Learning Nuclei Representations with Masked Image Modelling

Piotr Wójcik[1,2], Hussein Naji[1,2], Adrian Simon[4], Reinhard Büttner[4], and Katarzyna Bożek[1,2,3]

[1] Institute for Biomedical Informatics, Faculty of Medicine and University Hospital Cologne, University of Cologne, Germany
[2] Center for Molecular Medicine Cologne (CMMC), Faculty of Medicine and University Hospital Cologne, University of Cologne, Germany
[3] Cologne Excellence Cluster on Cellular Stress Responses in Aging-Associated Diseases (CECAD), University of Cologne, Germany
[4] Institute of Pathology, University Hospital Cologne, Germany

Abstract. Masked image modelling (MIM) is a powerful self-supervised representation learning paradigm, whose potential has not been widely demonstrated in medical image analysis. In this work, we show the capacity of MIM to capture rich semantic representations of Haemotoxylin & Eosin (H&E)-stained images at the nuclear level. Inspired by Bidirectional Encoder representation from Image Transformers (BEiT) [1], we split the images into smaller patches and generate corresponding discrete visual tokens. In addition to the regular grid-based patches, typically used in visual Transformers, we introduce patches of individual cell nuclei. We propose positional encoding of the irregular distribution of these structures within an image. We pre-train the model in a self-supervised manner on H&E-stained whole-slide images of diffuse large B-cell lymphoma, where cell nuclei have been segmented. The pre-training objective is to recover the original discrete visual tokens of the masked image on the one hand, and to reconstruct the visual tokens of the masked object instances on the other. Coupling these two pre-training tasks allows us to build powerful, context-aware representations of nuclei. Our model generalizes well and can be fine-tuned on downstream classification tasks, achieving improved cell classification accuracy on PanNuke dataset by more than 5% compared to current instance segmentation methods.

Keywords: nuclei classification · masked image modelling · self-supervised learning.

1 Introduction

In recent years, Transformer architectures [15] have demonstrated the capability to achieve competitive results in computer vision [5]. However, their data-hungry training paradigm hinders many potential applications, especially in the field of digital pathology, where annotated data is limited. Several self-supervised solutions have been proposed to tackle this problem and proved to be very efficient

in building good representations of visual data based on Transformers (e.g. [2]). The work of H. Bao *et al.* [1] helped to bridge the gap between training paradigms used in natural language processing, e.g. in BERT [4], and computer vision tasks. BERT randomly masks some word tokens and then reconstructs the missing tokens with the use of encoded representations of visible portions of the text. The fundamental difficulty in reproducing this pre-training schema for images is the magnitude of potential vocabulary for even small image patches (e.g. 16×16 pixels). At the same time, large portion of the low-frequency visual detail is spurious for semantic decoding of an image. Bao et al. [1] addressed this problem by using a semantic-aware discrete image tokenizer [12] which opened possibilities for efficient self-supervised learning of high-level, context-aware visual features via Transformers.

Concurrently, many approaches have been proposed to address a fundamental task in digital pathology – segmentation and classification of nuclei. These approaches can roughly be categorised according to their backbone into Convolutional Neural Network (CNN)-based [8,7] and, more recently, Transformer-based [14]. For the task of nuclei segmentation, Generative Adversarial Network was proposed as a method for synthesizing images with labels [10]. Despite promising advances in detection, classification of nuclei into various cell types is still considered as a by-product of instance segmentation and needs to be improved. Moreover, CNN-based methods are inherently limited by the locality of the view. This limitation poses a problem for pathological analysis, since the shape, size and density are not the sole indicators of a cell being cancerous. For example, in the diagnosis of diffuse large B-cell lymphoma (DLBCL), a highly heterogeneous disease both morphologically and clinically, neighboring cells as well as the subtle relationships between nucleus and cytoplasm are important to distinguish between healthy large leukocytes and lymphomatic cells. It is worth noting that several Transformer-based solutions have been proposed for learning good slide-level representations, with remarkable results achieved by R. Chen *et al.* [3], albeit no research known to us directly addresses the challenge of building context-aware representation of already segmented nuclei.

