sensors

MDPI

Article

Enhancing Bottleneck Concept Learning in Image Classification

Xingfu Cheng [1], Zhaofeng Niu [1], Zhouqiang Jiang [2] and Liangzhi Li [1,*]

[1] Computer Science Department, Qufu Normal University, Rizhao 276826, China; xingfucheng@qfnu.edu.cn (X.C.); zhaofengniu@qfnu.edu.cn (Z.N.)
[2] Osaka University, Osaka 565-0871, Japan; zhouqiang@is.ids.osaka-u.ac.jp
* Correspondence: liliangzhi@ieee.org

Abstract: Deep neural networks (DNNs) have demonstrated exceptional performance in image classification. However, their "black-box" nature raises concerns about trust and transparency, particularly in high-stakes fields such as healthcare and autonomous systems. While explainable AI (XAI) methods attempt to address these concerns through feature- or concept-based explanations, existing approaches are often limited by the need for manually defined concepts, overly abstract granularity, or misalignment with human semantics. This paper introduces the Enhanced Bottleneck Concept Learner (E-BotCL), a self-supervised framework that autonomously discovers task-relevant, interpretable semantic concepts via a dual-path contrastive learning strategy and multi-task regularization. By combining contrastive learning to build robust concept prototypes, attention mechanisms for spatial localization, and feature aggregation to activate concepts, E-BotCL enables end-to-end concept learning and classification without requiring human supervision. Experiments conducted on the CUB200 and ImageNet datasets demonstrated that E-BotCL significantly enhanced interpretability while maintaining classification accuracy. Specifically, two interpretability metrics, the Concept Discovery Rate (CDR) and Concept Consistency (CC), improved by 0.6104 and 0.4486, respectively. This work advances the balance between model performance and transparency, offering a scalable solution for interpretable decision-making in complex vision tasks.

Keywords: visual concept; explainable artificial intelligence; image classification

Academic Editor: Loris Nanni

Received: 3 March 2025
Revised: 6 April 2025
Accepted: 7 April 2025
Published: 10 April 2025

Citation: Cheng, X.; Niu, Z.; Jiang, Z.; Li, L. Enhancing Bottleneck Concept Learning in Image Classification. *Sensors* **2025**, *25*, 2398. https://doi.org/10.3390/s25082398

1. Introduction

Interpreting the behavior of deep neural networks (DNNs) has emerged as a critical challenge in the deployment of these models, particularly in high-stakes domains such as healthcare [1] and autonomous vehicles [2]. Despite their success in achieving state-of-the-art performance, DNNs remain predominantly "black-box" models: their decision-making processes are opaque and difficult to comprehend [3]. This lack of interpretability hinders trust and impedes verification, making it challenging to ensure model reliability in safety-critical applications [4]. In sensor-based systems, such as those employed in autonomous vehicles, medical imaging, and environmental monitoring, the demand for explainable AI is particularly critical. These systems rely extensively on sensor data to make real-time decisions that directly affect human safety. Explainable AI (XAI) [5,6] offers a promising solution by providing transparency through per-pixel relevance information, thereby elucidating the basis for model decisions.

A substantial body of XAI research has concentrated on providing feature-level explanations, particularly at the pixel or patch level for vision-related tasks [7]. These methods assign relevance scores to input features—such as individual pixels in an image—indicating

their significance in the model's decision-making process. Widely recognized approaches, such as saliency maps [8], Grad-CAM [9], and integrated gradients [10], are commonly employed to generate these feature-based explanations. While these methods offer valuable insights, they are often criticized for being challenging to interpret without domain expertise. The relevance information is typically presented at a very granular level, which can be abstract and unintuitive for non-expert users.

To address this gap, recent advancements in explainable AI (XAI) have introduced concept-based methods that aim to represent model behavior using high-level, human-understandable concepts [11]. These approaches seek to align a model's reasoning with human cognitive processes by linking model outputs to interpretable concepts, such as objects, attributes, or scenes. By focusing on the relationship between these concepts and the model's decisions, these methods facilitate more transparent explanations that are easier for humans to comprehend [12]. However, most existing concept-based methods rely on the explicit definition of concepts or supervision, which limits their generalizability and scalability. The need for large quantities of labeled data to predefine concepts or reliance on human expertise in defining meaningful concepts presents significant challenges in real-world applications.

We propose a novel concept-based explainability method, the Enhanced Bottleneck Concept Learner (E-BotCL), which leverages self-supervised learning to address the limitations of traditional concept-based methods. E-BotCL represents images by learning the presence or absence of concepts directly from the target task, without depending on manually defined concepts or external supervision. E-BotCL encourages the model to discover task-relevant, human-interpretable semantic concepts. This self-supervised learning framework enhances E-BotCL's scalability and explainability, making it suitable for a broad range of applications without the need for manual annotation or domain expertise. Through our proposed framework, we aim to make strides toward more interpretable, reliable, and transparent AI models that can be deployed effectively in real-world scenarios, with a focus on enhancing both the model's accuracy and the explainability of its decisions.

2. Related Works

2.1. Explainable AI

Explainable artificial intelligence (XAI) seeks to improve the transparency of decision-making processes in machine learning models [5,8,12–16], such as deep neural networks, in response to the growing demand for interpretability and trustworthiness, particularly in high-risk domains. XAI methods are typically divided into two broad categories [17]: post hoc explanations and intrinsic interpretability.

Post hoc explanation methods aim to provide insights into a model's decision-making after the model has been trained, without modifying the model itself. This category includes techniques such as saliency maps [8], which highlight regions in the input data (e.g., image pixels) that have the greatest impact on the model's predictions, and feature attribution methods such as LIME [3] and SHAP [18], which approximate decision boundaries by training interpretable surrogate models on localized data regions. Although post hoc explanations are valuable, they are often limited by their lack of semantic clarity and the risk of misinterpretation, as the relevance maps generated may not always align with human-understandable concepts [19].

In contrast, intrinsic interpretability [20] aims to design models that are interpretable by their very structure. These models are typically simpler, and their decision-making processes are more directly comprehensible. Examples of intrinsically interpretable models include decision trees, linear models, and rule-based systems [21]. However, these models often face a trade-off between interpretability and performance, as they may fail to capture

the complexity of data as effectively as more sophisticated, opaque models such deep neural networks. Recent research has focused on improving the balance between performance and interpretability, with some efforts dedicated to developing models that retain high accuracy while maintaining transparent internal processes [22,23].

Despite the progress made in XAI, several challenges remain. A key issue is ensuring the trustworthiness of explanations. Explanations must be not only interpretable but also accurate and consistent with the model's underlying decision-making process. Moreover, there is an ongoing need for improved tools and metrics to evaluate the quality of explanations, as well as to assess the usability, effectiveness, and potential biases in different XAI methods.

2.2. Concept-Based Framework for Interpretability

The concept-based framework for interpretability has emerged as a promising approach to enhancing the transparency of deep learning models, particularly by providing human-understandable explanations for complex decision-making processes [12,24]. In contrast to pixel-based or feature-based methods, which explain decisions at a granular level, the concept-based framework aims to offer high-level, semantically meaningful explanations by associating model predictions with a set of interpretable concepts. This approach is inspired by human cognition, where decisions are often based on abstract concepts that are more comprehensible than raw features.

