

ORIGINAL RESEARCH

Open access

Interpretable Deep CNNs through Binary Tree Structure Convolution and Decision-Path Learning

Pieter Botha^{1*}, Anel Van Wyk¹, Johan Marais²

Abstract

While deep convolutional neural networks (DCNNs) have achieved substantial breakthroughs in computer vision, these models are frequently criticized for their lack of interpretability during decision-making. A core issue is that their underlying operational mechanism is generally perceived as a "black-box." To address this, the authors introduce the binary tree structure convolution (BTSC) module and control the activation thresholds of specific neurons to construct an interpretable DCNN framework. First, the authors develop the BTSC module, where every parent node branches into two distinct child layers, which are subsequently integrated into a standard DCNN architecture. The primary benefits of the BTSC module include: 1) child nodes emerging from separate parent nodes operate without mutual interference; 2) parent and child nodes are capable of knowledge inheritance. Second, by managing neural activation states, the authors design an information coding objective that directs neural nodes toward acquiring a predefined, targeted information coding scheme. Empirical evaluations demonstrate that: 1) the decisions executed by both ResNet and DenseNet models can be effectively clarified via the "decision information flow path" (termed the decision-path) established within the BTSC module; 2) this decision-path offers a logical explanation for the decision reversal mechanism (Robustness mechanism) inherent in DCNN models; 3) the reliability of a decision can be quantified by evaluating the alignment between the actual and anticipated decision-paths.

Keywords Interpretability, Deep Convolutional Neural Networks (DCNNs), Binary Tree Structure Convolution (BTSC), Decision-path

*Correspondence:

Pieter Botha
pieter.botha@sun.ac.za

¹ Department of Interpretable Deep Learning and Decision Paths, Faculty of Engineering, Stellenbosch University, Stellenbosch, South Africa

² Department of Binary Tree Convolutional Networks, Faculty of Engineering, University of Pretoria, Pretoria, South Africa

Introduction

Driven by recent breakthroughs in deep learning within computer vision, convolutional neural networks (CNNs) have garnered significant interest. They are currently being implemented across a diverse range of applications, matching or surpassing human performance levels in numerous environments [1-5]. Nonetheless, due to factors such as structural depth (massive parameters), complex interlayer dependencies, and nonlinear activation functions, their internal learning and decision-making mechanisms remain categorized as a "black box" [6]. Consequently, apprehensions persist regarding the deployment of deep CNN models across critical fields where the models fail to provide sufficient transparency and safety assurances. For instance, artificial intelligence (AI) frameworks are increasingly central to domains like autonomous transportation, facial recognition systems, and AI-assisted healthcare [7]. Given these challenges, the utility and security of AI applications will remain constrained until they can be adequately understood,

explained, and regulated. Accordingly, theoretical advancements in deep CNN architectures are required to enhance their overall interpretability.

In the context of human-computer collaboration, the primary requirement is ensuring that the feature spaces extracted during model training align with human understanding. Most contemporary CNN models lack this interpretability due to two fundamental factors: 1) The foundational optimization criteria of deep neural architectures ignore human-centric explanation needs, focusing strictly on performance metrics rather than interpretability goals; 2) The feature representations learned by deep neural networks do not correspond to human knowledge structures. As a result, when applied to image recognition, DCNN models are viewed as unexplainable due to the following limitations:

- **Reason 1 (R1):** The specific categories of knowledge acquired by a DCNN model during a task remain fundamentally unknown, forcing users to passively accept whatever internal representations the model develops.
- **Reason 2 (R2):** The operational logic within the hidden layers of a DCNN model is typically ambiguous, leaving users without clear criteria for when to trust the model's outputs.

Insights from biological neural information coding indicate that while certain neurons within a neural circuit process multiple information types, others are dedicated strictly to encoding specific signals [8]. Consequently, a primary objective in cognitive science is determining the precise functions of individual neurons during cognitive tasks. Conversely, deep artificial neural networks comprise countless neurons that often simultaneously encode diverse task-related variables. This high degree of coupling and disorder within neural network information coding complicates the explanation and comprehension of the model's decision-making. Drawing inspiration from neuroscience, restructuring the DCNN architecture to explicitly define its information coding mechanisms represents a promising strategy for developing interpretable neural networks.

This study aims to resolve the aforementioned challenges (R1 and R2) by integrating a specialized neural module into standard DCNN architectures. The augmented models are designed to maintain performance levels competitive with their baseline counterparts. Drawing on the highly interpretable nature of decision tree architectures, we introduce a binary tree structure convolution (BTSC) layer. Within the BTSC setup, child layers branching from distinct parent layers are isolated from mutual interference, and knowledge inheritance is maintained between parent and child tiers. Unlike conventional convolutional layers, which require layer-by-layer feature decoupling due to highly entangled semantic spaces, our approach enforces specific, pre-determined decision information coding across pairs of child nodes derived from a common parent. Ultimately, experimental results validate that this framework successfully clarifies the internal decision-making process.

Related Works

Visualisation methods

One direct approach to understanding the features or concepts captured by a neural model involves mapping internal layers into human-interpretable imagery, which aids in clarifying the underlying mechanics of deep neural networks. For instance, Yosinski *et al.* operated on the premise that highly active neurons capture essential knowledge from input graphics during training [9]; they analyzed neural networks by rendering the primary features resolved by these dominant neurons [10, 11]. Selvaraju *et al.* introduced a method to generate and visualize gradient-based thermal maps from convolutional layers, highlighting and identifying influential pixels within the source image [12]. Chatfield *et al.* evaluated sensitivity to minute pixel alterations by computing partial derivatives of classification scores relative to input features [13, 14]. William *et al.* introduced alternative analytical techniques and tools to evaluate classification landscapes, providing visualizations to represent these outcomes clearly [15]. Additionally, Yang *et al.* developed a generalized framework enabling any standard convolutional neural network to produce text descriptions of its latent filter-level representations, allowing for the analysis of trained networks independently of their original training datasets [16]. These visualization techniques offer direct insight into the internal operations of deep neural networks.

Feature importance analysis

Shape hypothesis: A prevalent view holds that convolutional neural networks construct structural shapes by compounding low-level features layer by layer until object classification becomes feasible. This perspective is mirrored by LeCun [17], who states that "intermediate layers in a CNN model recognise parts of familiar objects and subsequent layers can detect objects as combinations of the parts." This shape-centric hypothesis is reinforced by multiple empirical studies; for example, visualization techniques [18, 19] regularly isolate object components within the higher layers of a CNN. Furthermore, Ye *et al.* introduced a dual features network architecture wherein one sub-network captures gradient feature information while the other functions as a conventional texture-biased network, thereby enhancing the shape bias to better match human visual interpretation [20]. Similar manifestations of "shape bias" have been documented in CNN implementations for image segmentation, image classification, and human object recognition, suggesting that structural shape properties frequently carry greater significance than alternative feature types [16, 21-27].

Texture bias: Conversely, competing research suggests that texture features may hold greater significance. This perspective traces back to the bag-of-words framework [28], which asserts that localized characteristics like texture provide higher utility than global representations. Multiple investigations have demonstrated that CNN models maintain high performance even when the spatial topology of an image is disrupted [29]. Furthermore, conventional CNN configurations experience a decline in object recognition performance when shapes are preserved but texture details are stripped away [30]. Additional findings indicate that CNNs trained on ImageNet exhibit an inherent inclination toward texture patterns, and that amplifying this texture bias enhances both classification accuracy and structural robustness [31]. Finally, utilizing style transfer algorithms, Robert *et al.* [32] demonstrated that texture properties exert a more pronounced influence on the ResNet-50 architecture than other feature dimensions.

Model diagnosis

Methods of proxy model: Due to their inherent complexity, DNN models are generally difficult to interpret and understand. Consequently, one prominent research direction focuses on mapping these complex architectures into more comprehensible rules or secondary models. For instance, LIME (local interpretable model-agnostic explanations) [33] seeks to approximate the non-linear classification boundaries of an original model by utilizing linear approximations on local instances. Kim [34] introduced an approach where a decision tree replicates the input-output mappings realized by a neural network, thereby clarifying the decision process via the class probability distributions across leaf nodes. Zhou Zhihua *et al.* [35-37] investigated the viability of utilizing non-neural network components to design deep learning architectures, aiming to simultaneously enhance representational capacity and interpretability. Ding [38] segmented a deep learning architecture into a cognitive system and a perceptual system, routing the intuitive predictions generated by the perceptual component into the cognitive framework. Hehn *et al.* [39] developed an end-to-end training methodology for decision trees and decision forests, improving structural interpretability through spatial regularization while uncovering training trajectories within the decision tree during optimization. Nauta *et al.* [40] introduced an intrinsically interpretable deep learning strategy called ProtoTree. Similarly, Wan A *et al.* [41] proposed the Neural-Backed Decision Tree (NBDT), which preserves the primary feature-extracting segments of a network but substitutes the final fully connected layer with a decision tree; it executes classification decisions by calculating the inner product of feature vectors at each sub-node, thereby improving overall network interpretability.

Feature space decoupling: Feature decoupling refers to the semantic isolation of feature representations acquired by a neural network. This method forces specific encoding modules to capture distinct semantic concepts to achieve clear, interpretable feature expressions. Upchurch P [42] suggested learning an interpolative transformation across distinct categories using external class labels, enabling the modulation of high-level semantics and generated outputs. Cai *et al.* [43] extracted discrete and continuous features via clustering algorithms, utilizing visualization techniques to interpret spatial characteristics. M. Connor [44] employed generative models to map latent variables and data patterns within a

manifold space, capturing non-linear data distributions to establish a direct mapping between semantic details and raw data. Some investigations attempt to discover a linear combination of features to represent a pre-established concept [45], or to identify concepts autonomously by clustering image patches and designating those clusters as novel concepts [46]. Ding [47] decoupled the latent variables of a Variational Autoencoder (VAE) by incorporating guidance data, achieving interpretability at a semantic layer. Guan *et al.* [48] integrated collaborative filtering into StyleGAN, introducing feature loss and latent space loss during training to enable more precise extraction of latent space attributes, thereby clarifying the feature extraction process. Zhi Chen *et al.* [49] implemented a mechanism termed concept whitening to decouple the feature space within a designated layer of a CNN, aligning the axes of the latent space directly with specific target concepts.

