HiLL PUBLISHING

Future Trends in AI Research, 2025, 2(1), 26-36
https://www.hillpublisher.com/Journals/ftair/
ISSN Online: 3066-3083

A Comprehensive Survey on Visualization Techniques for Convolutional Neural Networks

Yonten Jamtsho

Gyalpozhing College of Information Technology, Royal University of Bhutan, Thimphu 11001, Bhutan.

How to cite this paper: Yonten Jamtsho. (2025) A Comprehensive Survey on Visualization Techniques for Convolutional Neural Networks. *Future Trends in AI Research*, 2(1), 26-36.
DOI: 10.26855/ftair.2025.12.006

Received: October 11, 2025
Accepted: November 21, 2025
Published: December 27, 2025

***Corresponding author:** Yonten Jamtsho, Gyalpozhing College of Information Technology, Royal University of Bhutan, Thimphu 11001, Bhutan.

Abstract

As CNNs become increasingly complex, understanding their decision-making processes is crucial for improving model transparency. This study evaluates various visualization tools to enhance interpretability, including saliency maps, Grad-CAM, and feature map visualizations. The goal is to compare these tools based on effectiveness, usability, transparency, and integration with deep learning frameworks. The methodology involved reviewing these tools to assess how well they provide insights into CNNs' learned features and decision pathways. Key findings reveal that NetDissect emerges as the top-performing tool due to its comprehensive capabilities in visualization, user interface, data processing, transparency, and research impact. Other top performers, such as SHAP, AI Explainability 360, and LIME, demonstrate strong performance, particularly in transparency, data processing, and integration with frameworks. Tools such as TensorBoard and Activation Atlas also perform well, excelling in visualization and interactivity. However, tools such as Foolbox and CNN-Fixations have limitations in transparency and interpretability, making them less suitable for in-depth model analysis. The study concludes that NetDissect, SHAP, AI Explainability 360, and LIME are the most effective tools for interpreting CNN models, with NetDissect standing out as the most comprehensive and impactful for research.

Keywords

Convolutional Neural Networks; Deep Learning; Model Interpretability; Feature Visualization; Explainable Artificial Intelligence

1. Introduction

CNNs are a primary type of neural network employed for image recognition and classification tasks [1]. CNNs use a hierarchical architecture of convolutional, pooling, and fully connected layers to learn automatically and extract complex patterns from data. This architecture enables CNNs to model spatial relationships in images, making them highly effective across a range of AI tasks. Due to their remarkable performance, CNNs are regarded as the backbone of many modern AI applications.

Despite their widespread success, CNNs face significant interpretability challenges. Their deep, complex architecture often leads to models that function as “black boxes,” making it difficult to understand how decisions are made [2]. This lack of transparency poses risks in applications that require accountability, such as healthcare, finance, and autonomous vehicles [3]. Researchers are actively developing techniques, such as visualization tools and interpretability frameworks, to uncover the inner workings of CNNs. These efforts aim to provide more insights into what the network learns and how it makes decisions, and to build trust in the models used for critical applications [4].

Visualization plays a critical role in understanding the decision-making processes of CNNs. As these models are often considered "black boxes," visualization techniques help interpret their internal workings by highlighting the features and patterns influencing predictions. This enhances transparency, builds trust, and helps diagnose issues such as overfitting or reliance on irrelevant features, ultimately leading to improved model performance [5].

This research aims to explore and categorize existing CNN visualization techniques, providing a structured understanding of their diversity and applications. It also seeks to evaluate the usability and effectiveness of popular visualization tools, assessing their practicality and contribution to improving CNN interpretability. Additionally, the study identifies current limitations in the field and proposes potential directions for future research to address these gaps and advance the development of more effective visualization methods.