At its core, the concept-based framework operates by defining a set of concepts that capture significant patterns or structures in the input data. These concepts can either be predefined or learned directly from the data. Predefined concepts often stem from domain knowledge, such as medical terminology in healthcare or object categories in image recognition tasks. In contrast, data-driven methods seek to discover these concepts automatically, typically through unsupervised or semi-supervised learning techniques [25–27]. Once the concepts are defined, the model's decision-making process is articulated in terms of the presence or absence of these concepts, providing a more intuitive explanation for human users.

A prominent structure within this framework is the Concept Bottleneck Model (CBM) [24], which directly links model predictions to concept activations. The CBM introduces a bottleneck layer that forces the model to rely on a limited set of concepts to make decisions. As a result, the decision-making process is tightly coupled with the presence or absence of these predefined concepts, offering a transparent mechanism for interpretation. In this structure, the classifier is trained not on raw input data but on the binary activations of concepts, ensuring that the model's decisions can be traced back to human-understandable features. This concept has been extended and refined in several studies to handle more complex datasets, such as images and text, with notable success.

Despite the advantages of concept-based frameworks in providing semantically rich explanations, designing an appropriate set of concepts remains a critical challenge. Handcrafting concepts through domain expertise, while ensuring interpretability, can be labor-intensive and may not scale well to large, complex tasks. To address this, recent research has focused on automated concept discovery methods [26,27], where concepts are learned directly from data. Techniques such as clustering, factorization, and self-supervised learning have been employed to uncover meaningful concepts that align with human intuition. For example, in image classification, methods such as attention-based mechanisms and unsupervised feature learning have been used to discover high-level object concepts without requiring manual annotations [12]. However, while these automatically discovered concepts can improve scalability and flexibility, they may not always align with human-

understandable semantics, potentially leading to interpretations that are less intuitive or harder to verify [28,29].

Moreover, the semantic gap between learned concepts and human understanding remains a persistent challenge. Although concept-based frameworks aim to provide explanations that are easier for humans to interpret, concepts learned through data-driven methods may still lack the richness and clarity of those defined by human experts. To address this, some studies have proposed hybrid approaches that combine predefined and learned concepts, thereby balancing interpretability with model flexibility [12].

2.3. Contrastive Learning

Contrastive learning has become a cornerstone technique in self-supervised representation learning, particularly in the field of computer vision. The core idea behind contrastive learning is to learn discriminative features by minimizing the distance between similar samples while maximizing the distance between dissimilar ones. Early approaches to contrastive learning, such as SimCLR [30] and MoCo [31], have significantly advanced the state of the art by utilizing a large number of negative samples or maintaining a memory bank to sustain informative contrast. However, this reliance on numerous negative samples or memory storage introduces substantial computational overhead and complexity, posing challenges for scalability and efficiency, especially in large-scale settings.

To mitigate these drawbacks, SimSiam [32] introduces a more streamlined approach that eliminates the need for negative pairs altogether. Instead of relying on negative samples, SimSiam employs a stop-gradient mechanism to prevent representation collapse during training, while still enabling the model to learn meaningful features. This approach involves two key components: a projector network that transforms the learned representations into a latent space and a predictor network that attempts to predict the representation of one view from another. Notably, SimSiam's reliance on positive pairs, which are different augmentations of the same image, demonstrates that negative pairs are not a necessary condition for obtaining effective representations. This finding challenges the traditional paradigm in contrastive learning, offering a more computationally efficient framework for self-supervised learning.

SimSiam's success largely depends on the stop-gradient mechanism, which prevents trivial solutions, such as the collapse of all embeddings into a single point—a common issue in contrastive learning when negative samples are absent. Empirical evaluations on the ImageNet dataset reveal that SimSiam not only achieves competitive performance when compared to methods such as MoCo v2 [30] and BYOL [33] but also requires fewer hyperparameters and less computational resources. As a result, SimSiam presents an attractive alternative for practitioners seeking efficient self-supervised learning methods. Moreover, its simplicity makes it highly scalable and easier to implement than more complex methods that depend on negative sampling or large memory banks.

In the domain of interpretable AI, SimSiam's ability to learn structured, high-quality representations is particularly promising for concept-based learning frameworks. Concept-based learning seeks to align model representations with human-interpretable concepts, such as object parts, textures, or semantic categories. Unlike traditional supervised approaches, where concepts must be manually defined or pre-annotated, self-supervised contrastive methods such as SimSiam have the potential to autonomously discover meaningful features from the data. This self-discovery of features not only enhances model interpretability but also offers the potential to bridge the gap between high performance and explainability.Furthermore, recent works [34] have underscored the importance of model interpretability in real-world applications, where transparency is critical to ensuring trust and accountability. Thus, SimSiam's capacity for learning structured representations

could provide valuable insights into how deep learning models can achieve both strong performance and greater transparency.

Despite these compelling advantages, there remain open questions and areas for further improvement. For example, although SimSiam's reliance on positive pairs reduces the need for negative samples, the method still necessitates careful design of augmentation strategies to ensure the diversity of positive pairs. Furthermore, although SimSiam's simplicity enhances its computational efficiency, its scalability in highly complex tasks beyond image classification, such as object detection and video processing [35,36], remains to be fully evaluated.

3. Model

3.1. Preliminary

Figure 1 provides an overview of the BotCL framework [12] for training concept-based models. The process begins with the input image x, from which feature maps F are extracted using a backbone convolutional neural network (CNN). These feature maps are subsequently passed to the concept extractor, which performs two critical tasks: it generates the concept bottleneck activation vector t, representing the activation probabilities of various visual concepts, and it extracts concept features G corresponding to specific regions of interest in the image. The vector t is then forwarded to a classifier, which produces the final score s for the image classification. Throughout the training, the concept prototypes are constrained using self-supervised and regularization techniques, with both t and G guiding the learning process.

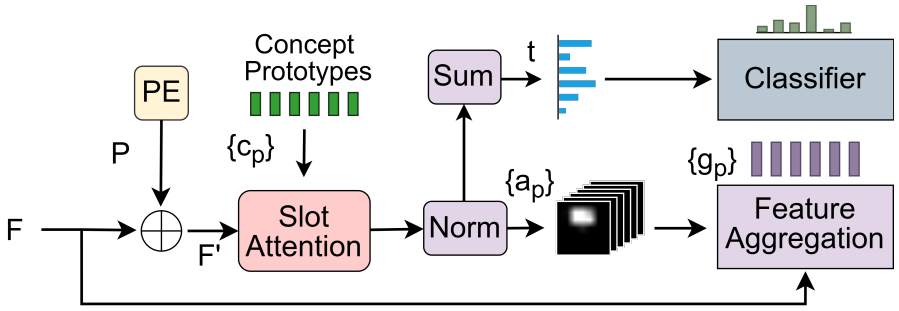

Figure 1. Overview of the concept extractor framework.

The concept extractor leverages a slot attention mechanism [37] to identify and extract relevant visual concepts from images. Initially, positional encodings P are incorporated into the feature map F to preserve spatial information, yielding a modified feature map $F' = F + P$. This modified map is then flattened into a 2D tensor of dimensions $l \times d$, where $l = hw$ represents the number of spatial locations, while d is the dimensionality of the feature vectors. The slot attention mechanism computes the attention weight a_p for each concept p across the spatial dimensions, which indicates the spatial distribution of each concept. The features in F corresponding to concept p are aggregated to form the concept feature g_p, which is calculated as the attention-weighted average of image features along the spatial dimension.

