Explainable Convolutional Neural Networks: A Taxonomy, Review, and Future Directions

RAMI IBRAHIM and M. OMAIR SHAFIQ, School of Information Technology, Carleton University, Ottawa, Ontario, Canada

Convolutional neural networks (CNNs) have shown promising results and have outperformed classical machine learning techniques in tasks such as image classification and object recognition. Their human-brain like structure enabled them to learn sophisticated features while passing images through their layers. However, their lack of explainability led to the demand for interpretations to justify their predictions. Research on **Explainable AI** or **XAI** has gained momentum to provide knowledge and insights into neural networks. This study summarizes the literature to gain more understanding of explainability in CNNs (i.e., Explainable Convolutional Neural Networks). We classify models that made efforts to improve the CNNs interpretation. We present and discuss taxonomies for XAI models that modify CNN architecture, simplify CNN representations, analyze feature relevance, and visualize interpretations. We review various metrics used to evaluate XAI interpretations. In addition, we discuss the applications and tasks of XAI models. This focused and extensive survey develops a perspective on this area by addressing suggestions for overcoming XAI interpretation challenges, like models' generalization, unifying evaluation criteria, building robust models, and providing interpretations with semantic descriptions. Our taxonomy can be a reference to motivate future research in interpreting neural networks.

CCS Concepts: • **Human-centered computing** → **Visualization**; *Accessibility*; • **Computing methodologies** → *Artificial intelligence*; *Machine learning*; *Computer vision*;

Additional Key Words and Phrases: Explainable AI, convolutional neural networks, Interpretable AI, survey

ACM Reference format:
Rami Ibrahim and M. Omair Shafiq. 2023. Explainable Convolutional Neural Networks: A Taxonomy, Review, and Future Directions. *ACM Comput. Surv.* 55, 10, Article 206 (February 2023), 37 pages.
https://doi.org/10.1145/3563691

1 INTRODUCTION

Convolutional neural networks have a complicated architecture described as black-box. There is a lack of transparency in their internal mechanism. Therefore, it is hard for humans to understand the reason behind making a certain decision [1–3]. The absence of human interaction can impact trust in applications like autonomous cars [4]. For instance, it is crucial to explain to passengers when a vehicle suddenly changes lanes or reduces speed. In addition, providing explanations in convolutional neural networks can help AI experts know when a model succeeds or fails. Moreover,

This research has been funded by Natural Sciences and Engineering Research Council of Canada (NSERC) and Carleton University, Canada. This paper is based on and part of the PhD thesis of Rami Ibrahim, supervised by M. Omair Shafiq, Carleton University.

Authors' address: R. Ibrahim and M. O. Shafiq, Carleton University, 1125 Colonel By Drive, Ottawa, ON, Canada. K1S5B6; emails: {ramif.ibrahim, omair.shafiq}@carleton.ca.

0360-0300/2023/02-ART206
https://doi.org/10.1145/3563691

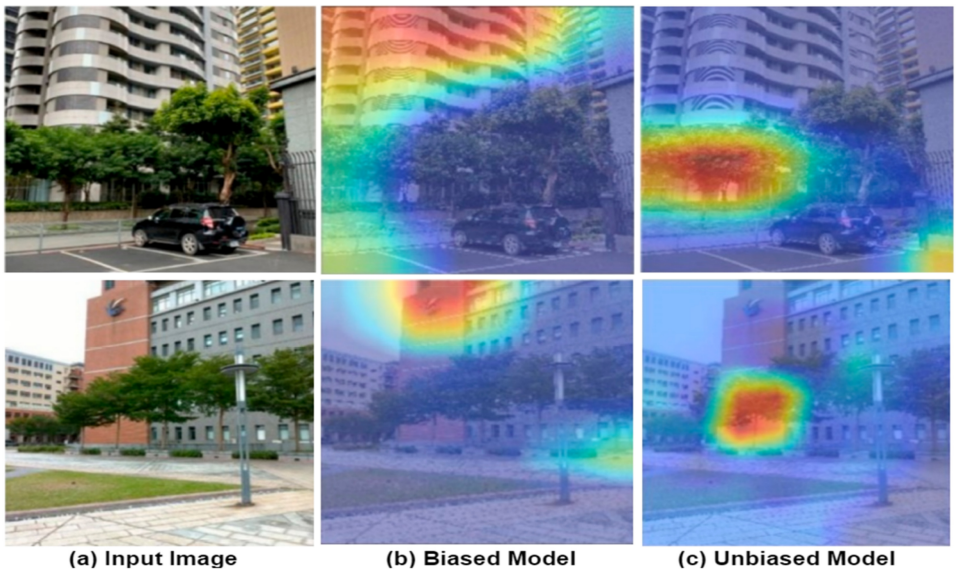

Fig. 1. Tree classification explanations [7].

regulators can use these explanations to detect unfaithful AI systems like biased decisions against gender, race, or age.

Therefore, **explainable AI** or **XAI** was presented to improve the explainability and interpretability in convolutional neural networks [5]. We refer to such works as Explainable Convolutional Neural Networks. Building explainable neural networks can provide valuable insights to end-users. Since human skills prefer visual data, explaining CNNs can be less complex than other models [5]. Meanwhile, some issues related to XAI models were addressed in the literature, like the need to identify the level of explanations provided in CNNs and the feasibility of embedding models like decision trees and linear regression in neural networks [6]. Additionally, providing various levels of explanations to stakeholders is critical [2, 5]. For example, providing technical details for end-users can be overwhelming. Whereas providing abstract explanations can make the model less transparent and impact the end-user trust. To demonstrate different levels of explanations, we analyze the explanations in the image and tabular classification tasks. In image classification, XAI models generate class activation maps to explain CNNs and interpret their decisions. These activation maps are useful for developers and end-users. Let us review the study that increased agricultural sustainability by identifying plants in Taipei, Taiwan [7]. In this study, the authors classified a dataset of 2,332 images with 14 tree species. Most of the tree images were captured with buildings in the background, as shown in the left column of Figure 1.

Therefore, despite having an accuracy of 74% on the InceptionV3 network, XAI activation maps successfully uncovered the bias in the CNN decision and showed that the CNN was looking at the building, not the tree itself, as shown in the middle column of Figure 1. These activation maps are useful for the developer to mitigate the bias in training data by adding tree images without buildings in the background. The right column represents the unbiased activation maps after fixing the training data. We can notice that the CNN in the right column focused on the tree, not the building in the background. After mitigating the bias in training data, the activation map explains the CNN decision to the end-user. In addition, some efforts were made to add a higher level of explanation to the activation maps by providing semantics that helped the end-user identify some object parts [8].