2. Literature Review

2.1 Convolutional Neural Networks

The development of CNNs can be traced back to the 1980s when Yann LeCun introduced the concept of using gradient-based neural networks [6]. This innovation led to the creation of LeNet-5, one of the first CNN architectures, successfully used for handwritten digit recognition. The architecture of LeNet-5 is shown in Figure 1:

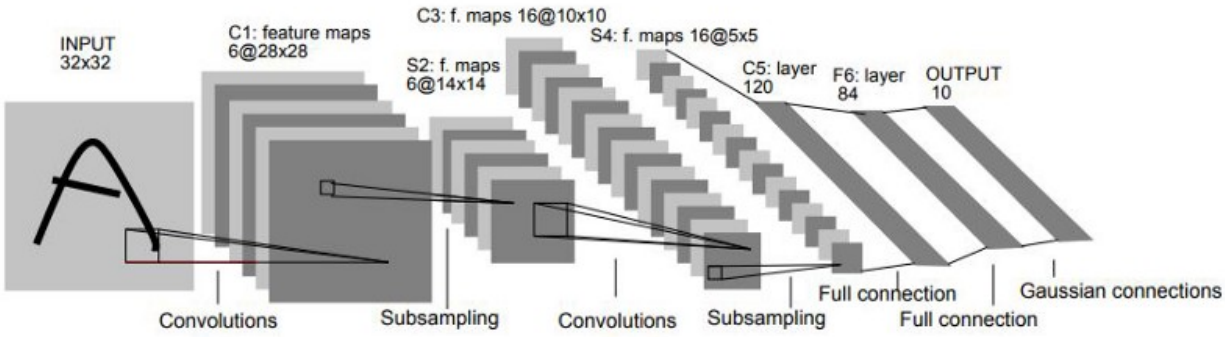

Figure 1. Architecture of LeNet-5.

Despite its success, the practical application of CNNs was limited during this period due to computational constraints and a lack of large datasets. The evolution of CNNs accelerated in the 2010s with advancements in hardware, particularly GPUs, and the availability of large-scale datasets like ImageNet. The breakthrough came in 2012 when AlexNet, developed by Krizhevsky et al., achieved state-of-the-art performance in the ImageNet Large Scale Visual Recognition Challenge [7]. This success marked a new era for deep learning, demonstrating the potential of CNNs for large-scale image classification tasks. Subsequent innovations, such as VGGNet [8], ResNet [9], and Inception [10], introduced deeper and more efficient architectures, solving challenges like vanishing gradients and improving computational efficiency. Today, CNNs are integral to various applications, including computer vision, natural language processing, and medical imaging, continually evolving to tackle increasingly complex tasks.

2.2 Overview of CNN Architecture and Components

CNNs are composed of specialized layers designed to process and analyze spatial hierarchies in data, particularly images. The architecture typically includes three main types of layers: convolutional layers, pooling layers, and dense (fully connected) layers, each playing a distinct role in feature extraction and classification.

2.2.1 Convolutional Layers

Convolutional layers are the core building blocks of CNNs. They apply filters (kernels) to input data to detect patterns such as edges, textures, and shapes. These filters slide over the input, performing a convolution operation that results in feature maps, which highlight the presence of specific features in the data. The learned filters are critical for capturing spatial hierarchies and patterns. The convolution operation between the input image and filter is shown in Figure 2.

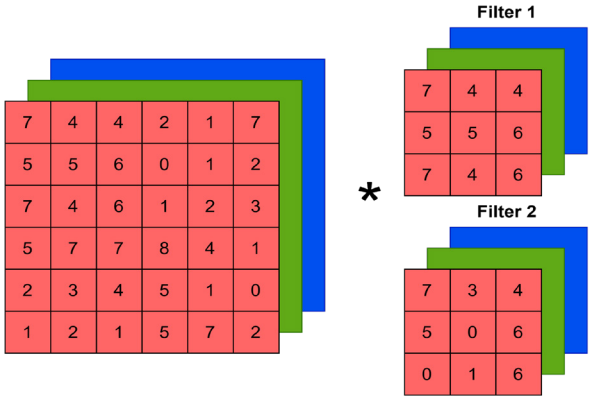

Figure 2. Convolution Operations.

2.2.2 Pooling Layers

Pooling layers reduce the spatial dimensions of feature maps, making the network more computationally efficient and less prone to overfitting. Max pooling, one of the most common pooling methods, selects the maximum value from a region of the feature map, preserving the most prominent features. This operation helps in retaining essential information while discarding less significant details. The depiction of Max Pool is shown in Figure 3.

