2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

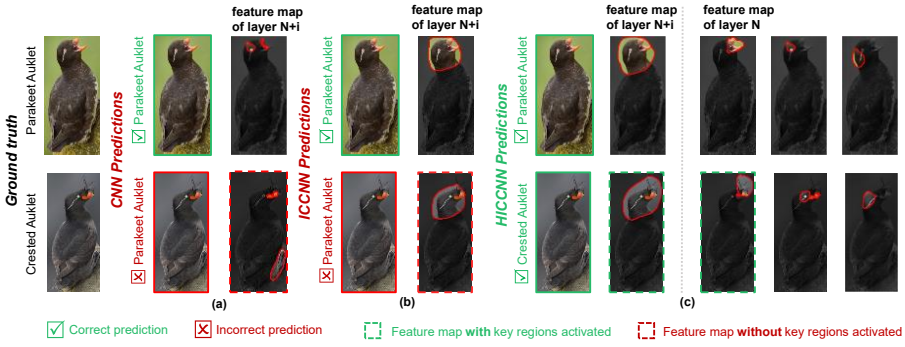
HICCNN: A Hierarchical Approach to Enhancing Interpretability in Convolutional Neural Networks

Yinze Luo[1], Yuxiang Luo[1], Bo Peng[2]✉, Lijun Sun[1]✉

[1]Tongji University, Shanghai 201804, China
[2]Shanghai Municipal Bureau of Public Security, Shanghai, China
2250409@tongji.edu.cn

Abstract. Convolutional Neural Networks (CNNs) frequently exhibit limited interpretability, which presents significant challenges to their deployment in high-stakes applications. Although existing methods such as ICCNN incorporate interpretability mechanisms, these approaches are typically confined to a single network layer and thus fail to capture the hierarchical nature of visual semantics. To overcome this limitation, we propose **H**ierarchical **I**nterpretable **C**ompositional **C**onvolutional **N**eural **N**etworks, a novel approach that facilitates layer-wise hierarchical interpretability without requiring any modifications to the original network architecture. Specifically, our method allows CNNs to learn semantically meaningful and fine-grained features in a structured hierarchy, thereby enhancing the model's interpretability. Extensive quantitative experiments demonstrate that our model not only offers superior interpretability compared to existing methods, but also enhances classification performance—particularly in complex multi-class tasks—by effectively leveraging the hierarchical compositional structure of the learned features. Moreover, we compare our method against Grad-CAM and demonstrate that our model achieves comparable semantic localization quality while offering built-in interpretability during inference, thereby eliminating the need for additional post-hoc explanation modules.

Keywords: Hierarchical interpretability, Neural network interpretability, Convolutional neural networks



✉Corresponding Authors

The work is partially supported by the National Nature Science Foundation of China (No. 62376199, 62206170).

Fig. 1. Comparison of three models—(a) a standard CNN, (b) ICCNN, and (c) the proposed HICCNN—on distinguishing two visually similar bird species. The CNN and ICCNN models often **fail** to accurately highlight key semantic regions such as the *Crested Auklet's unique feather crest*, leading to misclassification. HICCNN leverages a hierarchical semantic structure to consistently capture discriminative parts.

1 Introduction

Convolutional Neural Networks (CNNs) have achieved remarkable success in visual recognition tasks largely owing to their hierarchical architecture, which facilitates the learning of multi-level feature representations. However, their limited interpretability remains a significant obstacle, especially in critical domains where model transparency and trustworthiness are paramount.

Existing efforts to enhance CNN interpretability primarily focus on visualizing network activations or identifying pixel-level input-output correlations. Despite these advances, rendering the intermediate features semantically interpretable remains a fundamental challenge. Addressing this issue is crucial for aligning learned representations with human cognition, thereby yielding explanations that are both intuitive and verifiable.

Building upon the Interpretable Compositional CNN (ICCNN) proposed by Shen et al. [1], which learns meaningful patterns in a single intermediate layer without relying on part or region annotations, we propose a novel approach that extends interpretability across multiple layers of a CNN. Specifically, we introduce a hierarchical interpretability framework that enables the model to capture fine-grained and compositional semantics from shallow to deep layers, thereby mirroring the human perceptual processes. This not only enhances interpretability but also improves performance, particularly in fine-grained multi-class classification tasks.

As illustrated in Figure 1(c), our method enables deeper layers to focus on broader object parts (e.g., bird head or body), while shallower layers attend to more localized details (e.g., eyes or beak). Feature maps across different layers are organized into distinct semantic clusters, forming a coherent hierarchical structure that enhances both transparency and reasoning. For instance, in distinguishing Crested Auklet from Parakeet Auklet, the model effectively captures the presence of the Crested Auklet's unique feather crest—a key discriminative feature—without relying on any part-level supervision.

To achieve this, we incorporate an end-to-end training framework that introduces a novel hierarchical loss on top of ICCNN's original objective. This loss leverages activation regions from deep layers to guide shallow-layer feature grouping, ensuring that low-level features align with higher-level semantics. As a result, the model learns to organize semantic features across layers automatically, **without** requiring any manual annotations.

