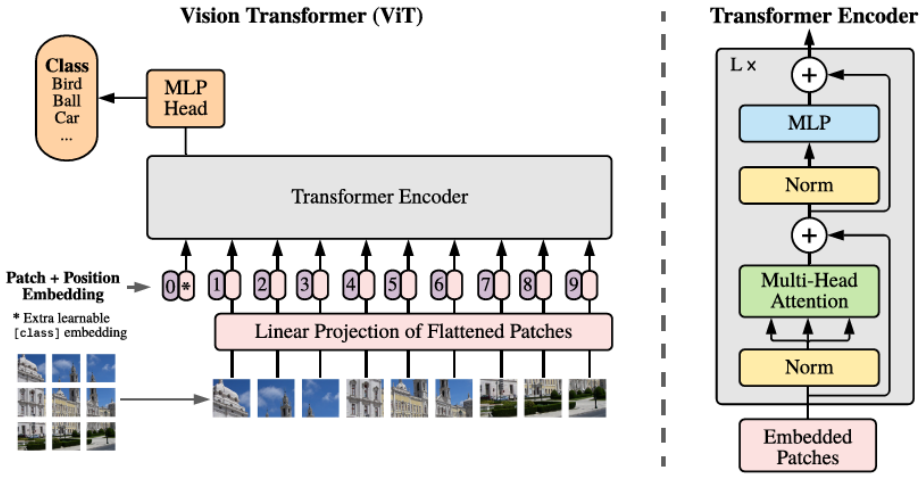

Figure 1: ViT architecture

Given an input image XR^{HWC}, it is reshaped into a sequence of 2D image patches $XR^{N(P^2C)}$. Once the input has been transformed, a class token and different position tokens are used to record extra meaningful information which will be used for inference. In [9] (Pol) more information about this process can be obtained. . If we add the input, we obtain the following formulation:

$$z_0 = [x_{cls}; x_p^1 * E; ...; x_p^N * E] + [E_{pos}^{cls}; E_{pos}^1; ...; E_{pos}^N]$$

where x_{cls} is the class token in R^D, E is a linear projection of each set of pixels X_p in $R^{D(P^2C)}$ and E_{pos}^i R^D is the learnable position embedding for the i^{th} token.

Once the input is defined, it is sent into several sequential transformer blocks, which are the following: If the layer is $l\{0, ..., L-1\}$ being L the number of transformed blocks:

$$z_{l+1} = z_l + MSA(LN(z_l))\ z_{l+1} = z_{l+1} + MLP(LN(z_{l+1}))$$

MLP includes two fully connected layers using GELU as an activation function, while LN(·) is a layer-normalization module and MSA is the following module:

$$MSA(z) = [SA_1(z)+;; SA_H(z)]U_{msa}$$

where $SA_i = *(\frac{Q*K^T}{sqrt(d_k)}) * V$, z being the input, $[Q, K, V] = zU_{qkv}^i, U_{qkv}^i R^{D(3D_h)}$ projects the D-dimensional input z to D_h dimensional Q, K, and V in the head i, (·) is the softmax function, and $U_{msa}R(HD_h)D$ re-casts the output from H heads of the MSA module into one D-dimensional output. There are different variants of MSA available.

When comparing Convolutional Neuronal Networks to Visual Transformers in Computer Vision, some main differences can be identified, most of them were explained by Fan H., et al. [5] (Pol) and Naseer M., et al. [6] (Pol). In this section, pros and cons from ViT when comparing to CNN will be discussed.

Visual transformers pros:

- ViT achives noteworthy results when comparing them to the ones achieved by CNN while making use of fewer computational resources for pre-training.
- ViT structure is generally more adaptive or suitable for most of the fields, while CNN can not always be used in all fields, specially because of its complexity and difficult explanation of the results. The mentioned ViT architecture is suitable for most of the fields, because it was originally designed for text-based tasks.
- ViT performance is extraordinarily good when trained on enough data, and when compared to CNN, it needs almost 4x fewer computational resources.
- The self attention in Visual Transformers, achieves that even the first layer of information processing makes connections between distant image locations as explained at Khan, S., et al.[**?**].
- CNN needs convolution, which makes the computational cost higher. On the other hand, ViT does not need to use convolutions.
- ViT is able to incorporate more global information than ResNet architecture (CNN) at lower layers, which leads to quantitatively different features.