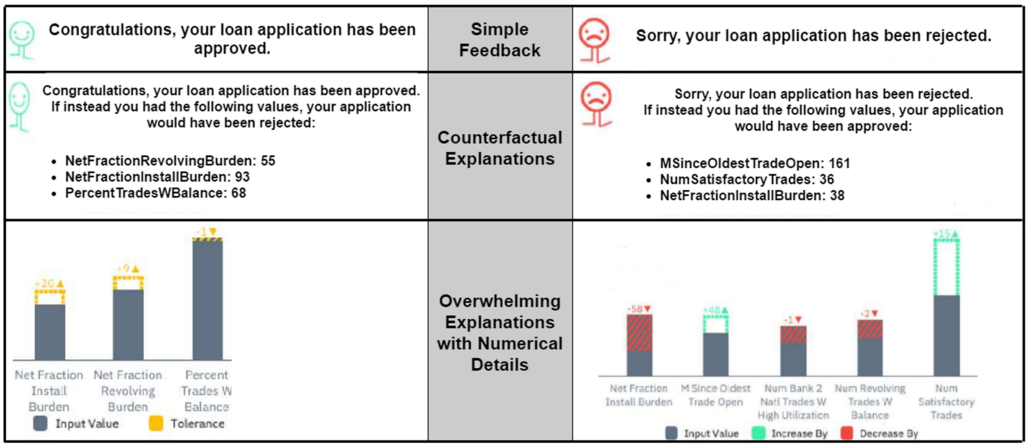

Fig. 2. Explanation levels for loan approval [9].

In tabular classification, textual feedback is provided to the end-user to explain the decision made by the classifier. Let us study a classifier that takes a loan application and decides if the loan is approved or rejected [9]. Providing the bank client with simple feedback (i.e., approved or rejected) can impact the trust in the system since the client has the right to know the reason behind this decision [10]. Therefore, XAI models like SHAP [11] can provide counterfactual explanations that justify the classifier's decision. The XAI model identifies the important features which contributed to this decision. After that, it calculates the minimum change required to flip the classifier's decision [10]. In Figure 2, we can notice that the first row provides simple feedback. The second row is a higher-level explanation generated by the XAI model. This explanation shows the decision in addition to the feature margins that can flip the decision. For instance, if the bank approves the loan application, the classifier shows features that could lead to rejection. In contrast, if the bank rejects the loan application, the classifier shows features that could lead to acceptance in the future. Therefore, these explanations help the client avoid certain issues that can cause rejection (e.g., Net fraction revolving burden of 55, Net fraction install burden of 93, Percent trades with a balance of 68). In contrast, explanations help the client fix some issues that can cause future approval (e.g., M since oldest trade open of 161, Num satisfactory trades of 36, Net fraction install burden of 38). The last row shows bar charts with feature increments/decrements required to flip the classifier's decision. This explanation level is overwhelming for the bank client as the plots are hard to interpret. Another issue is the lack of collaboration between software engineering and **human-computer interaction (HCI)** to build neural networks that provide explanations like contrastive, selective, conversational, and counterfactual explanations [1, 5, 6].

Current XAI models are still in their infancy stage. Therefore, they are not adopted widely in real-life applications. However, XAI experiments proved to have promising results in areas like self-driving cars, marketing, and bias detection [1, 6, 12]. For instance, autonomous vehicles can provide explanations along with their decisions. Voice or textual feedback can explain the reason for changing the vehicle's direction. In bias detection, the XAI model can identify the image pixels the CNN was looking to predict the class. For example, it was shown that the network trained to classify wolves and husky dogs relied on snow in the background, not on the features of a wolf or a dog. Most of the wolves' images in training data had snow in the background, which caused a bias and led to misclassifying a husky dog with a snowy background as a wolf [13]. Furthermore, XAI models were used in 3D action recognition, visual question answering, image classification, and image captioning [14, 15].

XAI models in convolutional neural networks can fall under an umbrella of two categories, interpreting the CNN decision and interpreting the CNN architecture. For decision interpretation, XAI models apply forward or backward passes to highlight significant parts in the input image, which leads to the predicted class. However, in the architecture interpretation, XAI models access the network layers and analyze their behavior. Moreover, XAI models in the two categories adopt various approaches to interpreting CNNs. This survey will discuss methods like architecture modification, CNN simplification, feature relevance, and visualization. We conducted this survey to summarize XAI models applied to convolutional neural networks. Previous surveys focused on reviewing XAI taxonomy, evaluation metrics, and application areas. However, they lacked a detailed analysis of XAI in convolutional neural networks. Therefore, we believe that this is the first specialized survey that studies XAI in CNNs. To address this knowledge gap, we proposed this survey with the following contributions:

(1) We conducted a structured search method and significant terms analysis to study the trend of XAI publications over the past years.
(2) We introduced a novel hierarchical taxonomy for XAI models that interpreted convolutional neural networks.
(3) We identified the structure, scope, and dependence for each XAI model we reviewed.
(4) We highlighted the correlations among XAI models in CNNs by building a Sankey chart that maps XAI taxonomy with structure, scope, and dependence.
(5) We discussed challenges that face XAI models in convolutional neural networks. In addition, we proposed some future directions to improve XAI models and address the research gaps.

Paper Organization: The remaining of the paper is organized as follows. Section 2 discusses our structured search method. Section 3 describes the XAI background. The XAI taxonomy for convolutional neural networks is presented in Section 4. Section 5 discusses the XAI evaluation metrics in convolutional neural networks. Section 6 discusses the application areas of XAI in convolutional neural networks. Section 7 discusses gaps and limitations in the existing XAI models and presents some future directions. Conclusions and closing remarks are drawn in Section 8.

2 RELATED EFFORTS

2.1 Search Query

The primary goal of this survey is to study XAI models in convolutional neural networks. Therefore, we investigate areas like XAI taxonomy, evaluation metrics, applications, and existing research gaps. We started by identifying the keywords for our search. We used combinations of three terms, “explainable”, “interpretable”, and “convolutional neural networks”. The search was conducted on Google Scholar and Scopus databases to explore relevant publishers, conferences, and journals. The search query was built with combinations of the three terms since previous studies used different terminologies to describe XAI.

2.2 Selection Criteria

In our search process, we excluded papers with the following criteria:

(1) Papers that are not in the English language
(2) XAI Papers not related to convolutional neural networks
(3) Papers implementing XAI models on a specific domain
(4) Papers that lack a full text or are not accessible
(5) Short papers

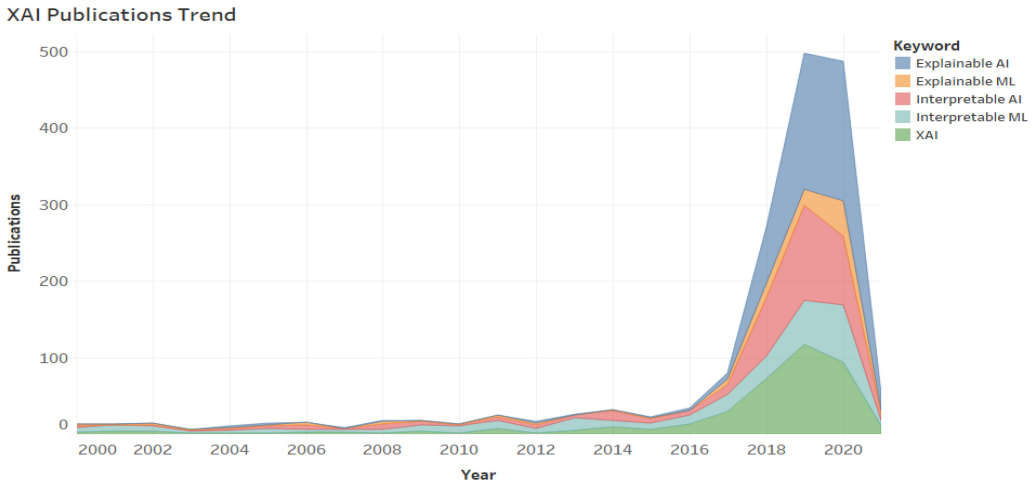

Fig. 3. XAI publications trend in the last two decades.

