Neural Clustering based Visual Representation Learning

Guikun Chen[1], Xia Li[2], Yi Yang[1], Wenguan Wang[1*]
[1] ReLER, CCAI, Zhejiang University [2] ETH Zürich
https://github.com/guikunchen/FEC/

Abstract

We investigate a fundamental aspect of machine vision: the measurement of features, by revisiting clustering, one of the most classic approaches in machine learning and data analysis. Existing visual feature extractors, including ConvNets, ViTs, and MLPs, represent an image as rectangular regions. Though prevalent, such a grid-style paradigm is built upon engineering practice and lacks explicit modeling of data distribution. In this work, we propose feature extraction with clustering (FEC), a conceptually elegant yet surprisingly ad-hoc interpretable neural clustering framework, which views feature extraction as a process of ***selecting representatives*** *from data and thus automatically captures the* ***underlying data distribution****. Given an image, FEC alternates between grouping pixels into individual clusters to abstract representatives and updating the deep features of pixels with current representatives. Such an iterative working mechanism is implemented in the form of several neural layers and the final representatives can be used for downstream tasks. The cluster assignments across layers, which can be viewed and inspected by humans, make the* ***forward*** *process of FEC* ***fully transparent*** *and empower it with promising ad-hoc interpretability. Extensive experiments on various visual recognition models and tasks verify the effectiveness, generality, and interpretability of FEC. We expect this work will provoke a rethink of the current de facto grid-style paradigm.*

1. Introduction

The measurement of features, which explores how to extract abstract, meaningful features from high-dimensional image data, is a topic of enduring interest in machine vision throughout its history [1–3]. This pursuit, initially dominated by manually engineered descriptors [4–9], has evolved under the influence of deep learning paradigms, transitioning from convolutional landscapes [10, 11] to the frontiers of attention-driven mechanisms [12, 13] and MLP-based approaches [14, 15]. Convolutional networks (ConvNets, Fig. 1a) treat an image as rectangular regions and execute in a sliding window manner. Attention-based methods (Fig. 1b) usually divide an image into several non-overlap patches and use an additional [CLS] token to represent the whole image. MLP-based backbones (Fig. 1c) also follow the grid-style paradigm while extracting features without convolution or attention operations.

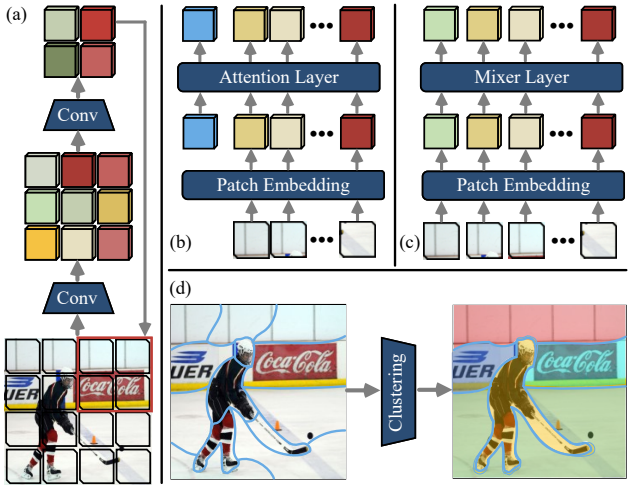

Figure 1. How to represent an image in a low-dimensional space and what could explain it? (abc) Existing visual backbones rely on the computational modeling of rigid grids. (d) Derived from a neural clustering view, FEC reformulates the procedures of feature extraction as clustering, thereby representing the image with its representatives. Our approach possesses promising *ad-hoc* interpretability and demonstrates the emergence of segmentation despite being trained only on the classification task.

Upon observing the array of visual backbones shown in Fig. 1, the following questions naturally arise: ❶ *What is the relation between them?* and more critically, ❷ *If these neural networks indeed implicitly capture some intrinsic properties of image data, might there exist a more transparent and interpretable method to measure the visual features?*

The pursuit of the question ❶ uncovers a persistent adherence to a grid-centric view in the realm of image data

[1]Corresponding author: Wenguan Wang.

analysis [16–18]. Concretely, **the basic elements during the forward process of existing backbones are rectangular image regions**, *e.g.*, the kernels (filters), sliding window, and receptive field in convolution-based backbones, or the image patches in vision Transformers (ViTs) and MLPs. Such a widespread paradigm, though instrumental in the evolution of convolutional networks and their successors, seems to be based more on engineering convention than on the emulation of natural image structures. Most existing efforts are expected to generate more abstract features as the network's layers deepen, while nobody knows how they make it [19]. Therefore, question ❷ becomes more fundamental: ❸ *What are the inherent limitations of this grid-style paradigm?* and ❹ *Can we evolve beyond the grid-based uniformity assumption that fails to encapsulate the organic structure of images?*

Driven by question ❸, we uncover two critical limitations: **First**, the grid model is at odds with the true nature of pixel organization, thereby failing to grasp the complexity of data distribution [20]. **Second**, the black-box nature of deep feature extractors impedes interpretability, veiling the rationale behind feature selection and significance. This leads us to the question ❹, probing the opacity of current methodologies and their divergence from human perception and cognition [21–23], which possess a remarkable ability to break down visual scenes into semantic-meaning components. The goal is to build feature extractors that can better capture the data distribution of pixels and mirror the cognitive processes of human vision so as to enhance both interpretability and transparency. To bridge the identified gaps, a fundamental paradigm shift is imperative: **i**) pivoting from the grid-view of image representations towards a more fluid model that embraces the dynamic nature of visual data; and **ii**) stepping away from the black-box models towards an ambitious hybrid that integrates powerful representation learning and interpretable feature encoding.

In this vein, we introduce FEC (§3), a *mechanistically interpretable* backbone that roots in the principle of clustering. It begins with a window-based pooling to generate pixel blocks that serve as initial elements. Afterward, FEC iterates through two key processes: **i**) Clustering-based Feature Pooling. Neural clustering is used to model representatives of the given inputs (pixel blocks or previous clusters), leading to more abstract (**growing**) clusters. Due to the clustering nature, a representative (cluster) *explicitly* represents a set of pixels in **any position**. This is where FEC differs from grid-style paradigm. **ii**) Clustering-based Feature Encoding. Here the representatives are first estimated and then used for redistributing features to each pixel given the similarity between the pixel and its representative. Within such a clustering based framework, the basic elements during FEC's forward process are **gradually growing clusters**.

FEC exhibits several compelling characteristics: **First**, enhanced simplicity, and ***transparency***. The streamlined design, coupled with the semantic meaning of clustering during feature extraction, renders FEC both conceptually elegant and straightforward to implement. The mechanism by which representatives are modeled ensures that the forward process of FEC is fully transparent. **Second**, automated discovery of ***underlying data distribution***. The deterministic clustering reveals the latent relationships between pixels of the image data, capturing the varying semantic granularity that standard backbones might overlook. As depicted in Fig. 1d, FEC can learn to distinguish non-grid semantic regions autonomously *without explicit supervision of cluster assignments*. **Third**, ***ad-hoc interpretability***. If further inspecting the *cluster assignments* in each feature pooling and combining them together, FEC can interpret its prediction based on the aggregated clusters during forward process and allow users to intuitively view the semantic components. Such *ad-hoc* interpretability is valuable in safety-sensitive scenarios and provides a feasible way for humans to understand the *forward* process of feature extraction.