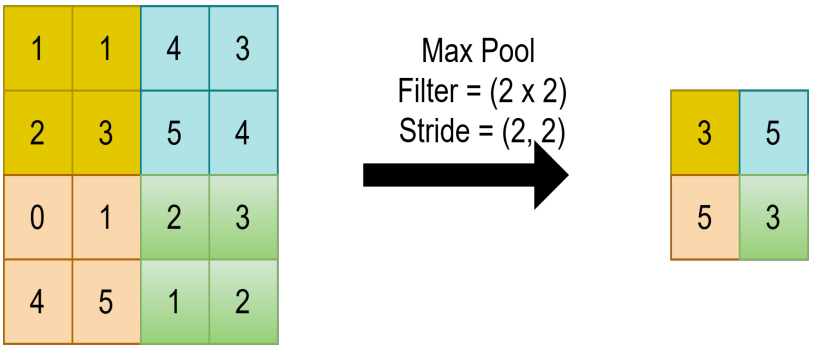

Figure 3. Pooling Operations.

2.2.3 Dense (Fully Connected Layers)

Dense layers are typically used toward the end of the CNN architecture. They flatten the high-dimensional feature maps into a single vector and use it for classification or regression tasks. Fully connected layers combine features learned in previous layers to make predictions, providing the network's final output. The depiction of FC Layers is shown in Figure 4.

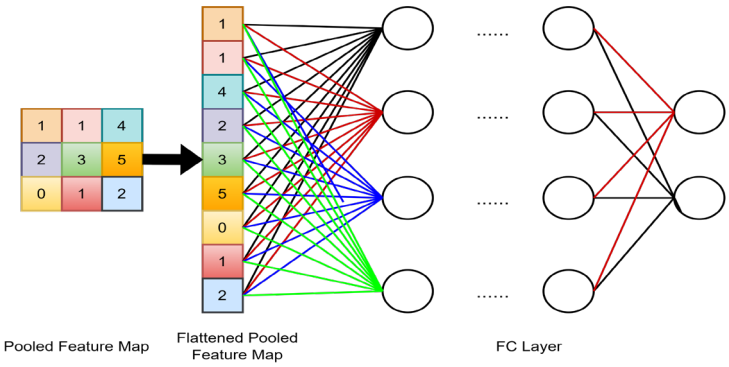

Figure 4. Fully Connected Layers.

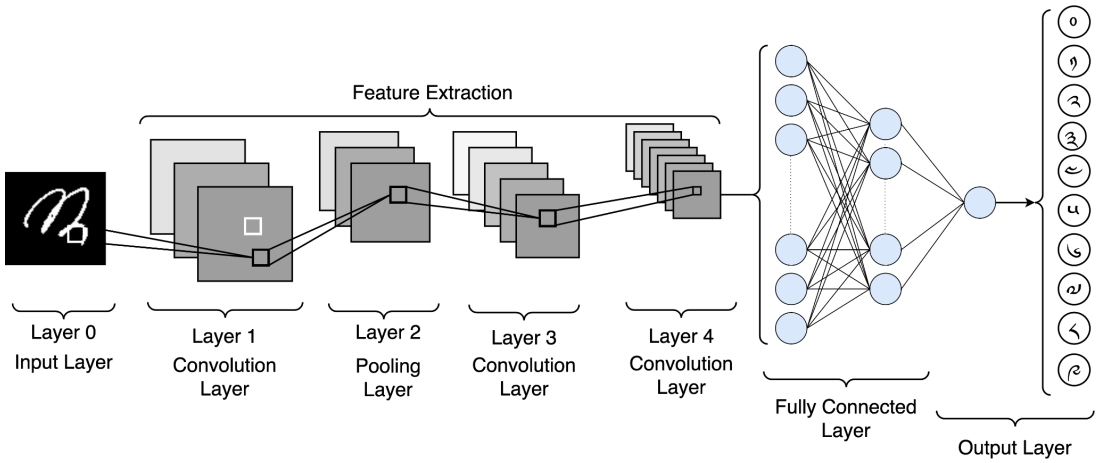

Figure 5. A Simple CNN Architecture [11].

