Interpretability of self-attention maps in self-supervised representation learning

Yanran Zhang
Chair for Data Processing, Technical University of Munich
yanran.zhang@tum.de

***Abstract*—The development of Transformer[1] in the area of image processing and Computer Vision was facilitated by the Vision Transformer(ViT) proposal[2], due to their capacity to contextualize information via methods of co-attention and self-attention mechanisms. It is generally assumed that attention can be used to detect information that models considered essential because attention layers explicitly weight the representations of input components. To discover the information that models considered important, we can visualize the self-attention maps for various heads from the last layer in ViT. In fact, the attention map is assumed as a method to explain the predictions. However, utilizing attention as the interpretability of model results is still highly dubious. In order to properly provide visual guidance regarding which important factor is playing a greater role in the decision-making of the corresponding task, an ideal interpretable attention model should not only be able to quantify the importance level of a particular part of the neural network but also be capable of identifying precise local regions of the input. Therefore, we will explore the factors that influence the attention map. in this study, we will investigate the effects of image resolution, data augmentation, and multi-crop training on self-attention maps using self-supervised representation learning. We'll also suggest some potential future study directions at the end.**

Keywords*—*self-supervised learning, representation leanring, self-attention map, interpretability

I. Introduction

Recently, Transformer, and specifically attention mechanisms, are becoming increasingly popular in Computer Vision field. The proposal of Vision Transformer(ViT)[2] makes Transformer widely used in image processing. In supervised learning, ViT is applied for image classification and achieve better results than other traditional architecture, such as ResNet and VGG. However, the success of ViT highly depends on a large size of dataset, which means training a ViT model in supervised learning requires extensive labeling work that is not always possible in the real world. In the previous study, self-supervised learning has shown amazing results in learning visual representation from unlabeled image[3][4][5][6][7][8]. Therefore, current state of the art methods combines Transformer with self-supervised learning and trains on the unlabeled dataset in order to get a better performance[9]. For example, Sara Atito et al. proposed SiT[10]. In this approach, several parts of the input image are corrupted, based on the context from the whole visual field, the corrupted part can be recovered. In advance, Facebook AI researchers raised a new method called DINO: Self-Distillation with no labels[3]. Through this paper, this approach has achieved an excellent performance on other downstream tasks by data augmentation and self-distillation, especially on image segmentation's task, which is one of the biggest challenge in Computer Vision. By visualizing the self-attention maps on the heads of last layer, we can see the model automatically learns class-specific features leading to unsupervised object segmentation, the output maps obtained from DINO even more correct and clear than that obtained with other supervised learning methods. It requires the model to be able to fully understand what is in the image.

Attention layers explicitly weight input components' representations. It's often assumed that attention can be used to identify information that models found important. Thanks to the development of deep learning, many applications enjoyed a big leap in performance in the last decade. Nevertheless, these deep learning based methods are commonly considered as 'black box' approaches, where the internal functioning is unclear. As one of the solutions to remedy this lack of interpretability, attention mechanism are frequently employed for explaining the deep model. However, in fact, attention mechanism itself is still a black box. Whether it can use attention maps to explain the model is still a research hotspot[11][12][13].The goal of explanation is often to determine what inputs are the most relevant to the prediction. An ideal interpretable attention model should not only be able to quantify the importance level of a particular part of the neural network but also be capable of identifying precise local regions of the input.

In general, whether the attention layer is a suitable interpretability model, we need to explore the main factors influencing it to obtain high-quality human-level interpretable attention maps. As mentioned before, DINO has an excellent performance on color image segmentation by visualizing the self-attention maps on the last layer. Therefore, in this study, in place of color images, we utilize binary images to examine the impact of various components on the self-attention mechanism with self-supervised representation learning and to determine whether DINO is as effective in segmenting binary images.