For the inclusion, we had the following criteria:

(1) Relevant full papers
(2) Relevant in progress papers (arXiv)

In the search process, an initial screening was applied as a first filter to exclude non-relevant papers. After that, another screening was done by reading sections like abstract, methodology, and conclusion. The search was conducted in February 2021 and returned 56 relevant publications.

2.3 XAI Publications Trend in Past Few Years

To highlight the growing attention to explainable AI, we analyzed the publications in the past few years. In this analysis, we used the Web of Science academic database as a source of our analysis. To cover all possible terms of XAI, we used various keywords in the search engine. The five keywords we searched for were "XAI", "Explainable AI", "Interpretable AI", "Explainable ML", and "Interpretable ML". After that, we analyzed the search query results.

We summarized the number of publications in the last 20 years (i.e., 2000 to 2021). Figure 3 shows the trend of XAI publications over the last two decades. We can observe the hike in XAI publications from 2018 to 2020. Therefore, it is evident that explainability and interpretability in Artificial Intelligence are attracting more researchers. Hence, there is a need for transparent AI systems that deliver faithful decisions, preserve users' privacy, and promote our community. Moreover, we highlighted the significant terms in relevant publications. We exported the abstracts from 2018 to 2021 using the Web of Science academic database. In addition, we identified the following key terms, "XAI CNN", "XAI deep learning", "XAI convolutional neural network", "explainable CNN", "explainable deep learning", "explainable convolutional neural network", "interpretable CNN", "interpretable deep learning", and "interpretable convolutional neural network". After that, we used the exported text file to build a word cloud that visualizes frequent terms in the abstracts, as shown in Figure 4. We can notice some interesting terms in this figure. For instance, the "image" term was highly recurrent in the abstracts. We believe that it is due to the images being the type of data CNNs use to make predictions. Another interesting term is "classification", which represents the image classification task. We believe this recurrence is because most interpretable CNN models were evaluated based on image and text

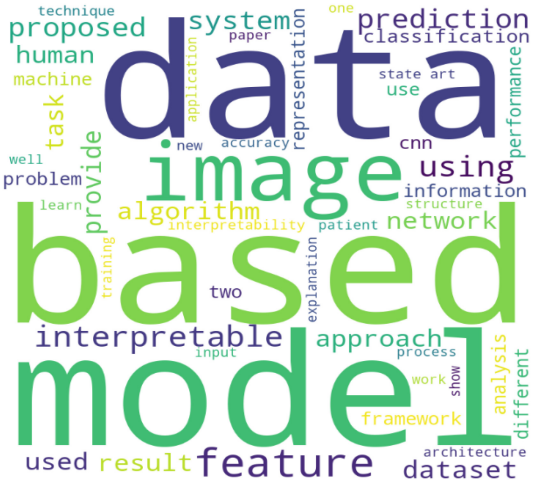

Fig. 4. Word Cloud for relevant terms in papers abstracts.

classification more than other applications. Another interesting recurrent term is "feature". We believe that this term represents the feature maps in CNNs. Apparently, XAI models relied on CNN features to interpret their prediction. The term "human" highlights the demand for CNN interpretations that involve humans in the loop. The terms "technique", "system", "algorithm", "model", and "approach" describes synonyms for the prototype that interprets CNNs.

3 XAI BACKGROUND

In this section, we discuss various XAI terms and definitions. Also, we review taxonomies adopted in previous studies.

3.1 XAI Concepts and Terminology

AI systems can be explainable by nature (intrinsic) or by adding supplementary XAI models (post-hoc) [12]. Explainability was defined as the interface that provides explanations to humans. However, interpretability was defined as the cognition of these explanations [5]. Therefore, XAI models should provide explanations that are interpretable and perceivable [16]. Moreover, XAI explanations are hard to generalize due to different domains and stakeholders (e.g., end-users, AI experts). Therefore, people of various disciplines like computer scientists, HCI, and social scientists need to collaborate to generate explanations with proper levels [1, 3, 5, 17]. AI experts can receive technical explanations for the model. Meanwhile, end-users can receive explanations for the decision made by the model.

Despite the increasing acceptance of AI systems, users still lack the human awareness of understanding their nature [16]. For instance, some users linked AI systems to robots and did not consider recommendation systems as AI. This knowledge gap should motivate XAI models to provide human-friendly explanations [12]. These explanations can be contrastive, conversational, selective, and counterfactual [1]. Moreover, XAI explanations should comply with AI principles described in the European **General Data Protection Regulation (GDPR)** [1, 5]. The purpose of these principles is to provide XAI explanations that preserve AI systems' faithfulness by detecting biased decisions. Additionally, XAI explanations should protect the privacy of AI systems besides improving their performance.

3.2 XAI Models Categorization

Previous studies categorized XAI models using various criteria. They relied on factors like scope, structure, dependence, and dataset. For structure, XAI models could be a part of the network (i.e., intrinsic) or could be attached to explain the network (i.e., post-hoc) [12, 18]. For example, intrinsic XAI models could embed autoencoders in convolutional neural networks to interpret them [19, 20]. For the scope, the XAI model could access the data to provide explanations (i.e., local) or analyze the network behavior (i.e., global) [1, 6, 12]. Local models accessed individual instances while global models studied the network architecture as a black box. For the dependence criteria, some XAI models were designed to work with specific AI systems (i.e., model-specific), while other models could generalize across several networks (i.e., model-agnostic) [6, 12]. For the dataset, XAI models could explain different data types, such as images, text, and tabular data [5]. For instance, XAI models produced saliency maps to explain image data. Meanwhile, they used other approaches like feature importance and visual plots to explain correlations among tabular data features.

Other studies categorized AI systems based on the existence of XAI models [3]. For instance, the black-box criteria indicated that the model was not transparent such as neural networks. The grey-box criteria meant that the XAI model was attached to the AI system. The white-box criteria indicated that the AI system was transparent, like linear regression and decision trees. Furthermore, some studies categorized XAI models based on the AI system deepness [5]. For instance, shallow XAI models existed in interpretable AI systems like linear and logistic regression. RuleFit [21] interpreted regression tasks by building new decision rules and ranking them based on their importance. In contrast, other XAI models interpreted regression tasks by calculating each feature's contribution to the class prediction [22, 23]. These models could explore the relationship between features and the average prediction by plotting their significance value (positive vs. negative). Semi-shallow XAI models were used with **Random Forest (RF)** and **Support Vector Machine (SVM)** [5]. The models in this area adopted approaches like architecture simplification and feature relevance. For instance, **Hyper-rectangle Rule Extraction (HRE)** applied clustering to generate prototypes for class samples [24]. They interpreted SVM by constructing hyperrectangle rules. For the deep XAI models, multiple models were proposed to interpret neural networks. In the upcoming sections, we review XAI models applied to convolutional neural networks.