We evaluate our approach on multiple CNN backbones, including VGG, ResNet, and DenseNet, and validate its effectiveness on the CUB-200-2011 and NABirds da-

tasets. Ablation experiments were conducted by removing the hierarchical interpretability and semantic clustering modules to produce ICCNN and standard CNN variants. These variants, along with the full model, were then trained and evaluated on the same datasets under identical conditions. Experimental results demonstrate that our model not only significantly enhances interpretability—measured by quantitative metrics—but also substantially improves accuracy compared to standard CNN baselines in fine-grained classification tasks.

In summary, this paper makes three key contributions:

1) We propose a **hierarchical interpretability framework** that aligns semantic features across layers, without modifying the network architecture.
2) Our model learns **finer-grained semantic representations**, leading to improved performance on multi-class classification tasks.
3) We design a **hierarchical loss function** that guides shallow-layer features using deep-layer activations, eliminating the need for part-level annotations.

2 Related Work

Learning interpretable features has long been a key focus in deep learning research. Early efforts approached this problem by introducing architectural innovations aimed at disentangling semantic representations. For example, Capsule Networks (CapsNets) [2] employed dynamic routing mechanisms between capsule structures to capture part-whole relationships. This design offers a degree of interpretability grounded in spatial hierarchies. Similarly, InfoGAN [3] and β-VAE [4] focused on generative models, learning disentangled latent variables that encode interpretable semantics. However, these models do not ensure that individual convolutional filters correspond to specific, localized visual patterns, limiting their applicability to standard CNN-based vision tasks.

To bridge this gap, researchers turned to filter-level interpretability in CNNs. One line of work proposed training class-specific filters [5][6], where each filter is encouraged to activate for a particular object category. Although this strategy improved discriminability and brought interpretability closer to task semantics, it remained insufficient for fine-grained understanding, as filters often failed to capture meaningful object parts or localized image regions. To address this, Zhang et al. [7] proposed Interpretable CNNs (ICNNs), which introduced an information-theoretic loss to enforce each intermediate-layer filter to correspond to a distinct object part. This approach significantly improved the part-level interpretability of CNNs, but it was constrained by its reliance on compact, blob-like activations, and struggled to model spatially diffuse or structurally ambiguous regions.

Building on these insights, Shen et al. [1] introduced the Interpretable Compositional CNN (ICCNN), which extends interpretability beyond object parts to more general visual regions without requiring part or region annotations. By learning compositional features across filters in intermediate layers, ICCNN overcame the structural rigidity of ICNNs and marked a significant advancement in aligning CNN representations with

human-recognizable patterns. Complementing this compositional perspective, hierarchical feature learning has also emerged as an effective paradigm for enhancing interpretability. Rangadurai et al. [9] proposed the Hierarchical Structured Neural Network (HSNN), which incorporates Modular Neural Networks (MoNNs) and a hierarchical indexing scheme to support efficient feature interaction and computation reuse. This architecture has demonstrated strong performance in handling occlusion, multi-scale structures, and large-scale retrieval tasks, underscoring the potential of hierarchical designs in complex real-world scenarios.

3 Method

The method enhances the interpretability of convolutional neural networks (CNNs) by enabling the network to automatically learn specific semantic patterns from intermediate feature maps.

The approach consists of three key components: (1) Grouping and clustering feature maps in intermediate layers to associate them with interpretable visual patterns. (2) Propagating deeper-layer group semantics to guide clustering in shallower layers, creating a hierarchical structure. (3) Using a surrogate loss function to enforce this hierarchical interpretability while maintaining the network's performance.

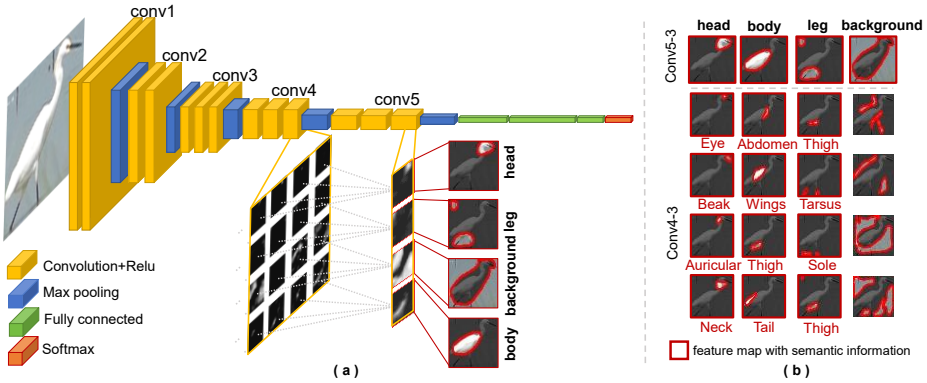


Fig. 2. (a) The architecture of the VGG16 convolutional network is shown here. We perform group clustering on the feature maps of layers *conv4-3* and *conv5-3*, which enhances the interpretability of these feature maps. Additionally, the feature map information from earlier layers is allocated into corresponding deep-layer feature groups, achieving a hierarchical interpretability effect. (b) The detailed visualization of activated regions shows that different parts of the image trigger specific patterns. The conv5-3 layer captures coarse semantic regions like the head and body, while the conv4-3 layer focuses on finer details such as the beak and wings within those regions.