II. Related Work

A. DINO

DINO algorithm is proposed from Facebook AI researchers, they wondered whether the success of the Transformers in Computer Vision stemmed from supervised training and

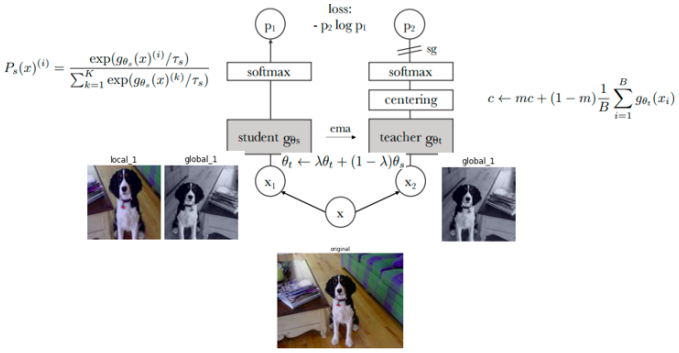

Fig. 1. DINO: Self-Distillation with No label

whether there was a way to build a self-supervised system that could be trained on unlabeled datasets. In this case, the approach chosen by the researchers was to use self-Distillation. Two networks as backbone with the same architecture but different parameters, one defined as a student and one as a teacher. These two networks will take as input two representations of the same image. In particular, for each image in the training set, a multi-crops augmentation is applied to extract two sets of images from it. Two patches of great dimensions and partially overlapped are obtained, able to give a global idea of the image in consideration, and a series of other smaller patches that will give instead a local representation of the image. Compare with knowledge-Distillation, the self-distillation is used in a quite different way, the teacher is not created prior but is trained by the student with exponential moving average (EMA). During the student training, a bit of information learned is propagated to the teacher which gradually learns from the views seen by the student but it has to perform classification based only on global views given to it. The centering step in teacher branch is used for avoid collapse. The architecture of DINO is shown in Figure 1.

B. Loss function

Contrastive learning algorithm aim to learn representations by enforcing similar elements to be equal and dissimilar elements to be different. In general, Noise Contrastive Estimator (NCE)loss and cross-entropy loss are two commonly loss function in contrastive learning. In fact, they are similar to each other.

Cross-entropy loss In DINO, Knowledge distillation is a learning paradigm where we train a student network $g_{\theta s}$ to match the output of a given teacher network $g_{\theta t}$, parameterized by θ_s and θ_t respectively. Given an input image x, both networks output probability distribution over K dimensions denoted by P_s and P_t. The probability P is obtained by normalizing the output of the network g with a softmax function,

$$P_s(x)^{(i)} = \frac{exp(g_{\theta s}(x)^{(i)}/\tau_s)}{\sum_{k=1}^{K} exp(g_{\theta s}(x)^{(k)}/\tau_s)} \quad (1)$$

with $\tau_s > 0$ a temperature parameter that controls the sharpness of the output distribution, and a similar formula holds for P_t with temperature τ_t. Given a fixed teacher network, we learn to match these distributions by minimizing the cross-entropy loss w.r.t parameters of the student network θ_s:

$$min_{\theta s} H(P_t(x), P_s(x)). \quad (2)$$

where $H(a, b) = -alogb$.

NCE loss The NCE loss function is widely used in contrastive learning, such as MoCo[8], simCLR[7].

$$NCE_{loss} = -log\frac{exp(sim(g(x),g(x^+))/\tau)}{exp(sim(g(x),g(x^+))/\tau)+\sum_{k=1}^{K} exp(sim(g(x),g(x_k^-))/\tau)} \quad (3)$$

As equation(3), the pair (x, x^+) represents a positive example. Usually, x^+ is the result of some transformation on x. This can be a geometric transform aimed to change the size, shape or orientation of x, or any type of data augmentation technique. Some examples include rotation, sheer, resize, cutout and more. On the other hands, the pair (x, x^-) represents a negative example,and they are meant to be uncorrelated.Note that for each positive pair (x,x+) we have a set of K negatives[14].The $sim(.)$ function is a similarity (distance) metric. It is responsible for minimizing the difference between the positives while maximizing the difference between positive and negatives. Often, $sim(.)$ is defined in terms of dot products or cosine similarities.Lastly, $g(.)$ is the framework to extract features, In DINO we consider teacher ans student network as backbone. The only difference between cross-entropy loss and NCE loss is that cross-entropy loss pass through K sample pairs while NCE loss pass through K+1 sample pairs.

C. Dataset

In this study, We training on STL-10 datast and cluttered-MNIST dataset.