Visual transformers cons:

- ViT generally shows a weaker inductive bias which results in increased reliance on data augmentation or model regularization, especially when training on smaller data files. Therefore, for ViT to be robust, long data sets are needed.

3 Transformers in Computer Vision

In recent years, transformers have gained significant attention in the field of computer vision due to their strong performance on tasks such as image classification, image segmentation, and image

generation. These tasks involve the analysis and interpretation of visual data, which is a crucial aspect of many real-world applications. One of the key advantages of transformers is their ability to capture global dependencies in the input data, which is particularly useful for image recognition tasks where the presence or absence of certain features may be important for correct classification.

In this section, we aim to explore the use of visual transformers for image recognition tasks and compare their performance to previous state-of-the-art results and architectures. Specifically, we will focus on the following tasks: image classification, image segmentation, and image generation. For each task, we will analyse experiments using different transformer architectures and compare their performance to existing methods. Additionally, we will investigate the use of fine-tuning techniques to further improve the performance of visual transformers on these tasks.

Overall, our goal is to provide a comprehensive analysis of the potential of visual transformers for image recognition tasks and identify the most promising directions for future research.

3.1 Methodology used for classification

Firstly, image classification was the main reason to bring people explore this innovative technique. It was presented in the work of Dosovitskiy, A., et al. [2](Jordi), where they performed an exhaustive study on how Visual Transformers may reach similar or even better results than SOTA[1] architectures, with much less computational power. Moreover, this paper introduced what it has been the way to go for the industry, to use pre-trained ViT on large datasets, and fine-tune[2] them to smaller datasets for concrete tasks, which yields in astonishing outcomes.

In this experiment, they made use of the ImageNet dataset, which contains 1000 classes and 1.3M images, a more sophisticated version called ImageNet-21k, with 21k classes and 14M images and the last one called JFT, with 18k classes and 303M images, and finally produced 3 variants of ViT, specified in the table 3.1, to check on each of them the test datasets. These 3 were used for training purposes, later we will see the results on other datasets, which were used for fine-tuning and testing purposes. Concretely, they used two versions of the ImageNet, the CIFAR-10 and 100, the Oxford datasets for flowers and pets and the VTAB one. As we already mentioned, they wanted to compare the Visual Transformers with the state-of-the-art CNNs, which were Big Transfer(BiT)-ResNets[3] and Noisy Student -EfficientNet[4].

Finally, all models were trained using Adam technique with batch sizes of 4096, as well as a SGD with momentum for fine-tuning. The metrics used to evaluate the models were mainly the fine-tuning accuracy, which indicates the accuracy of the model after being fine-tuned on the respective dataset.

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants

[1] State-Of-The-Art, term used to abbreviate the most cut-edge technologies

[2] Fine-tuning a transformer is the process of adapting a pre-trained transformer model to a specific task or dataset. This is typically done by "fine-tuning" the weights of the transformer model, which involves adjusting the model parameters to optimize performance on the specific task or dataset.

[3] A ResNet (short for Residual Network) is a type of convolutional neural network (CNN) that utilizes skip connections or shortcuts to improve the model's ability to learn and classify images. These skip connections allow the network to bypass layers of the network and directly access earlier layers, enabling the network to more easily learn from the data. This can be especially useful for training deep networks, as it helps mitigate the vanishing gradient problem and allows the network to more easily learn from the data.

[4] EfficientNets are a family of convolutional neural network models designed to achieve state-of-the-art accuracy while also being lightweight and efficient. They are designed using a combination of neural architecture search and compound scaling, which allows them to achieve improved accuracy while also reducing the number of parameters and computational complexity

3.2 Methodology used for object detection

It is time to investigate one of the most important areas of computer vision, object detection. As well as before, for the recent years the architecture which has been mostly used was CNN. However, M. Yang [10](Jordi) in his paper presents us an incredible way to use Transformers for this specific task, tweaking the backbone architecture, called DetTransNet.

DetTransNet model takes an image as input and divides it into n overlapping patches with m overlapping pixels. These patches are then embedded using a linear layer and processed through N Transformer encoder layers. The patches are then rearranged into an image, which is passed through a few residual blocks to obtain feature maps. A region proposal network is applied on top of the feature maps for object detection, with a classifier predicting the object category of bounding boxes and a box regressor outputting the coordinates of the bounding boxes.

In order to use this state-of-the-art model, M. Yang has also pretrained the DetTransNet using the ImageNet and JFT dataset, which usually improve the performance of the transformers. Furthermore, to test it they are going to use the COCO dataset, a set of images launched in 2017 specifically for object detection consisting of 10000 images and more than 20 categories.