3. Research Methodology

This section outlines the methodology employed to conduct a comprehensive survey of the existing visualization tools for CNNs. This survey aims to assess the practical utility and effectiveness of these tools in enhancing CNN interpretability. The methodology consists of several stages, including the identification of relevant tools, the establishment of evaluation criteria, and the analysis of each tool's performance and features.

3.1 Identification of CNN Visualization Tools and Techniques

The first phase of this survey involves identifying CNN visualization tools that are publicly available on the internet. A variety of online sources were explored, including open-source repositories, technical blogs, tutorials, and educational websites, to compile a comprehensive list of tools. Sources such as GitHub, Google Scholar, ArXiv, and popular deep learning platforms like TensorFlow and PyTorch were thoroughly searched to identify tools that have been developed for visualizing CNN architectures, feature maps, filters, and decision-making processes.

In this survey, the term visualization tools broadly refers to both software platforms and visualization techniques used for CNN interpretability. Some entries such as Saliency Maps, Grad-CAM, and DeepLIFT, are visualization or interpretability techniques implemented through libraries and frameworks, whereas TensorBoard, Netron, and AI Explainability 360 are standalone tools or toolkits.

The tools and techniques were selected based on several criteria, including public availability, relevance to CNN interpretability, and citation frequency in scholarly literature, GitHub repository activity, compatibility with major deep learning frameworks, and the availability of documentation or tutorials. Preference was given to tools that demonstrated active community support and reproducibility.

Table 1. Tools and Techniques for Visualizing and Interpreting Convolutional Neural Networks

Sl No	Tool/Technique Name	Description	Source
1	Convolution Demo	Demonstrates how convolution operations are performed between input images and kernels in CNNs.	[12]
2	Max Pool Demo	Demonstrates the max pooling process used for feature reduction in CNNs.	[13]
3	CNN Explainer	Interactive web-based tool for visualizing CNN concepts such as convolution, pooling, and feature maps.	[14]
4	3D Visualization of CNN	Provides interactive 3D visualization of CNN layers, activations, and feature extraction processes.	[15]

Table 1 Continued

Sl No	Tool/Technique Name	Description	Source
5	Netron	Open-source tool for visualizing and inspecting neural network architectures layer by layer.	[16]
6	TensorBoard	Visualization toolkit for monitoring, debugging, and analyzing deep learning models and CNN feature maps.	[17]
7	Grad-CAM	Generates heatmaps to highlight image regions influencing CNN predictions.	[18] [19]
8	SHAP	Explains CNN predictions by measuring the importance of input features.	[20]
9	Saliency Maps	Highlights significant input regions contributing to CNN decision-making.	[21]
10	AI Explainability 360	Open-source toolkit providing multiple methods for AI model interpretability and visualization.	[22]
11	VisualBackProp	Visualizes neuron contributions by propagating relevance through CNN layers.	[23]
12	DeepVis Toolbox	Visualizes feature maps, activations, and filters in pre-trained CNN models.	[24]
13	LIME	Generates local interpretable explanations for individual CNN predictions.	[25]
14	DeepLIFT	Explains CNN predictions by attributing importance scores to input features.	[26]
15	TensorSpace	Provides interactive 3D visualization of TensorFlow-based CNN architectures.	[27]
16	Activation Atlas	Visualizes activation patterns learned by CNNs across different input images.	[28]
17	Foolbox	Python library supporting adversarial analysis and CNN interpretability visualizations.	[29]
18	iNNvestigate	Library for interpreting CNNs using methods such as Layer-wise Relevance Propagation.	[30]
19	NetDissect	Analyzes CNN interpretability by mapping network units to visual concepts.	[31]
20	CNN-Fixations	Visualizes fixation and heatmaps showing regions influencing CNN predictions.	[32]

3.2 Criteria for Tool Evaluation

To ensure a systematic and objective evaluation of the selected tools, a set of evaluation criteria was established. These criteria were based on their ability to improve the interpretability and transparency of CNN models. The following criteria were used to assess the effectiveness and practical application of each tool (Table 2):