STL-10 The STL-10 is an image dataset derived from ImageNet and popularly used to evaluate algorithms of unsupervised feature learning or self-taught learning. Besides 100,000 unlabeled images, it contains 13,000 labeled images from 10 object classes (such as birds, cats, trucks). All the images are color images with 96×96 pixels in size[15]. Here, we training on the STL-10 dataset as prior to prepare some suitable hyperparameters in order to make sure our model make sense.

Cluttered MNIST The cluttered MNIST dataset is proposed from DeepMind. Based on an original handwritten digit image, we add various distortions around the true digit. In the previous study, this dataset is used to delete the distortions and classify the true label of the digit from the image. Here, we modify the original dataset and apply various experiments on it. It contains 50,000 training images and 10,000 test images and 10,000 validation images, each image has size 100 x 100. We randomly set 6 different distortions around the MNIST-digit and each distortion has size 9 x 9. We wonder, if the machine after training with DINO algorithm can successfully segment the true digit in an image and ignore other distortions at the same time.

III. Experiments

The model we build is based on DINO. As mentioned from DINO, We train with adamw optimizer and a batch size of 32. The learning rate is linearly ramped up during the first 3

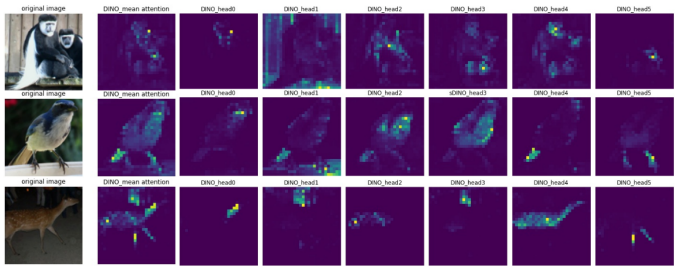

Fig. 2. Self-attention maps from the last layer of ViT with cross-entropy loss in DINO algorithm after training 10 epochs on ViT-S/8 model.

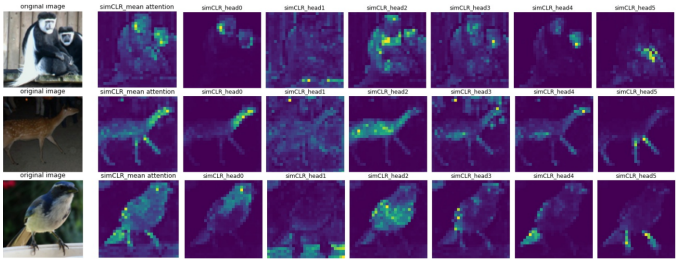

Fig. 3. Self-attention maps from the last layer of ViT with NCE loss in simCLR algorithm after training 10 epochs on ViT-S/8 model.

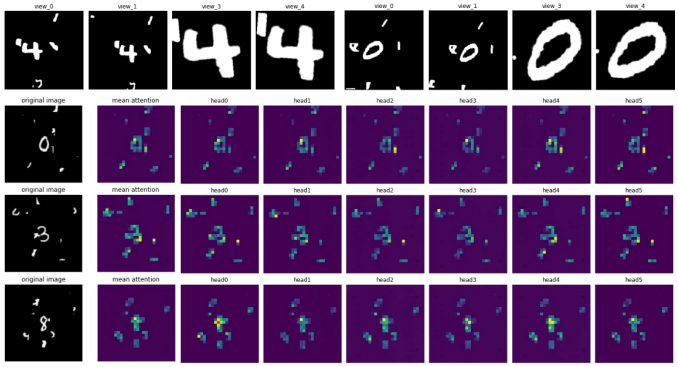

Fig. 4. The first row shows different crops and data augmentation we set to fed into the teacher branch and student branch. The second to fourth rows show the self-attention maps from the last layer of ViT with NCE loss in simCLR algorithm after training 20 epochs on ViT-S/8 model.

epochs to its base value determined with the following linear scaling rule: $lr = 0.0005 * batchsize/64$. After this warm up, we decay the learning rate with a cosine schedule. The weight decay also follows a cosine schedule from 0.04 to 0.4. The temperature τ is set to 0.1 at first.