Summary

We provide a comparative summary of the aforementioned research tracks. Visualization methodologies attempt to project hidden layers into human-comprehensible imagery, which partially resolves R1. Feature importance analyses focus on identifying which specific features contribute most heavily to a DCNN model's decision-making during a task; however, this provides an incomplete solution to both R1 and R2. Proxy model methods aim to translate complex neural networks into more intelligible rules or alternative models, which constitutes an approximate functional simulation rather than an explanation of the DCNN model's internal mechanics. Feature space decoupling attempts to isolate the feature space learned by a neural network, directing hidden neurons toward specific semantic properties to resolve R1. Nevertheless, feature space decoupling must be executed across every single semantic space layer by layer, and this decoupling technique remains detached from the underlying decision-making mechanism.

Our work focuses on engineering a DCNN model characterized by transparent information coding to directly explain its internal decision-making mechanism. We outline our core contributions below:

- **1)** We introduce a BTSC module designed for integration into standard DCNN frameworks to establish a partially interpretable model. Within this architecture, specific neural nodes are restricted to expressing clear decision information coding, which can be either preassigned or learned (R1).
- **2)** We propose learning interpretable knowledge structures modeled on anticipated decision-related information coding. This coding framework provides the foundation for clarifying the decision-making process and quantifying decision credibility (R2).
- **3)** The BTSC functions as a plug-and-play knowledge module that can be embedded into any standard DCNN architecture without degrading its original performance metrics.

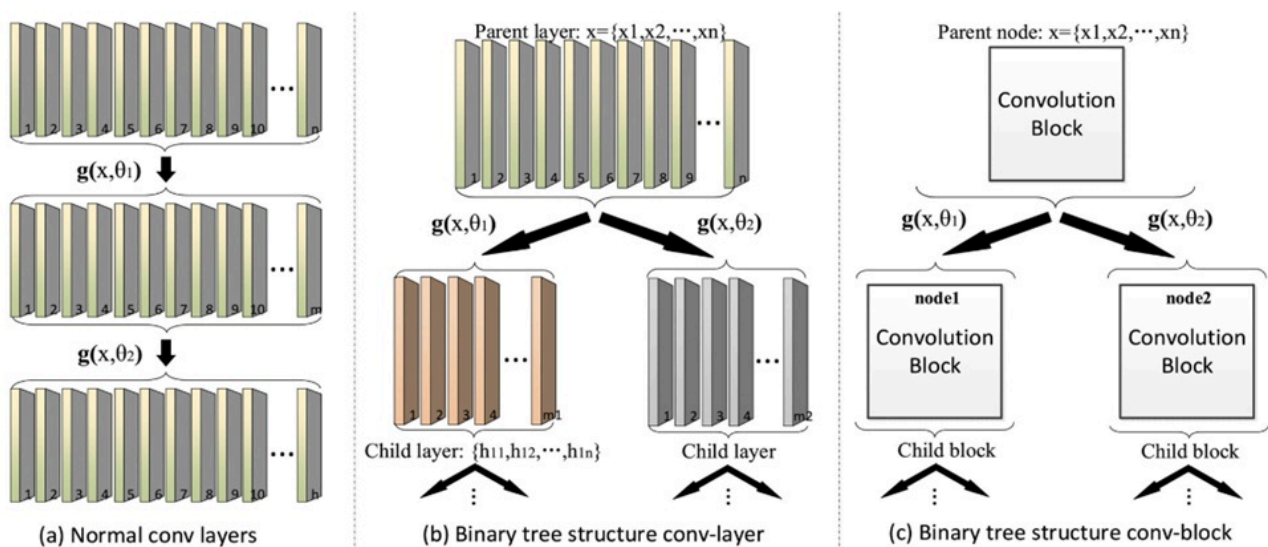


Figure 1. Normal conv-layers and BTSC module. (a) Normal conv-layers; (b) Binary tree structure conv-layer; (c) Practically, a convolution block that contains several conv-layers can be used to replace the single conv-layer in (b).

Materials and Methods

Binary tree structure conv-layer (BTSC) module

Standard neural network models generally fail to enforce conceptual isolation across hidden layers [49], causing information to become highly intertwined between consecutive stages. Consequently, current interpretability methods are typically limited to analyzing a single, isolated convolutional layer. Inspired by the transparent properties of decision tree models, we developed a BTSC module (Figure 1) wherein each parent convolutional layer branches into two self-contained child convolutional layers (nodes) using two independent weight matrices (see θ_1 and θ_2 in Figure 1b). The BTSC module functions as an interpretable component capable of being embedded into any conventional CNN framework.

This module introduces several important advantages for engineering interpretable neural networks. First, it ensures that all child layers derived from separate parent nodes operate independently, while establishing a semantic inheritance relationship between parent and child layers—a characteristic highly beneficial for developing explainable CNN models. Second, it allows us to guide each pair of child nodes within a parent structure to independently capture information coding via either supervised or unsupervised learning paradigms. Notably, this module departs from traditional decision tree models by utilizing a tree-structured neural network framework, successfully combining the positive attributes of both decision trees and neural networks. It is crucial to emphasize that our framework leverages an endogenous neural module, rather than an external non-neural proxy, to clarify the internal decision-making mechanisms of neural network models.

Decision information flow path

Overview

A critical element in designing the BTSC module involves assigning specific nodes to capture distinct information coding. This intentional configuration ensures that each node is dedicated to transmitting particular data streams, thereby establishing an anticipated route for decision information flow within the BTSC module, termed the expected decision-path. A non-zero activation level in a neural node indicates that a corresponding volume of information passes through it, which can influence the final prediction of the network. Within this framework, the route that channels the highest volume of information is defined as the primary decision-path. This primary decision-path maps out the underlying logical sequence of the decision-making process.

Task-related neural coding

In neuroscience, it is well established that biological neurons generally participate in encoding task-relevant information only after crossing a specific activation threshold. Furthermore, neural representations in biological systems typically rely on clusters of neurons rather than single isolated cells. Analogously, in a deep neural network, a node can influence subsequent layers and affect the ultimate output only when its average activation level (representing a group of features) exceeds zero, as formulated in Eq. 1:

$$f_{node}(x) = Avg(g(\theta_{node}, x)) \quad (1)$$

In Eq. 1, the function g represents the combined operations of linear convolution and non-linear mapping, while Avg denotes the global average activation level across all units in the final layer of a node. This metric serves as a quantitative indicator of the information volume allowed to pass through the neural node. For a given input sample, an activation state of $f_{node}(x) > c_1$ (where c_1 is a constant threshold) indicates that this specific Binary Tree layer node is actively involved in encoding information for the current input x . Conversely, if $f_{node}(x) < c_2$ (where c_2 is a minor value near zero), the node's contribution to encoding the current input can be neglected. If a node's activation level exceeds c_1 exclusively for samples belonging to specific target classes within a training set, it demonstrates that the node is uniquely dedicated to encoding decision-related information for those specific categories, as formalized in Eq. 2:

$$\forall x \in \{classes\}, f_{node}(x) > c_1 \ \& \ \forall x' \notin \{classes\}, f_{node}(x') < c_2 \quad (2)$$

A standard DCNN model may not achieve information coding separation

To construct an interpretable neural network grounded in clear information coding, the semantic spaces within its hidden layers must exhibit distinct separation. This clear delineation is necessary to explicitly determine the precise information encoded by individual neurons. However, separating information coding is generally not an implicit objective during standard neural network optimization. For example, visualizing the activation intensities of a single layer—as illustrated in Figure 2—using a randomly selected sample from each category reveals that the resulting activation profiles lack clear boundary separations across different classes. This blending occurs because the massive number of neurons within a deep neural network leads to highly complex and entangled information pathways, making the internal decision logic difficult to analyze or comprehend.

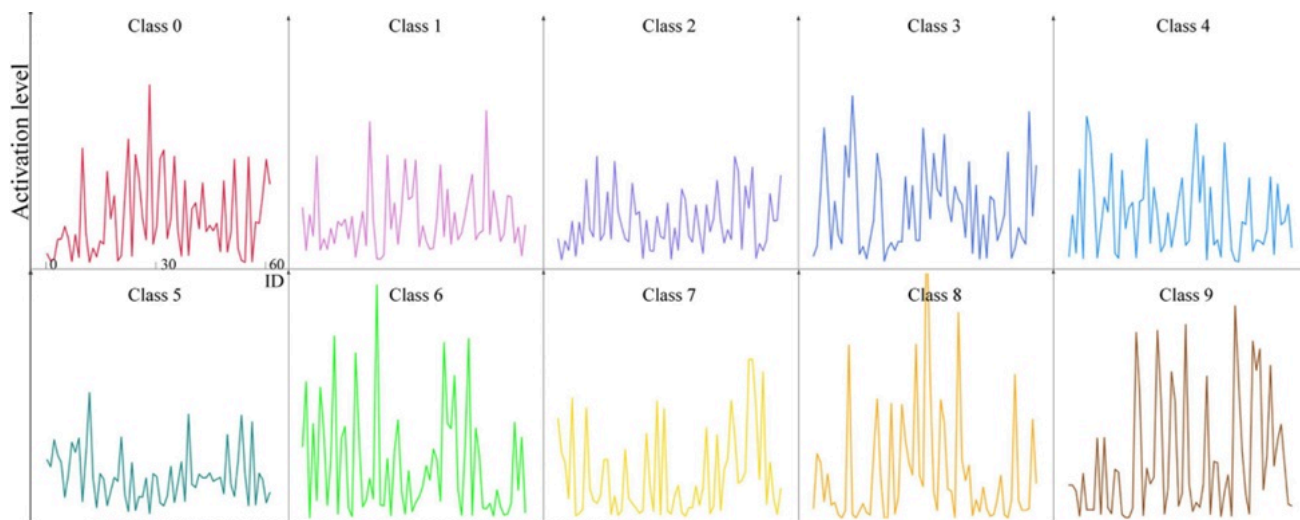


Figure 2. Activation level in the last layer of a standard ResNet-20 model for the samples selected from each class.