3.1 Grouping and Clustering

Our approach builds upon the methodology introduced by Shen et al. [1], which focuses on enhancing the interpretability of convolutional neural networks (CNNs) by learning

compositional features. We modify intermediate layers of the CNN by performing group clustering on the feature maps. This ensures that filters within the same group activate similar visual patterns, while filters across different groups represent distinct patterns.

Using VGG16 as an example, we perform filter clustering on the conv4-3 and conv5-3 layers. As the depth of CNNs increases, learned semantics become more abstract, and the focus shifts from local details to global structures. These layers are selected for group clustering to strike a balance between fine-grained feature capture and high-level semantic representation. Filters in deeper layers (e.g., conv5-3) capture global object parts (e.g., head or body), while those in shallower layers (e.g., conv4-3) focus on finer details (e.g., eyes or wings). A custom similarity metric ensures that filters within the same group correspond to similar visual concepts.

The filters in each layer are divided into K groups $(A_1, A_2, \dots, A_K)$, where $A_1 \cup A_2 \cup \dots \cup A_K = \Omega$ (the set of all filters), and $A_i \cap A_j = \emptyset$ for $i \neq j$. To enforce semantic coherence, we first define the normalized cross-correlation between filters i and j across all training images $\mathcal{I}$:

$$s_{ij} = \frac{\mathrm{cov}(X_i, X_j)}{\sigma_i \sigma_j} + 1 \tag{1}$$

Where $\mathrm{X_i} = \{\mathrm{x_i^I}\}_{\mathrm{I}\in\mathcal{I}}$ represents the activation maps of filter i after ReLU and spatial average pooling, $\mathrm{cov}(\mathrm{X_i}, \mathrm{X_j})$is the empirical covariance, and $\sigma_\mathrm{i}, \sigma_\mathrm{j}$ are the standard deviations of $\mathrm{X_i}$ and X_j, respectively. This metric quantifies co-activation patterns, with $\mathrm{s_{ij}} > 1$ indicating semantic synergy.

The grouping loss maximizes intra-group cohesion while suppressing inter-group interference:

$$L_{\mathrm{group}}(\theta, A) = -\sum_{k=1}^{K} \frac{S_{\mathrm{within},k}}{S_{\mathrm{all},k}} \tag{2}$$

Where measures intra-group similarity for cluster A_k:

$$S_{\mathrm{within},k} = \sum_{i,j\in A_k} s_{ij} \tag{3}$$

and $S_{\mathrm{all},k}$ quantifies inter-group divergence by aggregating similarities between A_k and all filters:

$$S_{\mathrm{all},k} = \sum_{i\in A_k, j\in\Omega} s_{ij} \tag{4}$$

This loss encourages filters within the same group to have high similarity and filters from different groups to have low similarity.

For multi-category classification, the loss is adjusted to ensure that filters in different groups represent object parts or image regions specific to different categories. The multi-category loss is:

$$L_{\mathrm{multi}}(\theta) = -\sum_{c=1}^{C} \sum_{p,q\in I_c} \frac{s_{pq}}{\sum_{p\in I_c, q\in I} s_{pq}} \tag{5}$$

where $s_{pq} = \left(z^{(p)}\right)^{\top} z^{(q)}$ measures activation consistency between images p and q, with $z^{(p)}$ being the normalized group activation vector for image p:

$$z_k^{(p)} = \frac{1}{|A_k|m} \sum_{i \in A_k} \sum_{u=1}^{m} x_{i,u}^{(p)} \tag{6}$$

The final objective function combines the classification loss L_{cls} with the grouping loss:

$$L(\theta, A) = \lambda \cdot \text{Loss}(\theta, A) + \beta \cdot L_{\text{multi}}(\theta) + \frac{1}{n} \sum_{I \in I} L_{\text{cls}}(y_I, y_I^*; \theta) \tag{7}$$

This method enables us to meaningfully group filters while maintaining the network's performance, ultimately improving the interpretability of CNNs by ensuring that each group of filters corresponds to a specific, interpretable visual concept.

3.2 Hierarchical construction

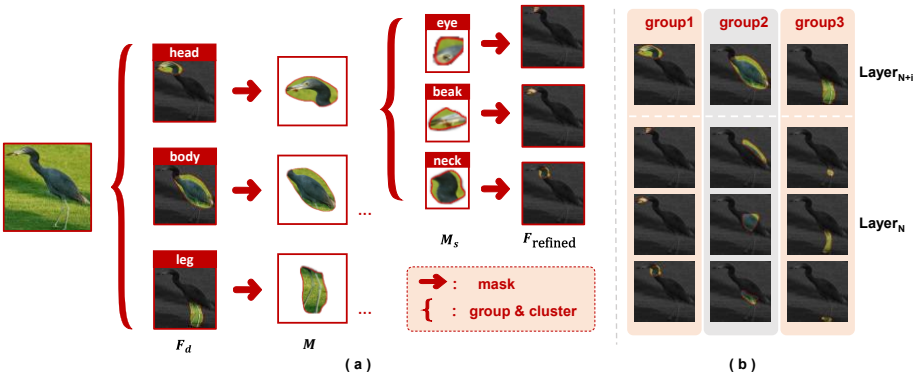


Fig. 3. (a) The hierarchical operation steps. (b) Grouped feature maps, each column representing a semantic group. The results highlight the model's ability to learn consistent and interpretable patterns.