Given a vision transformer, the size of each attention map of the last layer is image size/patch size. For example, if we set patch size equal to 8 and feed an image of size 112 x 112, the size of each output attention map is 14 x 14. This makes it difficult to interpret how the model was activated with respect to the image. A high resolution input image means a high quality attention map. However, due to the limitation of device, 448 x 448 is too big to training on Tesla P100-16GB GPU. Therefore, we choose the input image size equal to 224. As a result from DINO, The performance greatly improves as decreasing the size of the patch. So we set patch size to 8. Furthermore, we load the pretrained ViT-S/8 DINO weighting from PyTorch to speed up the training process.

A. Loss function

At first, we train on STL-10 dataset to make sure the parameters we set can make sense. Through the original paper from DINO, cross-entropy loss function is an important component for self-supervised ViT pretraining. Therefore, we train the model with both NCE and cross-entropy loss function to observe the results. After training with 5 epochs, the output self-attention maps are shown below in figure 2 and figure 3. We also run t-SNE with perplexity of 20 and present the resulting class embeddings. As a result, training with both NCE loss and cross-entropy loss can predict a quite well classification results. The self-attention maps of different heads we obtained also shown that different heads pay attention to different components of the object in an image. In addition, training with NCE loss can achieve a high level human-interpretable attention map. To make it easier, in the next experiments, we training our model with NCE loss.

B. Data augmentation

Since the binary images only have black and white pixels, the general data augmentation methods such as color jittering, solarization and flip don't work as well, it can make the digit indistinguishable. Here we only choose crop, rotation and Gaussian blur as augmentation methods. Contrastive learning between multiple views of the data has recently achieved state of the art performance in the field of self-supervised representation learning. Despite its success, the influence of different view choices has been less studied. In this experiment, we will focus on the optimal views for contrastive learning to find better attention maps.

1) teacher and student have same views: First, we fed the same views to both teacher and student branches. For each views we fed into 4 crops, which means 2 global crops with crop scale from 0.5 to 1.0, and two local crops with the true digit in the center of the crop as shown in the first row of figure 4. We resize the crops to size 224 x 224 and fed into the model.

To investigate whether the black background affects the segmentation of the foreground, we add different textured background images to the original cluttered MNIST dataset. This result in figure 5 shows that if we feed the same view in both branches, the machine can ignore the background image and pay attention to all the white pixels, i.e., the machine consider all distortions and true digit as objects, which is not what we expect when we segment only the digit and ignore the interference.

2) teacher and student have different views: Next, we fed different views to different branches. We randomly set the position of the number in the new image based on the same digit, and randomly set 6 different distortions around the digit, which means two views share same digit information but

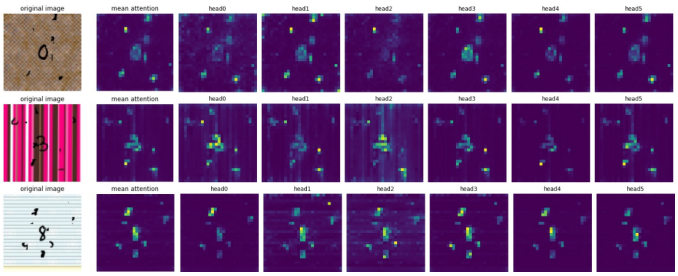

Fig. 5. Self-attention maps from the last layer of ViT with NCE loss in simCLR algorithm after training 20 epochs on cluttered MNIST dataset with different texture background.

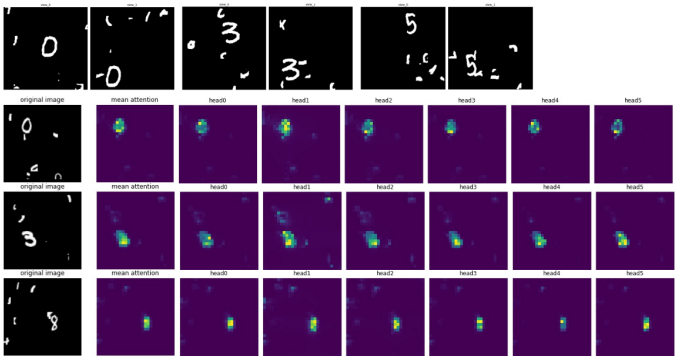

Fig. 6. The first row shows different crops and data augmentation we set. Different from the last experiment, in this experiment we will fed only the first, the third and the fifth images into the teacher branch and in other hand, fed the second, the fourth and the sixth images into student branch. The second to fourth rows show the self-attention maps from the last layer of ViT with NCE loss in simCLR algorithm after training 20 epochs on ViT-S/8 model.