By answering questions ❶-❹, we formalize visual feature extraction within a neural clustering-based, fully transparent framework, bridging the gap between classic clustering algorithm and neural network interpretability. We provide a literature review and related discussions in §4. FEC represents an intuitive and versatile feature extractor, seamlessly compatible with established visual recognition models and tasks, requiring no modifications. Experimental results in §5.1 show FEC achieves **72.7**% `top-1` accuracy on ImageNet [24] with only 5.5M parameters. In §5.2, with the modeled representatives, FEC can interpret how it captures the data distribution. In §5.3 and §5.4, the transferability and versatility of FEC are validated on three fundamental recognition tasks. Finally, we draw conclusions in §6.

2. Existing Visual Feature Extractors as Fixed Grid-style Parsers

Problem Statement. Here we study the standard classification setting. Let $\mathcal{X}$ be the input space (*i.e.*, image space for visual recognition), and $\mathcal{Y}=\{\texttt{cat},\cdots,\texttt{dog}\}$ denote the set of semantic categories, *e.g.*, $|\mathcal{Y}|=1000$ for ImageNet-1K[24].
Standard Pipeline. The current common practice of classification is to decompose the deep neural network $h:\mathcal{X}\mapsto\mathcal{Y}$ into $f:\mathcal{X}\mapsto\mathcal{F}$ and $g:\mathcal{F}\mapsto\mathcal{Y}$ that $h=g\circ f$, where f and g denote the feature extractor and the classifier, respectively. Given an input image $\boldsymbol{X}$, f maps it into a d-dimensional representation space $\mathcal{F}\in\mathbb{R}^C$, *i.e.*, $\boldsymbol{f}=f(\boldsymbol{X})\in\mathbb{R}^C$; and g further predicts the class prediction $\hat{y}$ based on the intermediate feature $\boldsymbol{f}$, *i.e.*, $\hat{y}=g(\boldsymbol{f})\in\mathcal{Y}$. This work focuses on the f only.
ConvNets. Convolution-based feature extractor has dominated academia and industry for years, whose detailed architectures are reviewed as follows. Formally, given an input image $\boldsymbol{X}\in\mathbb{R}^{3\times H\times W}$, ConvNets extract feature embed-

dings $\{\boldsymbol{F}^l\}_{l=1}^4$, where the resolutions are $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$ of the original image, respectively. These four feature embeddings are generated by four separate stages, each containing grid-style feature pooling and encoding. Taking the 2^{nd} stage of ResNet18 [11] as an example, given $\boldsymbol{F}^1 \in \mathbb{R}^{64\times56\times56}$ from the 1^{st} stage, a low-dimensional feature map is generated as:

$$\hat{\boldsymbol{F}}^2 = \texttt{grid\_pool}(\boldsymbol{F}^1) \in \mathbb{R}^{128\times28\times28}, \tag{1}$$

where `grid_pool` denotes a convolutional layer with a stride of 2, which can also be implemented with max pooling, average pooling, *etc*. After that, feature encoding is performed to get the outputs of this stage:

$$\boldsymbol{F}^2 = \texttt{encode}(\hat{\boldsymbol{F}}^2) \in \mathbb{R}^{128\times28\times28}, \tag{2}$$

where `encode` denotes several convolutional layers which keep the output resolution consistent. This step is the key essential to distinguish different backbones, which is implemented as self-attention in ViTs and token mixers in MLPs. **ViTs** [13] and **MLPs** [14] both commence their operations by generating visual token embeddings for all non-overlap patches of an image:

$$\boldsymbol{E} = \texttt{token\_emb}(\boldsymbol{X}). \tag{3}$$

After which, ViTs use the [CLS] token to represent the whole image while MLPs take the average of all patch embeddings to do so. Since the sequence of image patches is used throughout the forward process of feature extraction, we also categorize them as the grid-style paradigm.

In general, existing backbones are built upon the computational modeling of rigid grids, which uses regular regions to represent an image. However, this paradigm underestimates the dynamic nature of visual scenes, assuming a spatial uniformity that clashes with the *underlying data distribution of pixels*. In addition, it overlooks the essence of human perception, which does not bind itself to rigid grids but instead fluidly navigates through semantic context [25].

After tackling question ❸, in the next section we will detail our clustering based transparent visual feature extractor, which serves as a solid response to question ❹.

3. Feature Extraction with Clustering (FEC)

Algorithmic Overview. FEC is a neural clustering-based framework for visual feature extraction, building upon the idea of selecting representatives hierarchically. Concretely, given an input image, FEC initiates with a standard convolution whose kernel-size and stride are set to be 4 and 4, respectively. Subsequent feature extraction builds upon these resulted 4×4 pixel patches. Afterward, FEC alternates between the following steps for each given input feature:

- Clustering-based Feature **Encoding**, *i.e.*, $\mathbb{R}^{C\times W\times H} \mapsto \mathbb{R}^{C\times W\times H}$. It divides pixels from feature maps into several non-overlap clusters, by projecting the pixel features into a similarity space and using adaptive (the stride and kernel-size are automatically selected to adapt to the desired resolution) average pooling to initialize the cluster centers. As such, cluster assignments can be obtained according to the similarity between pixels and centers. Then, the pixel features are aggregated to construct cluster representations. Subsequently, feature dispatching, which uses the aggregated center to redistribute pixel features within the cluster, is employed to encode pixel-level features, *i.e.*, information communication. Hence elements (pixels) inside the same cluster become more consistent in the feature space.
- Clustering-based Feature **Pooling**, *i.e.*, $\mathbb{R}^{C\times W\times H} \mapsto \mathbb{R}^{C'\times W/2\times H/2}$. Similar to the feature encoding process, this module uses clustering to obtain the cluster assignments. The difference is that it directly returns the cluster representations to form low-dimensional feature maps without encoding features. These strategies not only preserve the compositional structures of varying semantic levels but also seamlessly integrate the concept of clustering into the feed-forward feature extraction.

To sum up, we formalize our target task — extracting deep features for visual inputs — as *representative selection*. By doing so, the intermediate representatives can be a natural substitute for `grid_pool`. Since these representatives are computed from the context of each input, it can also be used to communicate information by *feature dispatching*, which serves the same purpose as the `encode` operation. Next, we will detail the operations of those essential parts of FEC.

Center Initialization. Given an input feature map $\boldsymbol{F}\in\mathbb{R}^{N\times C}$, where $N{=}W{\times}H$, we first project it into *key* and *value* spaces using 1×1 convolutional layers, resulting in $\boldsymbol{K} \in \mathbb{R}^{N\times C'}$ and $\boldsymbol{V} \in \mathbb{R}^{N\times C'}$, respectively. Here C' is a hyperparameter to control the dimension. We then initialize the cluster centers with their key and value features:

$$\begin{aligned}[\boldsymbol{C}_1^k;\cdots;\boldsymbol{C}_O^k] &= \texttt{ada\_pool}_O(\boldsymbol{K}) \in \mathbb{R}^{O\times C'},\\ [\boldsymbol{C}_1^v;\cdots;\boldsymbol{C}_O^v] &= \texttt{ada\_pool}_O(\boldsymbol{V}) \in \mathbb{R}^{O\times C'},\end{aligned} \tag{4}$$

where $\texttt{ada\_pool}_O$ refers generating O feature centers in the projected spaces using adaptive average pooling. As such, the centers are initialized adaptively for each input itself and gradients can be passed through all indices.