Motivated by this need, we introduce a method for learning segmented nuclei representation which leverages the power of BEiT. Since its architecture is instance-agnostic, i.e. divides the image into a fixed number of regular grid-patches, we adopt the strategy presented in the work of S. Kim *et al.* [11]. Namely, we incorporate instance-level patches of varying sizes, containing separate cell nuclei, into the sequence of patches fed to the backbone Transformer. We demonstrate that the representations of nuclei indeed reflect their respective cell types and require small amount of fine-tuning to achieve improved classification results compared to current state-of-the-art segmentation and classification methods.
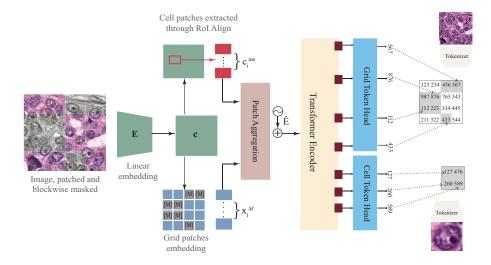
Fig. 1. Pipeline for self-supervised pre-training. We follow closely training schema proposed in [1], which converts an image into discrete tokens. In addition to patches obtained by dividing an image into a regular grid we also tokenize parts of the image inside bounding boxes B_i, where each box encloses a single nucleus. We randomly mask a portion of an image, replacing masked patches (gray squares) with a special token `[M]`. We then embed image patches into a feature map **c**. The same grid **c** is used to extract representations of nuclei through RoI Align [9]. Both grid patches and cell patches (represented by blue and red boxes, respectively) are fed into a vision Transformer backbone after aggregation. The pretext task aims at predicting discrete tokens for both input image and instance patches.

2 Method

2.1 Image Representation

As input, we are given an image $\boldsymbol{x}$ and bounding boxes of individual nuclei $B_i \in \mathcal{B}$, described by parameters $B_i = [x_{\min}^i, y_{\min}^i, x_{\max}^i, y_{\max}^i]$, i.e. by the coordinates of the upper-left and lower-right corners and $i \in 1, \ldots, N$ where N is the total number of bounding boxes contained within an image $\boldsymbol{x}$. We apply the standard protocol for vision Transformers [5] which involves splitting a 2D image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times C}$ into a sequence of grid patches $\boldsymbol{x}^p \in \mathbb{R}^{M \times (P^2 C)}$, where C stands for the number of channels, (H, W) represents the input size, (P, P) is the size of a single patch and $L = HW/P^2$. In this work we run all experiments with 3-channel images of size 448×448, divided into 28×28 square patches.

Grid Patch Embedding and Masking After obtaining grid patches, we use linear projection $\boldsymbol{E}\boldsymbol{x}^p$ (as depicted in Fig. 1) to embed grid patches into D-dimensional feature map $\boldsymbol{c}$, where $\boldsymbol{E} \in \mathbb{R}^{(P^2 C) \times D}$. Next, we randomly mask approximately

40% of patches, employing blockwise masking described in detail in [1]. Indices of patches chosen for masking are denoted by $\mathcal{M}$. Then, we replace grid patch embeddings with a shared learnable token $\boldsymbol{e}_{[\mathcal{M}]}$ of a size $1 \times 1 \times D$: $\boldsymbol{e}_i^p \colon \boldsymbol{x}_i^{\mathcal{M}} = \delta(i \in \mathcal{M}) \odot \boldsymbol{e}_{[\mathcal{M}]} + (1 - \delta(i \in \mathcal{M})) \odot \boldsymbol{x}_i^p$, where $\delta(\cdot)$ is the indicator function. The choice of the masking algorithm and the ratio was motivated by ablation studies in [1]. Specifically, the use of the blockwise masking algorithm was shown to improve the model's accuracy compared to selecting patches for masking independently at random.