4 XAI TAXONOMY IN CNNS

Convolutional neural networks (CNNs) are achieving significant advancements in the field of computer vision. CNNs can perform tasks like image classification, object detection, face recognition, and semantic segmentation [25]. They have an architecture that mimics the human brain. By forward passing through hundreds of convolutional layers and pooling layers, they can incrementally learn high-level features of an object [26]. In the end, fully connected layers map features with output class scores. Despite these achievements, the sophisticated structure of CNNs limited the ability to explore their internal representation and understand the reasons behind their decisions. Therefore, there is an increasing demand for CNNs explainability in computer vision areas like autonomous vehicles. In this survey, we analyze the latest research in explaining and interpreting CNNs. We taxonomize the literature in this area and discuss each category. After that, we discuss adopted qualitative and quantitative evaluation metrics and describe the applications of explainable CNNs. Finally, we identify the research gaps and propose some future directions. Previous studies related to explaining CNNs can be categorized as decision models and architecture models [5]. Decision models interpreted the CNN by applying backpropagation and mapping the predicted class with corresponding pixels in the input image. These models could identify the parts of an image that mostly contributed to the network decision. Meanwhile, architecture models explored

the network and analyzed the mechanism of layers and neurons. Decision models can be further divided into two subcategories, feature relevance, and visual explanation. Moreover, architecture models can be further divided into two subcategories, architecture modification, and architecture simplification. This survey considers the four subcategories as a taxonomy to categorize CNN explainable models in convolutional neural networks.

4.1 Architecture Modification

Explainable models in this category modify the CNNs architecture to improve their interpretability. This modification can replace some CNN parts like layers and loss functions or add new components to the CNN network like attention layers, autoencoders, and deconvolutional layers.

Various types of attention mechanisms were incorporated into CNNs architecture. **Global-and-local attention (GALA)** was integrated with neural networks like ResNet-50 to produce attention activity maps [27]. GALA could identify the important parts and features in the object by learning local saliency and global context. ClickMe.ai tool proved that interpretable visual features in GALA were like human features. In this tool, participants could interact with an image recognition task before and after applying GALA. ClickMe maps showed that the classification error in GALA was less than in state-of-art neural networks. The selection of network layers that need to use GALA could be challenging. There is a need for systematic analysis to identify the optimal layers and features to be selected. Moreover, GALA performed qualitative analysis and adopted a human-in-the-loop approach. However, there was a lack of quantitative analysis for attention activity maps like object localization. Attention mechanisms like DomainNet [28] considered two levels to enhance classification, object-level and part-level. The model aimed to find object parts to extract features. The object-level prediction followed top-down attention, while part-level prediction followed bottom-up attention. The model produced object-level predictions by converting a pre-trained CNN to FilterNet, a network that selected patches and then passed them to train another CNN called DomainNet. For part-level predictions, a part-based network was adopted. The DomainNet model did not use various layers in CNN to detect object parts. Therefore, different layers filters should be included to build a robust part-level prediction. Residual attention network [29] stacked attention modules inside networks like Inception and ResNeXt to produce attention-aware features. Each attention module (i.e., residual unit) consisted of a mask branch and trunk branch. The mask branch improved the trunk branch by applying top-down and bottom-up feedforward to weight output features. The trunk branch applied feature processing. The model proved that classification accuracy improved by adding more stacks of attention modules. Despite the accuracy improvement, there was a lack of complexity analysis to measure the cost of adding more residual attention stacks to the CNN.

Unlike previous attention mechanisms, Loss-based attention [30] did not add attention layers to CNN. It used the same CNN parameters to identify parts of the image that explain the CNN decision. The model connected with the CNN loss function by sharing parameters with fully connected layers. Moreover, it dropped the max-pooling layer to maintain spatial relationships among different patches. Furthermore, a new version of loss-function attention was proposed by replacing fully connected layers with two capsule layers. Experiments proved that loss-attention outperformed state-of-art networks in terms of classification accuracy, object localization, and saliency maps quality. A drawback of this method is that it could not locate multiple objects from the same class. Besides attention mechanisms in image classification, D-Attn [31] used text reviews to learn the features of users and items and predict their ratings. The model trained two CNNs, a user network, and an item network. Attention layers were added before convolutional layers in these networks. This dual architecture generated local attention maps for user preferences and item properties, and

global attention maps for the semantic of the entire user review. D-Attn improved the prediction accuracy and visualized words with high attention scores. A promising approach is to apply D-Attn to LSTM for long-range text reviews. Some studies replaced components of CNN architecture to improve their interpretability.

ALL-CNN [32] replaced max-pooling layers with increased stride convolutional layers. The size of the stride was set to 2×2 to reduce the network dimensionality. The authors argued that max-pooling could reduce overfitting and regularize the CNN, but it did not provide the desired result on small datasets. Moreover, they proved that using max-pooling layers was not essential for training large CNNs. The model used deconvolutional layers and guided backpropagation to generate saliency maps. However, choosing to drop or keep max-pooling layers is challenging as it depends on several factors, such as domain area, dataset, and network architecture. NIN [33] replaced convolutional layers and linear filters with a micro neural network. They argued that the level of spatial invariance in convolutional layers is low. The **micro convolutional layers** (i.e., **mlpconv layers**) had multiple fully connected layers with non-linear activation functions. NIN used the same approach of the convolutional layers window sliding. Therefore, each "mlpconv" layer used this approach to generate its feature map. After that, the averaged feature map was passed to the average pooling layer, and the output vector was sent to a SoftMax function. Their experiments proved that NIN had less accuracy than state-of-art networks, but its saliency maps were more interpretable. The experiments focused on classification accuracy and did not highlight the interpretability aspect. In addition, saliency maps were not evaluated in terms of class discrimination and object localization. CSG [34] replaced CNN filters with class-specific filters to avoid the overlapping of filters and classes. The model built a class-specific gate by assigning each filter in the last convolutional layer with one or more classes. They argued that transforming filters into a class-specific form could improve the interpretability of CNN decisions. They modified ResNet architecture to a CSG network and proved that it improved the classification accuracy, object localization, and saliency maps quality. Unlike previous models that focused on image classification, CSG evaluated the network robustness against adversarial examples. The classification drop for CSG was less than state-of-art networks. CSG model was evaluated on one type of CNNs (i.e., ResNet). Therefore, it is not evident if the model can be generalized across other types of CNNs. Attribute Estimation [35] added fully connected layers to CNN intermediate layers. The purpose was to apply attributes estimation to improve the interpretability of CNN. The task of generated attributes was to connect visual features with class information. Attribute Estimation improved the classification accuracy of the Inception-V3 network. However, adding extra layers and generating multiple attributes can impact the complexity of the neural network. Reducing the number of attributes should be carefully considered.

A different approach was to modify the CNN loss function to improve interpretability. Interpretable CNN [36] added the loss of feature map to all filters in the last convolutional layer. The purpose was to enforce each filter to encode distinct object parts. Therefore, this model did not require any annotations for object parts. Interpretable CNN outperformed state-of-art networks in terms of object localization and location instability. However, the single-class classification accuracy was lower than state-of-art networks. Therefore, there was a trade-off between accuracy and explainability in this model. Dynamic-K Activation [37] modified **stochastic gradient descent (SGD)** to interpret CNN. The model adopted a capsule NN EM routing approach and proposed an alternate optimization function called adaptive activation thresholding. The ResNet network was modified and trained using Dynamic-K Activation. Dynamic-K had a comparable classification accuracy and outperformed traditional ResNet in terms of interpretability and saliency maps quality. However, the Dynamic-K Activation model was evaluated on one network (i.e., ResNet). Therefore, it is not evident if the model can be generalized across other types of CNNs.