Representative Modeling. To **assign each element into a cluster**, the similarity matrix $\boldsymbol{M}$ is computed as:

$$\boldsymbol{M} = \langle \boldsymbol{K}, [\boldsymbol{C}_1^k;\cdots;\boldsymbol{C}_O^k] \rangle \in \mathbb{R}^{N\times O}, \tag{5}$$

where $\langle\cdot,\cdot\rangle$ denotes the cosine similarity. Each element is assigned to a cluster *exclusively* according to $\arg\max(\boldsymbol{M})$, resulting in an assignment matrix $\boldsymbol{A}$ which contains N one-hot vectors. With the **cluster assignments**, the deep features of the o-th representative (cluster) are aggregated by:

$$\boldsymbol{R}_o = \big(\boldsymbol{C}_o^v + \Sigma_{n=1}^N \boldsymbol{A}_{no}\boldsymbol{V}_n\big) / \big(1 + \Sigma_{n=1}^N \boldsymbol{A}_{no}\big) \in \mathbb{R}^{C'}. \tag{6}$$

So far, we've obtained the low-dimensional features $\boldsymbol{R} = [\boldsymbol{R}_1;\cdots;\boldsymbol{R}_O]$ (*i.e.*, representatives), which can seamlessly replace the `grid_pool` module in grid-style paradigm.

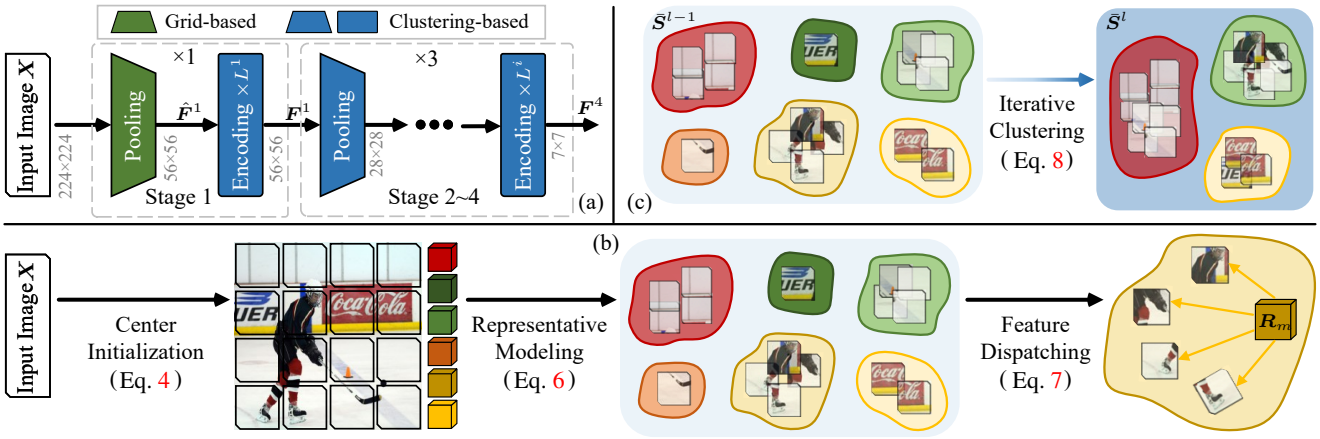

Figure 2. (a) Overall framework of FEC (§3). Each stage i contains L^i clustering-based `encode` layers. (b) Illustration of our clustering-based feature pooling and encoding. (c) The basic elements during FEC's forward process are *growing clusters* instead of *image patches*.

Feature Dispatching. With the insight that elements inside the same cluster shall have similar properties, we propose to propagate the information within each cluster to enhance this phenomenon. Concretely, we choose to achieve this with modulated propagation with respect to the similarity with the corresponding center [26, 27]. For element n in cluster o, we update its feature $\boldsymbol{F}_n \in \mathbb{R}^C$ by:

$$\boldsymbol{F}'_n = \boldsymbol{F}_n + \texttt{MLP}(\sigma(\alpha \boldsymbol{M}_{no} + \beta)\boldsymbol{R}_o) \in \mathbb{R}^C, \quad (7)$$

where σ denotes the sigmoid function. α and β are learnable parameters to scale and shift the similarity. The updated features $[\boldsymbol{F}'_1; \cdots; \boldsymbol{F}'_N]$ are the outputs of the `encode` operation in FEC (can be repeated many times). Since the center features are adaptively sampled from a group of elements, this dispatching process enables effective communication between elements within a cluster and those formed in the cluster center, leading to the overall understanding of the underlying data distribution and context of the image. From a higher perspective, FEC can be seen as an *exclusive* variant (non-overlap clusters) of self-attention [12], *e.g.*, center initialization *vs* key and value matrices, representative modeling *vs* attention scores, and feature dispatching *vs* weighted aggregation. More implementation details are left in the appendix.

Automated Discovery of Underlying Data Distribution. The cluster assignments elucidate not only the relationship between elements and their representatives but also the underlying data distribution within the feature maps. Pixels assigned the o-th centroid at the l-th level $\{n \mid \boldsymbol{A}^l_{no} = 1\}$ coalesce into a cluster (segment) $\boldsymbol{S}^l_o$, thereby $\{\boldsymbol{S}^l_o \mid 1 \le o \le O\}$ decomposing the entire feature map into O discernible segments at the l-th layer. By linking the clusters across sequential layers as follows:

$$\bar{\boldsymbol{S}}^l_h = \texttt{Union}(\{\boldsymbol{S}^{l-1}_o \mid \boldsymbol{A}^l_{oh} = 1\}), \quad (8)$$

we create a hierarchical pyramid $[\bar{\boldsymbol{S}}^1, \bar{\boldsymbol{S}}^2, \cdots, \bar{\boldsymbol{S}}^L]$ which coalesces pixels into increasingly larger segments, and explicitly reveals the underlying data distribution.

***Ad-hoc* Interpretability.** The described configuration facilitates a direct forward process $l = 1 \rightarrow L$ that yields a linked spatial decomposition $[\bar{\boldsymbol{S}}^1, \bar{\boldsymbol{S}}^2, \cdots, \bar{\boldsymbol{S}}^L]$, intuitively conveying image parsing to the observer. In contrast, earlier techniques (*e.g.*, Grad-CAM [28]) demand a retrospective process to accentuate the activated regions. These methods typically require complex post-processing to elucidate the concealed parsing mechanism. However, FEC is *mechanistically interpretable* because its forward process based on gradually growing clusters (segments) is *fully transparent*. See [29] for a more detailed discussion.

