Image as Set of Points

Esther Choi, Vinh Son Pho, Miray Senyuz

Project of Advanced Machine Learning & Deep Learning

Introduction

Feature extraction depends on the way we see an image:

- Convolution Neural Networks (CNNs) are based on the paradigm that images can be seen as 2D signals on which we can use convolutions to extract features.
- Vision Transformers (ViTs) [1] see images as sequences of patches and use an attention mechanism [3] to extract features.
- In this paper [2], we see images as sets of points and this new paradigm allows us to use **clustering** to extract features.

Model

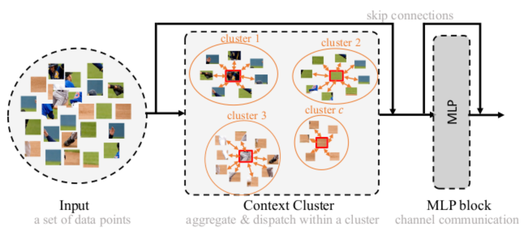

Fig. 1: Context Cluster Block

The **Context Cluster block** is composed of three steps. First, a context cluster operation is used to group a set of disorganized data points. Then, communication between points within the clusters takes place. Finally, a MLP (2 layers) is applied. This block can be compared to the transformer encoder block used in ViTs.

The **clustering operation** is done by first selecting c cluster centers, using adaptive average pooling. Each point is assigned to the cluster whose center is the closest in terms of cosine similarity. This operation is similar to the attention used in ViTs. It can also be implemented using multi-head computing.

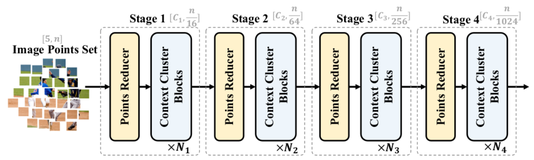

Fig. 2: Context Cluster Architecture with four stages

The **Context Cluster architecture** consists of four stages. It takes a set of image points and gradually decreases their number while extracting deeper features. Each stage starts with a **points reducer** which is done by a weighted average of k neighbouring points (2D convolution of size and stride $\sqrt{k}$). The point reducer is then followed by a series of context cluster blocks that are employed to extract features. Classification is done by averaging all points of the last block and using a FC layer.

Results

The authors proposed three sets of hyperparameters (CoC-Tiny, CoC-Small and CoC-Medium) with an increasing number of parameters. Among them, we tested the **CoC-Tiny** and the **CoC-Small** models on two datasets :

- **CIFAR-10** (32x32 images) : from scratch over 300 epochs
- **CalTech-101** (224x224 images) : over 20 epochs using pretrained weights from the authors for the feature extractor (due to technical issues, we could not train the whole model ourselves).

We also tested a smaller version of CoC-tiny that we called **CoC-Supertiny** containing only three stages.
The table below shows the test accuracy.

Model	CIFAR10	CalTech101
CoC-Supertiny	85%	91%
CoC-Tiny	80%	91%
Coc-Small	-	92.5%

We discarded the first `PointReducer` for images with very low resolution. Reducing the number of stages also improves the performance of the model. This is mainly due to the fact that if we had kept the 4 `PointReducer` modules, the last stage would get a very small output of size 2x2.

Clustering Maps

During the feature extraction process, clusters of pixels are created. The figures below show these clusters at each of the three stages.

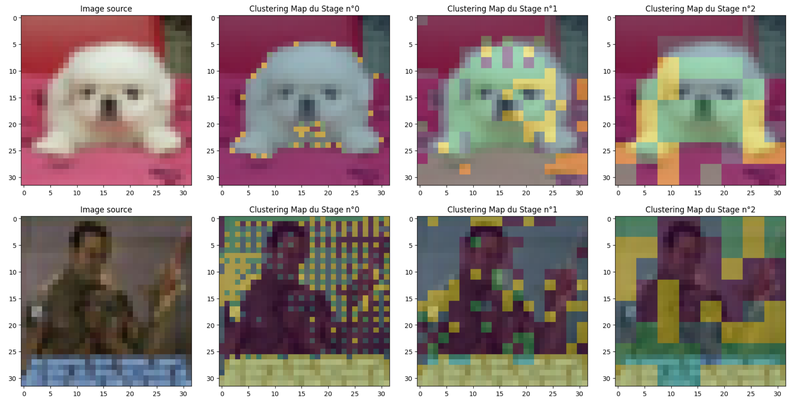

Fig. 3: Clustering Maps of the CoC-Small model on the CIFAR datasets

Source image and clustering maps at stage 1, 2 and 3

The Context Cluster model is able segment the different parts of the images, making it interpretable. These clustering maps are similar to attention maps in Vision Transformers.

Ablation Studies

We then performed an ablation study to better understand the effects of the clustering layers on the performance of the models. Below are the test accuracy of our model "CoC-Supertiny" with and without the clustering operation.

Model	CIFAR10	CIFAR100
w/ Clustering	85%	65%
w/o Clustering	80%	60%

The use of the clustering layers is able to boost our test performances on these two datasets by 5 points. This highlights the capacity of the clustering operation.

Remarks

- Though the CoC models are very interesting, they are hard to train : even with 300 epochs, we could not achieve achieve competitive performance on datasets with low resolution.
- However, as we can see with the clustering maps, CoC models are able to capture the location of important elements of an image, and thus also perform well on other visual tasks like image detection, image segmentation and semantic segmentation, as demonstrated in the original article
- The models presented here are very similar to ViTs, especially the Context Cluster block which can be seen as computing an "attention score" between all the points of the image and the c cluster centers.
- Below, the configuration used for the CoC Models :

	CoC-Supertiny	CoC-Tiny	CoC-Small
Embedding sizes	[5,32,128]	[32,64,196,320]	[64,128,320,512]
Number of Context Cluster blocks	[4,6,4]	[3,4,5,2]	[2,2,6,2]
Head Counts	[4,4,8]	[4,4,8,8]	[4,4,8,8]
Head Dimensions	[16,16,16]	[24,24,24,24]	[32,32,32,32]
MLP hidden layers ratio	[16,8,4]	[8,8,4,4]	[8,8,4,4]
Parameter count	900k	4.8M	13M

References

[1] Alexander Kolesnikov et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: 2021.

[2] Xu Ma et al. “Image as Set of Points”. In: (2023). URL: https://openreview.net/forum?id=awnvqZja69.

[3] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.