As shown in Figure 3, the hierarchical implementation is achieved by constraining shallower feature maps with the grouped deeper feature maps. Specifically, grouping information from the deep feature maps of the previous training round is used to mask the shallow feature maps, guiding them to focus on regions corresponding to each deep-layer group. The shallow feature maps are then clustered within these regions to learn more refined semantic groups.

Initial Group Clustering in Deeper Layers The initial group clustering in deeper layers is accomplished using *formula (1)*, which ensures that filters within each group activate similar visual patterns, thereby optimizing the clustering outcome. This results in the deep feature maps $\boldsymbol{F_d}$.

In the following, $\boldsymbol{F_d}$ denotes the deep feature maps obtained from the previous training round of the deep neural network.

Masking Shallow Layer Feature Maps After performing the initial clustering in the deep layers, we apply a mask to the shallow layer feature maps. Let $\boldsymbol{M}$ represent the masks generated from the deep feature maps $\boldsymbol{F_d}$, obtained by selecting regions with activation values exceeding a threshold. The shallow feature maps $\boldsymbol{F_s}$ are masked by the masks to ensure that only the relevant parts of the shallow feature maps are used in the subsequent operations. This masking operation is defined as:

$$\boldsymbol{M_s} = \boldsymbol{M} \times \boldsymbol{F_s} \tag{8}$$

This ensures that each shallow feature map only contributes to the deep-layer clusters that are semantically aligned with it.

Refining Group Clustering in Deeper Layers After applying the mask to the shallow feature maps, the next step is to refine the group clustering in the deeper layers using the masked shallow feature maps. By doing so, the deep layers are encouraged to activate regions that are aligned with the semantic features defined by the shallow layers. The updated deep-layer feature map, after applying the mask and re-clustering, is calculated by:

$$\boldsymbol{F}_{\mathbf{refined}} = \boldsymbol{M}_{\mathbf{channel}}(\boldsymbol{C}_{\mathbf{deep}}, \boldsymbol{F_s}) \tag{9}$$

Where $\boldsymbol{C_{deep}}$ defines the rough cluster mapping for the deeper layers, and $\boldsymbol{M_{channel}}$ is the masking operation applied to the deep-layer feature map using the shallow features as guidance.

Weighted Loss Computation After the hierarchical clustering and masking, the loss function computes a weighted loss for each cluster. The final loss is the sum of the individual cluster losses, each of which accounts for both intra-group consistency and the semantic relevance of the features:

$$L(\theta, A) = \sum_{k=1}^{K} L_{\text{cluster},k} \tag{10}$$

Each cluster loss $L_{cluster,k}$ is defined as:

$$L_{\text{cluster},k} = \frac{S_{\text{within},k}}{S_{\text{all},k}} \times F_{\text{normed}} \tag{11}$$

Where F_{normed} is the normalized version of the masked feature map across the clusters.

Final Hierarchical Loss The total hierarchical loss is the weighted sum of all individual cluster losses, where each cluster's loss is weighted based on its importance. The final loss function is given by:

$$L_{\text{final}} = \sum_{i=1}^{K} \left(L_{\text{cluster},i} \times \frac{F_{\text{normed},i}}{\sum F_{\text{normed}}} \right) \tag{12}$$

This ensures that clusters with higher activations and greater semantic relevance contribute more significantly to the final loss, leading to the learning of more interpretable feature maps.

3.3 Algorithm

Based on the grouping and hierarchical construction methods described above, our algorithm proceeds in three main stages: (1) Clustering feature maps in selected layers to

learn interpretable group representations. (2) Constructing hierarchical correspondences between shallow and deep layers by applying masks and refining the groupings. (3) Computing Loss Functions and Training via Backpropagation. The pseudocode is provided in **Algorithm 1**.