Cell Patch Embedding So far, we closely followed steps described in [1]. To encode not only the overall image structure but also the key semantic elements withing it - cells, we introduce the following modification to the baseline architecture inspired by the work [11]. Namely, given bounding boxes B_i and the feature map $\boldsymbol{c}$ (see Fig. 1), we extract cell instance features through RoI Align module [9]:

$$\boldsymbol{c}_i^{\text{ins}} = \text{RoIAlign}(\boldsymbol{c}; B_i) \in \mathbb{R}^{k \times k \times D} \tag{1}$$

where $k \times k$ is the size of extracted features. In our work we set $k = 3$. Note that some cell instance features may be sampled from masked patches if their bounding boxes intersect with a masked portion of the image. Subsequently, we process the $\boldsymbol{c}_i^{\text{ins}}$ to obtain embedding vectors of dimension $1 \times 1 \times D$. We achieve this by applying a convolution on $\boldsymbol{c}_i^{\text{ins}}$, as proposed in [11]:

$$\boldsymbol{p}_i^{\text{ins}} = \text{Conv}(\boldsymbol{c}_i^{\text{ins}}) \in \mathbb{R}^{1 \times 1 \times D} \tag{2}$$

Patch Aggregation In our architecture, we concatenate a sequence of grid patch embeddings $\{\boldsymbol{c}_i^{\mathcal{M}}\}_{i=1}^L$ (blue squares in Fig. 1) and a sequence of N cell patch embeddings $\{\boldsymbol{p}_i^{\text{ins}}\}_{i=1}^N$. Following [1], we prepend a special, learnable token [CLS] to the concatenated sequence. Since the number of nuclei in a particular image may vary, we pad cell embedding sequences to a maximum length using [PAD] token. Overall, in this module we use three special tokens ([M], [PAD], [CLS]), each of them is shared, learnable and has size $1 \times 1 \times D$. After aggregation, the input sequence to the positional encoding module has the following form:

$$\boldsymbol{H}^0 = [\texttt{[CLS]}, \boldsymbol{c}_1^{\mathcal{M}}, \dots, \boldsymbol{c}_L^{\mathcal{M}}, \underbrace{\boldsymbol{p}_1^{\text{ins}}, \dots, \boldsymbol{p}_N^{\text{ins}}, \texttt{[PAD]}, \dots, \texttt{[PAD]}}_{\text{fixed maximum length}}] \tag{3}$$

2.2   Positional Encoding

To encode the information about the positions of patches (both grid and cell), we use the technique proposed in [11]. The position and the shape of any patch, regardless of its regular grid or a bounding box identity, can be represented by center coordinates (x, y) and patch width w and height h, as exemplified in Fig. 2. We use a sinusoidal mapping $\gamma \colon \mathbb{R} \to \mathbb{R}^{D/4}$ defined as $\gamma(t) := [\sin(2^0 \pi t), \cos(2^0 \pi t), \dots, \sin(2^{D/8-1} \pi t), \cos(2^{D/8-1} \pi t)]$. For every patch token, positional encoding is constructed by concatenating all spatial information embedded by γ, namely $\hat{\boldsymbol{E}}_i = [\gamma(x_i), \gamma(y_i), \gamma(w_i), \gamma(h_i)]$. Positional encoding
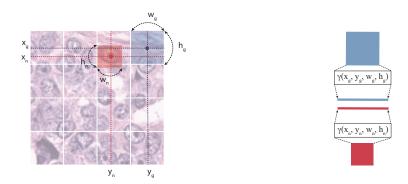
Fig. 2. Building positional encoding for both grid and cell patches. A tuple (x_g, y_g, h_g, w_g) describes center coordinates, width and height of a grid patch (blue), while (x_n, y_n, h_n, w_n) describes the position and shape of a cell patch (red).

is further added to the corresponding token embedding: $\boldsymbol{H}_i^0 + \hat{\boldsymbol{E}}_i$. We handle [CLS] token separately, with a learnable positional encoding of shape $1 \times 1 \times D$.

