Proceedings of CONF-SPML 2026 Symposium: The 2nd Neural Computing and Applications Workshop 2025
DOI: 10.54254/2755-2721/2026.TJ28969

Through Scatter Plots to Probe Layer-wise Classification in CNNs: An Explainable Visualization Scheme

Zhekai Yu

Institute of Software and Artificial Intelligence, Guangzhou University of Software, Guangzhou, China
Kaiyu476@gmail.com

Abstract. Although convolutional neural networks (CNNs) are very good at classifying images, they are still challenging to understand at the intermediate representation level. This paper proposes a pragmatic, layer-wise scatter-plot scheme that turns high-dimensional convolutional features into readable two-dimensional maps, enabling analysts to judge class separability and boundary errors across layers. We train a compact three-block CNN on MNIST and, for each convolutional block, extract flattened activations, reduce their dimensionality by PCA (100D) → t-SNE (2D), and render paired scatter plots that show correctly classified samples and overlay misclassifications as red crosses. To move beyond anecdotal visuals, this study attaches Silhouette coefficients per layer and per-class error rates. The implementation (PyTorch + scikit-learn + Matplotlib) is lightweight and reproducible. Empirically, separability increases monotonically from Conv1 to Conv3 (Silhouette ≈ 0.21→0.33→ 0.42), while residual errors cluster near inter-class borders (e.g., 9 vs. 4/7), matching human intuition. By mitigating overplotting and coupling visuals with a quantitative index, the method provides actionable diagnostics for hyper-parameter and architecture tuning without modifying the network. The scheme complements feature-visualization and loss-landscape methods by revealing how the population of samples arranges across layers rather than what individual filters detect.

Keywords: explainable AI, t-SNE, class separability, scatter-plot visualization, CNN

1. Introduction

CNNs underpin modern visual recognition and are now standard in applications from document digitization to autonomous perception [1]. Yet a persistent engineering problem remains: where inside a trained network do classes become separable, and why does a small subset of samples remain hard to classify [2] ?

Low-dimensional embedding tools such as t-SNE and UMAP enable visual inspection of learned features by mapping high-dimensional vectors to two dimensions that humans can readily interpret [3]. However, naïvely projecting features and plotting all points leads to overplotting and purely qualitative judgments. Without a small set of quantitative indicators, it is hard to translate “looks better” into model changes that generalize [4].

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.TJ28969

This paper addresses that gap with a compact, layer-wise visualization scheme. After training a small CNN, we extract activations after each convolutional block, apply PCA→t-SNE, and render paired scatter plots per layer—one for correctly classified samples, and one that overlays misclassifications on a faint background of correct points [5]. This study then computes the Silhouette coefficient to summarize separability, providing a simple number to accompany the visual story. The resulting dashboard reveals how class structure evolves from shallow to deep layers and where boundary errors concentrate, thereby offering day-to-day actionable guidance for architecture and training decisions.

Our contribution complements existing interpretability work that visualizes individual filters or global loss geometry. This paper demonstrates the scheme on MNIST, but the method is model-agnostic and readily portable to other datasets and architectures.

2. Background and problem formulation

2.1. Embedding visualization, overplotting, and separability

A common pathway to making representations visible is to embed them in 2-D while preserving meaningful neighborhoods. t-SNE constructs pairwise similarities p_{ij} in high dimensions and q_{ij} in low dimensions, then minimizes their Kullback–Leibler divergence [6]:

$$\min_{Y} C(Y) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}, \quad q_{ij} = \frac{\left(1+\|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1+\|y_k - y_l\|^2\right)^{-1}} \tag{1}$$

UMAP follows a related manifold-learning approach that tends to preserve more global structure and scales well to large datasets [7]. In both cases, directly plotting all points can yield visual clutter. Points crowd near dense regions, classes overlap in the display even when distances differ subtly in high-D, and boundary cases vanish beneath large clusters. To turn visuals into evidence, we incorporate cluster validity indices—primarily the Silhouette—which for each sample ii is

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{2}$$

$a(i)$ =mean intra-class distance, $b(i)$ =nearest other-class mean distance.

Values near 1 indicate compact, well-separated clusters; values near 0 indicate overlap. Cluster tightness and separation can be further described by two complementing indices: Davies–Bouldin (lower is better) and Calinski–Harabasz (higher is better). We focus on Silhouette for its interpretability and ease of communication with non-specialists [8, 9].