Algorithm1 *Hierarchical Group Clustering for Interpretable CNNs*

Initialize network parameters θ

Repeat until convergence

Group clustering on deeper layer l_d:

Extract feature map F_d

Cluster filters into K groups $\{A_1, A_2, \dots, A_K\}$

If multi-class:

  Compute: $L_{\text{multi}} = -\sum_{c=1}^{C} \sum_{p,q \in I_c} \frac{s_{pq}}{\sum_{q \in I} s_{pq}}$

Compute: $L_{\text{group}} = -\sum_{k=1}^{K} \frac{S_{\text{within},k}}{S_{\text{all},k}}$

Hierarchical constraint on shallow layer l_s:

Extract feature map F_s

Compute average activation maps from F_d

Generate mask M , apply to shallow map: $M_s = M \times F_s$

For each cluster k , compute:

$$(L_{\text{cluster},k} = \frac{S_{\text{within},k}}{S_{\text{all},k}} \times F_{\text{normed},k})$$

Compute hierarchical loss: $L_{\text{hier}} = \sum_{k=1}^{K} \left(L_{\text{cluster},k} \cdot \frac{F_{\text{normed},k}}{\sum F_{\text{normed}}} \right)$

Joint optimization:

If multi-class:

  Compute total loss: $L_{\text{total-multi}} = \lambda L_{\text{group}} + \beta L_{\text{multi}} + \gamma L_{\text{hier}} + L_{\text{cls}}$

else:

  Compute total loss: $L_{\text{total}} = \lambda L_{\text{group}} + \gamma L_{\text{hier}} + L_{\text{cls}}$

Update θ via backpropagation

We successfully applied this algorithm in our experiments to CNNs with hierarchical structures (such as *VGG16*, *VGG19*, *ResNet*, and *DenseNet*). By identifying the corresponding deep (l_d) and shallow (l_s) layers, this method can constrain the network to learn hierarchical interpretable representations.

4 Experiments

In this section, we conduct a series of quantitative experiments to evaluate the interpretability and classification performance of the proposed approach. Specifically, we present **three** types of evaluations: (1) direct comparison with conventional CNNs and ICCNNs to assess the impact of hierarchical constraints on filter interpretability; (2) benchmarking against Grad-CAM-based methods to demonstrate the superiority of our model in producing semantically consistent and structured visual explanations; and (3) analysis on multi-category classification tasks to investigate the scalability and generalization capabilities of the approach in more complex scenarios.

To this end, the proposed hierarchical interpretability approach was integrated into four widely used CNN architectures—VGG-13, VGG-16, ResNet-18, and DenseNet-121—and evaluated on both single-category binary and multi-category classification tasks. Throughout all experiments, we performed ablation comparisons by varying the loss terms in $L_{\text{total}} = \lambda L_{\text{group}} + \gamma L_{\text{hier}} + L_{\text{cls}}$. In particular, setting $\gamma = 0$ reduces the model to the original ICCNN, and setting $\gamma = \lambda = 0$ reduces it to a standard CNN.

When training the HICCNN in the single-category setting, the compositional loss weight λ was set to 1.0 for most models. A smaller value (λ = 0.1) was used for VGG-16 due to its lack of residual connections, which makes it more challenging to optimize and more sensitive to large loss weights. In the multi-category setting, both λ and the multi-task loss weight β were uniformly set to 0.1 across all architectures. The hierarchical constraint was assigned a fixed loss weight γ = 0.01 to ensure its magnitude remained comparable to other loss terms and contributed meaningfully without dominating the optimization process.

All models were trained using the Adam optimizer with an initial learning rate of 1×10^{-4}, which decayed by a factor of 0.8 every 100 epochs. Training was performed for a total of 2000 epochs with a mini-batch size of 64, using a single NVIDIA RTX A6000 GPU. To facilitate fair comparisons, we also trained ICCNN variants for each architecture by removing the hierarchical loss component, while keeping all other settings unchanged. This consistent experimental setup enabled rigorous comparisons among baseline CNNs, ICCNNs, and the proposed HICCNNs, allowing us to systematically assess the effectiveness of the hierarchical interpretability framework.

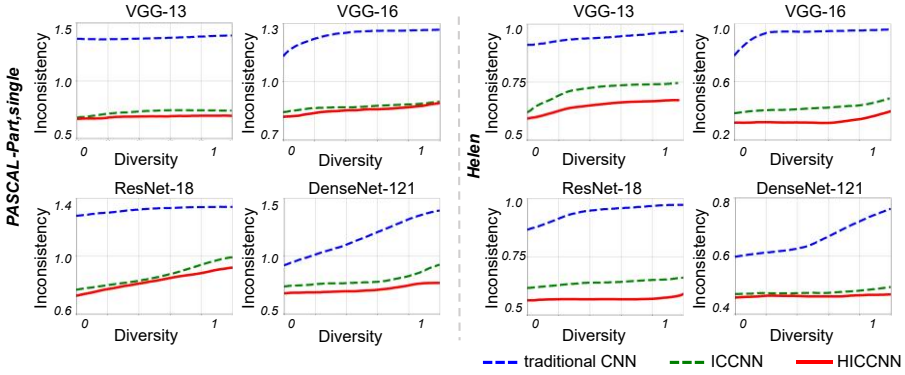


Fig. 4. The inconsistency-diversity curves across four mainstream network architectures (VGG-13, VGG-16, ResNet-18, and DenseNet-121).

4.1 Quantitative Evaluation of Diversity and Inconsistency

To quantitatively assess the interpretability of filters in convolutional networks, we conducted a comprehensive comparative study across four representative CNN architectures: VGG-13, VGG-16, ResNet-18, and DenseNet-121. We evaluated three types of models: (1) standard convolutional neural networks (CNNs), (2) interpretable CNNs

(ICCNNs), and (3) our proposed Hierarchical Interpretable Compositional CNNs (HICCNNs). All models were trained on the PASCAL-Part dataset under the setting of binary classification for a single object category.