Learning expected neural coding for the BTSC module

For a standard DCNN architecture, our methodology first integrates the BTSC module by embedding it into designated convolutional layers or utilizing it to substitute conventional layers. Following this, specific nodes within the BTSC module are assigned to encode decision information for predefined target classes. Due to the geometric constraints of the tree topology, information originating from samples of specific classes is expected to traverse a distinct, pre-allocated sequence of nodes. This route constitutes the expected decision-path.

Guided by this framework and the principles of task-related neural coding, customized objective functions are implemented to force the BTSC module to learn these expected decision-paths during classification tasks. The two child nodes residing within the same layer of the Binary Tree are governed by distinct objective functions, where $f_1(\cdot)$ corresponds to node1 and $f_2(\cdot)$ corresponds to node2, as shown in Eqs. 3 and 4:

$$J_1(\theta) = \text{Min} \left(\sum_{x_i} (f_1(x_i) - \delta)^2 + f_2(x_i)^2 \right), x_i \in \{\text{classes}_i\} \quad (3)$$

$$J_2(\theta) = \text{Min} \left(\sum_{x_j} f_1(x_j)^2 + (f_2(x_j) - \delta)^2 \right), x_j \in \{\text{classes}_j\} \quad (4)$$

During optimization, when an input sample x belongs to classes_i (comprising one or multiple categories), the value of $f_1(x)$ is driven toward a positive constant δ , while $f_2(x)$ converges toward 0. This behavior indicates that only the neurons in node 1 are stimulated by samples from classes_j , while the units in node 2 remain dormant, as expressed in Eq. 3. Conversely, when the input sample x belongs to classes_j , the value of $f_1(x)$ approaches 0, while $f_2(x)$ is driven toward δ (Eq. 4). By introducing these constraints into the training loop, only the pre-assigned neural nodes will exhibit high activation in response to task-specific input features. These responsive nodes collectively map out the decision-path. The objective functions J_1 and J_2 are applied exclusively to the final layer of each pair of child nodes within the BTSC module, where the symbol θ represents all learnable parameters within the BTSC module, and the primary classification loss is appended at the terminal output layer.

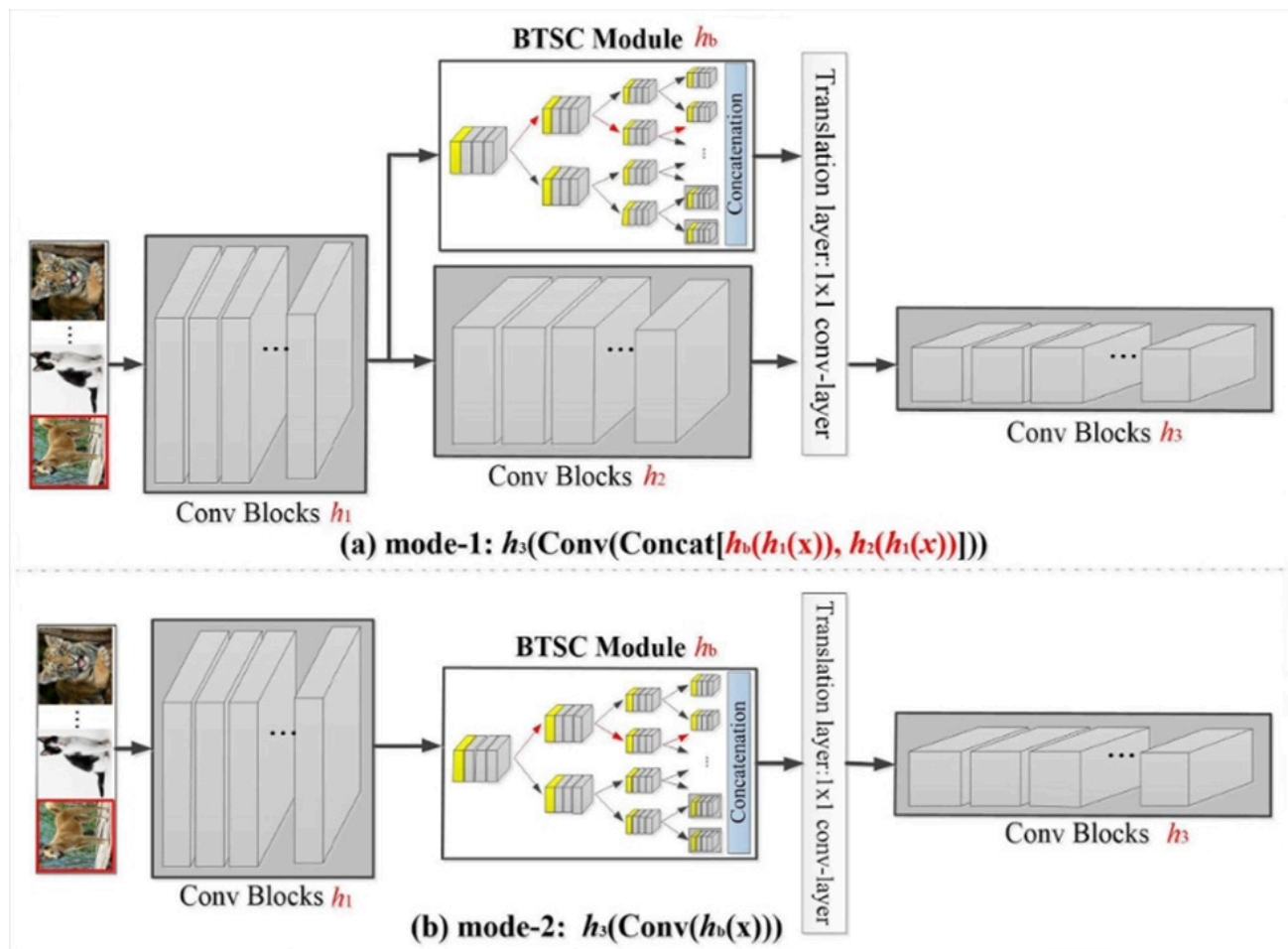


Figure 3. Two typical usages (mode-1 (a) and mode-2 (b)) in a normal DCNN model.

Use of the BTSC module in a standard DCNN

This paper introduces two distinct integration strategies. As illustrated in [Figure 3a](#), the BTSC module can be integrated as an auxiliary interpretability component at various depths within a CNN architecture. Alternatively, it can directly replace conventional operations to function as a standard processing module within the network, as shown in [Figure 3b](#). Compared to a conventional convolutional block of matching depth and width, our module contains fewer parameters, which can result in a minor reduction in total capacity for knowledge representation.

Consequently, we apply the module in two distinct configurations: first, as an external interpretability unit that preserves the base performance of the original network; second, as a core structural unit, which may introduce a slight performance trade-off in highly complex tasks. To mitigate this limitation and expand the learning capacity of our architecture, the structural width of the BTSC module can be scaled upward.

Understanding decision-making based on decision-path

Ideally, according to [Eqs.3 and 4](#), the activation response of each child node within the BTSC module should polarize toward two distinct endpoints (i.e., either highly active or completely suppressed). In practical implementations, however, not all operations or nodes satisfy these ideal convergence states. The activation levels of certain nodes may fail to reach definitive high or low boundaries, causing the BTSC module to formulate multiple concurrent decision pathways.

For any given input, we assume that the primary decision-path—formed by the sequence of nodes exhibiting the highest activation levels within the BTSC module (termed the main decision-path)—maps the actual classification process. The underlying decision logic is executed along this highest-activation route. It is important to note that the main decision-path does not always align with the expected decision-path. When the model yields a correct prediction, the main decision-path typically matches the expected decision-path; during misclassifications, these paths tend to diverge.

From a practical standpoint, if the activation level of the primary nodes along the main decision-path is not significantly higher than that of competing nodes, it indicates that the processed information does not fit cleanly within the expected decision-path. Under these conditions, the active route becomes highly sensitive to structural shifts or minor input perturbations, which can lower the reliability of the resulting prediction. To quantify this behavior, we establish that closer alignment between the main decision-path and the expected decision-path indicates a more reliable model decision. This study introduces the evaluation metric R (where $R > 1$) to calculate decision credibility. This metric computes the maximum ratio between the activation intensities of two child nodes ($f_1(x)$ and $f_2(x)$) sharing a common parent within the BTSC module, as defined in [Eq. 5](#):

$$R = \max\left(\frac{\max(f_1(x), f_2(x))}{\min(f_1(x), f_2(x))}\right), f_1(x) \& f_2(x) > \epsilon \quad (5)$$

A larger value of R denotes tighter alignment with the expected decision-path, indicating a more credible classification outcome. The symbol ϵ represents a baseline activation threshold, ensuring that both node responses must exceed this minimal value to be considered.

Note that:

- **Expected decision-path:** The information flow route intentionally designed during the construction of the BTSC module, representing the targeted knowledge structure to be acquired.

- **Main decision-path:** The actual information flow route observed during inference, tracking the path through which the maximum volume of information propagates during a decision event.