Versatility. After modeling the feature extraction process with representatives, one may wonder about the applicability of this new paradigm to dense prediction tasks. For instance, in the detection task, the training process of widely-used models like YOLO [30] and Faster-RCNN [31], relies on the grid-based label assignments (anchors). To keep the necessary grid-based information, we introduce the residual connection [11] to the feature of representatives:

$$\boldsymbol{R}_i = \texttt{ResConn}(\boldsymbol{F}_i) + \boldsymbol{R}_i \in \mathbb{R}^{C'}. \quad (9)$$

This modification allows each $\boldsymbol{R}_i \in \mathbb{R}^{C'}$ to represent both the rectangular regions in grid-style paradigms and the selected representatives in our clustering-based paradigm. As shown in Fig. 2a, we also adopt four stages during feature extraction as in existing standard backbones, resulting in the same resolutions of output features. In a nutshell, FEC marks a fundamental paradigm shift for visual feature extraction while retaining full compatibility with previous works. The computational pipeline is also detailed in §5.2.

Adaptation to Downstream Tasks. As aforementioned, with the introduction of residual connections, FEC can be seamlessly incorporated into dense prediction tasks like detection and segmentation without any architectural modifications. In terms of classification, we use the standard classification head over the final feature map $\boldsymbol{F}^L$, *i.e.*, a single-layer MLP, which takes the average of all representatives.

More details are left in the appendix. We expect the recent set-prediction architectures (*e.g.*, DETR [32]) can better utilize the modeled representatives and leave it as future works.

4. Related Work

Clustering stands as a foundational technique in machine learning that involves grouping akin data points together based on their intrinsic characteristics. Given numerous data, the goal of clustering is to model meaningful clusters (given pre-defined numbers or not), which can be viewed as the summary of the raw data. Subsequently, the similarities between each data and the cluster representations can be quantified. Clustering has been widely used in various fields, *e.g.*, scene understanding [33–36], point clouds [37–39], segmentation [40–44], and AI4Science [45–48].

In a departure from previous endeavors that harnessed clustering mechanisms as add-on heads to facilitate specific tasks, our approach pioneers the idea of learning universal visual representations from the clustering view. The proposed clustering-based feature extraction shares a kinship with the classic vision technique of clustering similar pixels [49]. Nonetheless, our innovation lies in its capacity to capture underlying data distribution and generate continuous representations for downstream tasks. Via reformulating the entire process of feature extraction as selecting representatives in a clustering fashion, FEC capitalizes on the robustness of end-to-end representation learning while preserving the transparent nature inherent to clustering. Inspired by biological systems that process inputs from different modalities simultaneously, Perceivers [50, 51] model a set of *latent vectors* correlated with inputs. FEC is motivated from the classic idea of clustering, and its intermediate elements have *explicit* meaning, *i.e.*, segments (Eq. 8).

Feature Extractors for Vision. Feature extraction is a pivotal aspect of machine learning, serving as a crucial step in transforming raw data into informative representations conducive to computational analysis. While some pre-deep learning methods [52, 53] also aim to get rid of grids, they are not *fully* end-to-end and do not scale well. Since the 2010s, convolutional networks [54] have reigned supreme in computer vision, with significant milestones like VGG [10], ResNet [11], ConvNeXt [55] and so on [56–58] consistently pushing the envelope by deepening and optimizing convolutional layers, enabling the extraction of hierarchical and abstract features. Innovations like depthwise convolution [59] and deformable convolution [60] have further enhanced feature extraction within these networks. More recently, the breakthroughs in NLP ushered in a revolution in feature extraction, introducing attention-based Transformer architectures [12]. Vision Transformer (ViT) is a time-honored work [13], which adopts self-attention mechanisms to image classification, achieving remarkable results. Note that ViTs' effectiveness often hinges on large-scale training datasets [61, 62]. Recent advancements have endeavored to bridge the gap between convolution and attention through hybrid models like CoAtNet [63] and Mobile-Former [64], which seamlessly combine both design paradigms. Besides, models employing multi-layer perceptrons (MLPs) [14, 15, 65–68] for spatial interactions and techniques such as shifting [69, 70] or pooling [71] for local context have emerged to explore the power of MLPs.

Though impressive, existing feature extractors are built upon rectangular receptive fields (as in CNN and variants, which represent an image as a grid of regular regions), or queries (as in ViTs, which incorporate an additional query token to represent an image), or rectangular image patches (as in MLPs, which use mixer layers to perform communication between patches). As far as we know, CoC [26] is the only backbone using a clustering algorithm. Compared to CoC, we want to highlight that our essential paradigm (*i.e.*, representing an image as representatives *vs* rectangular regions), goal (*i.e.*, *transparent* and *ad-hoc interpretable* backbone *vs* clustering based method for information exchange), and core techniques (*i.e.*, modeling *gradually growing* representatives *vs* modeling context clusters at specific resolution) are different. FEC takes the *first* step towards *fully* clustering based feature extraction, by reformulating its workflow as selecting representatives.

Neural Network Interpretability. The opaque nature of deep neural networks (DNNs) has posed challenges to their adoption in decision-critical applications, sparking a surge of interest in enhancing their transparency and interpretability. One key distinction in interpretable methods lies in whether they produce posterior explanations for pre-trained DNNs or aim to develop inherently interpretable DNNs from the outset. Posterior explanations, though prevalent, face criticism for their nature of approximation and limited capacity to truly elucidate the inner workings of DNNs [29, 72–75]. Representative works involve reverse-engineer importance values [28, 76–83] and sensitivities of inputs [84–87]. Recent advances in self-supervised learning, such as DINO [88, 89], have revealed emergent segmentation properties in ViTs. On the other hand, methods focused on ad-hoc explainability strive to incorporate more interpretable elements into black-box DNNs [90–92] or impose specific properties on the model's representations [93–95] to bolster interpretability. A notable instance is the deep nearest centroids [96], which introduces a nonparametric classifier and uses case-based reasoning, thereby presenting a novel and explainable paradigm for visual recognition.

In a related vein, CRATE [97] adopts the white-box Transformer [98] to delve into visual attention and the emergence of segmentation on the supervised classification task, showcasing considerable results. Nonetheless, we argue that CRATE falls within the realm of posterior explanations, whereas our FEC possesses an inherent capacity to eluci-

date the reasons behind its feature extraction process. Such innate transparency sets FEC apart from counterparts that solely offer *post-hoc* explainability. This work represents a small yet solid stride towards empowering the forward process of feature extraction with *ad-hoc* interpretability through an integrated clustering-based backbone and offers a fresh perspective on deep visual representations.

5. Experiment

FEC is proposed as the first framework to support feature extraction in a fully clustering-based manner. We evaluate the proposed backbone for four recognition tasks *viz*. image classification, object detection, semantic segmentation, and instance segmentation on three prevalent benchmarks (*i.e*., ImageNet-1K [24], MS COCO [99], and ADE20k [100]).

FEC combines *ad-hoc* interpretability with promising performance across diverse recognition tasks and datasets. Note that our objective is not to chase the state-of-the-art performance like ConvNeXt [55], but rather to assess FEC's capabilities (*e.g*., effectiveness, efficiency, transferability, *etc*.) through a comprehensive series of experiments.

5.1. Experiments on Image Classification

Dataset. ImageNet-1K [24] is a well-benchmarked image dataset. Following conventional procedures, it is divided into 1.28M/50K/100K images for `train`/`val`/`test`.