Table 2. Evaluation Criteria and Scoring Scale for CNN Visualization and Interpretability Tools

Criteria	1 - Poor	2 - Fair	3 - Good	4 - Very Good	5 - Excellent
Effectiveness of Visualization	Visualizations are unclear, not insightful, or lack functional utility.	Visualizations provide some value but lack clarity or robustness.	Visualizations are functional and offer moderate insights into CNN behaviour.	Visualizations are clear, insightful, and significantly aid in understanding.	Highly effective, offering advanced, detailed insights with exceptional utility.
User Interface & Interactivity	The interface is unintuitive, poorly designed, and lacks interactive features.	The interface is basic, minimally interactive, and somewhat unintuitive.	A functional and moderately intuitive interface with some interactive features.	An intuitive interface offering useful interactivity to enhance usability.	Flawless, highly intuitive interface with robust interactivity.
Data Processing Capabilities	Limited or no data processing support, requiring extensive manual effort.	Partial data processing capabilities but with significant manual intervention.	Moderate automation with support for basic preprocessing tasks.	Efficient data handling with broad support for automated preprocessing.	Advanced capabilities for handling diverse and large datasets effortlessly.
Transparency & Interpretability	Offers limited interpretability, making model behaviour challenging to understand.	Provides some insight but lacks comprehensive explanations.	Offers moderate transparency with limited interpretative depth.	Ensures clear insights into model decisions, enhancing interpretability.	Fully transparent, offering in-depth and comprehensible explanations.
Performance & Efficiency	Inefficient, with slow processing and high resource consumption.	Performs moderately but struggles with large-scale datasets.	Handles average workloads efficiently but with some resource constraints.	Processes tasks effectively, suitable for most workloads.	Highly efficient, and capable of handling complex and large-scale tasks smoothly.
Integration with Deep Learning Frameworks	No support for major frameworks such as TensorFlow or PyTorch.	Basic framework support with limited or less streamlined integration.	Adequate framework integration, requiring minor manual configuration.	Smooth integration with multiple frameworks.	Comprehensive compatibility with all major frameworks and effortless usability.
Limitations & Drawbacks	Severe limitations that significantly hinder functionality.	Noticeable limitations that restrict usability.	Functional with minor limitations that do not impact major operations.	Reliable with only a few minor, non-critical limitations.	Virtually no limitations, with optimal reliability and usability.
Novelty & Research Impact	Lacks originality and makes no significant contribution to research.	Limited novelty with minimal research impact.	Demonstrates moderate innovation and contributes meaningfully to the field.	Exhibits substantial novelty and has a positive research impact.	Highly innovative, establishing new standards in deep learning interpretability.

All evaluation criteria were assigned equal weight in this study. This approach was adopted because there is currently no universally accepted framework for prioritizing CNN visualization evaluation criteria. Equal weighting was used to maintain neutrality and reduce subjective bias in the comparative analysis.

4. Results and Discussion

4.1 Comparative Analysis

The following table provides a comprehensive evaluation of 20 CNN visualization tools/techniques, based on their

strengths and limitations across eight key criteria. These tools address diverse needs, ranging from conceptual understanding of CNN operations to advanced interpretability in real-world applications.

To ensure clarity and readability, the analysis has been divided into Table 3 and Table 4:

- Table 3 focuses on the criteria of Effectiveness of Visualization, User Interface and Interactivity, Data Processing Capabilities, and Transparency and Interpretability. These aspects emphasize usability, accessibility, and the ability of tools to make CNN operations comprehensible.
- Table 4 evaluates Performance and Efficiency, Integration with Deep Learning Frameworks, Limitations and Drawbacks, and Novelty and Research Impact. This part highlights the technical robustness, adaptability to frameworks, and potential impact of each tool in research and application.