Results and Discussion

Implementation

Dataset

Optimization experiments were conducted utilizing the training dataset, while performance evaluations were executed on the test dataset. The primary objective of this investigation is to demonstrate the interpretability of the introduced framework rather than optimizing for state-of-the-art accuracy metrics. For image classification applications, any standard benchmark dataset is suitable. In this research, we selected the CIFAR-10 dataset—which contains 50,000 training images and 10,000 test images distributed across 10 categories (comprising class0-aeroplane, class1-automobile, class3-cat, class4-deer, class5-dog, class6-frog, class7-horse, class8-ship, class9-truck)—and the 15-Scene dataset, which encompasses 3133 training images and 1352 test images across 15 distinct categories.

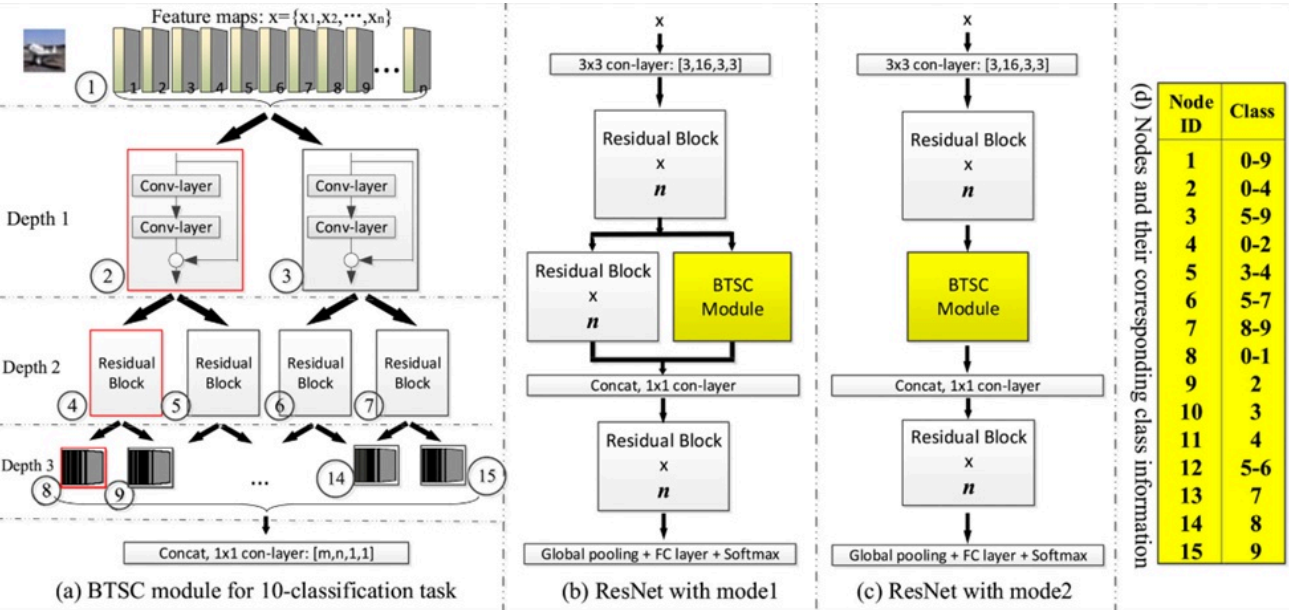


Figure 4. Our proposed applied on ResNet (The same way can also be used on DenseNet model.).

Neural network model and training configuration

As illustrated in **Figure 4a**, within the BTSC module, every individual child node incorporates a single 3 \times 3 residual block (structural specifications are detailed in **Table 1**), with each node designated to encode specific class-related information (**Figure 4d**). **Figure 4d** demonstrates one methodology for allocating targeted class information encoding to specific nodes within the BTSC structure. For instance, an input image depicting an airplane maps to class 0 in the CIFAR-10 dataset. Based on our node assignment logic, the second, fourth, and eighth nodes are responsible for processing the information encoding of this particular sample, while the remaining nodes remain inactive (see the red-bordered nodes in **Figure 4a**).

Figures 4b and 4c depict two standard integration configurations for embedding a BTSC component into a conventional ResNet architecture. For data preprocessing, we standardized the input samples by subtracting the mean pixel values and implemented data augmentation techniques [50] during the training phase. Weights were initialized in accordance with ref. [21], and the models were trained from scratch. Optimization was performed via an SGD optimizer with a mini-batch size of 128 and an initial learning rate of 0.1, which was reduced by a factor of 10 at 32,000 and 48,000 steps. The momentum parameter was set to 0.9, and the weight decay coefficient was 0.0001.

Table 1. Structural details of BTSC (Mode1: $k_0, k_1, k_2, k_3 = 4, 2, 1, 0.5$; Mode2: $k_0, k_1, k_2, k_3 = 4, 4, 4, 4$).

Depth 0	Depth 1	Depth 2	Depth 3
$\begin{bmatrix} 3 \times 3, 16 * k_0 \\ 3 \times 3, 16 * k_0 \end{bmatrix} \times 1$	$\begin{bmatrix} 3 \times 3, 16 * k_1 \\ 3 \times 3, 16 * k_1 \end{bmatrix} \times 1$	$\begin{bmatrix} 3 \times 3, 16 * k_2 \\ 3 \times 3, 16 * k_2 \end{bmatrix} \times 1$	$\begin{bmatrix} 3 \times 3, 16 * k_3 \\ 3 \times 3, 16 * k_3 \end{bmatrix} \times 1$
		$\begin{bmatrix} 3 \times 3, 16 * k_2 \\ 3 \times 3, 16 * k_2 \end{bmatrix} \times 1$	$\begin{bmatrix} 3 \times 3, 16 * k_3 \\ 3 \times 3, 16 * k_3 \end{bmatrix} \times 1$
	$\begin{bmatrix} 3 \times 3, 16 * k_1 \\ 3 \times 3, 16 * k_1 \end{bmatrix} \times 1$	$\begin{bmatrix} 3 \times 3, 16 * k_2 \\ 3 \times 3, 16 * k_2 \end{bmatrix} \times 1$	$\begin{bmatrix} 3 \times 3, 16 * k_3 \\ 3 \times 3, 16 * k_3 \end{bmatrix} \times 1$
		$\begin{bmatrix} 3 \times 3, 16 * k_2 \\ 3 \times 3, 16 * k_2 \end{bmatrix} \times 1$	$\begin{bmatrix} 3 \times 3, 16 * k_3 \\ 3 \times 3, 16 * k_3 \end{bmatrix} \times 1$
			$\begin{bmatrix} 3 \times 3, 16 * k_3 \\ 3 \times 3, 16 * k_3 \end{bmatrix} \times 1$
			$\begin{bmatrix} 3 \times 3, 16 * k_3 \\ 3 \times 3, 16 * k_3 \end{bmatrix} \times 1$
			$\begin{bmatrix} 3 \times 3, 16 * k_3 \\ 3 \times 3, 16 * k_3 \end{bmatrix} \times 1$
			$\begin{bmatrix} 3 \times 3, 16 * k_3 \\ 3 \times 3, 16 * k_3 \end{bmatrix} \times 1$
			$\begin{bmatrix} 3 \times 3, 16 * k_3 \\ 3 \times 3, 16 * k_3 \end{bmatrix} \times 1$

Main objective accuracy

Figure 5 displays the training and validation accuracy trajectories for the ResNet and DenseNet models augmented with the BTSC module during the optimization process. These empirical tests were executed across both the CIFAR-10 and 15-Scene datasets. The evaluations utilized identical model structures and BTSC integration approaches, differing exclusively in the configuration of the final classification layer. The learning capacity and generalization performance of the neural networks integrated with the BTSC module are nearly identical to those of the baseline architectures.

The evaluation outcomes for both the modified and baseline networks are presented in **Table 2**, confirming that the new models yield test dataset accuracies comparable to the original architectures. Because our objective is to validate the interpretability of DCNN decision mechanisms rather than maximizing raw performance benchmarks, we can preserve the accuracy of the modified models by tuning the depth and width parameters of the BTSC module. Consequently, our approach introduces a marginal expansion in the overall parameter footprint of the model. The experimental findings validate that embedding the BTSC module preserves the core performance of the base model.

