# Image as Set of Points

Xu Ma, Yuqian Zhou, Huan Wang, Can Qin, Bin Sun, Chang Liu, Yun Fu (Northeastern University, Adobe Inc.) – ICLR 2023

Jeeyoung Kim

University of Ulsan College of Medicine,

Asan Medical Center

77imjee@gmail.com

2023.05.19 Fri

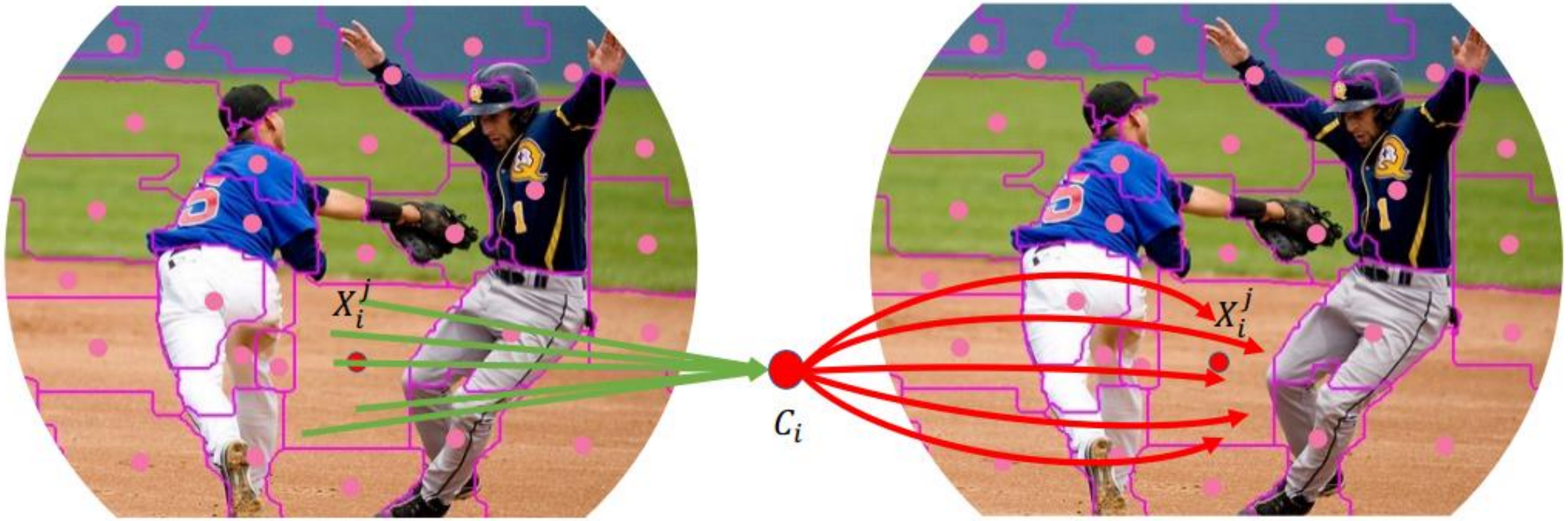
# Introduction

- Convolutional Networks
  - a collection of arranged pixels in a rectangle form
  - extract local features using convolution in a sliding window fashion
  - inductive biases like locality made CNNs to be more efficient and effective
- Vision Transformers
  - a sequence of patches
  - extract features via attention mechanism in a global range
  - inherent inductive biases are abandoned
- Hybrid Networks (CNN + ViT)
  - scan images in grid (conv) / mutual relationships of a sequence (attention)
  - locality prior (conv) without sacrificing global reception (attention)
  - But the insights and knowledge are still restricted to CNNs and ViTs

# Introduction

- New paradigm of feature extraction except CNN and ViT

  → **Context Clusters (CoCs)**




- Great generalization to different data domains
- Provide nice interpretability by visualizing each cluster
- Achieves competitive performance compared with CNNs and ViTs