The final score for each tool was calculated by summing the scores across all eight evaluation criteria. The graphical representation of the evaluation of the tool is shown in Figure 6:

Table 3. Comparative Evaluation of CNN Visualization and Interpretability Tools and Techniques

Tool/Techniques	Effectiveness of Visualization	User Interface and Interactivity	Data Processing Capabilities	Transparency and Interpretability
Convolution Demo	4	4	3	4
Max Pool Demo	4	4	3	3
CNN Explainer	5	5	4	5
3D Visualization of CNN	5	5	4	4
Netron	5	5	4	4
TensorBoard	5	5	5	5
Grad-CAM	4	5	4	5
SHAP	5	4	5	5
Saliency Maps	4	3	4	4
AI Explainability 360	5	4	5	5
VisualBackProp	4	4	3	5
DeepVis Toolbox	5	4	4	5
LIME	5	5	5	5
DeepLIFT	4	5	4	4
TensorSpace	5	5	4	5
Activation Atlas	5	5	4	5
Foolbox	4	3	5	3
iNNvestigate	4	4	5	4
NetDissect	5	5	5	5
CNN-Fixations	4	3	3	3

Table 4. Comparative Evaluation of CNN Visualization and Interpretability Tools and Techniques

Tool/ Techniques	Performance and Efficiency	Integration with Deep Learning Frameworks	Limitations and Drawbacks	Novelty and Research Impact	Final Score
Convolution Demo	4	3	3	4	29
Max Pool Demo	4	3	3	3	27
CNN Explainer	5	4	3	5	36
3D Visualization of CNN	4	4	3	5	34
Netron	5	5	4	3	35
TensorBoard	5	5	3	4	37
Grad-CAM	4	5	3	4	34
SHAP	5	5	4	5	**38**
Saliency Maps	3	3	5	4	30
AI Explainability 360	5	5	4	5	**38**
VisualBackProp	4	4	3	4	31
DeepVis Toolbox	4	4	4	5	35
LIME	5	5	3	5	**38**
DeepLIFT	4	4	3	4	32
TensorSpace	5	4	3	5	36
Activation Atlas	5	5	3	5	37
Foolbox	3	3	5	3	29
iNNvestigate	4	4	4	5	34
NetDissect	5	5	4	5	**39**
CNN-Fixations	3	3	5	3	27