SAD/FAD [38] proposed spatial activation diversity loss functions to make CNN more discriminative. Two loss functions, **spatial activation diversity (SAD)** and **feature activation diversity (FAD)** were applied to two different CNNs to recognize faces. SAD loss function enhanced structured feature responses, while the FAD loss function made responses insensitive to occlusions. Visualizing the average location of the filter on the face image proved the high consistency of responses over various face poses. In this model, CASIA-Net and ResNet-50 were trained as branches of a Siamese network. By using combinations of networks as branches, the model can prove if it can generalize across other types of CNNs. FBI [39] proposed the forward-backward interaction activation loss function as a regularization function. This loss function helped CNNs to be more interpretable. Unlike traditional CNNs that performed only a forward pass, the FBI trained CNN by making forward pass, computing pass, and backward pass. In each pass, the sum of layer-wise differences between neuron activations was calculated. Qualitative experiments proved that the FBI enabled CNN to learn significant regions of the image. For quantitative experiments, the FBI had higher confidence and lower confusion than state-of-art networks. Moreover, the network computation for performing three passes could be significant. Therefore, conducting a complexity analysis for the FBI model can prove its effectiveness and generalization.

Another approach was to dissect the image to extract object parts semantics. AOG [40] proposed a graphical model using And-Or graphs to rearrange convolutional layers representations semantically. This model opened the black-box by adding four layers to the CNN, semantic part, part template, latent pattern, and CNN unit. The model was evaluated on two variations, three-shots AOG (i.e., three annotations), and AOG with more annotations. Experiment metrics included part detection, center prediction, localization accuracy, and prediction accuracy. AOG model outperformed state-of-art networks. The AOG model required a subset of annotated object parts. Selecting images and object parts to annotate can be challenging and time-consuming since it requires domain knowledge. Moreover, it is useful to conduct a complexity analysis for the AOG model since adding four layers to the CNN can increase its computation. ProtoPNet [41] proposed a prototypical part network to dissect the image and find prototypical parts before making the final classification. The model added a prototype layer between the convolutional layers and the fully connected layers. CNN learned the image prototypes during the training. In the end, each class was associated with a set of prototypes. The ProtoPNet classification accuracy was comparable with state-of-art networks. Moreover, class activation maps of ProtoPNet were finer with higher quality. However, a drawback of this model was the high number of generated prototypes. Therefore, ProtoPShare [42] was proposed to reduce the number of prototypes generated by ProtoPNet [41]. ProtoPShare applied a merge-pruning approach to share prototypes between classes. It had two stages, initial CNN training, and prototype pruning. In the pruning stage, prototypes with the same semantics were merged. Thus, this model succeeded in pruning up to 30% of generated prototypes without impacting CNN accuracy. The experiments proved that using a data-dependent similarity measure was more consistent than a data-independent measure (i.e., inverse Euclidean norm). A different approach for interpreting CNNs was to integrate their architecture with other machine learning models.

For example, the Explainer model added autoencoders to interpret intermediate layers of pre-trained CNNs [19]. The encoder received feature maps in intermediate layers and decomposed them into several object parts. After that, the decoder inverted decomposed feature maps into reconstructed feature maps. The model used a filter loss to enforce the representation of object parts through interpretable filters. Experiments showed that feature maps of the Explainer model were more interpretable than state-of-art networks. Moreover, the localization instability of the model was lower than other CNNs. However, the classification accuracy of this model was lower than traditional CNNs. Adding an autoencoder to intermediate layers of CNN could impact the network

computation. There is a need for complexity analysis to study the Explainer model computation. XCNN [20] was another model that employed autoencoders in CNNs. An autoencoder was used to find **regions of interest (ROI)** in an image. The XCNN model had two components: an autoencoder and a CNN classifier. The autoencoder generated interpretable heatmaps that were passed to a CNN classifier. XCNN heatmaps were evaluated qualitatively using class discrimination and quantitatively using object localization. Methods like LRP and Guided-Backpropagation proved the high quality of XCNN heatmaps. However, the classification accuracy of XCNN was less than state-of-art networks. Also, there is a need to measure the complexity of the XCNN model.

The **Adaptive Deconvolutional model (Adaptive DeConv)** [43] was proposed to decompose an image into feature maps and reconstruct the input image again. This model integrated deconvolutional layers with max-pooling layers. After that, it was combined with a CNN classifier for object recognition. Images were reconstructed in CNN intermediate or high layers. The Adaptive DeConv model outperformed state-of-art networks and improved the object recognition accuracy. However, identifying useful layers (i.e., intermediate vs. high) in reconstructing an image could be challenging. There was a lack of comparison between selecting intermediate features and high-level features. Additionally, machine learning algorithms were combined with CNNs, like in the **Deep Fuzzy Classifier (FCM)** [44]. The FCM model incorporated fuzzy logic to classify data points. A fuzzy classifier was added after the last convolutional layer. This classifier applied fuzzy clustering and Rocchio's algorithm on the feature map to extract class representatives. The FCM model could visualize the saliency of each pixel w.r.t the predicted class. FCM saliency maps were more interpretable than traditional CNNs. However, the FCM classification accuracy was less than state-of-art networks.

Table 1 shows a review of models which interpret CNNs by modifying their architecture. We can notice that the models in this category were intrinsic, model-agnostic, and local. They were intrinsic since they modified CNNs architecture in training and compared the modified CNN with the traditional CNN. They were model-agnostic since they could generalize across various architectures of CNNs and were local as they required access to the dataset.

4.2 Architecture Simplification

Explainable models in this category rely on the rule extraction approach to generate human interpretable rules. Another approach is to apply network distillation and compression by pruning redundant features. Previous studies interpreted CNNs by creating hybrid models and incorporating linear models in their architecture. For example, decision trees were attached to high-level features to decompose them into semantic object parts [45]. Decision trees quantified the contribution of each filter to the CNN output score. After that, each filter was connected with a semantic object part label. However, this model required the manual labeling of object parts in each filter to calculate their contribution. This labeling could be challenging in medical imaging applications where objects and parts are tissues and cells. Moreover, the model ignored features that could be activated in some scenarios. Moreover, linear classifiers were combined with each intermediate layer in CNNs like Deep KNN [46]. This hybrid model used the training data to measure the non-conformity of a prediction on a test input. This measurement guaranteed that intermediate layers in training were consistent with the CNN prediction. K-NN classifier was attached to each layer to detect training data points that were like the test image. After that, learned training data points were compared to CNN output in the test time to provide interpretability. Their experiments proved that the Deep KNN model provided more insights and was more robust than other traditional CNNs. However, adding a KNN classifier to each layer can impact the network computation. Therefore, there is a need for complexity analysis to prove that training CNN with attached KNN classifiers is feasible. Another approach was to maintain the linear models'

Table 1. An Overview of Models which Interpreted CNNs by Modifying their Architecture