In the HICCNN models, we incorporate the proposed compositional loss into the top convolutional layers of each backbone. Specifically, the loss was added to the conv4-2 and conv5-2 layer in VGG-13 and ResNet-18, conv4-3 and conv5-3 layer in VGG-16, the third dense block and final dense block in DenseNet-121. For fair comparison, all HICCNN models were trained under identical initialization schemes and data conditions as their corresponding traditional CNNs, and pre-trained weights were loaded for all non-target layers.

We employed two complementary quantitative metrics to evaluate filter interpretability: (1) **Inconsistency** of visual patterns, which measures the semantic stability of a filter's activation across different images and is computed as the entropy of the filter's activation distribution over ground-truth semantic concepts; and (2) **Diversity** of visual patterns, defined as the proportion of pixels that are covered by activations from any filter in the model, thereby approximating the model's semantic coverage. Lower inconsistency values indicate that filters activate consistently on specific semantic parts or regions, while higher diversity reflects broader semantic representation capability. To obtain a set of inconsistency-diversity values, we varied the activation threshold τ and plotted the resulting curves for each model.

As illustrated in Figure 4, across all four network backbones, the HICCNN consistently outperformed both traditional CNNs and ICCNNs by achieving lower inconsistency under comparable diversity levels. Traditional CNNs exhibited relatively high inconsistency, indicating that their internal features lacked semantic stability. Although ICCNNs improved consistency to some extent, they demonstrated significantly lower diversity due to their restricted capacity to represent only a single layer's part-level semantics. In contrast, our proposed HICCNNs achieved a favorable trade-off between consistency and diversity, preserving broad semantic coverage while maintaining low inconsistency. These results substantiate the effectiveness and general applicability of HICCNNs in learning both stable and diverse interpretable visual representations across various CNN architectures.

4.2 t-SNE Visualization of Feature Space

To further evaluate the structural organization and semantic consistency of the learned representations in the feature space, we employ t-SNE (t-Distributed Stochastic Neighbor Embedding) to visualize the feature maps from intermediate convolutional layers of different models. Specifically, we select three models—traditional CNN, ICCNN, and our proposed HICCNN—all based on the VGG-16 architecture and trained on the bird category of the PASCAL-Part dataset. We extract the filter responses from the test images at designated convolutional layers and embed them into a two-dimensional space.

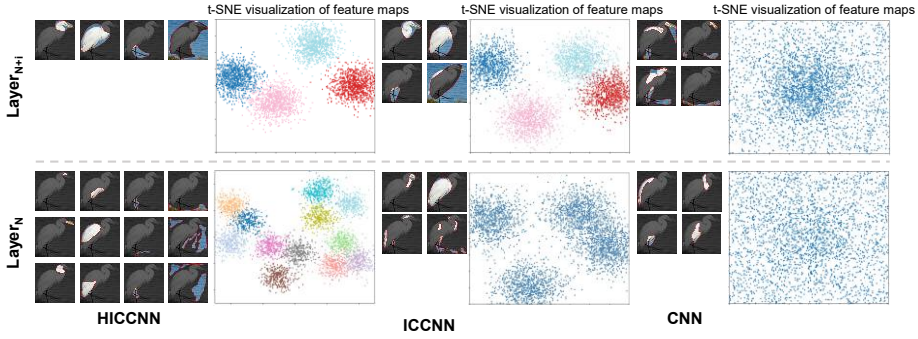


Fig. 5. t-SNE embeddings of feature maps from HICCNN, ICCNN, and traditional CNN models.

As shown in Figure 5, each point represents the activation response of a filter across different test images, treated as a high-dimensional vector. These vectors are embedded using t-SNE, and points are colored according to their assigned filter group. The results show that HICCNN yields a clear and structured clustering of filter features: filters within the same group form compact clusters, while distinct groups are well-separated. Notably, such structured groupings are observed not only in high-level convolutional layers but also consistently across lower layers, indicating that HICCNN is capable of learning semantically consistent filter groupings throughout the network hierarchy.

In contrast, ICCNN exhibits distinguishable feature groups only in the specific high-level layer where the interpretability loss is applied. Although certain shallow layers show weak grouping tendencies, they lack clear and consistent group representations. Traditional CNNs, on the other hand, present highly entangled and disordered distributions, with filter activations overlapping significantly and lacking any discernible semantic structure.

These results further validate the effectiveness of our proposed grouping constraint. Unlike models that impose interpretability only at a single layer, HICCNN enforces hierarchical grouping across multiple layers, leading to more consistent, semantically meaningful, and structurally organized feature representations in the embedding space.

4.3 Evaluation of Activation Accuracy Based on Semantic Masks

To further assess the spatial alignment between model activations and semantic regions, we introduce the **Standard-Region Activation Score (SRAS)** as a metric to quantify whether intermediate-layer activations are concentrated within human-annotated semantic regions. This metric reflects the model's ability to attend to semantically meaningful areas in a spatially interpretable manner.