2.2. Problem statement and research questions

From the vantage point of everyday CNN development, we face three practical questions:

RQ1: Depth vs. separability. How does class separability evolve from Conv1 to Conv3 in a compact CNN trained on MNIST?

RQ2: Boundary errors. Do misclassified samples cluster near inter-class boundaries in the embeddings, and how does their distribution change with depth?

RQ3: Actionable diagnostics. Can a lean visual-quantitative dashboard provide practical guidance for hyper-parameter and architecture tuning without modifying the model?

DOI: 10.54254/2755-2721/2026.TJ28969

3. Proposed method: layer-wise scatter-plot scheme

3.1. Model and training workflow

This study uses a compact CNN with three convolutional blocks (channels 8/16/32, 3×33\times3 kernels, ReLU), max-pooling after Conv1 and Conv2, and a linear 10-way classifier. The network is intentionally small to ensure that the visualization—not engineering horsepower—drives insights. Training uses Adam for 40 epochs with standard data loaders and logging. The codebase saves checkpoints, accuracy/loss curves, and all visual artifacts to facilitate reproducibility and side-by-side comparisons across runs [10].

3.2. Feature extraction and dimensionality reduction

For each image in the evaluation subset, we intercept flattened activations after Conv1, Conv2, and Conv3, forming matrices $X^{(l)} \in R^{N \times D_l}$. As a standard pre-processing step, we apply PCA to 100 dimensions to denoise and speed up downstream embedding; practical experience shows that PCA→t-SNE reduces crowding and improves convergence stability. We then run t-SNE (perplexity 30, fixed seed) to obtain 2-D coordinates $Y^{(l)}$. Although UMAP is a viable alternative, we keep t-SNE for comparability and because it often excels at preserving local neighborhoods for small to medium-sized datasets [11].

3.3. Scatter-plot design and overplotting mitigation

The visuals adopt three design decisions to combat clutter:

Paired views per layer. The top view shows correctly classified samples colored by ground-truth label, revealing cluster formation. The bottom view overlays misclassifications as red crosses on a faint alpha-blended background of correct samples. This makes boundary errors pop out without hiding global structure.

Context without occlusion. This study uses small markers and moderate transparency for the background points and reserves solid, high-contrast markers for errors.

3.4. Quantitative metrics

This study computes Silhouette per layer with Euclidean distances on the 2-D embeddings (Eq. 2). Where desired, we also compute per-class error rates and can add Calinski–Harabasz and Davies–Bouldin scores as secondary indicators [12].

3.5. Implementation notes and runtime

The end-to-end pipeline is fewer than two hundred lines of Python. On a modest GPU laptop, training for 40 epochs and producing all layer-wise embeddings and plots for a 1k-sample evaluation subset takes minutes. Deterministic seeds ensure run-to-run stability [13]. The method requires no modifications to the model or data pipeline; it simply “plugs into” trained checkpoints and produces comparable dashboards across experiments.

DOI: 10.54254/2755-2721/2026.TJ28969

4. Experimental validation

4.1. Dataset, protocol, and environment

We evaluate on MNIST, a 70k-image collection of 28×2828\times28 grayscale digits with ten classes. Although simple, MNIST remains instructive for analyzing representation geometry because digit confusions (e.g., 9 vs. 4/7) are well understood, and cluster boundaries correspond to recognizable stroke similarities. We use Python, PyTorch, scikit-learn, and Matplotlib with CPU/GPU auto-selection and fixed seeds. The evaluation subset comprises 1,000 test images to keep t-SNE computation swift while representative metrics. This study reports Silhouette per layer (Eq. 2) plus overall and per-class misclassification rates. The goal is not to chase the last percentage point of accuracy but to reveal how separability evolves and where errors concentrate.