Model	Methodology	Intrinsic	Post-hoc	Model-Agnostic	Model-Specific	Global	Local
GALA [27]	Embedded attention layers in CNNs to generate attention activity maps.	√	x	√	x	x	√
DomainNet [28]	Transformed pre-trained CNN to DomainNet and apply two attention levels for extracting object parts and features.	√	x	√	x	x	√
Residual Attention [29]	Stacked attention modules and integrate with CNNs to generate attention-aware features.	√	x	√	x	x	√
D-Attn [31]	Added attention layer before convolutional layer to learn local/global attentions for user reviews.	√	x	√	x	x	√
ALL-CNN [32]	Replaced max-pooling layer with convolutional layer and increased stride to reduce dimensionality.	√	x	√	x	x	√
NIN [33]	Replaced convolutional layers and linear filters with a micro network to enhance spatial invariance.	√	x	√	x	x	√
Interpretable CNN [36]	Added a loss of feature map to enforce each filter to encode distinct object parts.	√	x	√	x	x	√
AOG [40]	Added a graphical model to CNN to detect the semantic hierarchy of object representations.	√	x	√	x	x	√
CNN Explainer [19]	Added an autoencoder for each feature map to decompose object parts in the image and reconstruct feature maps.	√	x	√	x	x	√
Dynamic-K Activation [37]	Replaced stochastic gradient descent with adaptive activation thresholding to interpret the CNN.	√	x	√	x	x	√
XCNN [20]	Added Autoencoder before the CNN classifier to generate interpretable heatmaps.	√	x	√	x	x	√
Adaptive DeConv [43]	Combined a network of deconvolutional layers and max-pooling layers with CNN to decompose the image into feature maps, then reconstruct it.	√	x	√	x	x	√
CSG [34]	Combined class-specific gate with filters in CNN to assign each filter to one or more classes.	√	x	√	x	x	√
SAD/FAD [38]	Added SAD and FAD loss functions to CNNs to improve their discrimination in face recognition.	√	x	√	x	x	√
ProtoPNet [41]	Added prototype layer after last convolutional layer to assign object parts to various prototypes.	√	x	√	x	x	√
FBI [39]	Added new loss function to regularize CNN and improve its interpretability. It trained the CNN using three passes to learn important regions.	√	x	√	x	x	√
Attribute Estimation [35]	Added fully connected layers to CNN intermediate layers for generating attributes that enable interpretability of CNN.	√	x	√	x	x	√
FCM [44]	Added a fuzzy classifier layer after the last convolutional layer. The classifier applies clustering and Rocchio's algorithm to classify data points.	√	x	√	x	x	√
ProtoPShare [42]	Shared prototypes between classes to reduce the number of prototypes generated by ProtoPNet.	√	x	√	x	x	√
Loss Attention [30]	Removed max-pooling layer in CNN and added loss-based attention to identifying which parts of the image explain the CNN decision.	√	x	√	x	x	√

properties in CNN architecture. **Self-Explaining Neural Networks (SENN)** [47] applied a bottom-up mechanism to interpret CNNs. The model consisted of three components, a concept encoder, an input-dependent parametrizer, and an aggregation function. The input was transformed into a set of representative features, and relevant scores were calculated. After that, these scores were used to make the prediction. The experiments proved that the SENN model was robust, faithful, and intelligent. However, there was no evaluation for the SENN class discrimination, and

it lacked the classification accuracy comparison with state-of-art networks. Another hybrid approach was embedding clustering in CNNs to improve their interpretation. CNN-INTE [48] used meta-learning to generate meta-level test data. This model selected layers in CNN and applied clustering on two levels: base learning and meta-learning.

In base learning, the network was trained on original training data, while in meta-learning, the network was trained on predictions of base learning along with the true class of training data. Moreover, the overlap in the clustering plots indicated if the class was wrongly classified. However, finding the optimal clustering algorithm for generating meta-level data requires further analysis. Also, initializing clustering parameters could be challenging since it relies on the domain and the dataset context. Furthermore, different approaches were proposed to simplify the structure of CNNs and improve their interpretability. Examples of these approaches were network pruning, compression, and dissection. For the network pruning, extracting subnetworks was applied to detect semantics in CNN layers [49]. Pre-trained CNNs were pruned to produce subnetworks that connected CNN prediction with data features to improve interpretability. The subnetworks extraction was applied on two levels, sample, and class. The sample-specific subnetworks ensured that individual predictions were consistent with the CNN. The class-specific subnetworks measured the CNN prediction on a single class. Meanwhile, the sample-specific subnetwork applied hierarchical clustering to reflect input patterns. The class-specific subnetworks produced saliency maps to interpret the prediction. Applying hierarchical clustering can be computational. Therefore, other clustering algorithms can be considered, like K-means. Moreover, selecting the number of clusters can be challenging when interpreting deep CNNs and large datasets.

For the CNN network compression, the CAR model [50] was proposed to make CNN smaller and more interpretable. The CAR model compressed pre-trained CNNs by pruning filters with insignificant contributions to the CNN prediction. Similar visual filters in each layer were grouped into subsets like shape-based and color-based filters and were ranked based on their CAR importance index. After that, visual filters with low CAR index (i.e., redundant) were identified and pruned. This pruning process improved the prediction accuracy for pre-trained networks like AlexNet. Experiments showed that the CAR network outperformed state-of-art networks, improving classification accuracy by 16%–25%. Furthermore, CAR^c index was proposed to enhance the interpretability of pre-trained networks. CAR^c index highlighted the importance of each filter w.r.t the class label c. Visualizing layer 5 filters of AlexNet proved that filters with highest CAR^c index frequently appear in predicted classes (e.g., smooth curvature filter appears in top classes such as a steep bridge or soup bowl). The CAR model had a greedy approach by pruning all filters in CNN. A promising approach is to build a selective compression model that prunes filters based on a given criterion. Moreover, CNN network dissection was used to extract intermediate layers semantics [51]. The model used the Broden dataset that has a ground truth set of visual concepts. The model collected CNN intermediate layers responses to these visual concepts. After that, CNN layers were quantified by applying binary segmentation against visual concepts. This model required no training as the dissection was applied after training (i.e., post-hoc). Their experiments proved that deeper networks had better interpretability, and factors like dropout and batch normalization could affect CNN interpretability. However, this model heavily relied on the visual concepts of the Broden dataset. Therefore, the poor quality of visual concepts can impact the level of interpretability. An interesting simplification approach is LIME [52]. This model is general in terms of architecture and tasks. It was applied to tasks like text and image classification. It simplified CNN by generating feature analysis visualization. For text classification, LIME visualized each feature's positive and negative contributions to improve CNN interpretability. In image classification, the model highlighted pixels that contributed to class prediction. A promising approach is to utilize

Table 2. An Overview of Models which Interpreted CNNs by Simplifying their Architecture

Model	Methodology	Intrinsic	Post-hoc	Model-Agnostic	Model-Specific	Global	Local
Decision Trees [45]	Decomposed high-level features into object parts by using a decision tree to calculate filters numerical contribution.	**x**	√	√	x	x	√
SENN [47]	Interpreted CNN during training by transforming input to a set of interpretable features and combining transformed features with their relevant scores to make a prediction.	√	x	√	x	x	√
Deep KNN [46]	Combined KNN classifier with each layer to measure non-conformality of a prediction in the training stage.	√	x	√	x	x	√
CNN-INTE [48]	Applied clustering on hidden layers to generate meta-level test data and learn classifier results.	x	√	√	x	√	**x**
Subnetwork Extraction [49]	Extracted semantic information for CNN layers by pruning unimportant channels. Subnetworks are extracted on sample and class levels.	x	√	√	x	x	√
CAR [50]	Pruned all filters with the insignificant contribution in a greedy way to make CNN smaller and more interpretable.	√	x	√	x	√	**x**
Network Dissection [51]	Extracted semantics of intermediate layers by relying on Broden dataset visual concepts.	x	√	√	x	x	√
LIME [52]	Provided positive/negative contributions of features in text classification to improve interpretation. Highlighted pixels with significant contribution to class prediction in image classification to improve interpretation.	x	√	√	x	x	√

parallel processing platforms to deploy LIME in real-time applications. Table 2 shows a detailed review of models that interpreted CNNs by simplifying their architecture.