Given a set of intermediate feature maps $F \in R^{C\times H\times W}$, where $F_c \in R^{H\times W}$ denotes the activation map of the $c-th$ channel, and a binary semantic mask $M^{\text{standard}} \in \{0,1\}^{H\times W}$ indicating the target semantic region, the activation score of channel c within the standard region is defined as:

$$\langle F_c, M^{\text{standard}} \rangle = \sum_{x=1}^{H}\sum_{y=1}^{W} F_c(x,y)\cdot M^{\text{standard}}(x,y) \tag{13}$$

This measures the total activation of F_c that falls within the semantic region. The total activation of the same channel is computed as:

$$||F_c||_1 = \sum_{x=1}^{H}\sum_{y=1}^{W} F_c(x,y) \tag{14}$$

The SRAS is then defined as the average normalized activation over all channels:

$$S_{\text{activation}}\left(F, M^{\text{standard}}\right) = \frac{1}{C}\sum_{c=1}^{C}\frac{\left\langle F_c, M^{\text{standard}}\right\rangle}{|F_c|_1} \tag{15}$$

This score lies in the range [0,1], where a higher value indicates that the model's activation is more concentrated within the desired semantic region, while a lower value suggests that the activations are scattered or misaligned.

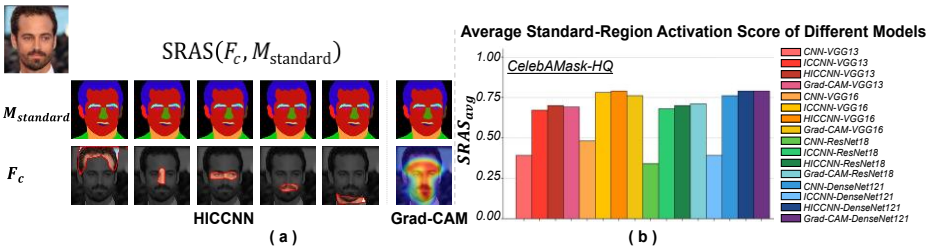


Fig. 6. Visualization of activation maps and corresponding semantic regions on face datasets.

We conduct SRAS-based evaluations on two high-quality facial segmentation datasets: **CelebAMask-HQ**. The datasets provide precise pixel-level semantic masks for facial components such as eyes, nose, mouth, cheeks, and background, which serve as M^{standard} in our evaluation. For each test image, we extract intermediate-layer feature maps from three different models—traditional CNN, ICCNN, and our proposed HICCNN—and compute their SRAS scores using the above formulation.

In addition, we also compute SRAS scores based on Grad-CAM heatmaps generated from the output of the final classification layer of each model. This allows us to directly compare our method with an established post-hoc explanation technique in terms of semantic alignment.

As illustrated in Figure 6, the bar charts on the right report SRAS scores across different facial regions for all methods. The results demonstrate that HICCNN consistently achieves higher SRAS scores across most key semantic regions, indicating that its activations are highly aligned with human-annotated semantic structures. While ICCNN achieves moderately high SRAS in certain regions, its interpretability is mostly confined to the high-level layer where constraints are explicitly applied. Traditional CNNs, by contrast, show overall lower SRAS scores, suggesting unfocused and poorly aligned activation patterns.

Notably, the Grad-CAM heatmaps exhibit SRAS scores that are close to those of HICCNN, especially in salient regions such as the nose and eyes. This suggests that our model's internal activations inherently capture similar semantic alignment to what Grad-CAM highlights post-hoc, but without relying on backpropagation or external visualization modules.

This experiment validates the effectiveness of HICCNN in achieving precise spatial localization of semantic regions. The activations it learns are not only semantically consistent but also inherently interpretable at the feature level, offering comparable localization quality to Grad-CAM while providing built-in interpretability during forward inference.

4.4 Performance on Multi-Class Classification Tasks

To assess the effectiveness of HICCNN in multi-class classification settings, we evaluate its performance on two widely used fine-grained datasets, CUB-200-2011 and NABirds, along with a proposed dataset specifically designed to reflect more challenging classification scenarios. We compare three models: the original CNN (Ori CNN), the interpretable compositional CNN (ICCNN), and our hierarchical CNN (HICCNN).

Table 1. Comparisons of Multi-Class classification accuracy between ICCNNs and HICCNNs revised from different classic CNNs.

Dataset	Model	Ori CNN	ICCNN	HICCNN
CUB200	VGG16	77.21	77.32	**78.01**
	ResNet18	78.87	78.05	**79.22**
	DenseNet121	80.03	80.19	**81.04**
Na-Birds	VGG16	88.42	88.17	**88.83**
	ResNet18	88.64	**88.80**	**88.80**
	DenseNet121	89.56	89.49	**90.11**
Proposed Dataset	VGG16	10.22	12.19	**49.32**
	ResNet18	10.98	11.22	**48.01**
	DenseNet121	11.01	10.92	**48.65**

As shown in Table 1, HICCNN consistently outperforms both Ori CNN and ICCNN across all backbone architectures on the CUB200 and NABirds datasets. These results indicate that the introduction of hierarchical grouping and compositional constraints not only enhances interpretability, but also improves the model's ability to capture discriminative features, leading to better classification performance in standard fine-grained tasks.