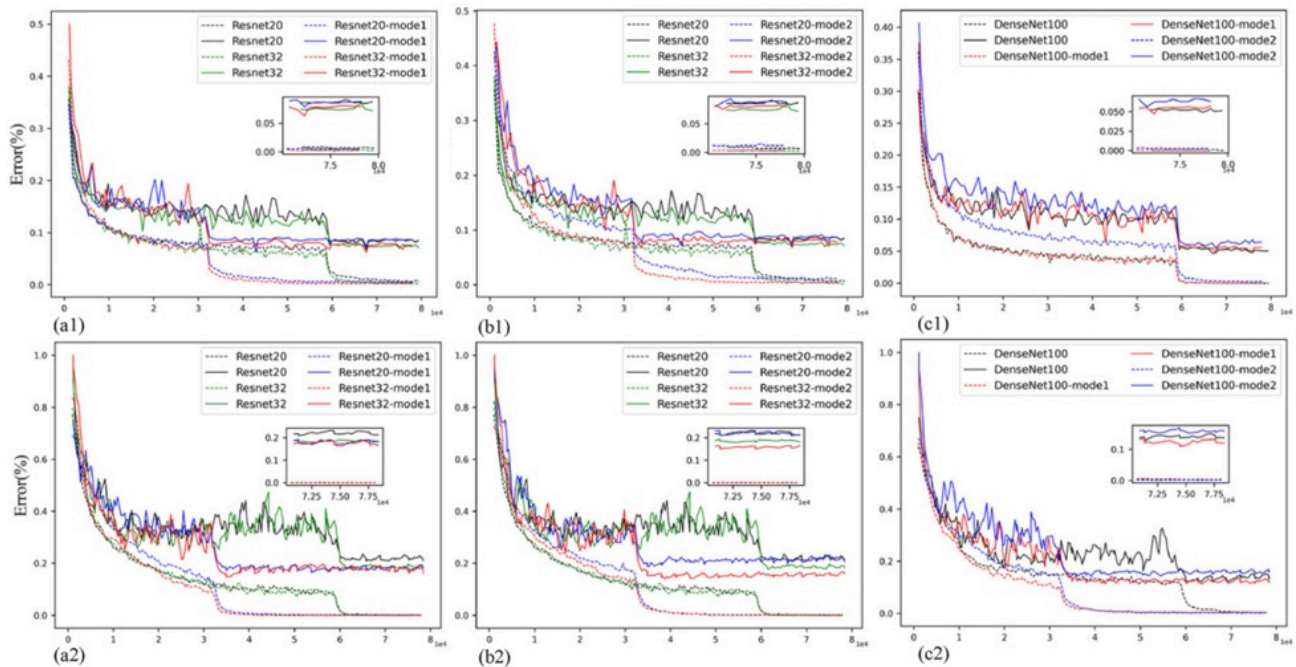


Figure 5. Training and testing on the datasets that Cifar10 (top row) and 15-Scene (bottom row). The dotted lines denote the training error, and the solid lines denote the testing error (a1)-(a2) Mode-1 of our method applied on ResNet20 and Resnet32 models (b1)-(b2) Mode-2 of our method applied on ResNet20 and ResNet32 models (c1)-(c2) Mode-1 and Mode-2 of our method applied on DenseNet100 models.

Table 2. Errors on the test dataset of Cifar10 and 15-Scene.

Dataset	Architecture	Proposed Method (Mode-2)		Proposed Method (Mode-1)		Baseline Model	
		Top-1 Error (%)	Parameters (M)	Top-1 Error (%)	Parameters (M)	Top-1 Error (%)	Parameters (M)
CIFAR-10	ResNet-20	8.49	0.66	8.38	0.40	8.75	0.27
	ResNet-32	7.83	0.79	7.27	0.60	7.51	0.46
	DenseNet-100	5.74	1.24	5.39	0.92	5.92	0.80
15-Scene	ResNet-20	16.79	0.66	13.54	0.43	16.49	0.27
	ResNet-32	15.98	0.79	13.46	0.60	14.79	0.46
	DenseNet-100	15.68	1.24	10.06	0.92	8.95	0.80

Note: Bold values can represent the optimal result among the three models.

Interpretability analysis

Interpretability of decision-making in BTSC module

Correct decisions: To analyze a correct classification produced by a trained DCNN model embedded with the BTSC module (specifically, ResNet-BTSC and DenseNet-BTSC), we visualized the complete set of feature maps within the terminal layer of the BTSC module (Figure 6). According to the preassigned learning constraints of the BTSC module, each feature map is designed to participate in encoding the decision criteria for specific classes. When an input sample is accurately classified by the DCNN framework, the internal information coding layout within the BTSC module should align with the predefined assignment.

As shown in Figures 6a–6d, the corresponding pre-allocated nodes (Figure 4d) display significant stimulation, achieving activation intensities that are distinctly higher than those of competing nodes within the BTSC module. This confirms that the main decision-path predominantly governs the encoding of this decision process as anticipated, executing its intended operational logic.

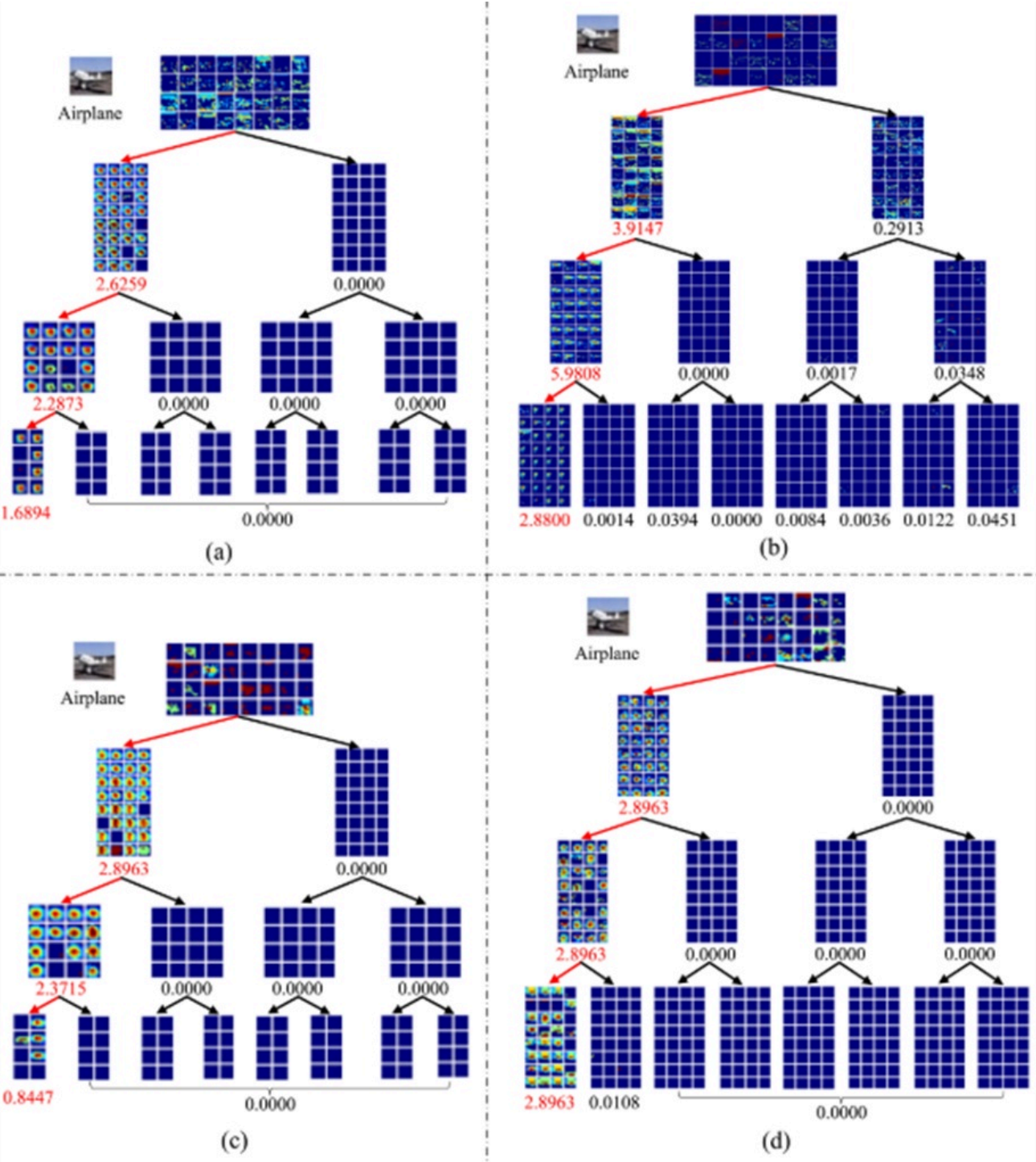


Figure 6. Visualisation of the activation level of nodes in the BTSC module embedded in ResNet20 and DenseNet100 for a correct decision-making; Red nodes indicate the main decision-path. (a) and (b) BTSC module embedded in ResNet20 model by mode-1 and mode-2, respectively; (c) and (d) BTSC module embedded in DenseNet100 model by mode-1 and mode-2, respectively.

Incorrect decisions: To examine an incorrect classification event, we similarly visualized all feature maps in the final layer of the BTSC module embedded within the ResNet20 and DenseNet100 architectures (Figure 7). Expectedly, during

a misclassification, the information coding profile within the BTSC module tends to diverge from the preassigned layout (the expected decision-path).

As illustrated in **Figure 7**, when a sample is incorrectly classified, the activation responses of the pre-allocated target nodes within the BTSC module fall below those of alternative nodes (**Figures 7a, 7c, and 7d**); that is, the main decision-path undergoes a severe deviation from the expected decision-path. Alternatively, another failure mode involves the hyper-activation of multiple nodes outside the main decision-path, which disrupts the model's classification sequence. These observations indicate that erroneous predictions in the neural network stem from a disordered state of information coding within the BTSC module.