# Related works

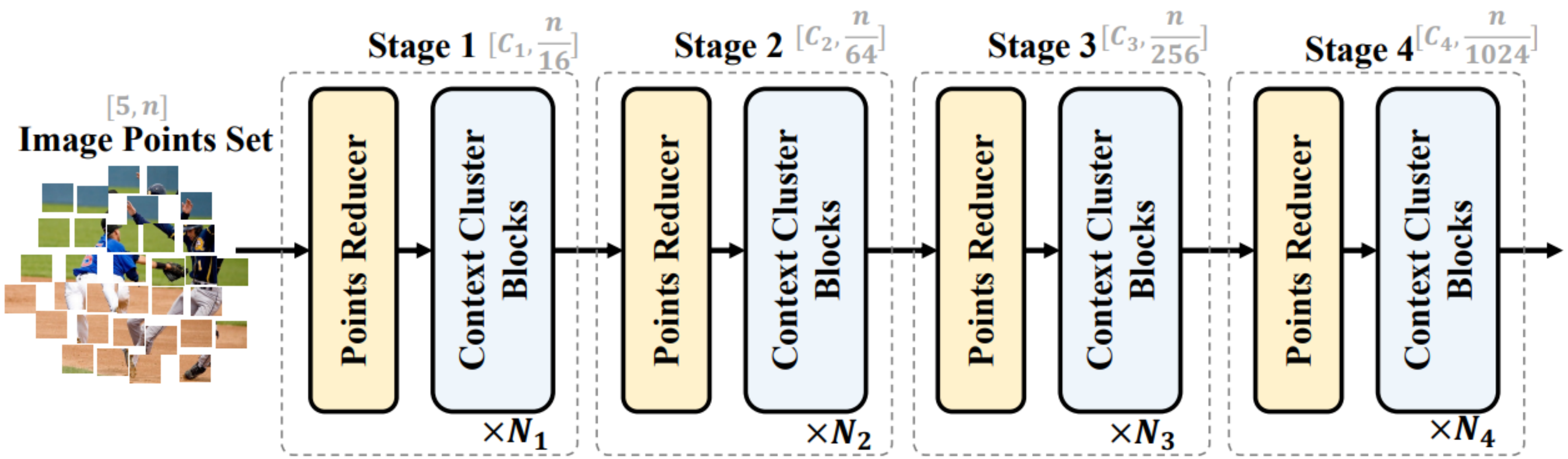
- SuperPixel (Ren & Malik, 2003)
  - segment an image into regions by grouping a set of pixels that share common characteristics
  - common practice for image preprocessing
  - clusters pixels over the entire image → **heavy computational cost**
- SLIC (Achanta et al., 2012)
  - limits the clustering operation in a local region
  - evenly initialized the K-means centers for better and faster convergence

→ for image processing or specific task

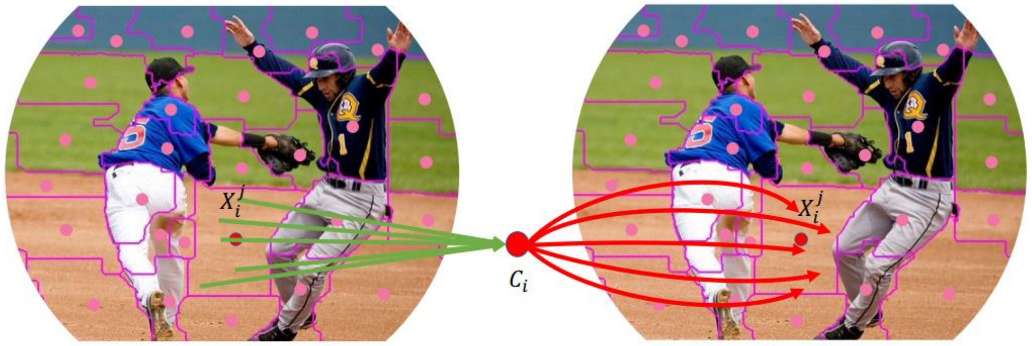
→ **No work was conducted for a general visual representation via clustering**

# Methods

- Context Cluster architecture




1. **From image to set of points**
2. **Points reducing**
3. **Context clustering**
4. **Feature aggregating and dispatching**



# Methods

- From image to set of points
  - given an input image, $\mathbf{I} \in \mathbb{R}^{3\times w\times h}$
  - $\mathbf{I}_{i,j} = \left[\frac{i}{w} - 0.5, \frac{j}{h} - 0.5\right]$ - It is feasible to investigate further positional augmentation techniques to potentially improve performance
  - converted to a collection of points (i.e., pixels) $\mathbf{P} \in \mathbb{R}^{5\times n}$
  - each point contains both feature (color) and position (coordinates) information → unordered and disorganized