4.2. Qualitative results: layer-wise structure

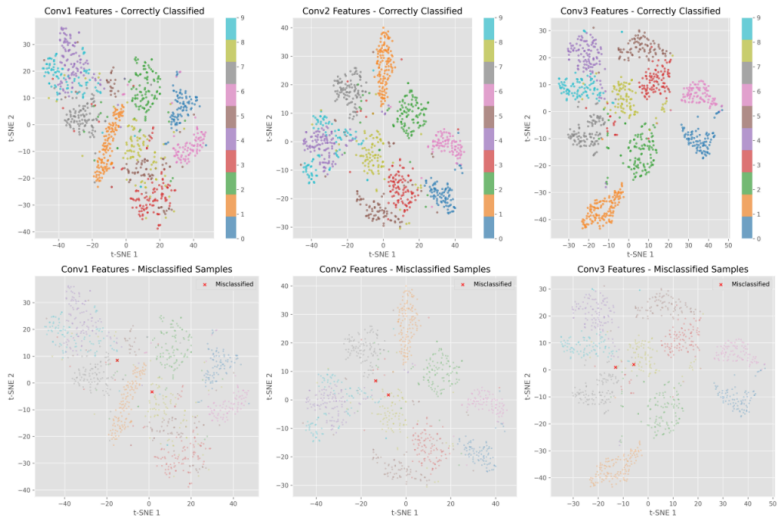

Figure 1. Layer-wise t-SNE embeddings of CNN features (top: correctly classified; bottom: misclassified)

Figure 1 illustrates the layer-wise t-SNE embeddings of CNN features. The top row ("Correctly Classified") shows the evolution of class separability across layers: Conv1 (Shallow Features) reveals broad, overlapping clusters, where some classes form loose neighborhoods, but the boundaries are still diffused. Conv2 (Middle Features) demonstrates more compact clusters with visible gaps between the classes, and the orientation of these clusters becomes more stable. Conv3 (Deep Features) exhibits well-separated, compact clusters with clear inter-cluster margins, indicating a highly discriminative representation.

In the bottom row ("Misclassified"), Figure 1 highlights the distribution of errors. Red crosses representing misclassified samples concentrate around the fringes of clusters, particularly at boundaries where visually similar digits meet (e.g., 9 vs. 4/7). This aligns with human intuition, showing that errors typically occur at decision boundaries. For example, the cluster for digit "8" remains compact and isolated, matching its near-zero error rate. The use of the paired-view design, where errors are overlaid on a faint background of correctly classified samples, makes these error-prone regions easily identifiable.

DOI: 10.54254/2755-2721/2026.TJ28969

4.3. Quantitative results: separability and error

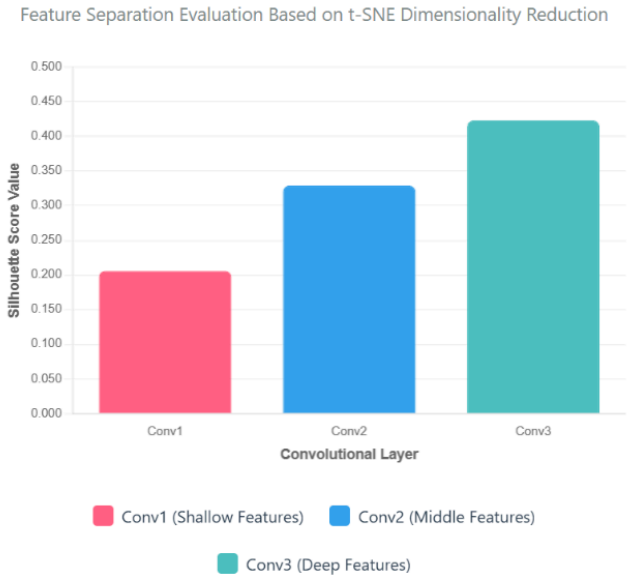

Figure 2. CNN layer silhouette scores comparison

Figure 2 presents a comparison of the Silhouette scores across the layers of the CNN. The Silhouette coefficients show a clear trend of improving separability as we move from Conv1 to Conv3: Conv1 has a Silhouette score of approximately 0.2058, indicating moderate separability with some overlap between classes. Conv2 sees an improvement, with the score rising to 0.3288, showing more distinct clustering. Conv3 achieves the highest score of 0.4227, reflecting a significant increase in separability as the feature space becomes more discriminative.

The Silhouette score quantitatively corroborates the observations from Figure 1, where the clusters in Conv3 are not only visually separated but also more distinct, as confirmed by the higher Silhouette value. Misclassifications, as shown in the detailed analysis of Figure 1, often occur at class boundaries where similar digits (such as 9 and 4/7) are confused. The trend in Figure 2 reinforces the conclusion that deeper layers enhance separability, which results in more accurate classifications overall.

4.4. Sensitivity and ablation

We explored three practical sensitivities: (1) Perplexity. Varying t-SNE perplexity from 5 to 50 changes local cluster granularity but does not alter the monotonic trend of Silhouette across layers. Small islands are exaggerated by lower confusion; the map is smoothed by higher values. (2) PCA dimension. Similar layer ordering is obtained by reducing to 50D or 200D prior to t-SNE; 100D provides a decent speed-stability trade-off. (3) Sample size. The qualitative picture stays the same from 500 to 3000 samples. However, with a bigger NN, the presentation looks better using density/hexbin or subsampling, both of which are supported by our plotting tools.