To further validate the advantages of HICCNN under more difficult conditions, we construct a proposed dataset by collecting images that were misclassified by either Ori CNN or ICCNN in the CUB200 and NABirds datasets. This curated dataset emphasizes challenging cases such as subtle inter-class differences, occlusion, or ambiguous part configurations.

Experimental results show that HICCNN significantly outperforms the other models on the proposed dataset, demonstrating its superior robustness and generalization ability in more complex multi-class classification tasks. The performance gap is especially evident in scenarios where traditional CNNs and ICCNNs struggle, highlighting the practical benefits of our hierarchical compositional design.

5 Limitations

The proposed method is inherently tailored for convolutional neural networks and leverages the spatial hierarchies in CNN feature maps. Consequently, it is not directly applicable to recent non-convolutional architectures such as Vision Transformers (ViTs), which do not exhibit the same hierarchical spatial structure. This limitation arises from the methodological focus on CNN-based interpretability.

6 Conclusion

In this work, we propose HICCNN, a hierarchical interpretable compositional framework that enables layer-wise semantic grouping without requiring any modifications to the original CNN architecture. By introducing a hierarchical loss that aligns shallow features with deep-layer semantics, our approach learns structured and semantically consistent representations across multiple layers. Extensive experiments demonstrate that HICCNN significantly enhances interpretability and delivers notable performance improvements on fine-grained and challenging multi-class classification tasks. Furthermore, it achieves semantic localization quality comparable to Grad-CAM while providing built-in interpretability. These findings highlight the potential of HICCNN as a practical and scalable solution for interpretable deep learning in CNN-based models.

References

1. Shen, Wen, et al. "Interpretable compositional convolutional neural networks." arXiv preprint arXiv:2107.04474 (2021).
2. Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In NeurIPS, 2017.
3. 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.
4. Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In ICLR, 2017.
5. Yuchao Li, Rongrong Ji, Shaohui Lin, Baochang Zhang, Chenqian Yan, Yongjian Wu, Feiyue Huang, and Ling Shao. Interpretable neural network decoupling. arXiv:1906.01166, 2020.
6. Haoyu Liang, Zhihao Ouyang, Yuyuan Zeng, Hang Su, Zihao He, Shu-Tao Xia, Jun Zhu, and Bo Zhang. Training interpretable convolutional neural networks by differentiating class-specific filters. In ECCV, 2020.
7. Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In CVPR, 2018.
8. Wang, Jianyu, et al. "Detecting semantic parts on partially occluded objects." arXiv preprint arXiv:1707.07819 (2017).
9. Rangadurai, Kaushik, et al. "Hierarchical Structured Neural Network for Retrieval." arXiv preprint arXiv:2408.06653 (2024).

10. Mavrovouniotis, Micheal L., and S. Chang. "Hierarchical neural networks." Computers & chemical engineering 16.4 (1992): 347-369.
11. Wah, Catherine, et al. "The caltech-ucsd birds-200-2011 dataset." (2011).
12. Van Horn, Grant, et al. "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
13. David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In CVPR, 2017.
14. Pu, Yifan, et al. "Fine-grained recognition with learnable semantic data augmentation." IEEE Transactions on Image Processing (2024).
15. Long (Leo) Zhu, Yuanhao Chen, Alan Yuille, and William Freeman. Latent hierarchical structural learning for object detection. In CVPR, 2010.
16. Austin Stone, Huayan Wang, Michael Stark, Yi Liu, D. Scott Phoenix, and Dileep George. Teaching compositionality to cnns. In CVPR, 2017.
17. Stone, Austin, et al. "Teaching compositionality to cnns." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
18. Sun, Ke, et al. "High-resolution representations for labeling pixels and regions." arXiv preprint arXiv:1904.04514 (2019).
19. Lee, Cheng-Han, et al. "Maskgan: Towards diverse and interactive facial image manipulation." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
20. Shen, Shiwen, et al. "An interpretable deep hierarchical semantic convolutional neural network for lung nodule malignancy classification." Expert systems with applications 128 (2019): 84-95.
21. Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." Proceedings of the IEEE international conference on computer vision. 2017.
22. Amorim, José Pereira, et al. "Evaluating Post-hoc Interpretability with Intrinsic Interpretability." arXiv preprint arXiv:2305.03002 (2023).
23. Gjelsvik, Elise Lunde, and Kristin Tøndel. "Increased interpretation of deep learning models using hierarchical cluster-based modelling." PloS one 18.12 (2023): e0295251.
24. Ibrahim, Rami, and M. Omair Shafiq. "Explainable convolutional neural networks: a taxonomy, review, and future directions." ACM Computing Surveys 55.10 (2023): 1-37.
25. Ji, Yang, et al. "A Comprehensive Survey on Self-Interpretable Neural Networks." arXiv preprint arXiv:2501.15638 (2025).