different distortions information. This time, we only use one global crop per view with crop scale from 0.5 to 1.0. As shown in figure 6, after training with only 1 epoch, although due to the low resolution of attention map, we cannot get the shape of the digit so clearly, the machine can find where the true digit is very accurately and can ignore all the other distortions, which is what we expect. However, after 10 epochs, the loss is still decreasing, but different heads focus on different information, some point to the digit, some even pay attention to the distortions, which is not we expect. In the future, we still have to continue investigate the meaning of different heads and deal with this question.

C. multi-crops training

Through the original DINO paper, multi-crops training is also an important component for self-supervised ViT pretraining. During the previous work, we trained the model with 4 crops per branch, 2 crops per branch and one crop per branch. The resulting findings are not significantly different. There may not be much information in the binary images, which is one potential reason.

IV. Conclusion

In this work, we train on cluttered MNIST dataset to observe the impact of different components in DINO algorithm for self-attention maps. As a result, for binary images, the size of the input images and the different views we fed into the model are two important components to obtain a high resolution and high quality self-attention maps. However, the performance on binary images is not as good as on color images. In the future, we plan to exploring more other factors. For example,

- Contrastive learning needs a big batch size. In simCLR, the batch size is set from 256 to 8192. Due to the limitation of device, in this study we only train with batch size equal to 32. Whether a large batch size is the key to obtaining high quality self-attention maps is one of the research directions.
- In this study, we set the number of heads equal to 6. What is the meaning of each head and how many heads do we really need to obtain a high-resolution attention maps is one of the next research directions.
- In this study, we only consider DINO algorithm and use the pretrained weighting ViT-S/8 from DINO. Recently, a lot of state of the art models are proposed, whether there is a better self-supervised learning model to obtain high-resolution self-attention maps is also the next research direction.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[3] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," *CoRR*, vol. abs/2104.14294, 2021. [Online]. Available: https://arxiv.org/abs/2104.14294

[4] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *CoRR*, vol. abs/1803.07728, 2018. [Online]. Available: http://arxiv.org/abs/1803.07728

[5] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," *CoRR*, vol. abs/1505.05192, 2015. [Online]. Available: http://arxiv.org/abs/1505.05192

[6] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," *CoRR*, vol. abs/1603.09246, 2016. [Online]. Available: http://arxiv.org/abs/1603.09246

[7] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," *CoRR*, vol. abs/2002.05709, 2020. [Online]. Available: https://arxiv.org/abs/2002.05709

[8] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," *CoRR*, vol. abs/1911.05722, 2019. [Online]. Available: http://arxiv.org/abs/1911.05722

[9] D. Coccomini, "Self-supervised learning in vision transformers," 2021. [Online]. Available: https://towardsdatascience.com/self-supervised-learning-in-vision-transformers-30ff9be928c

[10] S. A. A. Ahmed, M. Awais, and J. Kittler, "Sit: Self-supervised vision transformer," *CoRR*, vol. abs/2104.03602, 2021. [Online]. Available: https://arxiv.org/abs/2104.03602

[11] J. D. Janizek, P. Sturmfels, and S. Lee, "Explaining explanations: Axiomatic feature interactions for deep networks," *CoRR*, vol. abs/2002.04138, 2020. [Online]. Available: https://arxiv.org/abs/2002.04138

[12] S. Serrano and N. A. Smith, "Is attention interpretable?" in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2931–2951. [Online]. Available: https://aclanthology.org/P19-1282

[13] S. Jain and B. C. Wallace, "Attention is not Explanation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 3543–3556. [Online]. Available: https://aclanthology.org/N19-1357

[14] Thalles, "Exploring simclr: A simple framework for contrastive learning of visual representations," 2020. [Online]. Available: https://sthalles.github.io/simple-self-supervised-learning/

[15] D. Wang and X. Tan, "C-svddnet: An effective single-layer network for unsupervised feature learning," *CoRR*, vol. abs/1412.7259, 2014. [Online]. Available: http://arxiv.org/abs/1412.7259