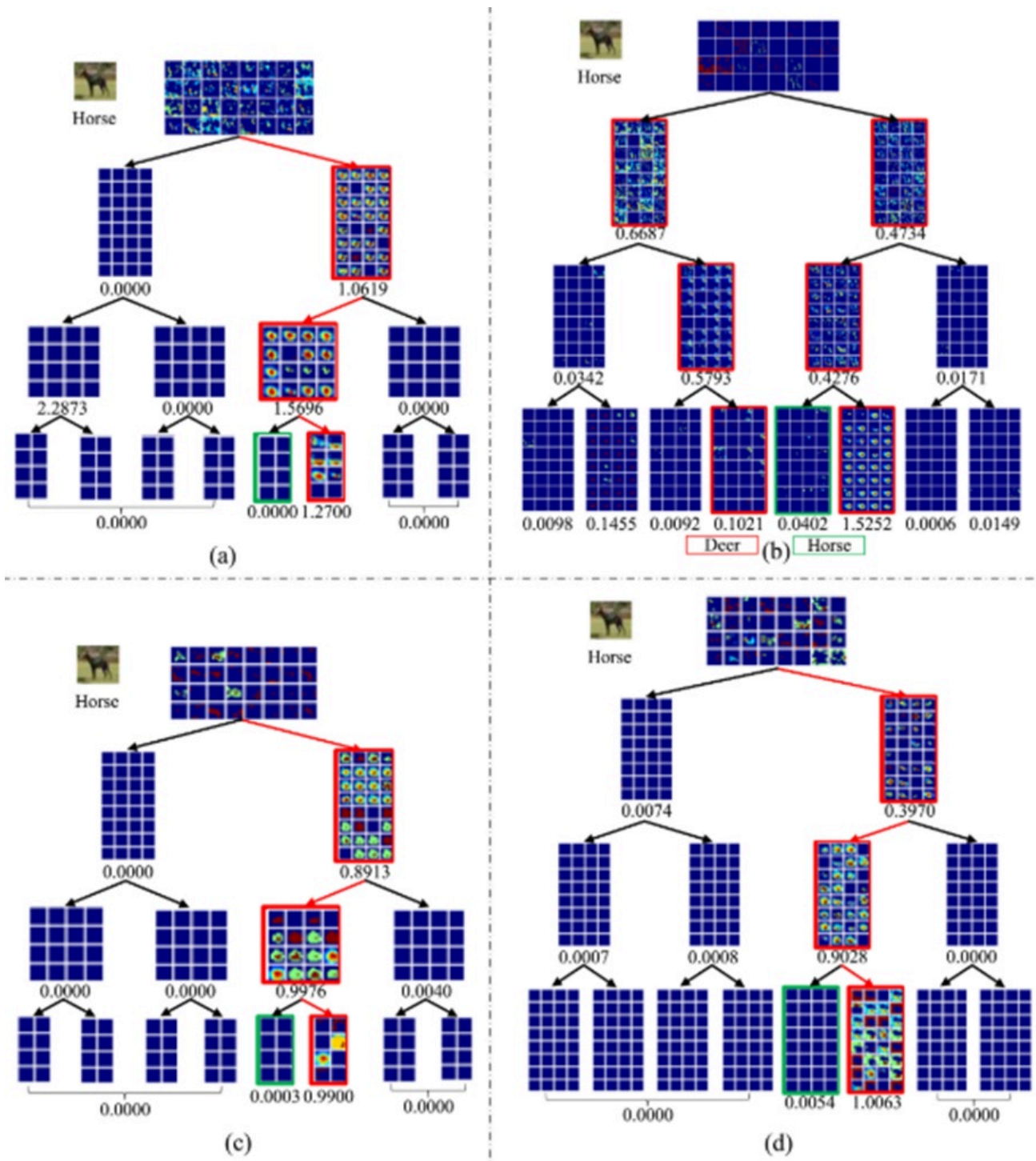


Figure 7. Visualisation of the activation level of nodes in the BTSC module embedded in ResNet20 and DenseNet100 for an incorrect decision-making; Red nodes indicate the main decision-path, green nodes indicate the expected decision-path. (a) and (b) BTSC module embedded in the ResNet20 model by Modes 1 and 2, respectively; (c) and (d) BTSC module embedded in the DenseNet100 model by Modes 1 and 2, respectively.

Statistical analysis of consistency: Based on the structural design of the BTSC module, the internal data flow during inference is intended to consistently propagate along the nodes exhibiting maximum activation levels (as exemplified in Figures 6 and 7); hence, each main decision-path is engineered to encode a singular classification category.

Concurrently, the activation levels of nodes situated outside this main decision-path should remain minimal and converge toward zero. In practice, alternative decision pathways driven by highly active nodes can also impact the final classification output of the model.

Statistical distributions mapping the alignment between the model's final prediction and the decision details encoded by the main decision-path are detailed in **Figure 8**. We derive the following insights from these metrics:

1. In general, a substantial majority of the decision-making processes—exceeding 70% on the 15-Scene dataset and reaching above 90% on CIFAR-10—adhere directly to our pre-established learning objective. This indicates that the core decision logic is successfully encoded by the main decision-path formed by the highest-activating nodes.
2. Occasionally, several concurrent nodes at the same depth of the BTSC module exhibit minor differences in high activation levels, causing information to propagate across multiple decision pathways; in these instances, the final prediction can still align with alternative decision pathways. Over 90% (and up to 95%) of the decision-making pipeline is governed by the units holding the top-2 (or top-3) highest activation intensities (**Figures 8(b*)** and **(c*)**).

In **Figure 8(a*)**, the alignment consistency of the ResNet model using the second mode falls below that of the alternative configurations, an effect that may be linked to the specific structural position where the module is embedded within the ResNet architecture. In **Figure 8(a2)**, using an identical model and BTSC integration strategy, the alignment consistency evaluated on the 15-Scene dataset is lower than that on CIFAR-10 at the Top-1 metric, yet displays relative similarity across the Top-2 and Top-3 thresholds. This discrepancy may stem from reduced compatibility between this specific integration approach and the characteristics of the 15-Scene dataset.

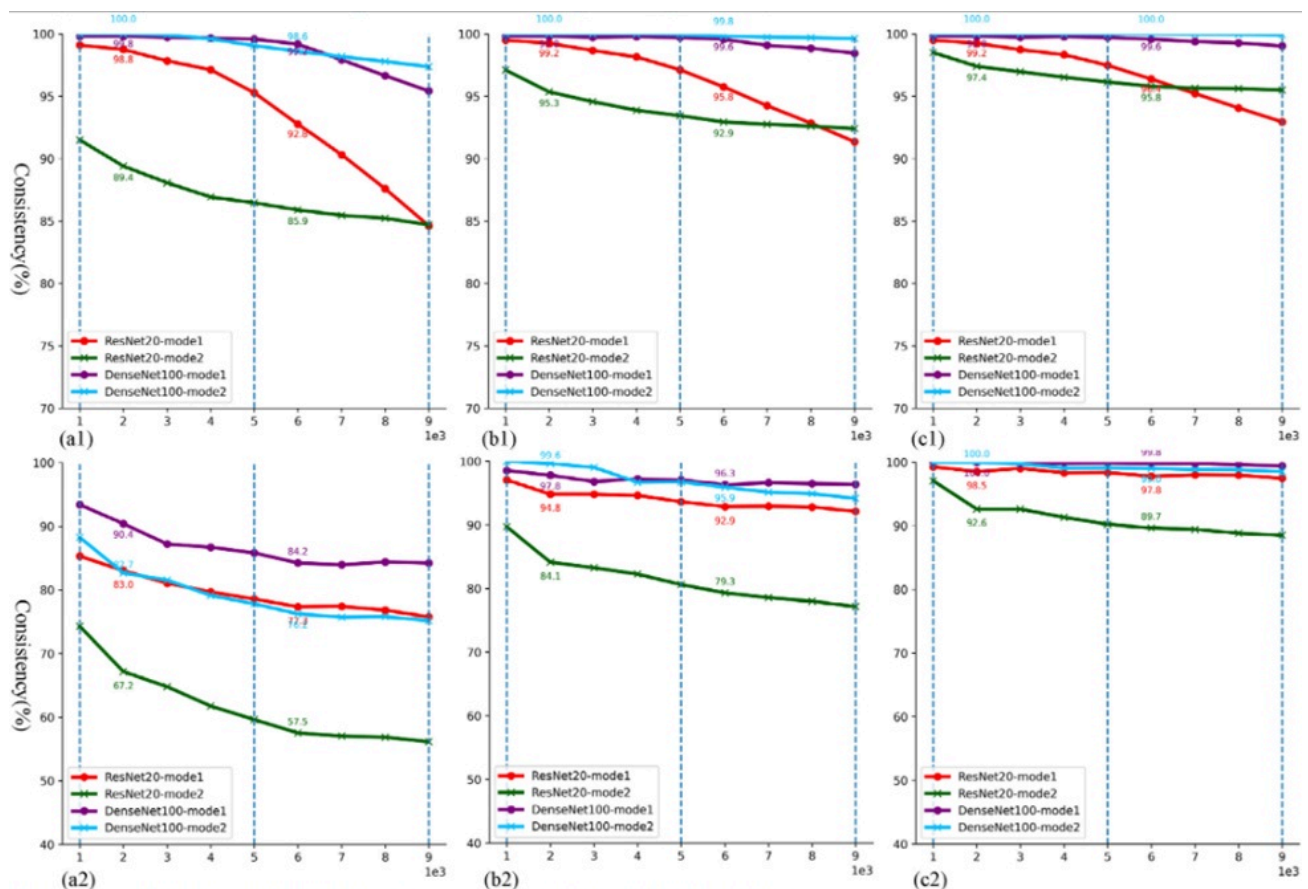


Figure 8. Consistency statistics for information coding of nodes with Top-k largest activation level in the last depth of the BTSC module and model's final prediction, this is experiments on CIFA-10 (top row) and 15-Scene (bottom row) datasets.

The horizontal axis represents the ranking of credibility measure (R) from large to small, the longitudinal axis indicates the percentage of the consistency statistics (a1)-(a2) as (a*) (b1)-(b2) as (b*), and (c1)-(c2) as (c*) represents Top-1, Top-2, and Top-3, respectively.

Summary: We have engineered a specialized neural component (the BTSC module) to construct a predictable and interpretable information coding framework, which is seamlessly integrated into standard DCNN structures. Empirical results confirm that most nodes within the BTSC architecture successfully learn their preassigned (expected) information coding functions, meaning that each distinct main decision-path explicitly maps out a specific classification outcome. This structured propagation route within the hierarchical module clearly delineates the decision logic executed by the model for any given input sample. For instances where the decision-making process fails to strictly follow our pre-allocated routes (the expected decision-path), the system becomes highly prone to making erroneous predictions. Based on these foundations, this work focuses on providing a clear explanation for the internal decision mechanisms of DCNN architectures.