For classification, a simple fully connected (FC) layer, without any bias terms, is employed. The concept activation vector $t = (t_1, t_2, \ldots, t_k)^\top$ serves as the input to the classifier, which models the concept bottleneck. Let M represent the learnable weight matrix. The predicted class label $\hat{y}$ is computed as

$$\hat{y} = Mt. \tag{1}$$

Here, M is the vector corresponding to class k, and M_{kp} denotes the pth element of this vector. A positive value of M_{kp} suggests that concept p frequently co-occurs with class k in the dataset, supporting the classification of the image as belonging to class k. Conversely, a negative value of M_{kp} implies that concept p rarely co-occurs with class k, offering less support for the classification.

Given the absence of concept labels, a self-supervised learning approach is employed for concept discovery. To address various target tasks, two distinct loss functions are employed: one for learning visual representations and another for capturing relationships between concepts.

Reconstruction Loss: The SENN [25] framework adopts an autoencoder-like structure to learn more accurate representations. This structure assumes that the visual elements in an image are tightly connected to their spatial locations, enabling discrete concepts to reconstruct the original image effectively. A reconstruction loss is designed based on this assumption, where the decoder D receives the concept activation t and reconstructs the image. The reconstruction loss is formulated as

$$l_{\text{rec}} = \frac{1}{|B|} \sum_{x \in B} \|D(t) - x\|^2, \tag{2}$$

where $|B|$ is the mini-batch of images.

Contrastive Loss: Since the composition of natural images is inherently arbitrary, the information in the concept activations t alone may not suffice for accurate reconstruction. To address this, a contrastive loss function is introduced using image-level labels from the target classification task. Let $\hat{t} = 2t - 1_k$ be a vector of ones. If a pair of concept activations $(\hat{t}, \hat{t}')$ corresponds to the same class (i.e., $y = y'$, where y and y' are the labels corresponding to $\hat{t}$ and $\hat{t}'$, respectively), they are expected to be similar, as the images should share a similar set of concepts. Conversely, if they belong to different classes, the activations should be dissimilar. The contrastive loss is then formulated as

$$l_{\text{ret}} = -\frac{1}{|B|} \sum \alpha(y, y') \log J(\hat{t}, \hat{t}', y, y'), \tag{3}$$

where α is a weight term that adjusts the contribution of each class to the overall loss, addressing class imbalance. The function J is defined as

$$J(\hat{t}, \hat{t}', y, y') = \begin{cases} \sigma(\hat{t}^\top \hat{t}') & \text{for } y = y' \\ 1 - \sigma(\hat{t}^\top \hat{t}') & \text{otherwise} \end{cases} \tag{4}$$

A concept regularizer is introduced to constrain the concept prototypes $\{c_p\}$ and their corresponding features $\{g_p\}$. This regularizer ensures that each concept is stable across images, particularly when $t_p$is close to 1. The consistency loss is defined using cosine similarity as

$$l_{\text{con}} = -\frac{1}{p} \sum_{p} \sum_{g_p, g'_p} (\text{sim}(g_p, g'_p) \frac{1}{|H_p|(|H_p| - 1)}), \tag{5}$$

where the second summation iterates over all concept features within the set H_p, ensuring similarity between features for similar concept activations.

To ensure diversity among concepts, a diversity loss term is introduced. This encourages each concept to correspond to distinct visual elements. The diversity loss is formulated as

$$l_{\text{dis}} = \sum_{p, p'} (\text{sim}(\bar{g_p}, \bar{g_{p'}}) \frac{1}{p(p-1)}), \tag{6}$$

where the summation is taken over all pairs of concepts, ensuring that different concepts correspond to different visual features.

Finally, a quantization loss is introduced to enforce binarization of the concept activation vector t. This loss ensures that the activation values are close to 0 or 1, which is beneficial for interpretability:

$$l_{\text{qua}} = \frac{1}{p|\mathcal{B}|} \sum_{x \in \mathcal{B}} \left\| \text{abs}(\hat{t}) - 1_p \right\|^2 \tag{7}$$

where $\text{abs}(\cdot)$ represents the element-wise absolute value operation, while $\| \cdot \|$ denotes the Euclidean norm.

For the target classification task, the softmax cross-entropy loss, denoted as l_{cls}, is applied. The overall loss function, L_{base}, combines the classification loss with various regularization terms:

$$\mathcal{L}_{\text{base}} = l_{\text{cls}} + \lambda_R l_R + \lambda_{\text{con}} l_{\text{con}} + \lambda_{\text{dis}} l_{\text{dis}} + \lambda_{\text{qua}} l_{\text{qua}}, \tag{8}$$

where l_R is either l_{rec} or l_{ret}, depending on the target domain, and λ_{qua}, λ_{con}, λ_{dis}, and λ_R are the regularization coefficients that balance the contributions of each term.

3.2. E-BotCL

E-BotCL is an enhanced iteration of the original BotCL framework [12], designed to improve concept discovery and classification robustness by integrating a dual-path contrastive learning strategy, inspired by SimSiam [32]. Given a dataset $S = \{(x_i, y_i) \mid i = 1, 2, \ldots, N\}$, where x_i represents an image and y_i is the target class label associated with x_i from the set Ω. Figure 2 details the architecture of the Contrastive Concept Extractor, where PE denotes position embedding.

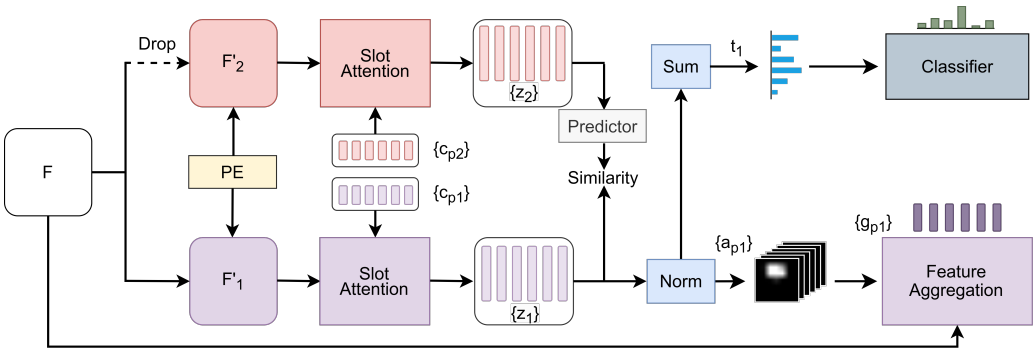

Figure 2. Overview of the contrastive concept extractor framework.

Given an input image x, the backbone convolutional neural network B extracts a feature map $F = B(x) \in \mathbb{R}^{d \times h \times w}$. This feature map F is then passed through the Contrastive Concept Extractor e_C, where C is a matrix whose pth column vector c_p represents a learnable concept prototype. The Contrastive Concept Extractor produces a concept bottleneck activation $t_p \in [0, 1]^l$, indicating the presence of each concept, as well as concept features $G \in \mathbb{R}^{d \times p}$ corresponding to the regions where each concept is present. The concept activation t_1 is subsequently used as input to the classifier to compute the classification score $s \in [0, 1]^{|\Omega|}$.