Training. We use *timm* [101] as our codebase and follow the standard training protocols as detailed in [26, 63, 71]. We use an AdamW [102] optimizer using a cosine decay learning rate scheduler and 5 epochs of warm-up. The momentum and weight decay are set to 0.9 and 0.05, respectively. A batch size of 1024 and an initial learning rate of 0.001 are used. More details are left in the appendix.

Test. Following [26], we use one input image scale of 224×224 with center cropping without any data augmentation. By default, the models are trained on 4 NVIDIA Tesla A100 GPUs with 80GB memory per card. Testing is conducted on the same machine.

Metric. We report the parameters used, FLOPs, and Top-1 classification accuracy on a single crop following previous works [11, 103]. Throughput (image/s), or FPS, is measured using the same script as in [103, 104] on a single V100 GPU using a batch size of 256.

Performance Comparison. Table 1 presents our classification results on ImageNet [24] `val`. As illustrated, FEC achieves competitive performance and efficiency, outperforming ResNet18 [11] by **2.9**% using less than half the number of parameters that ResNet18 has. ResNet achieves high FPS due to optimizations for convolutional operators.

5.2. Study of Ad-hoc Interpretability

We have empirically demonstrated FEC's effectiveness and efficiency in image classification. By redefining visual feature extraction as a clustering process, where representatives are iteratively selected during the forward process, FEC departs from the traditional grid-style paradigms. This shift suggests that FEC offers exceptional *ad-hoc* interpretability, a quality we will now explore further.

Table 1. Quantitative results on ImageNet-1K [24] `val` for **image classification** (§5.1). All models are trained and tested at 224×224 resolution, except ViT-B [13] and ViT-L [13].

	Method		#Param (M)	FLOPs (G)	Top-1 (%)↑	FPS (img/s)↑
Clu.	CoC-Tiny	[26]	5.3	1.1	71.8	1146.7
	CoC-Small	[26]	14.0	2.8	77.5	852.1
	CoC-Medium	[26]	27.9	5.9	81.0	345.7
MLP	ResMLP-12	[15]	15.0	3.0	76.6	1499.0
	ResMLP-24	[15]	30.0	6.0	79.4	741.6
	ResMLP-36	[15]	45.0	8.9	79.7	484.6
	MLP-Mixer-B/16	[14]	59.0	12.7	76.4	387.7
	MLP-Mixer-L/16	[14]	207.0	44.8	71.8	114.2
	gMLP-Ti	[105]	6.0	1.4	72.3	1440.0
	gMLP-S	[105]	20.0	4.5	79.6	650.5
Attention	ViT-B/16	[13]	86.0	55.5	77.9	86.4
	ViT-L/16	[13]	307	190.7	76.5	26.6
	PVT-Tiny	[106]	13.2	1.9	75.1	-
	PVT-Small	[106]	24.5	3.8	79.8	-
	Swin-Tiny	[103]	29	4.5	81.3	631.1
	Swin-Small	[103]	50	8.7	83.0	374.9
Conv.	ResNet18	[11]	12	1.8	69.8	4284.9
	ResNet50	[11]	26	4.1	79.8	1206.0
	ConvMixer$_{512/16}$	[107]	5.4	-	73.8	-
	ConvMixer$_{1024/12}$	[107]	14.6	-	77.8	-
	ConvMixer$_{768/32}$	[107]	21.1	-	80.2	139.7
Clu.	FEC-Small (Ours)		5.5	1.4	72.7$_{\pm 0.06}$	1042.9
	FEC-Base (Ours)		14.4	3.4	78.1$_{\pm 0.00}$	754.1
	FEC-Large (Ours)		28.3	6.5	81.2$_{\pm 0.06}$	342.1

We first detail the procedure to aggregate cluster assignments across layers. Given an image (224×224), a standard convolution-based pooling is used to generate a low-dimensional feature map (56×56) where each pixel represents a 4×4 region in the raw image. As for the first clustering-based pooling layer, those **pixel blocks** (56×56, each with 4×4 pixels) will be assigned to one of a total of 28×28 clusters $\{\boldsymbol{S}_i^1, 1 \leq i \leq 28\times 28\}$. Afterward, the subsequent pooling layers perform the same clustering procedure based on **previous clusters**. In this way, the clusters at different levels can be aggregated across layers according to Eq. 8, leading to 7×7 segments $\{\bar{\boldsymbol{S}}_i^4, 1 \leq i \leq 7\times 7\}$ in the last step. While humans can view and examine the final cluster assignments, 49 clusters are simply too many to comprehend for a 224×224 image. Therefore, we use K-Means to further reduce the number of clusters. Concretely, we use the deep features ($\boldsymbol{R}$ in Eq. 6) of those 49 clusters as their representations. Default hyperparameters in the scikit-learn [110] implementation are adopted. Medium filtering

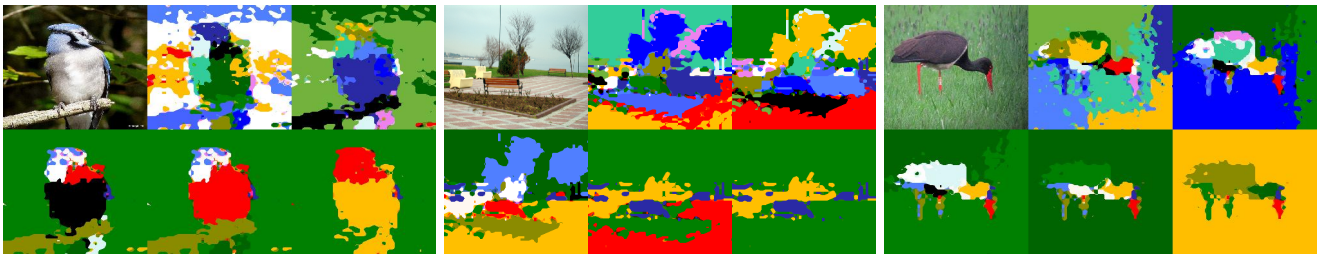

Figure 3. Inspection of the modeled representatives (§5.2) on ImageNet-1K [24] `val`. Different colored masks indicate different clusters. As the number of clusters decreases, each cluster tends to represent an entire object or a portion of an object, suggesting that FEC effectively captures the underlying data distribution of visual scenes.

Table 2. Quantitative results on ADE20K [100] `val` for **semantic segmentation** (§5.3). Semantic FPN [108] is adopted.

Backbone	#Param (M)	ADE20K mIoU(%)↑
ResNet18 [11]	15.5M	32.9
ResNet50 [11]	28.5M	36.7
PVT-Tiny [106]	17.0M	35.7
PVT-Small [106]	28.2M	39.8
CoC-Small/4 [26]	17.6M	36.6
CoC-Small/25 [26]	17.6M	36.4
CoC-Small/49 [26]	17.6M	36.3
CoC-Medium/4 [26]	31.5M	40.2
CoC-Medium/25 [26]	31.5M	40.6
CoC-Medium/49 [26]	31.5M	40.8
FEC-Small	9.1M	$35.3_{\pm 0.15}$
FEC-Base	18.0M	$37.7_{\pm 0.15}$
FEC-Large	31.9M	$40.5_{\pm 0.10}$

Table 3. Quantitative results on COCO [99] `val2017` for **object detection** and **semantic segmentation** (§5.4). We use Mask RCNN [109] to evaluate the performance of the proposed backbone on two tasks.