4.3 Feature Relevance

Models in this category rely on ranking the importance of features against the CNN prediction. Their feature space analysis improves interpretation by identifying significant features. Previous studies searched for features in CNN layers and grouped them using techniques like clustering and similarity measures. For example, the EBANO model [53] clustered hyper columns selected from high-level layers using K-means. Each clustered group of pixels identified an interpretable feature. After that, interpretable features were used to perturb the input image passed to a pre-trained CNN. The network classified the perturbed image and provided useful transparency details. IR and IRP indices were used to evaluate the EBANO model. The IR index calculated the probability of the class in the original image w.r.t the perturbed image. In comparison, the IRP index calculated the influence of each feature on the set of classes. However, initializing the value of k in the k-means algorithm can be challenging for medical images and large datasets. In addition, other clustering algorithms can be applied as an alternate. Another similar approach was to use k-nearest observation for measuring the similarity of stored features [54]. This model trained a CNN to detect features in the first pooling layer and store them in a database. After that, the test image features were extracted using the same CNN and compared with the features database. The similarity was measured using k-nearest observation with cosine and Euclidean distance. Experiments proved that cosine with $k = 3$ achieved the highest classification accuracy for the model. A drawback of this model was the features extraction in low levels (i.e., first pooling layer), and the ignorance of high-level features with more semantics. Moreover, features were stored in the database without being ranked, which levels their contribution. DGN-AM model [55] synthesized images to identify features learned by neurons. The model used a **deep neural network (DNN)**

to generate images similar to the real image. After that, it applied backpropagation using the generated image to search for the neuron with maximum activations. The experiments proved that the DNN network could generalize across different types of datasets. Moreover, DGN-AM proved to enhance the CNN ability to learn features on the neuron level. However, searching for neurons with maximum action can be challenging because of the computation and the similarity in deep space. In addition, DGN-AM could only visualize features properly if the images were canonical.

Other studies visualized pixels' contribution to the CNN prediction. The LRP model [56] decomposed the output on the feature and pixel levels. It applied layer-wise backpropagation and Taylor-type decomposition to redistribute each neuron's contribution and calculate the features/pixels relevance scores. The generated heatmaps corresponded to the pixel's contribution w.r.t the predicted class. LRP was evaluated qualitatively by visualizing the saliency maps. However, there was a lack of quantitative evaluation, like object localization and faithfulness. Integrated gradients model [57] argued that LRP broke the implementation invariance by using discrete gradients and backpropagation. Therefore, integrated gradients proved to satisfy CNN sensitivity to capture relevant features and implementation invariance. The model was generalized by identifying path models. The integrated gradients model was used in multiple applications like object recognition, diabetic retinopathy detection, and question classification. The saliency maps were clearer than other gradient models. However, there was a lack of quantitative evaluation, like localization and faithfulness.

The DeepLIFT model [58] was proposed to decompose CNN prediction w.r.t the input image by backpropagating the features' contribution. The model argued that LRP suffered from gradients saturation issue since it applied elementwise product between gradients and input. Moreover, the model argued that the Integrated gradients model was highly computational when extracting high-quality integrals. Therefore, DeepLIFT relied on domain knowledge to select the reference input. The experiments proved that DeepLIFT outperformed gradient and Integrated gradients models in terms of saliency maps quality. However, DeepLIFT saliency maps were not evaluated in terms of object localization and faithfulness. A different approach was to attach a feedback CNN to the original CNN to reconstruct features in a hierarchical mode [59]. The **feature extraction and reconstruction CNN (FER-CNN)** built a response field reconstruction by finding the activity of a neuron w.r.t other neurons. Then, it applied feature interpolation by clustering features at a layer and storing clusters in the response field. The FER-CNN had two networks, an encoder for extracting features (i.e., original CNN), and a decoder for reconstructing features (i.e., feedback CNN). The results proved that its saliency maps outperformed LRP in their quality. Moreover, FER-CNN outperformed other neural networks in classification accuracy. However, initializing hyperparameters for encoder and decoder CNNs could be challenging. Furthermore, it is hard to choose the combination of CNNs architectures in terms of layers and networks. Table 3 shows a detailed review of models which interpreted CNNs by applying feature relevance.

4.4 Visual Explanation

Models in this category interpret CNNs by generating saliency maps or class activation maps.

4.4.1 Saliency Maps. Explainable models in this category generate heatmaps (i.e., saliency maps) to interpret the CNN prediction. They learn features contributions to the prediction w.r.t each pixel in the image. A saliency maps model [60] was proposed to rank the input image pixels by relying on their influence on the gradients' score. This gradient-based model calculated gradient scores w.r.t the input image by applying backpropagation. The saliency maps were visually evaluated for various classes. However, saliency maps' quality and color segmentation lacked the quantitative evaluation of the object localization. Moreover, saliency maps were noisy as it

Table 3. An Overview of Models which Interpreted CNNs by Applying Feature Relevance

Model	Methodology	Intrinsic	Post-hoc	Model-Agnostic	Model-Specific	Global	Local
EBANO [53]	Clustered hyper columns in high-level layers to identify interpretable features in the image.	x	√	√	x	x	√
Feature Similarity [54]	Extracted similar features from the training database by applying cosine and Euclidean distance measures.	√	x	√	x	x	√
DGN-AM [55]	Synthesized image similar to the input image and applied backpropagation on it to search for neurons with maximum activations.	√	x	√	x	x	√
Integrated Gradients [57]	Combined gradients implementation invariance with sensitivity to identify important features w.r.t the input pixels.	x	√	√	x	x	√
FER-CNN [59]	Attached a feedback CNN to the original CNN to reconstruct features in a hierarchical approach.	x	√	√	x	x	√
LRP [56]	Decomposed CNN output prediction into feature/pixel relevance scores by applying layer-wise backpropagation and Taylor-type decomposition.	x	√	√	x	x	√
DeepLIFT [58]	Decomposed CNN output prediction w.r.t the input by backpropagating through each feature in the input and choosing an input reference.	x	√	√	x	x	√

was challenging to localize captured objects. The deconvolutional network model (Deconv) [61] adopted a top-down approach to synthesize the image based on the reconstructed feature maps of a specific layer. This generative model consisted of three layers of feature maps. The first layer learned Gabor-style filters, the second layer learned V2-like elements, and the third layer learned high-diverse features.