3.3. Contrastive Concept Extractor

The feature map F is first processed through a 1×1 convolutional layer to project it into a latent space, followed by batch normalization and ReLU activation. This operation yields the base feature representation:

$$\mathbf{F}_{\mathrm{i}} = \mathrm{ReLU}(\mathrm{Norm}(\mathrm{Conv}_{1\times1}(\mathbf{F}))). \tag{9}$$

Inspired by Siamese networks, the model employs two augmented views ($\mathbf{F}_1$, $\mathbf{F}_2$) of $\mathbf{F}_{\mathrm{i}}$ for self-supervised contrastive learning: Branch 1 retains the original features $\mathbf{F}_1$. Branch 2 applies stochastic dropout (`simsiam_drop`) to $\mathbf{F}_2$ as a form of feature augmentation.

Both feature representations are then combined with position embeddings and reshaped into sequential features:

$$\mathbf{F}_i^{'} = \mathrm{Reshape}(\mathbf{F}_i + \mathbf{P}), \quad i \in \{1, 2\}. \tag{10}$$

The slot attention mechanism [37,38] is employed to compute the spatial attention of concept p between c_{pi} and F_i'. Let $Q(c_{pi}) \in \mathbb{R}^d$ and $K(F_i') \in \mathbb{R}^{d\times l}$ represent the nonlinear transformations of c_{pi} and F_i', respectively. These transformations are implemented using multilayer perceptrons (MLPs) composed of three fully connected (FC) layers with ReLU activation between them. The attention $a_{p1} \in [0, 1]^l$ is computed using a regularization function φ as follows:

$$a_{p1} = \varphi(Q(c_{p1})^\top K(F_1')). \tag{11}$$

This attention mechanism identifies the spatial location of concept p in the image. If concept p is absent, the corresponding entries of a_{p1} remain close to zero. To quantify the presence of each concept, we compute the concept activation score t_p by aggregating the spatial dimension of a_{p1} as $t_p = \tanh\left(\sum_n a_{p1_n}\right)$, where a_{p1_n} is the nth element of a_{p1}.

3.4. Slot-Based Feature Aggregation

During training, we aggregate the features in F corresponding to concept p into the concept feature g_{p1}, as follows:

$$g_{p1} = F a_{p1}, \tag{12}$$

which provides the weighted average of the image features in the spatial dimension, with attention applied.

3.5. E-BotCL Loss

The following pseudocode outlines the process for calculating the contrastive learning loss in a PyTorch-like framework (Algorithm 1).

The slot-updated prototypes $\mathbf{z}_1$ and $\mathbf{z}_2$ are passed through a prediction network h, which projects them into a shared representation space. To align the cross-branch representations, a negative cosine similarity loss function is employed. This is expressed as

$$L_{\mathrm{cont}} = 1 - \frac{1}{2}[D(p_1, \mathbf{z}_2) + D(p_2, \mathbf{z}_1)], \tag{13}$$

where $D(p, z)$ represents the negative cosine similarity measure, $p_1 = h(\mathbf{z}_1)$ and $p_2 = h(\mathbf{z}_2)$ are the predictions for $\mathbf{z}_1$ and $\mathbf{z}_2$ obtained from the prediction network h, and $\mathbf{z}_1$ and $\mathbf{z}_2$ are the slot-updated prototypes. The operation D computes the cosine similarity between the predictions and the stop-gradient versions of the prototypes (i.e., $\mathbf{z}_1$ and $\mathbf{z}_2$ are detached during the loss calculation). The loss function aims to maximize the similarity between the projected representations of $\mathbf{z}_1$ and $\mathbf{z}_2$ across different branches by minimizing the negative cosine similarity.

The overall loss is defined as the sum of the contrastive loss and the base loss:

$$\mathcal{L} = \mathcal{L}_{\text{cont}} + \mathcal{L}_{\text{base}}. \tag{14}$$

Algorithm 1: Contrastive Loss Pseudocode, PyTorch-like

Input: Input batch x
Output: Contrastive Loss
Require: Backbone network f, projection mlp g, prediction mlp h, cosine similarity loss D

Function `Contrastive Loss(`x`):`
- $x_1 \leftarrow \text{backbone}(x)$;
- $x_2 \leftarrow \text{clone}(x_1)$;
- $x_2 \leftarrow \text{dropout}(x_2)$;
- $x_1^{pe} \leftarrow x_1 + \text{position\_encoding}(x_1)$;
- $x_2^{pe} \leftarrow x_2 + \text{position\_encoding}(x_2)$;
- `/* Slot attention for both views */`
- $updates_1, \text{attn}_1 \leftarrow \text{slot\_attention}(x_1^{pe}, x_1)$;
- $updates_2, \text{attn}_2 \leftarrow \text{slot\_attention}(x_2^{pe}, x_2)$;
- $z_1 \leftarrow \text{reshape}(updates_1, \text{flatten})$;
- $z_2 \leftarrow \text{reshape}(updates_2, \text{flatten})$;
- `/* Prediction heads (mlp) */`
- $p_1 \leftarrow h(z_1)$;
- $p_2 \leftarrow h(z_2)$;
- $z_1 \leftarrow z_1.\text{detach}()$;
- $z_2 \leftarrow z_2.\text{detach}()$;
- `/* contrastive loss computation */`
- $L_{\text{cont}} \leftarrow 1 - \left(\frac{D(p_1,z_2)+D(p_2,z_1)}{2}\right)$;
- **return** L_{cont};

Function `Cosine Similarity(`p, z`):`
- $z \leftarrow z.\text{detach}()$;
- $p \leftarrow \text{normalize}(p)$;
- $z \leftarrow \text{normalize}(z)$;
- **return** $-(p \cdot z).sum(dim = 1).mean()$;

4. Results

4.1. Experimental Settings

We evaluated E-BotCL on the CUB200 [39] and ImageNet [40] datasets. For both CUB200 (using the same data split as in [24]) and ImageNet, we employed a pre-trained ResNet-18 [41] as the backbone, reducing the channel size from 512 to 128 using a 1×1 convolutional layer. We selected $p = 20$ as the number of concepts for both datasets. The images were resized to 256×256 and then cropped to 224×224. During training, random horizontal flipping was applied as the sole data augmentation technique. The weights for each loss term were set to the default values: $\lambda_{qua} = 0.1$, $\lambda_{con} = 0.01$, $\lambda_{dis} = 0.05$, and $\lambda_R = 0.1$. The learning rate was set to 0.0001, the number of epochs was set to 60, and the batch size was set to 128.

4.2. Interpretability

Figure 3 visually contrasts the top five salient concepts identified by the E-BotCL and BotCL models in bird image recognition through heatmap analysis. This comparison

highlights the differences between the two models in terms of interpretability and conceptual accuracy. From the perspective of concept identification, E-BotCL demonstrates a qualitative improvement over BotCL. Specifically, E-BotCL achieves finer granularity in locating key features within the image. For instance, in the recognition of a bird's head and back, the heatmap produced by E-BotCL shows highly focused activation areas, distinctly separating these two features. In contrast, BotCL often exhibits numerous regions with no concept activation. Moreover, BotCL's concept activation map sometimes confuses the leg region with the background or abdomen, whereas E-BotCL correctly identifies the legs as distinct concepts, forming well-defined attention regions. This ability to capture spatially separated yet semantically related concepts significantly enhances the interpretability of the model's decision-making process.