Backbone	#Param (M)	COCO AP^{box} ↑	COCO AP^{box}_{50} ↑	COCO AP^{box}_{75} ↑	COCO AP^{mask} ↑	COCO AP^{mask}_{50} ↑	COCO AP^{mask}_{75} ↑
ResNet18 [11]	31.2M	34.0	54.0	36.7	31.2	51.0	32.7
ResNet50 [11]	44.2M	38.0	58.6	41.4	34.4	55.1	36.7
PVT-Tiny [106]	32.9M	36.7	59.2	39.3	35.1	56.7	37.3
PVT-Small [106]	44.1M	40.4	62.9	43.8	37.8	60.1	40.3
CoC-Small/4 [26]	32.7M	35.9	58.3	38.3	33.8	55.3	35.8
CoC-Small/25 [26]	32.7M	37.5	60.1	40.0	35.4	57.1	37.9
CoC-Small/49 [26]	32.7M	37.2	59.8	39.7	34.9	56.7	37.0
CoC-Medium/4 [26]	46.7M	38.6	61.1	41.5	36.1	58.2	38.0
CoC-Medium/25 [26]	46.7M	40.1	62.8	43.6	37.4	59.9	40.0
CoC-Medium/49 [26]	46.7M	40.6	63.3	43.9	37.6	60.1	39.9
FEC-Small	24.3M	$35.6_{\pm 0.06}$	$57.5_{\pm 0.10}$	$38.2_{\pm 0.15}$	$33.6_{\pm 0.10}$	$54.7_{\pm 0.10}$	$35.7_{\pm 0.15}$
FEC-Base	33.1M	$37.9_{\pm 0.06}$	$60.1_{\pm 0.06}$	$40.8_{\pm 0.15}$	$35.5_{\pm 0.06}$	$57.2_{\pm 0.06}$	$37.8_{\pm 0.10}$
FEC-Large	47.1M	$39.9_{\pm 0.10}$	$62.5_{\pm 0.06}$	$43.2_{\pm 0.12}$	$37.3_{\pm 0.06}$	$59.5_{\pm 0.15}$	$39.5_{\pm 0.12}$

is adopted for better visualization.

After the above introduction, we visualize cluster assignments in Fig. 3 to clarify FEC's principles. Two remarkable observations justify the *ad-hoc* interpretability and effectiveness of FEC: **i**) Across successive pooling layers, the clustering-based feature extraction progressively abstracts pixel blocks, mirroring human cognitive processes and providing a natural representation of visual data. **ii**) In final cluster assignments, we find consistent semantic representations, with clusters often corresponding to coherent objects or object parts. This aligns FEC with human perception and enhances feature interpretability, linking clusters to identifiable image elements.

5.3. Experiments on Semantic Segmentation

Dataset. ADE20K [100] is a prominent semantic segmentation dataset renowned for its extensive collection of images. It encompasses 150 object categories with a total of 25,000 images (20K/2K/3K for `train`/`val`/`test`).

Training. We use *mmsegmentation* [111] as our codebase and follow the standard training protocols as detailed in [26, 106]. We evaluate FEC on a classic segmentation method, *i.e.*, Semantic FPN [108], for its high efficiency. We utilize an AdamW [102] optimizer for 80K iterations using the polynomial decay learning rate scheduler with a power of 0.9. We set the batch size as 16 and the initial learning rate as 0.0001. More details are left in the appendix.

Test. We use one input image scale with a shorter side of 512 during inference without applying data augmentation.

Metric. Mean intersection-over-union (mIoU) is reported.

Performance Comparison. From table 2 we can observe that our approach yields remarkable performance on dense prediction tasks. For example, our FEC-Base outperforms ResNet18, ResNet50, and PVT-Tiny by **4.8**%, **1.0**%, and **2.0**% in terms of mIoU, respectively. In addition, FEC-Small can even outperform ResNet18, *i.e.*, **35.3**% *vs* **32.9**% in terms of mIoU. Such performance gains remain consistent as in the classification task and are significant considering the number of used parameters.

5.4. Experiments on Object Detection and Instance Segmentation

Dataset. The MS COCO 2017 benchmark[99] features a diverse collection of over 200K high-quality images, annotated across 80 common object categories in daily contexts. It is divided into 118K/5K/41K images for `train`/`val`/`test`.

Training. We use *mmdetection* [112] as our codebase and follow the training protocols as in [26]. We verify the effectiveness of FEC backbones on top of a milestone model, *i.e.*, Mask R-CNN [109]. AdamW [102] optimizer is used

Table 4. A set of ablative experiments (§5.1) on ImageNet [24] `val`, ADE20K [100] `val`, and COCO [99] `val2017`. The adopted hyperparameters are marked in red.

similarity measurement	#Param. (M)	Top-1 (%)↑	Top-5 (%)↑
Euclidean	5.46	72.2	90.9
Dot Product	5.46	72.7	91.1
Cosine	5.46	72.7	91.2

(a) Similarity measurement (Eq. 5)

feature dispatching	#Param. (M)	Top-1 (%)↑	Top-5 (%)↑
w/o α and β	5.46	72.1	90.8
w/o α	5.46	72.4	91.0
w/o β	5.46	72.2	90.9
Ours (Eq. 7)	5.46	72.7	91.2

(b) Parameters used in feature dispatching (§3)

feature dimension	#Param (M)	ADE20K mIoU(%)↑
(48, 48, 96, 96)	8.6M	34.9
(72, 72, 144, 144)	8.9M	35.2
(96, 96, 192, 192)	9.1M	35.3
(120, 120, 240, 240)	9.4M	35.5

(c) Number of channels in `encode` (Eq. 4)

feature dimension	#Param (M)	COCO AP^{box}↑	AP^{box}_{50}↑	AP^{box}_{75}↑	AP^{box}_{s}↑	AP^{box}_{m}↑	AP^{box}_{l}↑	COCO AP^{mask}↑	AP^{mask}_{50}↑	AP^{mask}_{75}↑
(48, 48, 96, 96)	23.8M	34.9	57.2	36.9	20.3	36.9	45.8	33.4	54.5	35.0
(72, 72, 144, 144)	24.0M	35.2	57.2	37.7	20.1	37.4	46.7	33.5	54.5	35.6
(96, 96, 192, 192)	24.3M	35.6	57.5	38.2	20.9	37.7	46.8	33.6	54.7	35.7
(120, 120, 240, 240)	24.5M	35.5	57.5	37.8	20.7	37.6	47.4	33.8	54.7	35.9

(d) Number of channels in `encode` (Eq. 4)

for 12 epochs (1× scheduler) and initialize the backbone with ImageNet [24] pre-trained weights. A batch size of 16 and an initial learning rate of 0.0002 are used. More details are left in the appendix.

Test. We use one input image scale with a shorter side of 800 during inference without applying data augmentation.

Metric. We report average precision (AP), AP_{50}, AP_{75} for both object detection and instance segmentation.