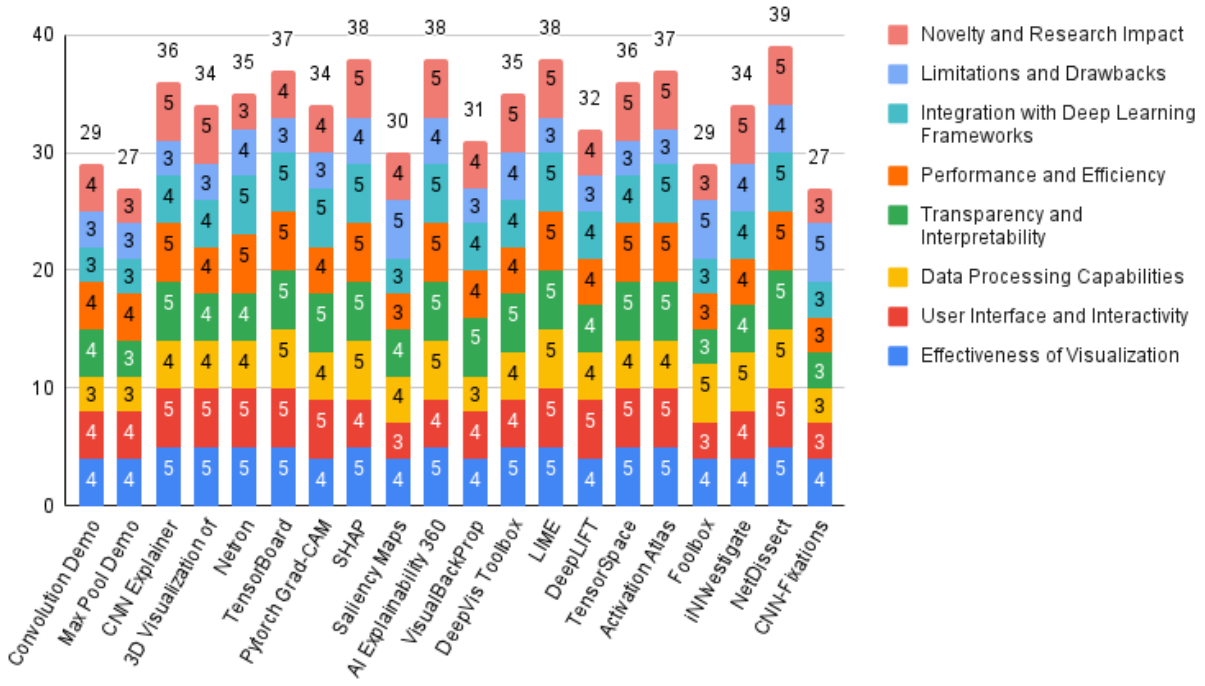

Figure 6. Comparative Performance Analysis of CNN Visualization and Interpretability Tools Based on Evaluated Criteria.

The key findings from the table highlight that NetDissect stands out as the top-performing tool, scoring the highest with 39 points. It excels across various categories, including Effectiveness of Visualization, User Interface and Interactivity, Data Processing Capabilities, Transparency and Interpretability, and Integration with Deep Learning Frameworks, while also making a significant contribution to Research Impact. Other top performers include SHAP, AI Explainability 360, and LIME, which all scored 38 points, demonstrating strong performance in areas like Transparency, Data Processing, and Integration with Frameworks. These tools are particularly noted for their capabilities in explaining deep learning models. Tools like TensorBoard and Activation Atlas (37 points each) also perform well, particularly in visualization and interactivity. On the lower end, tools like Foolbox (29 points) and CNN-Fixations (27 points) show limitations in Transparency, Interpretability, and Data Processing, suggesting that they may not be as effective for in-depth model analysis. Overall, the tools offering the best balance of visualization, interpretability, and integration with frameworks are NetDissect, SHAP, AI Explainability 360, and LIME, with NetDissect emerging as the highest-performing tool based on the selected evaluation criteria.

5. Conclusion

CNN visualization tools have evolved significantly, offering researchers and practitioners powerful methods to interpret and understand complex deep learning models. State-of-the-art tools such as NetDissect, SHAP, and AI Explainability 360 have demonstrated exceptional capabilities in enhancing model transparency, enabling effective debugging, and fostering trust in AI systems. These advancements underscore the critical role of visualization in advancing CNN research, from improving model accuracy and robustness to facilitating interdisciplinary applications in fields like medicine, finance, and autonomous systems.

Visualization is not just a supplementary tool but a driving force in comprehending the inner workings of CNNs, leading to better design and optimization of architectures. The ability to explain model decisions is crucial for responsible AI development, particularly in high-stakes domains where interpretability can directly impact outcomes. However, challenges remain, including the need for standardization, better integration with frameworks, and overcoming limitations in scalability and real-time performance.

By addressing these challenges and leveraging the opportunities presented by emerging visualization techniques, the field can continue to push the boundaries of AI research. Future efforts should focus on creating more user-friendly,

interpretable, and integrative visualization tools that cater to diverse stakeholders, from researchers and developers to end-users. In doing so, we can ensure that CNNs not only achieve technical excellence but also align with ethical and practical considerations in real-world applications.

References

[1] Taye MM. Theoretical understanding of convolutional neural network: concepts, architectures, applications, future directions. Computation. 2023;11(3):52.

[2] Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data. 2021;8(1):53.

[3] Parmar UPS, Surico PL, Singh RB, Goparaju AS. Artificial intelligence (AI) for early diagnosis of retinal diseases. Medicina. 2024;60(4):527.

[4] Ali S, Abuhmed T, El-Sappagh S, Muhammad K, Alonso-Moral JM, Confalonieri R, et al. Explainable artificial intelligence (XAI): what we know and what is left to attain trustworthy artificial intelligence. Inf Fusion. 2023;99:101805.

[5] 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). arXiv. 2017. Available from: https://arxiv.org/abs/1711.11279v5

[6] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278-324.

[7] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst. 2012;25:1097-105.