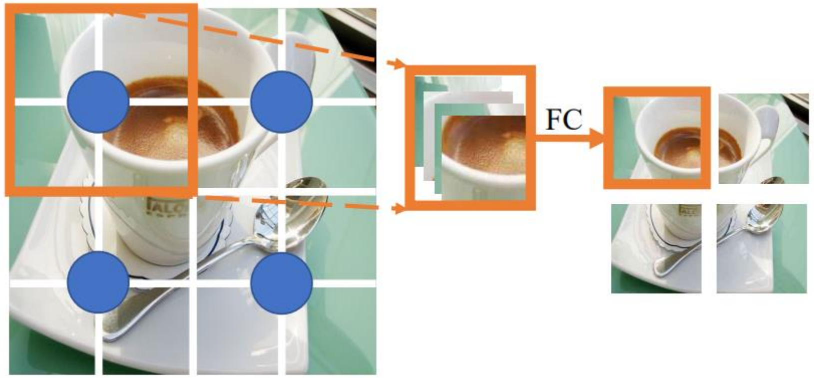
- Points reducing
  - $\mathbf{P} \in \mathbb{R}^{5\times n} \rightarrow \mathbf{P} \in \mathbb{R}^{n\times d}$
  - 16 points with 4 proposed anchors for point reduction,

  each of which takes its closest 4 neighbors into account.

  All neighbors are concatenated along the channel dimension,

  and a FC layer is used to lower the dimensional number and

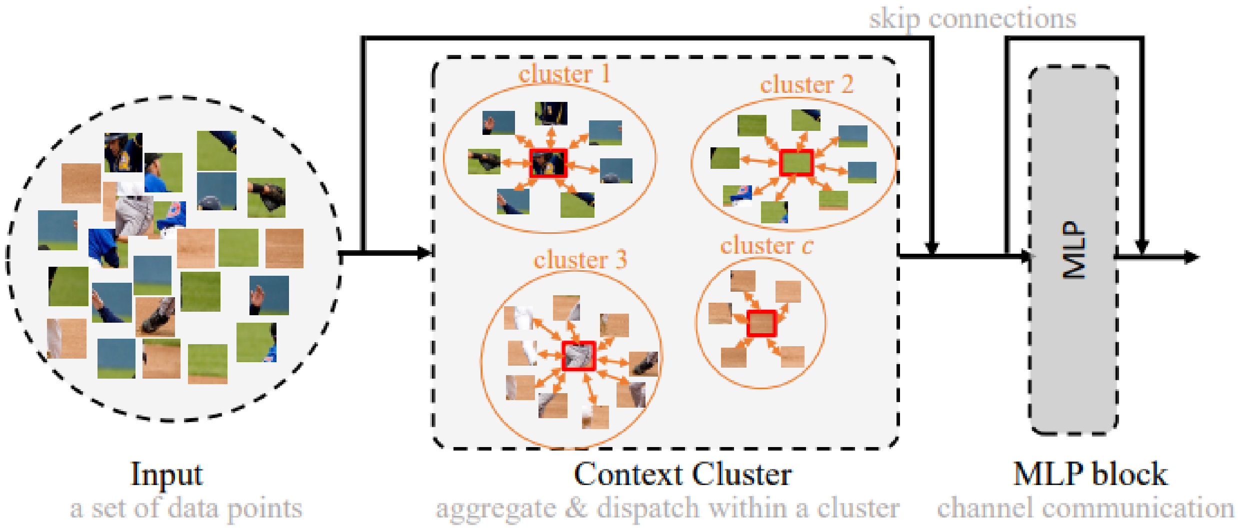
  fuse the information.

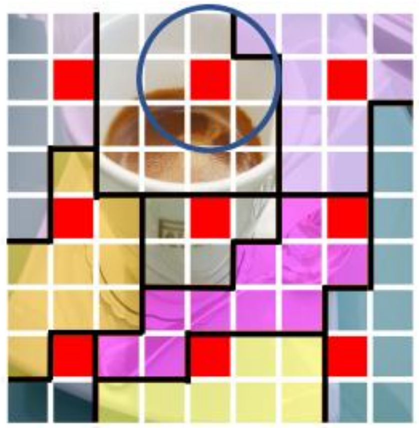



(a) Illustration of anchors for points reduction.