Credibility of decision-making

To evaluate this behavior, **Figure 9** presents two distinct cases evaluated via ResNet20 under Mode 1, representing exceptionally high and low magnitudes of R, respectively. In the scenario where the R value is minimal (depicted in **Figure 9a**), the primary computational pathway demonstrates profound instability. Even when subjected to uniform noise thresholds, the core decision-path varies unpredictably across separate evaluations, indicating that such outputs are fundamentally unreliable. In contrast, when the R value is elevated (**Figure 9b**), applying the identical noise configuration yields only marginal variations in the activation profiles of auxiliary nodes. Consequently, the core execution path persists without deviation, keeping the final output structurally robust.

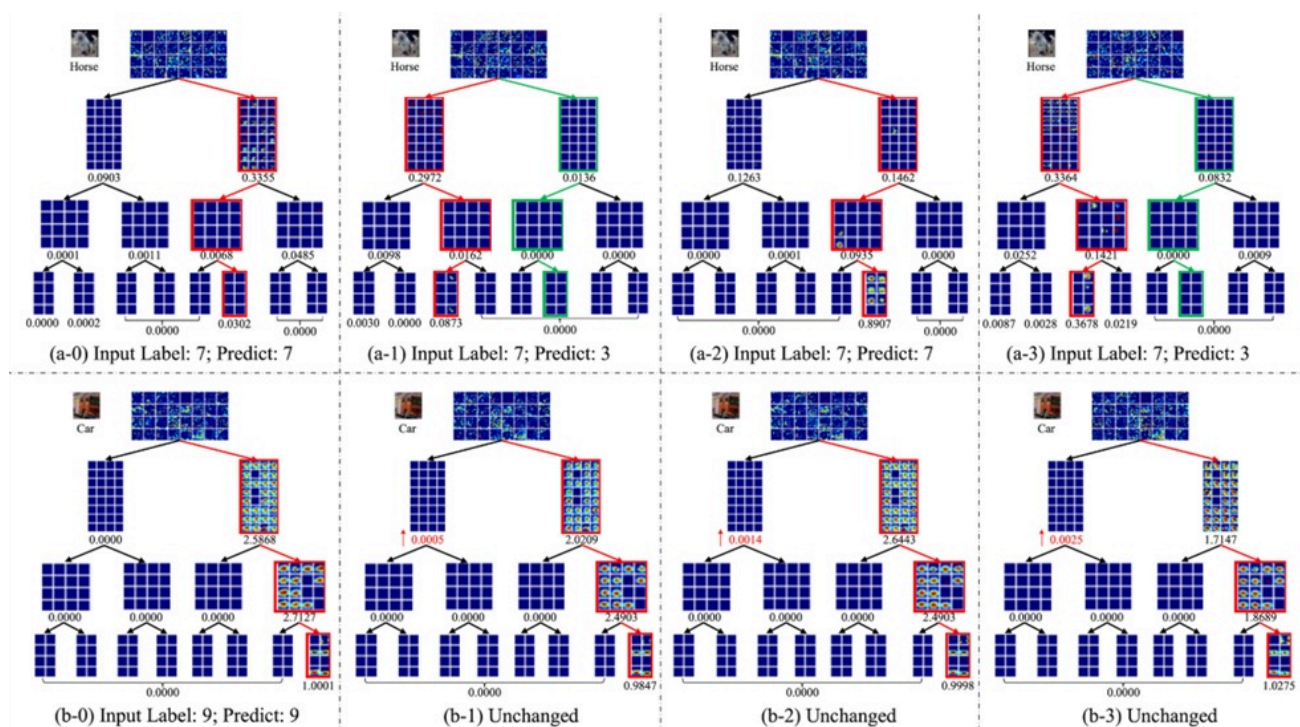


Figure 9. Robustness analysis for incredible and credible decisions by the BTSC module; Red nodes indicate the main decision-path, green nodes original one (a-0)~(a-3) indicate the activation level of all the nodes in the BTSC for the incredible decision and its actual decision-path under adding the same level of salt and pepper noise four times (b-0)~(b-3) for the credible decision under the same noise with (a-0)~(a-3).

To validate these observations across a broader distribution, varying intensities of input perturbations were applied to a cohort of the top-1000 highly credible and top-1000 poorly credible classifications, with results compiled in **Figure 10**. Low-credibility outputs demonstrate a much higher susceptibility to input variations, making them significantly more vulnerable to classification shifts than their high-credibility counterparts. This phenomenon is especially pronounced under minimal noise conditions and remains uniform across both the Cifa10 and 15-Scene benchmarks. Finally, **Figure 11** illustrates representative source images associated with these contrasting decision states. Visually, the data samples tied to high-credibility outcomes possess much clearer features, making them more amenable to automated recognition. Conversely, within the low-credibility subset, multiple instances contain erroneous ground-truth annotations, while the highly reliable examples tend to exhibit a certain level of structural pattern collapse.

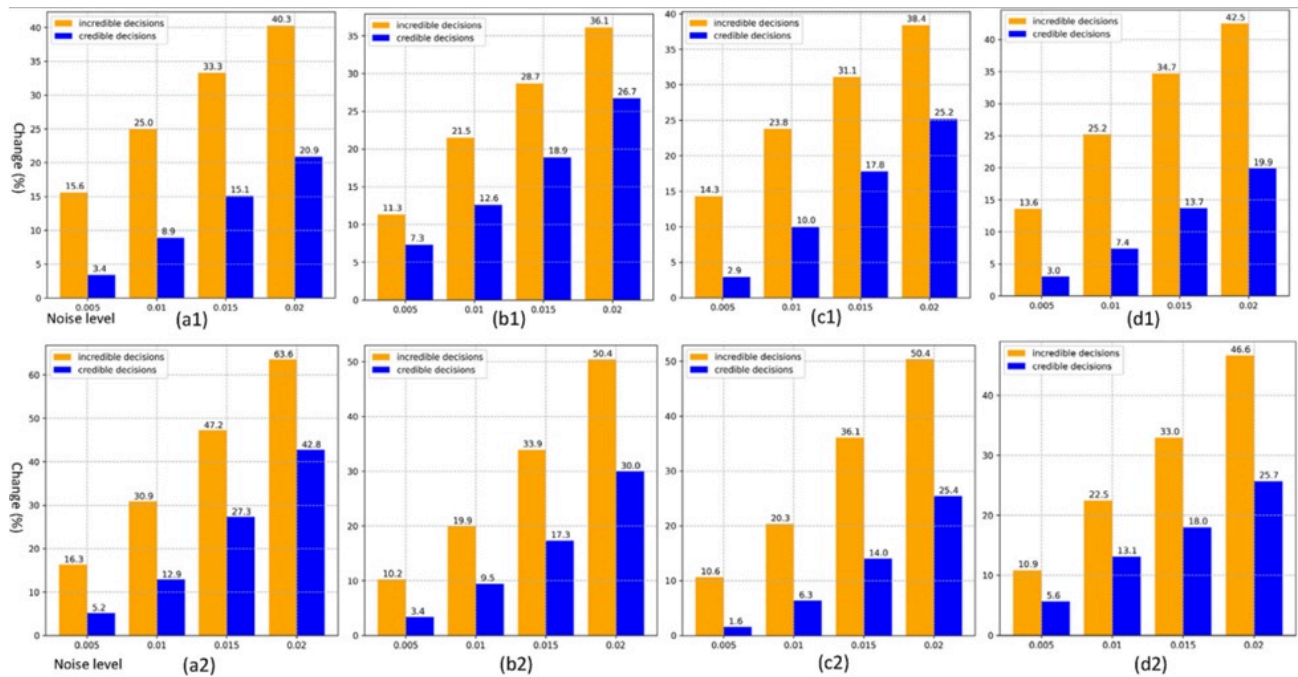


Figure 10. Robustness analysis of the credible and incredible decisions by adding different levels of noise on CIFA10 (top row) and 15-Scene (bottom row) (a1)-(a2) is Mode 1 and (b1)-(b2) is Mode two on ResNet (c1)-(c2) is Mode 1 and (d1)-(d2) is Mode two on DenseNet. The horizontal axis indicates the level of salt and pepper noise, and the longitudinal axis indicates the percentage of decisions that are changed by noise.



Figure 11. Input samples in the red and green boxes corresponding to incredible and credible decisions respectively. For the incredible decisions, even some examples that signed with small red boxes were mislabelled.

Conclusion

Despite the monumental milestones achieved by DCNN frameworks within computer vision, further theoretical progress is essential to fully unpack their internal operations. This investigation targets two fundamental vulnerabilities persistent in the field: 1) the obscure and opaque nature of DCNN reasoning pathways, and 2) the inability to quantitatively evaluate output reliability. Drawing principles from biological task-specific neural coding mechanisms, we developed an approach to isolate and learn decision-centric feature spaces. By leveraging the explicit logic and autonomous inheritance dynamics characteristic of hierarchical decision trees, we engineered the BTSC architecture and integrated it within standard DCNN topologies. Additionally, an information decoupling optimization loss was formulated to guide specific BTSC nodes toward extracting the intended classification representations. Practical validation demonstrates that this architecture successfully demystifies DCNN classification mechanics while introducing a viable metric for calculating output confidence.

Reflecting on how decision trees operate systematically, we anticipate that the BTSC structure will significantly enhance the transparency of deep models. This exploration utilized information decoupling via the BTSC framework to establish a more transparent DCNN classification paradigm. Moving forward, research should prioritize the formulation of more refined objective functions to help the BTSC isolate crisper knowledge topologies and distinct semantic spaces. Expanding the deployment of this architecture is particularly critical for specialized domains that mandate rigorous auditing of automated reasoning mechanisms.

Acknowledgements

None

Conflict of interest

None

Financial support

This work was sponsored by Nature Science Foundation of Chongqing (CSTB2023NSCQ-MSX0033), the State Key Program of National Nature Science Foundation of China (61936001), and the Special Foundation of ChongQing Postdoctoral Science (2021XM3113).