2.3 Discrete Tokenizer

The role of discrete tokenizer is to map an image to discrete tokens, *i.e.* natural numbers. Formally, an image $\boldsymbol{x}$ is transformed into a grid $\boldsymbol{z} = [z_i, \ldots, z_T] \in \mathcal{V}^{(H/R) \times (W/R)}$, where $\mathcal{V}$ denotes the discrete vocabulary, R is the resolution of the discrete tokenizer. In our work, we use pre-trained weights of a publicly available dVAE (`https://github.com/openai/DALL-E`) with $|\mathcal{V}| = 8192$. For the details of dVAE implementation and pre-training, please refer to [13]. We tokenize the input image to into 28×28 grid. We then crop each nucleus instance to its bounding box and resize obtained crops to a size 32×32. Nuclei views are subsequently tokenized into 4×4 grids.

2.4 Backbone Transformer Encoder

Our encoder shares all architectural details with ViT [5], except for additional attention masking for preventing attending to [PAD] tokens, which we implement according to BERT [4]. Embedded patches with positional encoding ($\boldsymbol{H}^0 + \hat{\boldsymbol{E}}$) are fed into 12 layers of Transformer blocks with 12 attention heads.

$$\boldsymbol{H}^{12} = [\boldsymbol{h}_{\texttt{[CLS]}}, \underbrace{\boldsymbol{h}_1^g, \ldots, \boldsymbol{h}_{28 \times 28}^g}_{\text{grid representations}}, \overbrace{\boldsymbol{h}_1^n, \ldots, \boldsymbol{h}_N^n}^{\text{nuclei representations}}, \boldsymbol{h}_{\texttt{[PAD]}}, \ldots, \boldsymbol{h}_{\texttt{[PAD]}}]. \qquad (4)$$

After discarding the representation of [CLS] and [PAD] tokens, $\boldsymbol{h}_{\texttt{[CLS]}}$ and $\boldsymbol{h}_{\texttt{[PAD]}}$ respectively, we are left with 28×28 representations of regular grid patches $\boldsymbol{h}_i^g$ and N representations of nuclei $\boldsymbol{h}_j^n$.
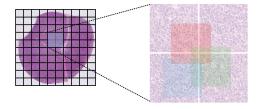
Fig. 3. Datasets augmentation strategy. **Left**: Self-supervised pre-training dataset consists of one H&E-stained WSI of a lymph node, divided into tiles of size 512×512. For each four adjacent tiles, we randomly generate overlapping crops. **Right**: To evaluate our model on the CoNSeP dataset, we divide 1000×1000 pixel-large images into four overlapping tiles of 448×448. Each nucleus is classified basing on its representation from one tile only.

Pre-training Loss Function We use two fully-connected layers as output heads (see Fig. 1). The first one, image head, predicts discrete tokens from the representation of every masked grid patch $\{\boldsymbol{h}_i^g : i \in \mathcal{M}\}$. Since the input image was tokenized into 28×28 natural numbers, there is one-to-one correspondence between grid patches and tokens. The second, cell instance head, predicts 4 tokens from $\boldsymbol{h}_j^n$, for each cell instance whose bounding box B_j intersects with masked portion of the image. The training MIM loss is defined as:

$$\mathcal{L}_{\text{MIM}} = - \sum_{(\boldsymbol{x}, \mathcal{B}) \in \mathcal{D}} \left(\overbrace{\sum_{i \in \mathcal{M}} \log(p(z_i^g | \boldsymbol{c_i^{\mathcal{M}}}))}^{\text{BEiT loss}} + \overbrace{\sum_{j \in \mathcal{B}_{\mathcal{M}}} \sum_{k=1}^{4} \log(p(z_{j,k}^n | \boldsymbol{p_j^{\text{ins}}}))}^{\mathcal{L}_{\text{inst}}} \right) \tag{5}$$

where $\mathcal{D}$ denotes pre-training set of images with bounding boxes, $\mathcal{B}_{\mathcal{M}}$ indicates the set $\mathcal{B}$ of bounding boxes that intersect with masked grid patches, z_i^g the discrete token for the i-th grid patch and $z_{j,1}^n, \ldots, z_{j,4}^n$ discrete tokens for j-th nuclei instance. Notice, that the first component of the loss function comes directly from [1] while the second, dubbed $\mathcal{L}_{\text{inst}}$, is proposed by us in order to sharpen the view on nuclei.