[8] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv. 2014. Available from: https://arxiv.org/abs/1409.1556

[9] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. arXiv. 2015. Available from: https://arxiv.org/abs/1512.03385

[10] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. arXiv. 2014. Available from: https://arxiv.org/abs/1409.4842

[11] Jamtsho Y, Yangden P, Wangmo S, Lhaden, Pema, Tshering K, et al. Deep learning-based Dzongkha handwritten digit classification. J Inf Technol Comput Eng. 2024;8(1):1-7.

[12] DEEPLIZARD. Interactive demo - convolution operation [Internet]. Available from: https://deeplizard.com/resource/pavq7noze2

[13] DEEPLIZARD. Interactive demo - max pooling operation [Internet]. Available from: https://deeplizard.com/resource/pavq7noze3

[14] Wang ZJ, Turko R, Shaikh O, Park H, Das N, Hohman F, et al. CNN explainer: learning convolutional neural networks with interactive visualization. IEEE Trans Vis Comput Graph. 2021;27(2):1396-406.

[15] Harley AW. An interactive node-link visualization of convolutional neural networks. In: Bebis G, Boyle R, Parvin B, Koracin D, editors. Advances in visual computing. Cham: Springer International Publishing; 2015. p. 867-77.

[16] Roeder L. Netron: visualizer for neural network, deep learning, and machine learning models [Internet]. Zenodo; 2017. Available from: https://doi.org/10.5281/zenodo.5854962

[17] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. TensorFlow: a system for large-scale machine learning. arXiv. 2016. Available from: https://arxiv.org/abs/1605.08695

[18] Gildenblat J. PyTorch library for CAM methods [Internet]. GitHub; 2021. Available from: https://github.com/jacobgil/pytorch-grad-cam

[19] Selvaraju RR, 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(2):336-59.

[20] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Adv Neural Inf Process Syst. 2017;30:4765-74.

[21] Ozbulak U. PyTorch CNN visualizations [Internet]. GitHub; 2019. Available from: https://github.com/utkuozbulak/pytorch-cnn-visualizations

[22] Arya V, Bellamy RKE, Chen PY, Dhurandhar A, Hind M, Hoffman SC, et al. One explanation does not fit all: a toolkit and taxonomy of AI explainability techniques. arXiv. 2019. Available from: https://arxiv.org/abs/1909.03012

[23] Bojarski M, Choromanska A, Choromanski K, Firner B, Jackel L, Muller U, et al. VisualBackProp: efficient visualization of CNNs. arXiv. 2017. Available from: https://arxiv.org/abs/1611.05418

[24] Yosinski J, Clune J, Nguyen A, Fuchs T, Lipson H. Understanding neural networks through deep visualization. arXiv. 2015.

Available from: https://arxiv.org/abs/1506.06579

[25] Ribeiro MT, Singh S, Guestrin C. "Why should I trust you?": explaining the predictions of any classifier. arXiv. 2016. Available from: https://arxiv.org/abs/1602.04938

[26] Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences. arXiv. 2019. Available from: https://arxiv.org/abs/1704.02685

[27] Zhang L, Zhang P, Ma X, Zhang S, Tao Y, Li W, et al. A generalized language model in tensor space. arXiv. 2019. Available from: https://arxiv.org/abs/1901.11167

[28] Carter S, Armstrong Z, Schubert L, Johnson I, Olah C. Activation atlas. Distill. 2019;4(3):e15.

[29] Rauber J, Zimmermann R, Bethge M, Brendel W. Foolbox native: fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch, TensorFlow, and JAX. J Open Source Softw. 2020;5(53):2607.

[30] Alber M, Lapuschkin S, Seegerer P, Hägele M, Schütt KT, Montavon G, et al. iNNvestigate neural networks! J Mach Learn Res. 2019;20(93):1-8.

[31] Bau D, Zhou B, Khosla A, Oliva A, Torralba A. Network dissection: quantifying interpretability of deep visual representations. arXiv. 2017. Available from: https://arxiv.org/abs/1704.05796

[32] Mopuri KR, Garg U, Babu RV. CNN fixations: an unraveling approach to visualize the discriminative image regions. arXiv. 2018. Available from: https://arxiv.org/abs/1708.06670