Ethics statement

None

Received: 26 Jul 2022

Revised: 18 Sep 2022

Accepted: 20 Oct 2022

Published online: 15 March 2023

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

References

- Chuang YY, Wu JM. Codebook design of all index modulation with deep reinforcement learning. In: 2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall). IEEE; 2022. p. 1-5.
- Tyagi AK, Mohapatra C, Das P, Makharia G, Mehra L, AP P, et al. DeGPR: deep guided posterior regularization for multi-class cell detection and counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2023. p. 23913-23.
- Rokon MSJ, Morol MK, Hasan IB, Saif AM, Khan RH. Food recipe recommendation based on ingredients detection using deep learning. In: Proceedings of the 2nd International Conference on Computing Advancements. 2022. p. 191-8.
<https://doi.org/10.1145/3542954.3542983>.
- Zhan ZH, Li JY, Zhang J. Evolutionary deep learning: a survey. *Neurocomputing*. 2022;483:42-58.
<https://doi.org/10.1016/j.neucom.2022.01.099>.
- Lei E, Hassani H, Bidokhti SS. Neural estimation of the rate-distortion function with applications to operational source coding. *IEEE J Sel Areas Inf Theory*. 2023.
<https://doi.org/10.1109/JSAIT.2023.3273467>.

Dai D, Yu L, Wei H. Parameters sharing in residual neural networks. *Neural Process Lett*. 2020;51(2):1393-410.
<https://doi.org/10.1007/s11063-019-10143-4>.

Dai D, Tang C, Wang G, Xia S. Building partially understandable convolutional neural networks by differentiating class-related neural nodes. *Neurocomputing*. 2021;452:169-81.
<https://doi.org/10.1016/j.neucom.2021.04.003>.

Wei H, Dai D, Bu Y. A plausible neural circuit for decision making and its formation based on reinforcement learning. *Cogn Neurodyn*. 2017;11(3):259-81.
<https://doi.org/10.1007/s11571-017-9426-4>.

Yosinski J, Clune J, Nguyen A, Fuchs T, Lipson H. Understanding neural networks through deep visualization. *arXiv [Preprint]*. 2015. arXiv:1506.06579.
<https://doi.org/10.48550/arXiv.1506.06579>.

Zhang Q, Cheng X, Chen Y, Rao Z. Quantifying the knowledge in a DNN to explain knowledge distillation for classification. *IEEE Trans Pattern Anal Mach Intell*. 2023;45(4):5099-113.
<https://doi.org/10.1109/TPAMI.2022.3200344>.

Wang H, Wang Z, Du M, Yang F, Zhang Z, Ding S, et al. Score-CAM: score-weighted visual explanations for convolutional neural networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*. 2020. p. 111-9.
<https://doi.org/10.1109/CVPRW50498.2020.00020>.

Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017. p. 618-26.
<https://doi.org/10.1109/ICCV.2017.74>.

Chatfield K, Simonyan K, Vedaldi A, Zisserman A. Return of the devil in the details: delving deep into convolutional nets. In: *Proceedings of the British Machine Vision Conference (BMVC)*. 2014. p. 6.1-6.12.

Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences. In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. PMLR; 2017. p. 3145-53.

Powers W, Shi L, Liebovitch LS. Tools and visualizations for exploring classification landscapes. In: *2023 57th Annual Conference on Information Sciences and Systems (CISS)*. IEEE; 2023. p. 1-6.
<https://doi.org/10.1109/CISS56502.2023.10089673>.

Yang J, Wickramasinghe U, Ni B, Fua P. ImplicitAtlas: learning deformable shape templates in medical imaging. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022. p. 15861-71.

Goodfellow I, Bengio Y, Courville A. *Deep learning*. Cambridge (MA): MIT Press; 2016.

Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: *Computer Vision – ECCV 2014*. Cham: Springer; 2014. p. 818-33.

Wei Y, Wang Z, Qiao X. Rice disease recognition and feature visualization using a convolutional neural network. In: *Proceedings of the 2022 11th International Conference on Computing and Pattern Recognition*. 2022. p. 20-5.

Ye Z, Gao Z, Cui X, Wang Y, Shan N. DuFeNet: improve the accuracy and increase shape bias of neural network models. *Signal Image Video Process*. 2022;16(5):1153-60.
<https://doi.org/10.1007/s11760-021-02065-3>.

Sun J, Darbehani F, Zaidi M, Wang B. SAUNet: shape attentive U-Net for interpretable medical image segmentation. In: Medical Image Computing and Computer Assisted Intervention – MICCAI 2020. Cham: Springer; 2020. p. 797-806.

Zhang Y, Zhang Y, Xu Q, Zhang R. Learning robust shape-based features for domain generalization. *IEEE Access*. 2020;8:63748-56.
<https://doi.org/10.1109/ACCESS.2020.2984279>.

Portelance E, Frank MC, Jurafsky D, Sordoni A, Laroché R. The emergence of the shape bias results from communicative efficiency. In: Proceedings of the 25th Conference on Computational Natural Language Learning (CoNLL). 2021. p. 607-23.

Narayanan M, Rajendran V, Kimia B. Shape-biased domain generalization via shock graph embeddings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2021. p. 1315-25.

Ding X, Zhang X, Zhou Y, Han J, Ding G, Sun J. Scaling up your kernels to 31×31: revisiting large kernel design in CNNs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022. p. 11963-75.

Sinha H, Kovashka A. Towards shape-regularized learning for mitigating texture bias in CNNs. In: Proceedings of the 2023 ACM International Conference on Multimedia Retrieval (ICMR). 2023. p. 325-34.

Jarvers C, Neumann H. Shape-selective processing in deep networks: integrating the evidence on perceptual integration. *Front Comput Sci*. 2023;5:1113609.
<https://doi.org/10.3389/fcomp.2023.1113609>.

Li Z, Imai J, Kaneko M. Facial-component-based bag of words and PHOG descriptor for facial expression recognition. *IEEE Trans Syst Man Cybern B Cybern*. 2009. p. 1353-8.

Mu J, De Mello S, Yu Z, Vasconcelos N, Wang X, Kautz J, et al. CoordGAN: self-supervised dense correspondences emerge from GANs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022. p. 10011-20.

Kalischek N, Caye Daudt R, Peters T, Furrer R, Wegner JD, Schindler K. BiasBed: rigorous texture bias evaluation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2023. p. 22221-30.

Geirhos R, Rubisch P, Michaelis C, Bethge M, Wichmann FA, Brendel W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations (ICLR). 2019.

Geirhos R, Medina Temme CR, Rauber J, Schütt HH, Bethge M, Wichmann FA. Generalisation in humans and deep neural networks. In: Advances in Neural Information Processing Systems. 2018;31.

Ribeiro MT, 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 (KDD). 2016. p. 1135-44.
<https://doi.org/10.1145/2939672.2939778>.

Kim S, Nam J, Ko BC. ViT-NeT: interpretable Vision Transformers with neural tree decoder. In: Proceedings of the 39th International Conference on Machine Learning (ICML). PMLR; 2022. p. 11162-72.

Feng J, Zhou ZH. AutoEncoder by Forest. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI). 2018. p. 2967-73.
<https://doi.org/10.1609/aaai.v32i1.11732>.

Feng J, Yu Y, Zhou ZH. Multi-layered gradient boosting decision trees. In: Advances in Neural Information Processing Systems. 2018;31:3555-65.

Zhou ZH, Feng J. Deep forest. *Natl Sci Rev*. 2019;6(1):74-86.

Ding M, Zhou C, Chen Q, Yang H, Tang J. Cognitive graph for multi-hop reading comprehension at scale. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL). 2019. p. 2694-703.

Hehn TM, Kooij JFP, Hamprecht FA. End-to-end learning of decision trees and forests. *Int J Comput Vis*. 2020;128(4):997-1011.
<https://doi.org/10.1007/s11263-019-01237-6>.

Nauta M, van Bree R, Seifert C. Neural prototype trees for interpretable fine-grained image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021. p. 14933-43.
<https://doi.org/10.1109/CVPR46437.2021.01469>.

Wan A, Dunlap L, Ho D, Yin J, Lee S, Petryk S, et al. NBDT: neural-backed decision tree. In: International Conference on Learning Representations (ICLR). 2021.

Upchurch P, Gardner J, Pleiss G, Pless R, Snavely N, Bala K, et al. Deep feature interpolation for image content changes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. p. 7064-73.

Cai J, Fan J, Guo W, Wang S, Zhang Y, Zhang Z. Efficient deep embedded subspace clustering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022. p. 1-10.

Connor M, Rozell C. Representing closed transformation paths in encoded network latent space. *Proc AAAI Conf Artif Intell*. 2020;34(4):3666-75.

Kim B, Wattenberg M, Gilmer J, Cai C, Wexler J, Viegas F, et al. Interpretability beyond feature attribution: quantitative testing with Concept Activation Vectors (TCAV). In: Proceedings of the 35th International Conference on Machine Learning (ICML). PMLR; 2018. p. 2668-77.

Ghorbani A, Wexler J, Zou J, Kim B. Towards automatic concept-based explanations. In: Advances in Neural Information Processing Systems. 2019;32:9277-86.

Ding Z, Xu Y, Xu W, Parmar G, Yang Y, Welling M, et al. Guided variational autoencoder for disentanglement learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020. p. 7920-9.

Guan S, Tai Y, Ni B, Zhu F, Huang F, Yang X. Collaborative learning for faster StyleGAN embedding. *arXiv [Preprint]*. 2020. arXiv:2007.01758.

Chen Z, Bei Y, Rudin C. Concept whitening for interpretable image recognition. *Nat Mach Intell*. 2020;2(12):772-82.
<https://doi.org/10.1038/s42256-020-00265-z>.

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 (CVPR). 2016. p. 770-8.
<https://doi.org/10.1109/CVPR.2016.90>.