These tests show that the dashboard is strong enough to be used for comparisons. Even when layouts are different, the order of layers and the location of mistakes stay the same.

DOI: 10.54254/2755-2721/2026.TJ28969

4.5. Actionable diagnostics for development

During model iteration, the dashboard provides tangible levers due to its layer-indexing: Consider extending training times with cosine scheduling, expanding mid-level channels, or employing larger kernels to expand receptive fields if Conv2 Silhouette lags while Conv1→Conv2 exhibits minimal improvement.

If Conv3 still exhibits overlap, the architecture's capacity might be reached or the data might not be sufficiently diverse; use margin-oriented losses (such as additive angular margin), class reweighting, or stronger augmentation.

If mistakes tend to cluster in a certain area (for example, 9 vs. 4/7), create targeted augmentations or gather additional data for those boundary situations. Teams can record if these activities truly shift incorrectly classified crossovers away from the boundary and increase the silhouette using the dashboard. These interventions offer a population-level representation perspective to the toolbox, complementing interpretability strategies that concentrate on filters or optimization landscapes.

4.6. Relation to prior paradigms and tools

The patterns that stimulate certain neurons or filters are revealed via feature visualization techniques (such as deconvolution and activation maximization). Loss-landscape visualizations, on the other hand, characterize global geometry to understand optimization and generalization. Our scatter-plot scheme occupies the middle ground: it examines how all samples are arranged in the representation space at each depth. The three views are complementary and, together, deliver a fuller picture of model behavior. For interaction, the 2-D embeddings can be uploaded to the Embedding Projector for brushing, linking, and image-level inspection, which is helpful during error triage and dataset curation [14].

4.7. Limitations and threats to validity

t-SNE prioritizes local neighborhoods; global distances can be misleading, and different runs can vary if seeds or parameters change. PCA pre-reduction may discard weak but discriminative directions. Silhouette depends on distance metrics and cluster shapes; very elongated or crescent-shaped clusters may receive lower scores despite being class-pure [15]. Finally, MNIST is simpler than natural-image datasets; absolute values will differ on harder corpora. Nevertheless, the directional story—improving separability with depth and boundary-localized errors—proved robust across our ablations and is precisely the level of evidence practitioners need when tuning models.

5. Conclusion

This study presents a layer-wise scatter-plot scheme that reveals how class structure emerges inside a CNN and where residual errors live. By extracting per-layer embeddings, applying a PCA→t-SNE pipeline, and pairing visuals with the Silhouette coefficient and per-class error rates, the method converts opaque intermediate activations into actionable diagnostics rather than merely aesthetic projections. On MNIST with a compact three-block CNN, separability increases monotonically from Conv1 to Conv3 (≈0.21→0.33→0.42). Misclassifications concentrate near inter-class boundaries—most notably between visually similar digits (9 versus 4/7)—while distinctive shapes such as 8 remain cleanly isolated. These findings are intuitive yet non-trivial: practitioners can now both see and quantify how depth helps, and where it does not, within the same reproducible dashboard [16].

DOI: 10.54254/2755-2721/2026.TJ28969

The pragmatic explainability has a wider value. While some interpretability techniques create visually appealing graphs that are difficult to connect to specific engineering decisions, many are insightful but heavyweight. We purposefully make our proposal model-agnostic, lightweight, and replicable. Through preprocessing (PCA prior to t-SNE) and design (alpha-blended context for successfully categorized samples and a discrete overlay for errors), it reduces overplotting. Interpretation is then anchored in a single, easily comprehensible index (Silhouette). Because the dashboard is layer-indexed, it naturally answers the everyday question "which layer needs work?" during architecture and hyper-parameter tuning. Teams can track how interventions—adding channels, modifying kernel sizes, extending training, adjusting learning-rate schedules, or strengthening augmentation—shift the Silhouette trajectory and re-arrange boundary errors, turning visualization into a closed-loop feedback signal for development [17].

More importantly, the approach enhances other viewpoints on model behavior rather than taking their place. Loss-landscape perspectives show how optimization acts worldwide, while filter-centric approaches describe what neurons react to. By revealing population structure in representation space across depth, our method strikes a compromise. This population view is often the missing piece when deciding between architecture variants that perform similarly on headline accuracy but differ in how they separate classes internally—a factor that can influence robustness and data efficiency [18].