Performance Comparison. Table 3 confirms again the transferability and versatility of FEC for the common instance-centric recognition tasks. On top of a relatively conservative baseline, *i.e.*, Mask-RCNN [109], our algorithm outperforms both types of rivals. For instance, the performance of FEC-Tiny is clear ahead compared to ResNet-18 [11] (*i.e.*, **35.6**% AP^{box} *vs* 34.0% AP^{box} and **33.6**% AP^{mask} *vs* 31.2% AP^{mask}), and FEC-Base achieves promising gains of **1.2**% AP^{box} and **0.4**% AP^{mask} against PVT-Tiny [106].

5.5. Diagnostic Experiment

This section ablates FEC's key components on ImageNet [24] `val`, ADE20K [100] `val`, and COCO [99] `val2017`. All experiments use the FEC-Small model.

Similarity Measurement. We first examine the similarity measurement in the clustering process (Eq. 5) by contrasting it with several standard similarity (distance) functions, *i.e.*, Euclidean distance and dot product (unnormalized cosine similarity). As shown in Table 4a, dot product and cosine similarity shows better performance than Euclidean distance, *e.g.*, **72.7**% *vs* 72.2% in terms of Top-1 accuracy. Cosine similarity is adopted by default.

Feature Dispatching. We then investigate the effects of the learnable parameters for similarity scaling and shifting (Eq. 7). In table 4b, the first three lines indicate the removal of the corresponding hyperparameters. We find that introducing these two factors can bring minor performance improvements, *e.g.*, **72.7**% *vs* 72.1% in terms of Top-1 accuracy and **91.2**% *vs* 90.8% in terms of Top-5 accuracy.

Feature Dimension. Last, we ablate the feature dimension C' in the encoding layer (Eq. 4). In the pooling layer, C' is used to increase the number of channels for the next stage. In the encoding layer, C' is used to control the complexity of the projected *key* and *value* space (the output channels can be further adjusted by the `MLP` during feature dispatching). The results are summarized in Table 4c and Table 4d. The four values of each row correspond to the four encoding phases. In a nutshell, the performance for downstream tasks improves as the feature dimension increases, *e.g.*, **35.5**% *vs* 34.9% in terms of mIoU, **35.6**% *vs* 34.9% in terms of AP^{box}, and **33.8**% *vs* 33.4% in terms of AP^{mask}, at the expense of parameter growth. For a fair comparison with previous work [26], the settings in the 3^{rd} row are adopted.

6. Conclusion and Discussion

In machine vision, extracting powerful distributed representations for visual data while preserving the interpretability and explicit modeling of data distribution, presents a perennial challenge. While the community has witnessed great strides in visual backbones, top-leading solutions remain bound to the computational confines of processing rectangular image patches — a stark contrast to the pixel organization observed in human perception. This study represents a significant leap forward by reformulating feature extraction as representative selection, resulting in a transparent and interpretable feature extractor. Our goal is to pave the way for vision systems that not only excel in performance but also possess an intrinsic understanding of the underlying data distribution of visual scenes, thereby enhancing both trust and clarity in their application.

Acknowledgements This work was supported by the National Natural Science Foundation of China (No. 62372405) and CCF-Tencent Open Fund.

References

[1] Wesley E Snyder and Hairong Qi. *Machine vision*, volume 1. Cambridge University Press, 2004. 1

[2] Christopher M Bishop. Pattern recognition. *Machine learning*, 128(9), 2006.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 1

[4] Pauline C Ng and Steven Henikoff. Sift: Predicting amino acid changes that affect protein function. *Nucleic acids research*, 31(13):3812–3814, 2003. 1

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417, 2006.

[7] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *ECCV*, pages 778–792, 2010.

[8] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, pages 2564–2571, 2011.

[9] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *ICCV*, pages 2548–2555, 2011. 1

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1, 5

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 3, 4, 5, 6, 7, 8

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 4, 5

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1, 3, 5, 6

[14] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. In *NeurIPS*, pages 24261–24272, 2021. 1, 3, 5, 6

[15] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE TPAMI*, 45(4): 5314–5321, 2022. 1, 5, 6

[16] Rafael C Gonzales and Paul Wintz. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 1987. 2

[17] Kenneth R Castleman. *Digital image processing*. Prentice Hall Press, 1996.

[18] David Beymer and Tomaso Poggio. Image representations for visual learning. *Science*, 272(5270):1905–1909, 1996. 2

[19] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018. 2

[20] Andreas Weinlich, Peter Amon, Andreas Hutter, and Andre Kaup. Probability distribution estimation for autoregressive pixel-predictive image coding. *IEEE TIP*, 25(3): 1382–1395, 2016. 2

[21] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94 (2):115, 1987. 2

[22] Brent R Beutter and Leland S Stone. Motion coherence affects human perception and pursuit similarly. *Visual neuroscience*, 17(1):139–153, 2000.

[23] Johannes Bill, Hrag Pailian, Samuel J Gershman, and Jan Drugowitsch. Hierarchical structure is employed by humans during visual motion perception. *Proceedings of the National Academy of Sciences*, 117(39):24581–24589, 2020. 2

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 2, 6, 7, 8

[25] Joost Rommers, Ton Dijkstra, and Marcel Bastiaansen. Context-dependent semantic processing in the human brain: Evidence from idiom comprehension. *Journal of Cognitive Neuroscience*, 25(5):762–776, 2013. 3

[26] Xu Ma, Yuqian Zhou, Huan Wang, Can Qin, Bin Sun, Chang Liu, and Yun Fu. Image as set of points. In *ICLR*, 2023. 4, 5, 6, 7, 8

[27] James Liang, Yiming Cui, Qifan Wang, Tong Geng, Wenguan Wang, and Dongfang Liu. Clusterfomer: Clustering as a universal visual learner. In *NeurIPS*, 2023. 4

[28] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 4, 5

[29] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85, 2022. 4, 5

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 4

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 4

[32] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 5

[33] Lin Li, Long Chen, Hanrong Shi, Hanwang Zhang, Yi Yang, Wei Liu, and Jun Xiao. Nicest: Noisy label correction and training for robust scene graph generation. *arXiv preprint arXiv:2207.13316*, 2022. 5

[34] Lin Li, Long Chen, Yifeng Huang, Zhimeng Zhang, Songyang Zhang, and Jun Xiao. The devil is in the labels: Noisy label correction for robust scene graph generation. In *CVPR*, pages 18869–18878, 2022.

[35] Guikun Chen, Lin Li, Yawei Luo, and Jun Xiao. Addressing predicate overlap in scene graph generation with semantic granularity controller. In *ICME*, pages 78–83, 2023.

[36] Lin Li, Guikun Chen, Jun Xiao, Yi Yang, Chunping Wang, and Long Chen. Compositional feature augmentation for unbiased scene graph generation. In *ICCV*, pages 21685–21695, 2023. 5

[37] Junbo Yin, Dingfu Zhou, Liangjun Zhang, Jin Fang, Cheng-Zhong Xu, Jianbing Shen, and Wenguan Wang. Proposalcontrast: Unsupervised pre-training for lidar-based 3d object detection. In *ECCV*, pages 17–33, 2022. 5

[38] Tuo Feng, Wenguan Wang, Xiaohan Wang, Yi Yang, and Qinghua Zheng. Clustering based point cloud representation learning for 3d analysis. In *ICCV*, pages 8283–8294, 2023.