# Methods

- Context clustering
  - linearly project feature points $\mathrm{P}$ to $\mathrm{P_s}$
  - Following SLIC, evenly propose $\mathrm{c}$ centers whose features are computed by averaging its k nearest points
  - calculate the pair-wise cosine similarity matrix $\mathrm{S}$ between $\mathrm{P_s}$ and set of $\mathrm{c}$
  - allocate each point to the most similar center, resulting in $\mathrm{c}$ clusters
  - each cluster may have a different number of points



(b) Demo of centers in CoC.

# Methods

- Feature aggregating
  - $m$ points in a cluster, similarity $s$ between the $m$ points and the cluster center
  - map the points to a value space of d'-dim, center $v_c$ in the value space
  - aggregated feature ($g$)

$$g = \frac{1}{C}\left(v_c + \sum_{i=1}^{m} \mathrm{sig}\left(\alpha s_i + \beta\right) * v_i\right), \qquad \text{s.t.,} \quad C = 1 + \sum_{i=1}^{m} \mathrm{sig}\left(\alpha s_i + \beta\right)$$

α, β: learnable scalars to scale and shift
C: control the magnitude

- Feature dispatching
  - match the feature dimension (d' to d) with FC layer

$$p_i' = p_i + \mathrm{FC}\left(\mathrm{sig}\left(\alpha s_i + \beta\right) * g\right)$$

  - adaptively dispatch to each point based on the similarity
  - the points an communicate with one another and share features from all points in the cluster

# Methods

- Architecture initialization
  - try to align with other networks and make CoCs compatible with most detection and segmentation algorithms
  - reduce the number of points by a factor of 16, 4, 4, 4
  - 16, 9, 9, 9 nearest neighbors for selected anchors in each stage
- Region partition
  - calculating the similarity between n d-dim points and c clusters → high computational cost
  - split the points into several local regions like Swin Transformer
- Fixed or dynamic centers for cluster
  - Fixed: inference efficiency, compromise between accuracy and speed
  - Dynamic: exorbitant computing costs, inference time increases exponentially
- Overlap or non-overlap clustering
  - allocate the points solely to a specific center
  - to demonstrate that the simple and traditional algorithm can serve as a generic backbone, adhere to the non-overlap clustering

# Results

- Image Classification on ImageNet-1K

Table 1: Comparison with representative backbones on ImageNet-1k benchmark. Throughput (images / s) is measured on a single V100 GPU with a batch size of 128, and is averaged by the last 500 iterations. All models are trained and tested at 224×224 resolution, except ViT-B and ViT-L.

| | Method | Param. | GFLOPs | Top-1 | Throughputs (images/s) |
|---|---|---|---|---|---|
| MLP | ♣ ResMLP-12 (Touvron et al., 2022) | 15.0 | 3.0 | 76.6 | 511.4 |
| | ♣ ResMLP-24 (Touvron et al., 2022) | 30.0 | 6.0 | 79.4 | 509.7 |
| | ♣ ResMLP-36 (Touvron et al., 2022) | 45.0 | 8.9 | 79.7 | 452.9 |
| | ♣ MLP-Mixer-B/16 (Tolstikhin et al., 2021) | 59.0 | 12.7 | 76.4 | 400.8 |
| | ♣ MLP-Mixer-L/16 (Tolstikhin et al., 2021) | 207.0 | 44.8 | 71.8 | 125.2 |
| | ♣ gMLP-Ti (Liu et al., 2021a) | 6.0 | 1.4 | 72.3 | 511.6 |
| | ♣ gMLP-S (Liu et al., 2021a) | 20.0 | 4.5 | 79.6 | 509.4 |
| Attention | ♦ ViT-B/16 (Dosovitskiy et al., 2020) | 86.0 | 55.5 | 77.9 | 292.0 |
| | ♦ ViT-L/16 (Dosovitskiy et al., 2020) | 307 | 190.7 | 76.5 | 92.8 |
| | ♦ PVT-Tiny (Wang et al., 2021) | 13.2 | 1.9 | 75.1 | - |
| | ♦ PVT-Small (Wang et al., 2021) | 24.5 | 3.8 | 79.8 | - |
| | ♦ T2T-ViT-7 (Yuan et al., 2021a) | 4.3 | 1.1 | 71.7 | - |
| | ♦ DeiT-Tiny/16 (Touvron et al., 2021) | 5.7 | 1.3 | 72.2 | 523.8 |
| | ♦ DeiT-Small/16 (Touvron et al., 2021) | 22.1 | 4.6 | 79.8 | 521.3 |
| | ♦ Swin-T (Liu et al., 2021b) | 29 | 4.5 | 81.3 | - |
| Convolution | ♠ ResNet18 (He et al., 2016) | 12 | 1.8 | 69.8 | 584.9 |
| | ♠ ResNet50 (He et al., 2016) | 26 | 4.1 | 79.8 | 524.8 |
| | ♠ ConvMixer-512/16 (Trockman et al., 2022) | 5.4 | - | 73.8 | - |
| | ♠ ConvMixer-1024/12 (Trockman et al., 2022) | 14.6 | - | 77.8 | - |
| | ♠ ConvMixer-768/32 (Trockman et al., 2022) | 21.1 | - | 80.16 | 142.9 |
| Cluster | ♥ Context-Cluster-Ti (ours) | 5.3 | 1.0 | 71.8 | 518.4 |
| | ♥ Context-Cluster-Ti‡ (ours) | 5.3 | 1.0 | 71.7 | 510.8 |
| | ♥ Context-Cluster-Small (ours) | 14.0 | 2.6 | 77.5 | 513.0 |
| | ♥ Context-Cluster-Medium (ours) | 27.9 | 5.5 | 81.0 | 325.2 |