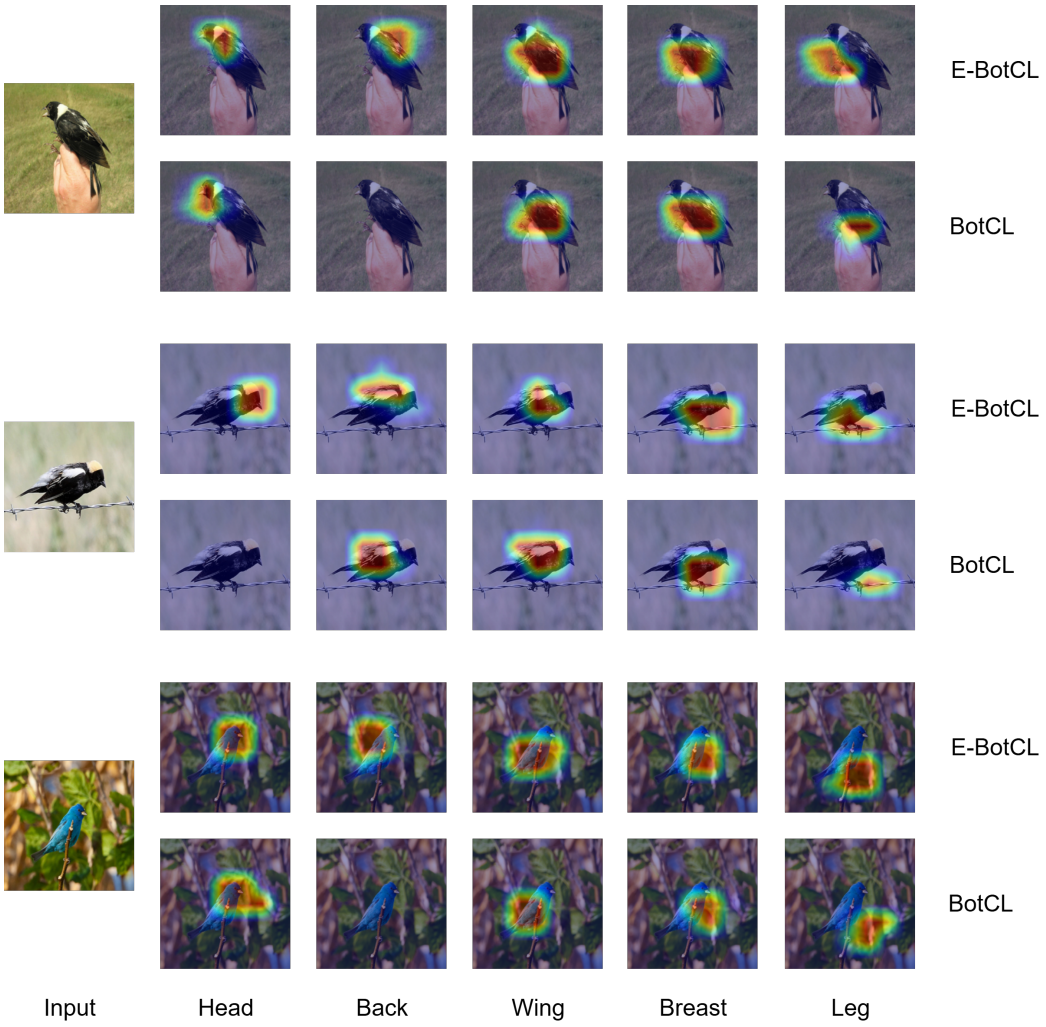

Figure 3. Visualization comparison of five key body parts between BotCL and E-BotCL on the same input image.

In addition, Figure 4 shows examples of concept activations learned by E-BotCL on the CUB200 dataset, further illustrating the model's capacity for fine-grained interpretability. E-BotCL not only distinguishes between various body parts of a bird as independent concepts (e.g., cpt0 representing the bird's wings and cpt10 representing the bird's head) but also identifies more subtle and intricate patterns present on the bird's body. For example, the concepts activated for the bird's spots (cpt12 and cpt13) and the stripes on its wings (cpt15) are clearly visible in the figure, reflecting the model's ability to recognize and isolate fine-grained features that are crucial for concept explanation. These results emphasize the

enhanced granularity and flexibility of the E-BotCL framework in learning both the broader structural components and the finer texture-based details, which contributes significantly to improving the model's ability to explain concepts.

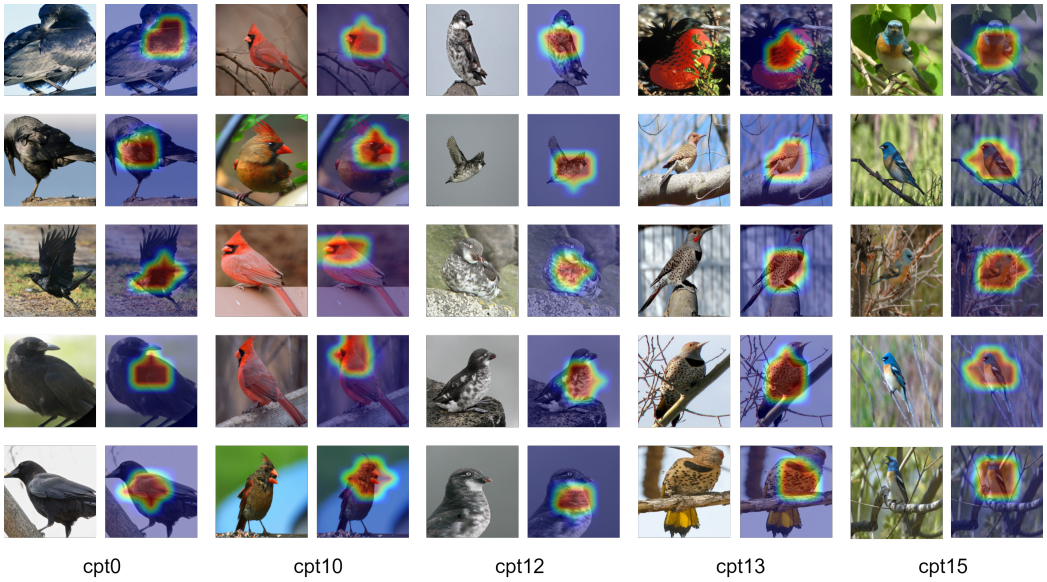

Figure 4. Examples of activated concepts learned from the CUB200 dataset.

By effectively capturing both high-level body parts and low-level texture patterns, E-BotCL not only enhances the transparency of the decision-making process but also strengthens the model's ability to provide clear and interpretable explanations for the concepts it identifies.

We set the number of concepts to 20 and selected the top 20 activated samples for each concept to analyze the activation patterns. The experimental results clearly demonstrate that all concepts in the E-BotCL method exhibited activation, as illustrated in Figure 5. This indicates a more consistent and robust activation across concepts than in the BotCL method, which showed notable shortcomings. Specifically, BotCL failed to activate samples for concepts cpt1, cpt14, and cpt17. Moreover, the activation distribution for the remaining concepts in BotCL was sparse, with concept cpt3 having only two activated samples; concepts cpt4, cpt6, cpt7, and cpt10 having four each; and concept cpt12 having three activated samples. This overall scarcity of activated samples suggests that the BotCL method struggles to generate meaningful and well-represented concepts, resulting in a less effective concept activation process.