It is important to note that for the COCO dataset, there are well-defined custom metrics such as the Average Precision(AP). AP is a metric used to evaluate the performance of object detection algorithms. It represents the average precision at different recall levels and is calculated by first calculating the precision and recall at different intersection over union (IoU) thresholds and then averaging these values. AP is typically used to compare different object detection algorithms and is commonly reported at different IoU[5] thresholds such as 0.5, 0.75, and 0.9. A higher AP value indicates better performance of the object detection algorithm, see the paper of M. Everingham, et al. [3](Jordi) for further explanation on metrics used.

3.3 Methodology used for image generation and style transfer

A transformer model applied to the problem of image generation and style transfer is proposed in [1] (Leonardo) with the StyTr^2 model. To utilize the ability of transformers to capture the long-range dependencies of image features for style transfer, the problem is framed as a sequential patch generation task. Content-aware positional encoding (CAPE), which is scale-invariant and better suited for style transfer tasks, is introduced. Unlike sinusoidal positional encoding (PE), which only takes into account the relative distance between patches, CAPE takes into account the meaning of image content.

The encoder captures the long-range dependencies of image patches using a transformer-based structure to learn sequential visual representations. Unlike other vision tasks, the input for the style transfer task comes from two different domains: natural images and artistic paintings. To handle this, StyTr^2 has two transformer encoders to encode domain-specific features, which are used to translate a sequence from one domain to another in the following stage.

Rather than directly up-sampling the output sequence to create the final results, a three-layer CNN decoder to refine the outputs of the transformer decoder was employed, as suggested in [11]

In order to assess the performance of the model and optimize it, different loss functions are considered. The content perpetual loss $\mathcal{L}_c$, the style perceptual loss $\mathcal{L}_s$, and two identity loss terms, $\mathcal{L}_{id1}$ and $\mathcal{L}_{id2}$. These are defined as follows

$$\mathcal{L}_c = \frac{1}{N_l}\sum_{i=0}^{N_l} \|\phi_i(I_0) - \phi_i(I_c)\|_2$$
$$\mathcal{L}_s = \frac{1}{N_l}\sum_{i=0}^{N_l} \|\mu(\phi_i(I_0)) - \mu(\phi_i(I_s))\|_2 + \|\sigma(\phi_i(I_0)) - \sigma(\phi_i(I_s))\|_2$$
$$\mathcal{L}_{id1} = \|I_{cc} - I_c\|_2 + \|I_{ss} - I_s\|_2$$
$$\mathcal{L}_{id2} = \frac{1}{N_l}\sum_{i=0}^{N_l} \|\phi_i(I_{cc}) - \phi_i(I_c)\|_2 + \|\phi_i(I_{ss}) - \phi_i(I_s)\|_2$$

The network then minimizes the following loss function

[5]Intersection over Union (IoU) is a measure used to evaluate the overlap between two bounding boxes in object detection. It is calculated as the ratio of the intersection of the bounding boxes to the union of the bounding boxes.

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s + \lambda_{id1} \mathcal{L}_{id1} + \lambda_{id2} \mathcal{L}_{id2}$$

4 Results and Discussion

In this section, we present the results of the experiments using visual transformers on a variety of tasks and data sets. We compare the results with those obtained using other state-of-the-art architectures and discuss the implications of our findings. Our goal is to provide insights into the capabilities and limitations of visual transformers and how they compare to other approaches in the field.

4.1 Experiments for classification

For classification, as we early introduced in the section 3.1, an exhaustive work has been performed to conclude if visual transformers might be the new SOTA for image classification in many of the most important and used image datasets of nowadays. In the table 4.1, we can easily see how Big Transfer and Noisy Student underperform the better results - marked in bold- which are from transformers.

Obviously, the best of them come from the heaviest transformer, although the huge transformer version only uses approximately a 25% of the computational power compared to CNNs[6]. But not only the ViT-H outperforms BiT-L, we can also see how ViT-L, which uses less than 10% power compared to its adversary, outperforms it in all tasks, which is simply amazing. Yet Noisy Student keep performing better than ViT-L in the ImageNet dataset, but using quasi 20 times more computational resources.