3   Experiments

Datasets We used three datasets in our work. For self-supervised pre-training, we used our in-house dataset of $37\,665$ H&E images (40x magnification) obtained from a single WSI of a DLBCL lymph node. Each 512×512 tile was segmented with a bounding box for each nucleus, albeit no cell labels were provided. The segmentation was performed automatically, without further verification. We resized every tile to 448×448 and randomly cropped additional tiles to increase

the number of training examples, as demonstrated in Fig. 3. After preprocessing, DLBCL dataset consists of 160 545 image tiles with segmented nuclei. For fine-tuning and testing we use CoNSeP and PanNuke datasets [8,6]. The CoNSeP dataset consists of 24 319 annotated nuclei from 41 H&E images. Nuclei are grouped into four types. The PanNuke dataset [6] contains 205 343 annotated nuclei of five types. Images in both datasets have size of 256×256 and originate from 19 different tissue types. We did not perform any pre-processing on these images except for resizing them to 448×448.

Pre-Training Setup The proposed network was pre-trained with similar set of parameters as BEiT [1]. We used 12-layer Transformer with 768 hidden size and 12 attention heads. We pre-trained the model for 800 epochs with a batch size 96. The codebase used for experiments was that of the original BEiT implementation `https://github.com/microsoft/unilm/tree/master/beit`. Instance embedding code was built upon InstaFormer implementation `https://github.com/KU-CVLAB/InstaFormer`. Around 550-th epoch, we added PanNuke Fold 1 dataset and CoNSeP training dataset to the initial DLBCL dataset to boost performance. Throughout the pre-training phase, we applied the same set of augmentations as in [1], which included RandomResizeAndCrop, color jittering with a parameter of 0.4 and RandomFlip. Additionally, all input images were normalized to the mean and standard deviation of ImageNet.

Fine-tuning on Annotated Datasets We fine-tuned our model and validated its performance on the two labelled datasets, CoNSeP and PanNuke. In nuclei classification task, we discarded two MIM pre-training linear layers and used softmax classifier on nuclei representations: $\text{softmax}(\{\boldsymbol{h}_i^n\}_{i=1}^N \boldsymbol{W}_C)$, where $\boldsymbol{h}_i^n$ stands for i-th nuclei representation, C is the number of nuclei classes and $\boldsymbol{W}_C$ denotes a linear layer parameter matrix.

Evaluation Protocol As a baseline model, we chose the widely used state-of-art nuclei detection and classification model HoVer-Net [8]. Since the authors only provided metrics relevant to the combined tasks of nuclei detection and classification, for a fair comparison, we modified the codebase to print both accuracy and F1 score for the subset of nuclei that were correctly segmented. Results presented in Tab. 1 were obtained with a use of published pre-trained weights `https://github.com/vqdang/hover_net` for the CoNSeP and PanNuke datasets. Subsequently, we ran evaluation of our model on the exact shapes of correctly detected nuclei.

Ablation Studies on $\mathcal{L}_{inst}$ We performed ablation studies to test the importance of $\mathcal{L}_{\text{inst}}$. We pre-trained two variants of the model (with and w/o $\mathcal{L}_{\text{inst}}$) for 250 epochs on the DLBCL dataset and compared linear probing results on Fold 3 of PanNuke dataset. We kept the models frozen and added a linear head to evaluate the self-supervised models [1]. The model pre-trained using the full $\mathcal{L}_{\text{MIM}}$ loss function achieved a significantly higher accuracy of 0.613 compared to the model pre-trained with only the BEiT component of the loss function, which achieved an accuracy of 0.517.
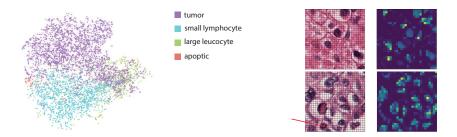
Fig. 4. Left: t-SNE visualisation of DLBCL nuclei representations without fine-tuning on any labeled dataset. **Right**: Upper row shows self-attention map for `[CLS]` token; lower row shows self-attention map for a reference point indicated with an arrow.