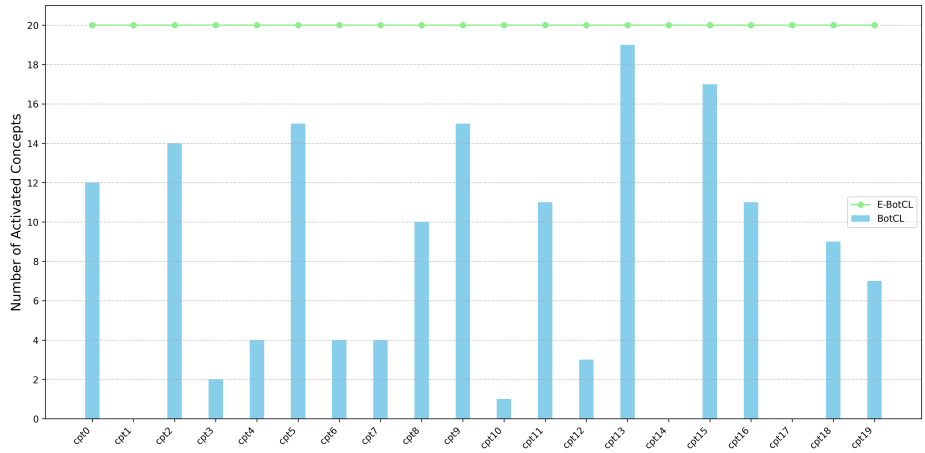

Figure 5. Concept activation status within each concept.

To further assess the concept explanation performance, we quantitatively compared the internal similarity within each concept for both E-BotCL and BotCL. Higher internal similarity indicates better alignment and coherence within the concept. As shown in Figure 6, the internal similarity for the concepts generated by E-BotCL was consistently superior to that of BotCL, signifying that E-BotCL produces more coherent and tightly defined concepts. This higher similarity is indicative of the method's ability to capture more accurate and consistent concept representations, which is essential for interpretability in model decision-making.

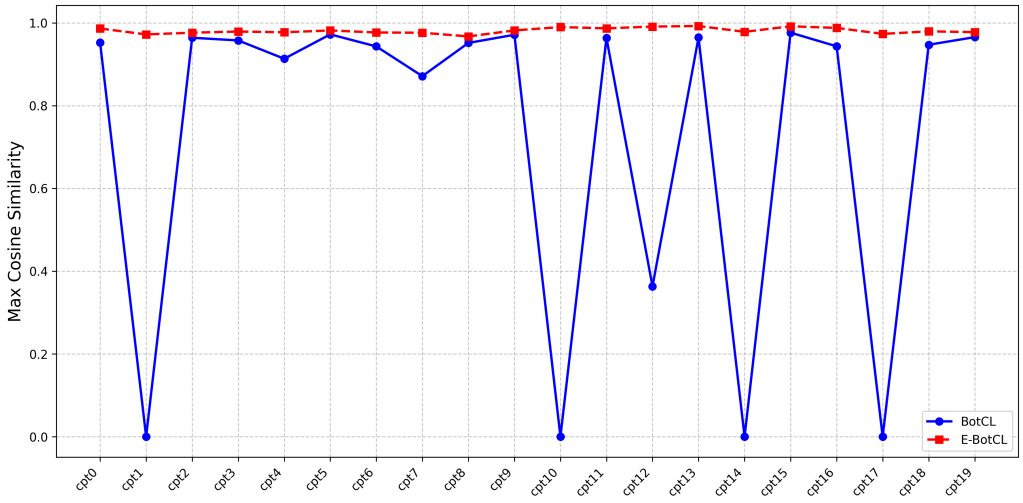

Figure 6. Maximum cosine similarity within each concept category.

Additionally, we evaluated the degree of independence between concepts, where a lower independence value indicates a higher degree of overlap and interaction between concepts—often a desirable characteristic in complex models that aim to reflect real-world semantic relationships. The Distinctiveness Average Similarity (DAS) for E-BotCL was 0.592, outperforming the 0.578 achieved by BotCL. This observed difference can be attributed to the fact that BotCL generates certain meaningless concepts that do not exhibit strong activation patterns or meaningful relationships with other concepts, leading to lower internal similarity and higher independence. In contrast, E-BotCL produces concepts that, while distinct, demonstrate a degree of overlap in their activations. This overlap suggests that the concepts in E-BotCL are more semantically coherent and interrelated, which ultimately leads to the observed higher overall similarity.

4.3. Classification Performance

We conducted a comprehensive comparison of the performance of E-BotCL with BotCL, k-means clustering, Principal Component Analysis (PCA) (re-implemented from [12,16]), and other leading concept-based models. The results are summarized in Table 1. Notably, E-BotCL outperforms all baseline methods, achieving the highest accuracy on both the CUB200 and ImageNet datasets. This reinforces our hypothesis that contrastive self-supervision plays a pivotal role in facilitating effective concept discovery, providing both interpretability and robustness to the learned representations.

To further understand the relationship between the number of concepts and classification accuracy, we explored this dynamic for both E-BotCL and BotCL on the CUB200 dataset. As depicted in Figure 7, E-BotCL maintains strong performance when the number of concepts is between 20 and 200. This range demonstrates that the method is capable of adapting well to datasets of varying sizes, consistently delivering competitive accuracy even for smaller to medium-sized concept sets. In particular, E-BotCL excels in situations

where the number of concepts is neither too small nor too large, offering a balance that ensures high-quality concept learning.

Table 1. Performance Comparison of Classification Accuracy. The best concept-based method is highlighted in bold. For ImageNet, the top 200 classes were used.

	CUB200	ImageNet
k-means [16]	0.063	0.427
PCA [16]	0.044	0.139
SENN [25]	0.642	0.673
ProtoPNet [42]	0.725	0.752
BotCL	0.725	0.768
E-BotCL	**0.726**	**0.770**