Interesting conclusions that this paper conclude and we would like to remark are:

1. Visual Transformers, in general, have much less image-specific inductive bias than CNNs. While CNNs incorporate locality, two-dimensional neighborhood structure, and translation equivariance into every layer, in ViT, only the multi-layer perceptron (MLP) layers exhibit these properties. The self-attention layers in ViT, on the other hand, are global in nature.

   Additionally, ViT relies on the two-dimensional neighborhood structure only at the beginning of the model and during fine-tuning for adjusting the position embeddings for images of different resolutions. Otherwise, the position embeddings at initialization time do not contain any information about the 2D positions of the image patches, and all spatial relations between the patches have to be learned from scratch.

2. Visual Transformer models pre-trained on larger datasets outperformed those pre-trained on smaller datasets in image classification tasks. When regularized, the ViT-Large models performed similarly to the ViT-Base models when pre-trained on ImageNet-21k, but showed a significant improvement when pre-trained on JFT-300M. These results suggest that pre-training on larger datasets is important for the performance of ViT models in image classification tasks.

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55 ± 0.04**	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.5
ImageNet ReaL	**90.72 ± 0.05**	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50 ± 0.06**	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	
CIFAR-100	9**4.55 ± 0.04**	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	
Oxford-IIIT Pets	**97.56 ± 0.03**	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	
Oxford Flowers-102	99.68 ± 0.02	**99.74 ± 0.00**	99.61 ± 0.02	99.63 ± 0.03	
VTAB (19 tasks)	**77.63 ± 0.23**	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Table 2: Comparison of different models on various benchmarks

[6]The last row indicates the computational power in terms of hardware per days. The number represents the TPU v3 cores (2 per chip) used for training multiplied by the training time in day

4.2 Experiments for object detection

In this section, we compare the performance of DetTransNet, a visual transformer model adapted for object detection, with previous state-of-the-art models such as R-CNN and YOLO. It is important to note that the field of object detection is constantly evolving, with newer models such as YOLO-v7 being released in recent months.

However, we may observe how a Visual Transformer properly adapted and pretrained has outperformed again CNNs, and in some cases for a substantial quantity, see table 5. There is non doubt that with DetTransNet state-of-the-art results are achieved, demonstrating the efficiency and effectiveness of the model.

Not only did they discover a new use-case for transformers, but they also propose an improvement to the typical ViT we have seen in classification, that is to overlap image patches instead of dividing them, clearly explained in the image 2.

Method	AP	AP small	AP medium	AP large
R-CNN	38.0	17.5	40.8	56.1
Fast R-CNN	39.8	18.4	42.3	58.7
Faster R-CNN	42.0	20.5	45.8	61.1
YOLO	41.9	19.6	45.6	60.8
DETR	42.0	20.5	45.8	61.1
De-DETR	43.8	21.2	46.5	61.9
DetTransNet	**45.4**	**22.5**	**47.8**	**62.1**

Table 3: Average precision values for various object detection methods

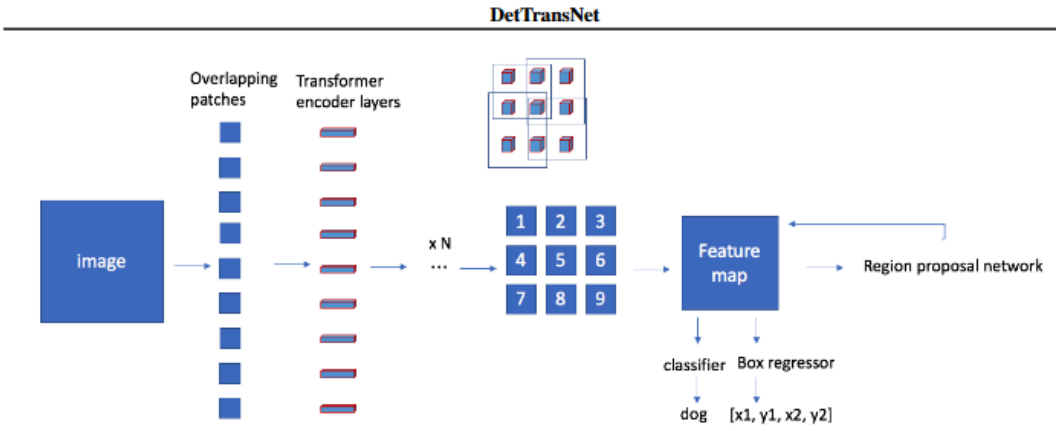

Figure 2: DetTransNet architecture

4.3 Experiments for image generation and style transfer

Transformers models have recently been applied also to problems of image generation and style transfer. Previously, this type of problem has been tackled using non-parametric algorithms and convolutional neural networks. The first makes use of resampling techniques over the pixels of a given source texture, while the second takes care of the main limitation of the first, which is using only low-level image features, and allows for the separation of the representations of content and style; a thorough study of the advantages of convolutional neural networks applied to this problem is presented in [4] (Leonardo)

An attempt in applying a transformer model to this problem and comparing its performance to other state-of-the-art models has been presented in [1]. (Leonardo) In order to assess the performance

of different models, an initial comparison of the average inference time of different models over different resolutions is considered.