4 Results and Conclusion

As demonstrated in Tab. 1, our model visibly outperforms HoVerNet in the task of nuclei classification. The difference is especially pronounced on the PanNuke dataset (0.05 on average) for all dataset which is much larger than CoNSeP. We find striking the capacity of our model to generalize to images from a pathological domain completely different from the DLBCL corpus used for pre-training. Fig. 4 shows the ability of self-supervised training to separate cells of different types while self-attention map demonstrates that our model learns long-distance relations between nuclei. Although our proposed training method requires a non-negligible amount of segmented images, these can be obtained automatically at a low cost. It should be also noted that MIM methods achieve competitive results on image segmentation [1], offering new possibilities for performing both tasks of nuclei detection and segmentation simultaneously in the future.

Table 1. Nuclei classification results on PanNuke dataset (Fold 1 - Fold 3 split) and CoNSeP. HoVerNet (HN) is used as a baseline model.

	PanNuke					
	All	**Neo.**	**Inflam.**	**Conn.**	**Dead**	**Epith.**
HN F1	0.780	0.856	0.714	0.719	0.399	0.765
Ours F1	**0.831**	**0.893**	**0.776**	**0.764**	**0.571**	**0.848**
HN Acc	0.780	0.748	0.555	0.562	0.249	0.619
Ours Acc	**0.832**	**0.807**	**0.634**	**0.618**	**0.400**	**0.735**
	CoNSeP					
	All	**Misc.**	**Inflam.**	**Epith.**	**Spindle**	
HN F1	0.872	0.734	0.817	0.939	0.856	
Ours F1	**0.885**	**0.804**	**0.822**	**0.954**	**0.864**	
HN Acc	0.872	0.580	0.690	0.884	0.749	
Ours Acc	**0.885**	**0.672**	**0.698**	**0.911**	**0.761**	

References

1. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: BERT pre-training of image transformers. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net (2022)

2. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021. pp. 9630–9640. IEEE (2021)

3. Chen, R.J., Chen, C., Li, Y., Chen, T.Y., Trister, A.D., Krishnan, R.G., Mahmood, F.: Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 16123–16134. IEEE (2022)

4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics (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: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021)

6. Gamper, J., Koohbanani, N.A., Graham, S., Jahanifar, M., Khurram, S.A., Azam, A., Hewitt, K., Rajpoot, N.: Pannuke dataset extension, insights and baselines. arXiv preprint arXiv:2003.10778 (2020)

7. Graham, S., Epstein, D.B.A., Rajpoot, N.M.: Dense steerable filter cnns for exploiting rotational symmetry in histology images. IEEE Trans. Medical Imaging **39**(12), 4124–4136 (2020)

8. Graham, S., Vu, Q.D., e Ahmed Raza, S., Azam, A., Tsang, Y., Kwak, J.T., Rajpoot, N.M.: Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. Medical Image Anal. **58** (2019)

9. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. IEEE Trans. Pattern Anal. Mach. Intell. **42**(2), 386–397 (2020)

10. Hou, L., Agarwal, A., Samaras, D., Kurç, T.M., Gupta, R.R., Saltz, J.H.: Robust histopathology image analysis: To label or to synthesize? In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 8533–8542. Computer Vision Foundation / IEEE (2019)

11. Kim, S., Baek, J., Park, J., Kim, G., Kim, S.: Instaformer: Instance-aware image-to-image translation with transformer. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 18300–18310. IEEE (2022)

12. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 8821–8831. PMLR (2021)

13. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 8821–8831. PMLR (2021)
14. Tang, Y., Zhang, N., Wang, Y., He, S., Han, M., Xiao, J., Lin, R.: Accurate and robust lesion RECIST diameter prediction and segmentation with transformers. In: Wang, L., Dou, Q., Fletcher, P.T., Speidel, S., Li, S. (eds.) Medical Image Computing and Computer Assisted Intervention - MICCAI 2022 - 25th International Conference, Singapore, September 18-22, 2022, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 13434, pp. 535–544. Springer (2022)
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 5998–6008 (2017)