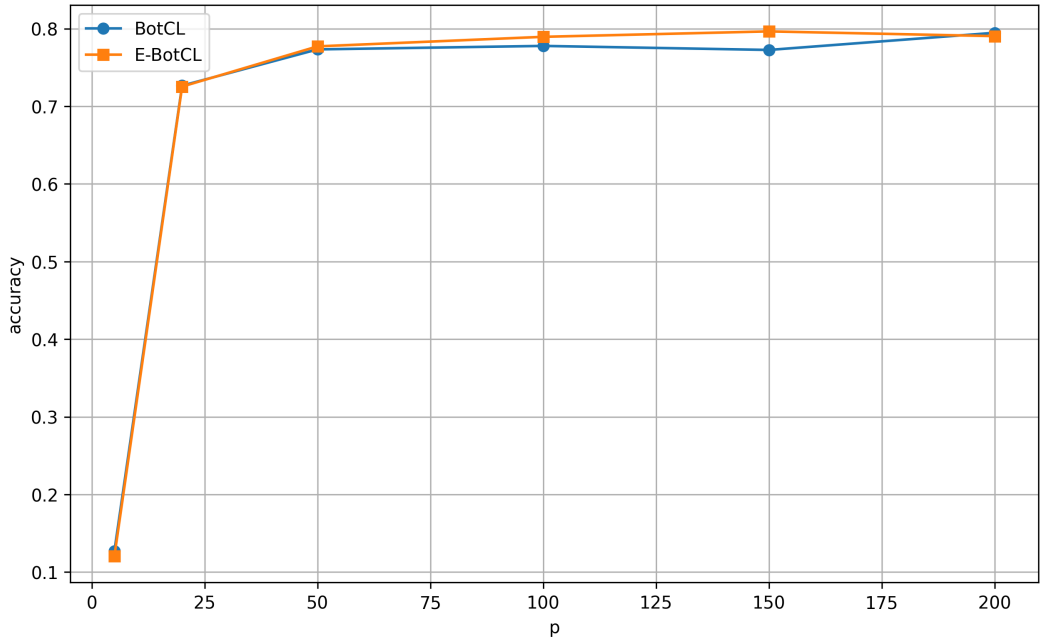

Figure 7. Impact of the number of concepts (p) on BotCL and E-BotCL classification accuracy.

On the other hand, when the number of concepts is either below 20 or above 200, BotCL emerges as the superior model. These observations suggest that, while E-BotCL is robust within an optimal concept range, BotCL may be more effective in scenarios that involve either a very small or a very large number of concepts. This indicates that the effectiveness of concept-based learning approaches is significantly influenced by the scale and distribution of concepts, with E-BotCL showing particular promise for moderate ranges. These results collectively confirm that the dual-path contrastive learning strategy employed by E-BotCL contributes significantly to both concept discovery and classification performance.

4.4. User Study

The user study aims to evaluate the performance of E-BotCL in human interpretability using real-world datasets. Participants were tasked with observing test images annotated with concept attention maps and selecting the phrase from a predefined vocabulary that most accurately describes the concept (i.e., the attended region). If no consistent visual element could be identified, participants were allowed to choose "none". For each concept in the CUB200 dataset, 20 participants were recruited for evaluation. For both E-BotCL and BotCL, we selected 200 concept attention maps for each method to be used in the user evaluation.

Table 2 compares the performance of two methods, BotCL and E-BotCL, across three key metrics [12]: **Concept Discovery Rate (CDR)**, **Concept Consistency (CC)**, and **Mutual Information between Concepts (MIC)**. These metrics were selected to provide a comprehensive assessment of the methods' effectiveness in concept learning and interpretability, particularly in terms of their ability to discover and express concepts.

Table 2. Results of user study.

	CDR (↑)		CC (↑)		MIC (↓)	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**
BotCL	0.3896	0.4527	0.2466	0.3361	0.2489	0.0787
E-BotCL	1.0000	0.0000	0.6952	0.1396	0.1706	0.0512

CDR measures the proportion of participants who successfully identify and generalize visual elements as valid concepts. A higher CDR indicates that participants are better at recognizing consistent and representative visual features from the data, thereby forming clearer concepts. E-BotCL performs exceptionally well in terms of CDR, suggesting that the method is highly consistent in the concept discovery process, with all participants successfully identifying and generalizing the concept. In contrast, BotCL shows significant variability in its CDR, indicating that BotCL has unstable performance in concept discovery and struggles to provide consistent visual feature guidance for all participants.

CC quantifies the degree of agreement between participants in their expressions of the same concept, reflecting the method's effectiveness in guiding participants toward a consistent understanding of the concept. A high CC value suggests that different participants use similar language and terminology to describe the same concept, indicating that the concept is both clear and stable. The experimental results reveal that E-BotCL achieves a CC mean of 0.6952 with a standard deviation of 0.1396, demonstrating its ability to effectively guide participants toward a highly consistent conceptual understanding, with good stability across different participants. In contrast, BotCL's CC is 0.2466 with a standard deviation of 0.3361, showing considerable fluctuation and highlighting its limitations in ensuring concept consistency, with substantial variation in participants' understanding.

MIC reflects the similarity of response distributions between different concepts, with lower values indicating greater differentiation between concepts and avoidance of overlap. For an effective concept learning method, the MIC should be as low as possible to ensure that each concept remains sufficiently distinct. E-BotCL excels in MIC, indicating that it effectively minimizes redundancy between concepts, preventing excessive overlap. In contrast, BotCL's MIC suggests some degree of overlap and information redundancy between concepts, leading to poorer differentiation.

Overall, E-BotCL outperforms BotCL on all three metrics, providing further evidence of its superiority in enhancing the quality of concept discovery and learning. Specifically, in the context of interpretability and model transparency, E-BotCL better supports model explainability, ensuring that the learned concepts not only exhibit high consistency in expression but also offer clearer and more distinguishable representations.

4.5. Ablation Study

The ablation study presented in Table 3 examines the impact of different components of the E-BotCL framework on the CUB200 dataset. Specifically, we evaluate the inclusion of Concept Learning (CL) and Multi-Task Loss (MTL) alongside the baseline BotCL approach in terms of accuracy, model complexity (number of parameters), training time, and GPU memory consumption.

Table 3. Ablation study of E-BotCL components on CUB200 dataset.

BotCL	CL	MTL	Acc	#Params	Training Time	GPU Memory
✓			0.7733	14.37 M	65 min	6.8 GB
✓	✓		0.7758	15.61 M	83 min	7.1 GB
✓		✓	0.7765	16.10 M	96 min	9.3 GB
✓	✓	✓	0.7772	17.34 M	107 min	9.6 GB

From the results, we observe that the baseline BotCL model achieves an accuracy of 0.7733 with a parameter count of 14.37 M. Introducing the CL component improves accuracy to 0.7758 but comes with an increase in model complexity (15.61 M parameters) and a rise in training time from 65 to 83 min. Similarly, adding the MTL component to BotCL results in an accuracy of 0.7765 while further increasing the parameter count to 16.10 M and requiring 96 min for training. The full E-BotCL model, which incorporates both CL and MTL, achieves the highest accuracy (0.7772). However, this comes at the cost of additional computational demands, with a parameter count of 17.34 M, a training time of 107 min, and increased GPU memory consumption of 9.6 GB.

These results indicate that both CL and MTL contribute to performance improvements, albeit at the expense of higher computational costs. The incremental accuracy gains suggest that the inclusion of these components enhances the model's interpretability and robustness without significantly compromising efficiency. Therefore, the full E-BotCL framework represents a balanced trade-off between accuracy and computational resources, making it a viable approach for interpretable image classification tasks.

5. Conclusions

This study addresses the critical challenge of balancing model performance and interpretability in deep learning by introducing the Enhanced Bottleneck Concept Learner (E-BotCL). By integrating self-supervised contrastive learning, attention mechanisms, and multi-task regularization, E-BotCL autonomously discovers human-interpretable semantic concepts, eliminating the need for manual annotations or predefined concept sets. The dual-path contrastive framework, inspired by SimSiam, facilitates robust concept prototype learning, while the slot-based attention mechanism and feature aggregation strategies ensure precise spatial localization and semantic alignment of the discovered concepts.