Resolution	StyTr^2	StyleFormer	IEST	AdaAttN	ArtFlow	MCC	MAST	AAMS	SANet	Avatar	AdaIN
256×256	0.116	0.013	0.065	0.104	0.142	0.013	0.030	2.074	0.015	0.260	0.007
512×512	0.661	0.026	0.092	0.213	0.418	0.015	0.096	2.173	0.019	0.470	0.008

Table 4: Average inference time (in seconds) of different methods at two output resolutions.

Furthermore, a quantitative comparison considering the average content loss and style loss produced the following results.

	StyTr^2	StyleFormer	IEST	AdaAttN	ArtFlow	MCC	MAST	AAMS	SANet	Avatar	AdaIN
$\mathcal{L}_c \downarrow$	**1.91**	2.86	<u>1.97</u>	2.29	2.13	2.38	2.46	2.44	2.44	2.84	2.34
$\mathcal{L}_s \downarrow$	<u>1.47</u>	2.91	3.47	2.45	3.08	1.56	1.55	3.18	**1.18**	2.86	1.91

Table 5: Quantitative comparisons. We compute the average content and style loss values of results by different methods to measure how well the input content and style are preserved.The best results are in bold while the second-best results are marked with an underline.

In conclusion, the proposed StyTr^2 model allows for a content transformer encoder and a style transformer encoder to be used to capture domain-specific long-range information. StyTr^2 addresses the issue of content leak in CNN-based models and offers a new perspective on the difficult problem of style transfer. Currently, the test-time speed of the proposed method is not as fast as some CNN-based approaches. It would be interesting to explore ways to incorporate some of the efficiency of CNNs to speed up computation in the future.

5 Conclusion

In conclusion, the experiments we have seen with Visual Transformers have shown that they are a powerful tool for various tasks in the field of computer vision. In the task of image classification, we have seen that ViT models, when properly regularized and pre-trained on large datasets, can achieve state-of-the-art results.
In the task of object detection, we have demonstrated that the proposed DetTransNet architecture, which builds upon the ViT model and includes overlapping patches and a region proposal network, is able to outperform previous state-of-the-art CNN-based approaches.
Additionally, we have explored the use of Visual Transformers for image generation and style transfer, and have found them to be a promising direction for future research. Specifically, the proposed StyTr^2 model obtains good results in terms of optimization of the loss function but leaves space for improvements on the average inference time. Also, further work on the robustness of image style transfer by implementing transformers and new techniques shows promising results and improvements. [8] (Leonardo)
Overall, our work highlights the versatility and effectiveness of Visual Transformers in the field of computer vision, and we believe they will continue to be valuable tools in various tasks and applications.

References

[1] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11326–11336, 2022.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, David Weissenborn, Xin Zhai, Thomas Unterthiner, Mohammad Dehghani, Markus Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[3] M. Everingham and et al. The pascal visual object classes (voc) challenge. *International Journal of Computer*, 88:303–338, 2010.

[4] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[5] Karttikeya Mangalam Yanghao Li Zhicheng Yan Jitendra Malik Christoph Feichtenhofer Haoqi Fan, Bo Xiong. Multiscale vision transformers. pages 6824–6835, 2021.

[6] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. pages 33–62, 2022.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Liam Jones, Antonio N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

[8] Pei Wang, Yijun Li, and Nuno Vasconcelos. Rethinking and improving the robustness of image style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 124–133, 2021.

[9] Yifan Xu, Huapeng Wei, Minxuan Lin, Yingying Deng, and Mengdan Zhang Sheng, Kekai, Fan Tang, Weiming Dong, Feiyue Huang, and Changsheng Xu. Transformers in computational visual media : A survey. pages 33–62, 2022.

[10] M. Yang. A visual transformer for object detection. In *International Conference on Learning Representations*, 2022.

[11] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.