‡ denotes a different region partition approach that we used to divide the points into [49, 49, 1, 1] in the four stages
Default: [64, 16, 4, 1]

# Results

- Visualization of activation map

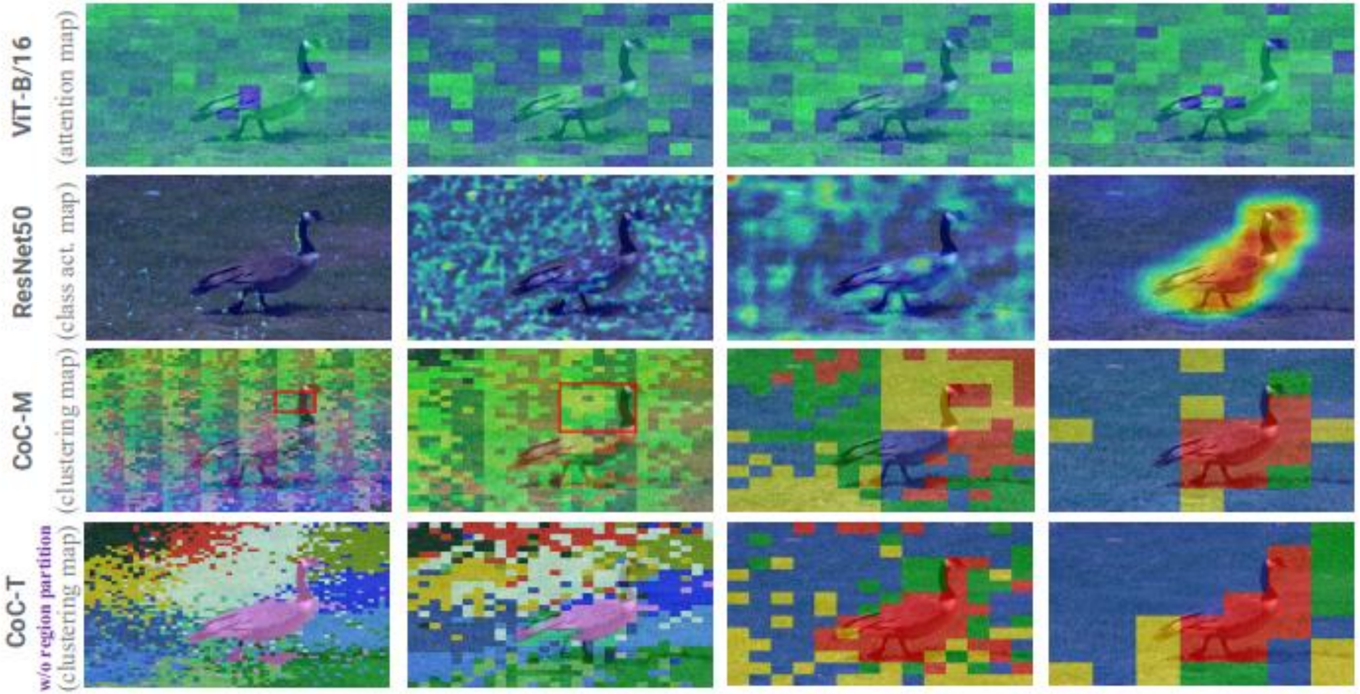



Figure 4: Visualization of activation map, class activation map, and clustering map for ViT-B/16, ResNet50, our CoC-M, and CoC-T without region partition, respectively. We plot the results of the last block in the four stages from left to right. For ViT-B/16, we select the [3rd, 6th, 9th, 12th] blocks, and show the cosine attention map for the `cls-token`. The clustering maps show that our Context Cluster is able to cluster similar contexts together, and tell what model learned visually.