Experimental results on the CUB200 and ImageNet datasets demonstrate the superiority of E-BotCL over existing concept-based models, achieving state-of-the-art classification accuracy rates of 72.6% and 77.0%, respectively, while maintaining high interpretability. Notably, E-BotCL excels in concept consistency and distinctiveness, as evidenced by quantitative metrics (e.g., higher intra-concept similarity) and qualitative visualizations (e.g., accurate localization of bird body parts and patterns). These findings underscore the framework's ability to bridge the semantic gap between low-level features and high-level, human-understandable concepts.

This work significantly advances the practical application of explainable AI in domains that necessitate transparent decision-making, including healthcare, autonomous systems, and sensor-driven technologies. By enhancing the interpretability of deep learning models, particularly within sensor-based applications, our approach contributes to the development of more reliable and trustworthy sensor systems. For instance, in the context of autonomous vehicles, the ability to explain how sensor data (from cameras, LiDAR, and radar) informs decision-making can substantially improve both safety and user trust. Similarly, in medical sensor technologies, offering interpretable AI-driven insights into diagnostic sensor data can empower clinicians to make more informed and accurate decisions. Future research could focus on extending E-BotCL to multimodal tasks, refining concept diversity through

adversarial training, or incorporating domain-specific constraints to enhance performance in specialized applications.

Author Contributions: Conceptualization, Z.N. and X.C.; methodology, X.C.; software, X.C. and Z.J.; validation, X.C. and Z.J.; formal analysis, X.C.; investigation, X.C.; resources, L.L.; data curation, Z.N.; writing—original draft preparation, X.C.; writing—review and editing, X.C.; visualization, X.C.; supervision, L.L.; project administration, L.L.; funding acquisition, L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Taishan Scholar Program, the Rizhao-Qufu Normal University Joint Technology Transfer Center, the Shandong Science Fund Program for Excellent Young Scientists (Overseas), and the Rizhao Science Fund Program for Excellent Young Scientists (Overseas). Additionally, this work was supported by the National Natural Science Foundation of China under grant no. 62372266.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All datasets used in this article are publicly accessible.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, Y.W.; Jain, L.C. Deep learning in healthcare. In *Paradigms and Applications*; Springer: Berlin/Heidelberg, Germany, 2020.
2. Zablocki, É.; Ben-Younes, H.; Pérez, P.; Cord, M. Explainability of deep vision-based autonomous driving systems: Review and challenges. *Int. J. Comput. Vis.* **2022**, *130*, 2425–2452. [CrossRef]
3. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
4. Caruana, R.; Lou, Y.; Gehrke, J.; Koch, P.; Sturm, M.; Elhadad, N. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 1721–1730.
5. Chattopadhay, A.; Sarkar, A.; Howlader, P.; Balasubramanian, V.N. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 839–847.
6. Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.R.; Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **2015**, *10*, e0130140. [CrossRef] [PubMed]
7. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
8. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv* **2013**, arXiv:1312.6034.
9. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. [CrossRef]
10. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic attribution for deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 3319–3328.
11. Poeta, E.; Ciravegna, G.; Pastor, E.; Cerquitelli, T.; Baralis, E. Concept-based explainable artificial intelligence: A survey. *arXiv* **2023**, arXiv:2312.12936.
12. Wang, B.; Li, L.; Nakashima, Y.; Nagahara, H. Learning bottleneck concepts in image classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 10962–10971.
13. Fong, R.; Patrick, M.; Vedaldi, A. Understanding deep networks via extremal perturbations and smooth masks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2950–2958.

14. Shrikumar, A.; Greenside, P.; Kundaje, A. Learning important features through propagating activation differences. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 3145–3153.
15. Wang, B.; Li, L.; Verma, M.; Nakashima, Y.; Kawasaki, R.; Nagahara, H. MTUNet: Few-shot image classification with visual explanations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 2294–2298.
16. Yeh, C.K.; Kim, B.; Arik, S.; Li, C.L.; Pfister, T.; Ravikumar, P. On completeness-aware concept-based explanations in deep neural networks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 20554–20565.
17. Ras, G.; Xie, N.; Van Gerven, M.; Doran, D. Explainable deep learning: A field guide for the uninitiated. *J. Artif. Intell. Res.* **2022**, *73*, 329–396. [CrossRef]
18. Lundberg, S. A unified approach to interpreting model predictions. *arXiv* **2017**, arXiv:1705.07874.
19. Samek, W. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv* **2017**, arXiv:1708.08296.
20. Akhtar, N. A survey of explainable ai in deep visual modeling: Methods and metrics. *arXiv* **2023**, arXiv:2301.13445.
21. Rudin, C.; Chen, C.; Chen, Z.; Huang, H.; Semenova, L.; Zhong, C. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Stat. Surv.* **2022**, *16*, 1–85. [CrossRef]
22. Carvalho, D.V.; Pereira, E.M.; Cardoso, J.S. Machine learning interpretability: A survey on methods and metrics. *Electronics* **2019**, *8*, 832. [CrossRef]
23. Nesvijevskaia, A.; Ouillade, S.; Guilmin, P.; Zucker, J.D. The accuracy versus interpretability trade-off in fraud detection model. *Data Policy* **2021**, *3*, e12. [CrossRef]
24. Koh, P.W.; Nguyen, T.; Tang, Y.S.; Mussmann, S.; Pierson, E.; Kim, B.; Liang, P. Concept bottleneck models. In Proceedings of the International Conference on Machine Learning, Virtual Event, 13–18 July 2020; pp. 5338–5348.
25. Alvarez Melis, D.; Jaakkola, T. Towards robust interpretability with self-explaining neural networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*.
26. Ge, Y.; Xiao, Y.; Xu, Z.; Zheng, M.; Karanam, S.; Chen, T.; Itti, L.; Wu, Z. A peek into the reasoning of neural networks: Interpreting with structural visual concepts. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 2195–2204.
27. Ghorbani, A.; Wexler, J.; Zou, J.Y.; Kim, B. Towards automatic concept-based explanations. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
28. Laugel, T.; Lesot, M.J.; Marsala, C.; Renard, X.; Detyniecki, M. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. *arXiv* **2019**, arXiv:1907.09294.
29. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 586–595.
30. Chen, X.; Fan, H.; Girshick, R.; He, K. Improved baselines with momentum contrastive learning. *arXiv* **2020**, arXiv:2003.04297.
31. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 9729–9738.
32. Chen, X.; He, K. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 15750–15758.
33. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. Bootstrap your own latent-a new approach to self-supervised learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21271–21284.
34. Fong, R.C.; Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3429–3437.
35. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
36. Redmon, J. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
37. Li, L.; Wang, B.; Verma, M.; Nakashima, Y.; Kawasaki, R.; Nagahara, H. Scouter: Slot attention-based classifier for explainable image recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1046–1055.
38. Locatello, F.; Weissenborn, D.; Unterthiner, T.; Mahendran, A.; Heigold, G.; Uszkoreit, J.; Dosovitskiy, A.; Kipf, T. Object-centric learning with slot attention. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 11525–11538.
39. Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; Perona, P. *Caltech-UCSD Birds 200*; Technical Report CNS-TR-2010-001; California Institute of Technology: Pasadena, CA, USA, 2010.
40. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.

41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
42. Chen, C.; Li, O.; Tao, D.; Barnett, A.; Rudin, C.; Su, J.K. This looks like that: Deep learning for interpretable image recognition. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.