[39] Tuo Feng, Ruijie Quan, Xiaohan Wang, Wenguan Wang, and Yi Yang. Interpretable3d: An ad-hoc interpretable classifier for 3d point clouds. In *AAAI*, 2024. 5

[40] Wenguan Wang, Jianbing Shen, Fatih Porikli, and Ruigang Yang. Semi-supervised video object segmentation with super-trajectories. *IEEE TPAMI*, 41(4):985–998, 2018. 5

[41] Chen Liang, Wenguan Wang, Jiaxu Miao, and Yi Yang. Gmmseg: Gaussian mixture based generative semantic segmentation models. In *NeurIPS*, pages 31360–31375, 2022.

[42] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking semantic segmentation: A prototype view. In *CVPR*, 2022.

[43] James Chenhao Liang, Tianfei Zhou, Dongfang Liu, and Wenguan Wang. Clustseg: Clustering for universal segmentation. In *ICML*, pages 20787–20809, 2023.

[44] Liulei Li, Wenguan Wang, Tianfei Zhou, Jianwu Li, and Yi Yang. Unified mask embedding and correspondence learning for self-supervised video segmentation. In *CVPR*, pages 18706–18716, 2023. 5

[45] Isabella N Grabski, Kelly Street, and Rafael A Irizarry. Significance analysis for clustering with single-cell rna-sequencing data. *Nature Methods*, 20(8):1196–1202, 2023. 5

[46] Inigo Barrio-Hernandez, Jingi Yeo, Jürgen Jänes, Milot Mirdita, Cameron LM Gilchrist, Tanita Wein, Mihaly Varadi, Sameer Velankar, Pedro Beltrao, and Martin Steinegger. Clustering predicted structures at the scale of the known protein universe. *Nature*, 622(7983):637–645, 2023.

[47] Hannah K Wayment-Steele, Adedolapo Ojoawo, Renee Otten, Julia M Apitz, Warintra Pitsawong, Marc Hömberger, Sergey Ovchinnikov, Lucy Colwell, and Dorothee Kern. Predicting multiple conformations via sequence clustering and alphafold2. *Nature*, 625(7996):832–839, 2024.

[48] Ruijie Quan, Wenguan Wang, Fan Ma, Hehe Fan, and Yi Yang. Clustering for protein representation learning. In *CVPR*, 2024. 5

[49] Ren and Malik. Learning a classification model for segmentation. In *ICCV*, pages 10–17, 2003. 5

[50] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *ICML*, pages 4651–4664, 2021. 5

[51] Joao Carreira, Skanda Koppula, Daniel Zoran, Adria Recasens, Catalin Ionescu, Olivier Henaff, Evan Shelhamer, Relja Arandjelovic, Matt Botvinick, Oriol Vinyals, et al. Hip: Hierarchical perceiver. *arXiv preprint arXiv:2202.10890*, 2022. 5

[52] Chunhui Gu, Joseph J Lim, Pablo Arbeláez, and Jitendra Malik. Recognition using regions. In *CVPR*, pages 1030–1037, 2009. 5

[53] Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, pages 430–443, 2012. 5

[54] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 5

[55] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11976–11986, 2022. 5, 6

[56] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 5

[57] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.

[58] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019. 5

[59] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. In *CVPR*, 2017. 5

[60] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *CVPR*, 2017. 5

[61] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022. 5

[62] Hanan Gani, Muzammal Naseer, and Mohammad Yaqub. How to train vision transformer on small-scale datasets? In *BMVC*, 2022. 5

[63] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. In *NeurIPS*, pages 3965–3977, 2021. 5, 6

[64] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. In *CVPR*, pages 5270–5279, 2022. 5

[65] Jianyuan Guo, Yehui Tang, Kai Han, Xinghao Chen, Han Wu, Chao Xu, Chang Xu, and Yunhe Wang. Hire-mlp: Vision mlp via hierarchical rearrangement. In *CVPR*, pages 826–836, 2022. 5

[66] Qibin Hou, Zihang Jiang, Li Yuan, Ming-Ming Cheng, Shuicheng Yan, and Jiashi Feng. Vision permutator: A permutable mlp-like architecture for visual recognition. *IEEE TPAMI*, 45(1):1328–1334, 2022.

[67] Yehui Tang, Kai Han, Jianyuan Guo, Chang Xu, Yanxi Li, Chao Xu, and Yunhe Wang. An image patch is a wave: Phase-aware vision mlp. In *CVPR*, pages 10935–10944, 2022.

[68] Shoufa Chen, Enze Xie, Chongjian Ge, Runjian Chen, Ding Liang, and Ping Luo. Cyclemlp: A mlp-like architecture for dense prediction. In *ICLR*, 2022. 5

[69] Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, 2021. 5

[70] Dongze Lian, Zehao Yu, Xing Sun, and Shenghua Gao. As-mlp: An axial shifted mlp architecture for vision. In *ICLR*, 2021. 5

[71] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *CVPR*, pages 10819–10829, 2022. 5, 6

[72] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. In *IJCAI*, 2019. 5

[73] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[74] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[75] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. In *NeurIPS*, 2019. 5

[76] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009. 5

[77] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[78] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[79] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015.

[80] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[81] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[82] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

[83] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017. 5

[84] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *KDD*, 2016. 5

[85] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *ICLR*, 2017.

[86] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017.

[87] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NeurIPS*, 2017. 5

[88] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 5

[89] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 5

[90] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In *NeurIPS*, 2018. 5

[91] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *ICML*, 2018.

[92] Tong Wang. Gaining free or low-cost interpretability with interpretable partial substitute. In *ICML*, 2019. 5

[93] Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. Spine: Sparse interpretable neural embeddings. In *AAAI*, 2018. 5

[94] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, 2016.

[95] Seungil You, David Ding, Kevin Canini, Jan Pfeifer, and Maya Gupta. Deep lattice networks and partial monotonic functions. In *NeurIPS*, 2017. 5

[96] Wenguan Wang, Cheng Han, Tianfei Zhou, and Dongfang Liu. Visual recognition with deep nearest centroids. In *ICLR*, 2023. 5

[97] Yaodong Yu, Tianzhe Chu, Shengbang Tong, Ziyang Wu, Druv Pai, Sam Buchanan, and Yi Ma. Emergence of segmentation with minimalistic white-box transformers. *arXiv preprint arXiv:2308.16271*, 2023. 5

[98] Yaodong Yu, Sam Buchanan, Druv Pai, Tianzhe Chu, Ziyang Wu, Shengbang Tong, Benjamin D Haeffele, and Yi Ma. White-box transformers via sparse rate reduction. *arXiv preprint arXiv:2306.01129*, 2023. 5

[99] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6, 7, 8

[100] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 6, 7, 8

[101] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 6

[102] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6, 7

[103] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 6

[104] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357, 2021. 6

[105] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. In *NeurIPS*, pages 9204–9215, 2021. 6

[106] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pages 568–578, 2021. 6, 7, 8

[107] Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022. 6

[108] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, pages 6399–6408, 2019. 7

[109] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 7, 8

[110] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011. 6

[111] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020. 7

[112] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 7