# Results

- Object Detection and Instance Segmentation on MS-COCO

Table 4: COCO object detection and instance segmentation results using Mask-RCNN (1×).

| Family | Backbone | Params | $AP^{box}$ | $AP^{box}_{50}$ | $AP^{box}_{75}$ | $AP^{mask}$ | $AP^{mask}_{50}$ | $AP^{mask}_{75}$ |
|---|---|---|---|---|---|---|---|---|
| Conv. | ♠ ResNet-18 | 31.2M | 34.0 | 54.0 | 36.7 | 31.2 | 51.0 | 32.7 |
| Attention | ♦ PVT-Tiny | 32.9M | 36.7 | 59.2 | 39.3 | 35.1 | 56.7 | 37.3 |
| Cluster | ♥ CoC-Small/4 | 33.6M | 35.9 | 58.3 | 38.3 | 33.8 | 55.3 | 35.8 |
| | ♥ CoC-Small/25 | 33.6M | **37.5** | **60.1** | **40.0** | **35.4** | **57.1** | **37.9** |
| | ♥ CoC-Small/49 | 33.6M | 37.2 | 59.8 | 39.7 | 34.9 | 56.7 | 37.0 |

Table 8: COCO object detection and instance segmentation results using Mask-RCNN (1×).

| Family | Backbone | Params | $AP^{box}$ | $AP^{box}_{50}$ | $AP^{box}_{75}$ | $AP^{mask}$ | $AP^{mask}_{50}$ | $AP^{mask}_{75}$ |
|---|---|---|---|---|---|---|---|---|
| Conv. | ♠ ResNet-50 | 44.2M | 38.0 | 58.6 | 41.4 | 34.4 | 55.1 | 36.7 |
| Atten | ♦ PVT-Small | 44.1M | 40.4 | 62.9 | 43.8 | **37.8** | **60.1** | **40.3** |
| Cluster | ♥ CoC-Medium/4 | 42.1M | 38.6 | 61.1 | 41.5 | 36.1 | 58.2 | 38.0 |
| Cluster | ♥ CoC-Medium/25 | 42.1M | 40.1 | 62.8 | 43.6 | 37.4 | 59.9 | 40.0 |
| Cluster | ♥ CoC-Medium/49 | 42.1M | **40.6** | **63.3** | **43.9** | 37.6 | **60.1** | 39.9 |

# Results

- Semantic Segmentation on ADE20K

Table 5: Semantic segmentation performance of different backbones with Semantic FPN on the ADE20K validation set.

| Backbone | Params | mIoU(%) |
|---|---|---|
| ♠ ResNet18 | 15.5M | 32.9 |
| ♦ PVT-Tiny | 17.0M | 35.7 |
| ♥ CoC-Small/4 | 17.7M | **36.6** |
| ♥ CoC-Small/25 | 17.7M | **36.4** |
| ♥ CoC-Small/49 | 17.7M | **36.3** |

Table 7: Semantic segmentation results of different backbones with Semantic-FPN on the ADE20K validation set.

| Family | Backbone | Params | mIoU(%) |
|---|---|---|---|
| Conv. | ♠ ResNet50 | 28.5M | 36.7 |
| Atten. | ♦ PVT-Small | 28.2M | 39.8 |
| Cluster | ♥ CoC-Medium/4 | 25.2M | **40.2** |
| Cluster | ♥ CoC-Medium/25 | 25.2M | **40.6** |
| Cluster | ♥ CoC-Medium/49 | 25.2M | **40.8** |

# Conclusion

- The authors proposed Context Cluster, a novel feature extraction paradigm for visual representation
- CoC is fundamentally distinct from CNNs and ViTs, no convolution or attention is involved
- Instead of chasing SOTA performance, CoCs can achieve comparable or even better results than CNN and ViT baselines on multiple tasks and domains
- Departing from the current framework on detection and segmentation to apply CoC philosophy to other tasks is also worthwhile direction to pursue