In summary, layer-wise scatter-plot diagnostics provide a simple, robust, and extensible path toward explainable CNN development. By combining careful visualization with compact, numeric evidence, they transform intermediate representations into a tool that guides decisions, documents progress, and ultimately improves the reliability and transparency of deep classifiers.

References

[1] Raj, R., Kos, A. (2025) An extensive study of convolutional neural networks: applications in computer vision for improved robotics perceptions. Sensors, 25: 1033.
[2] Shirane, K., Yamamoto, T., Tomiyama, H. (2021) A design methodology for approximate multipliers in convolutional neural networks: A case of MNIST. Intern. J. Reconfig. Emb. Syst., 10: 1.
[3] Sun, Y., Han, Y., Fan, J. (2023) Laplacian-based cluster-contractive t-SNE for high-dimensional data visualization. ACM Trans. Knowl. Discov. Data, 18: 1-22.
[4] Xia, L., Lee, C., Li, J.J. (2024) Statistical method scDEED for detecting dubious 2D single-cell embeddings and optimizing t-SNE and UMAP hyperparameters. Nat. Commun., 15: 1753.
[5] Bocker, M., Grushko, M.G., Arline, K.E. (2022) toward improved cancer classification using PCA+ tSNE dimensionality reduction on bulk RNA-seq data. Cancer Res., 82: 2708-2708.
[6] Meyer, B.H., Pozo, A.T.R., Zola, W.M.N. (2022) Global and local structure preserving GPU t-SNE methods for large-scale applications. Expert Syst. Appl., 201: 116918.
[7] Zhang, Q., Liu, Y., Fang, H. (2024) Manifold learning-based UMAP method for geochemical anomaly identification. Geochemistry, 84: 126157.
[8] Goldstein, A., Shahar, Y., Weisman Raymond, M., Peleg, H., Ben-Chetrit, E., Ben-Yehuda, A., Shalom, E., Goldstein, C., Shiloh, S.S., Almoznino, G. (2024) Multi-dimensional validation of the integration of syntactic and semantic distance measures for clustering fibromyalgia patients in the rheumatic monitor big data study. Bioengineering, 11: 97.
[9] Katarya, R., Saini, R. (2022) Enhancing the wine tasting experience using greedy clustering wine recommender system. Multimed. Tools. Appl., 81: 807-840.
[10] Wataya, T., Nakanishi, K., Suzuki, Y., Kido, S., Tomiyama, N. (2020) Introduction to deep learning: minimum essence required to launch a research. Jpn. J. Radiol., 38: 907-921.
[11] Zhang, L., Jiang, C., Chai, Z., He, Y. (2024) Adversarial attack and training for deep neural network based power quality disturbance classification. Eng. Appl. Artif. Intell., 127: 107245.
[12] Bagirov, A.M., Aliguliyev, R.M., Sultanova, N. (2023) Finding compact and well-separated clusters: Clustering using silhouette coefficients. Pattern Recognit., 135: 109144.

DOI: 10.54254/2755-2721/2026.TJ28969

[13] Wu, W., Shi, X., He, L., Jin, H. (2023) Turbognn: Improving the end-to-end performance for sampling-based gnn training on gpus. IEEE Trans. Comput., 72: 2571-2584.
[14] Maitra, C., Seal, D.B., De, R.K. (2024) NeuroDAVIS: A neural network model for data visualization. Neurocomputing, 573: 127182.
[15] Zhou, Y., Sharpee, T.O. (2022) Using global t-SNE to preserve intercluster data structure. Neural Comput., 34: 1637-1651.
[16] Liu, N., Lu, X., Wan, L., Huo, H., Fang, T. (2018) Improving the separability of deep features with discriminative convolution filters for RSI classification. ISPRS Int. J. Geoinf., 7: 95.
[17] Kalyanathaya, K.P. (2024) A novel method for developing explainable machine learning framework using feature neutralization technique. Scient. Temper, 15: 2225-2230.
[18] Xie, T., Chen, J., Yang, Y., Geniesse, C., Shi, G., Chaudhari, A., Cava, J.K., Mahoney, M.W., Perciano, T., Weber, G.H. (2024) LossLens: Diagnostics for Machine Learning Through Loss Landscape Visual Analytics. IEEE Comput. Graph. Appl.: 1-14.