4.4.2 Class Activation Maps. The **Class Activation Map (CAM)** [62] was proposed to interpret CNN prediction by generating activation maps (i.e., heatmaps). The CAM model modified the CNN architecture by adding a **global average pooling layer (GAP)** instead of the fully connected layer. The GAP layer calculated the average contribution of each feature map in the last convolutional layer. After that, it weighted the sum of vectorized averages to generate the final activation map. In the end, the CAM model overlayed the activation map on the input image to identify the areas of interest the CNN used to make its prediction. The CAM drawback was the architecture modification which impacted the prediction accuracy. Therefore, the Grad-CAM model [63] was proposed to overcome the drawbacks of CAM. The model maintained the fully connected layer and calculated the gradients of a predicted class in the last convolutional layer. The model proved to be more general since it did not change the CNN architecture. The Grad-CAM model captured features that positively influenced the class prediction since negative features were irrelevant to the class. The qualitative and quantitative experiments proved that Grad-CAM outperformed other gradient-based models. A drawback of the Grad-CAM model was the inability to capture multiple objects of the same class. Afterward, different modifications and extensions to the Grad-CAM model were proposed to enhance object capturing and class discrimination. The Grad-CAM++ [64] model was presented as a pixel-wise gradient-based approach. The model calculated gradient weights of pixels instead of features. Similar to Grad-CAM, this model calculated gradients in the last convolutional layer with respect to the image provided as input. The qualitative and quantitative experiments proved that Grad-CAM++ performed better than Grad-CAM in terms of faithfulness, human trust, and object localization. Furthermore, to improve object capturing and localization, the Smooth Grad-CAM++ model [65] was proposed. This model combined SMOOTHGRAD [66] and Grad-CAM. The model used Gaussian noise to add noise to the input image. After that, it

took the noised images and calculated their average gradients to generate the activation map. Unlike previous gradient-based models, the Smooth Grad-CAM++ could generate saliency maps for selected feature maps or neurons in any CNN layer. Saliency maps were produced for various layers and neurons. However, no quantitative evaluation like faithfulness and object localization was conducted. Another proposed variation of Grad-CAM was Augmented Grad-CAM [67]. This model adopted the image augmentation approach to generate different versions of the image provided as input. Each image was rotated and translated with a slight angle. The Augmented Grad-CAM proved that each augmented image carried some useful spatial information. Therefore, every augmented image generated a unique activation map. In the end, the augmented activation maps were combined in order to produce the final activation map. The experiments proved that Augmented Grad-CAM outperformed Grad-CAM in weakly object localization. However, the generation of activation maps for every augmented image could be highly computational. Therefore, there is a need for complexity analysis to address the feasibility of the Augmented Grad-CAM model.

Another gradient-based model was U-CAM [68]. This model was proposed to utilize uncertainty loss to improve the quality of saliency maps. The model was applied in **visual question answering (VQA)** to reduce model and data uncertainty. An attention network was added to the LSTM for calculating uncertainty loss and combining it with the cross-entropy loss. After that, gradients were calculated w.r.t the loss functions and corresponding features. The gradients were used to produce the class activation map. U-CAM outperformed other gradient-based models in ablation analysis (i.e., uncertainty reduction) and saliency map quality. Moreover, U-CAM improved the VQA network accuracy for all datasets. However, the model was not compared with other gradient-based models in applications like image classification. Also, the U-CAM model saliency maps were not evaluated in terms of localization and faithfulness.

In addition, Eigen-CAM [69] is a class activation map model that relies on extracted features rather than a classification network. It did not apply gradients propagation and visualized principal components of learned features. Eigen-CAM visualizations outperformed Grad-CAM in capturing multiple objects in the same image. Moreover, the class activation maps could localize objects even when the CNN misclassified the prediction. In weakly supervised localization, Eigen-CAM had a lower IoU error rate than Grad-CAM and backpropagation models. The model proved to be more robust against perturbed images produced by the DeepFool algorithm. However, the model was not evaluated in terms of faithfulness and human trust. Due to the lack of semantic descriptions in gradient-based models, the IBD model [8] was proposed to generate labeled heatmaps with corresponding probabilities that rank them from highest to lowest. This model incorporated semantic description in class activation maps using interpretable basis decomposition. The model generated heatmaps along with labels and rankings. This post-hoc model decomposed predicted class vectors into interpretable vectors. After that, it associated each activation map (i.e., basis vector) with labels and rankings. The labels were extracted from the Broden dataset, which has a set of object parts visual clues. The qualitative experiments on IBD proved that it could provide useful insights into CNN prediction. Moreover, the human study showed that IBD visualizations were more reasonable than Grad-CAM visualizations. A drawback of IBD is label extraction; finding appropriate labels in medical images is challenging. Also, the class activation maps were not quantitively evaluated.

4.4.3 Masking Visualizations. Unlike previous gradient models, the Score-CAM model [70] adopted a masking approach. This model argued that using gradients could have some limitations like gradients saturation and false confidence. The saturation issue could produce noisy saliency maps, while the false confidence was related to the fact that high weights of gradients did not

necessarily reflect the contribution to the class prediction. Therefore, Score-CAM relied on the increase of confidence metric and forward passing approach to generate class activation maps. After that, activation maps were upsampled to fit the input image size. Then, each activation map was multiplied with the input image to generate masked images, which were passed to CNN to calculate the scores. After that, calculated scores were linearly combined with their relevant activation maps to generate the final activation map. The qualitative and quantitative experimentation showed that Score-CAM performed better than Grad-CAM when compared for class discrimination, faithfulness, and object localization. We have proposed a method called Augmented Score-CAM [71], built on top of the existing Score-CAM [70]. Augmented Score-CAM adopted the image augmentation approach by producing augmented class activation maps and merging them into one activation map.

A similar masking approach to Score-CAM [70] was Mask [72]. This perturbation model interpreted CNN prediction by identifying significant input regions. Mask considered explanations as meta-predictors and predicted the behavior of CNN toward certain inputs. The model applied three techniques, replacing the input region with a fixed value, adding noise to the input image, and blurring portions of the input image. The quantitative experiments proved that Mask outperformed CAM, Grad-CAM, and Occlusion in terms of robustness, localization error, and pointing game. Moreover, the model could capture small areas that significantly impacted the CNN prediction. However, Mask lacked the evaluation of human trust and faithfulness. Also, there was no comparative analysis for various masking techniques.

4.4.4 Intrinsic Visualizations. The visualization models discussed earlier were post-hoc that attached an auxiliary part to interpret the CNN. However, some models interpreted CNNs by modifying their architecture. The Teacher-Student model [73] used Autoencoders to explain the important regions in a classified image. The model had two networks, one for encoding input image representations, and one for reconstructing an image with the same input image size. The model used the reconstructed image to visualize important parts of the classified image by using a binary threshold. The experiments proved that the model visualizations could identify plant disease symptoms. Moreover, this model outperformed Grad-CAM and LRP models in terms of visualizations' sharpness and perturbation curve metrics. However, the computation cost of the model was high since it applied two networks for reconstructing and visualizing important parts in plant disease classification. Furthermore, the HPnet model [74] used hierarchical prototypes for interpreting the CNN image classification. The model attached prototype layers to each parent node in the CNN; these layers were used in training to generate a set of prototypes. After that, the generated prototypes were distributed over classes in the fully connected layer. HPnet saliency maps could classify objects like forklifts by capturing important prototypes like wheels. However, HPnet classification accuracy was less than VGG16 for fine-grained and coarse-grained metrics.

Besides image classification, some visualization models interpreted CNNs in other applications. For example, the Equalizer model [75] identified bias in image captioning applications. The model used two methods to mitigate gender-biased descriptions for images, appearance confusion loss and confidence loss. The appearance confusion loss forced CNN to predict when the gender features were absent. In contrast, the confidence loss forced CNN to predict gender when its features were evident. The gender features were the ground truth for the two methods and were applied to each image. The experiments proved that the Equalizer model outperformed state-of-art networks in terms of classification error rate, gender ratio error, and pointing game metrics. However, annotating each image's ground truth could be challenging for image captioning in areas like race and age bias. It is hard to find useful features that distinguish each group in these areas. Other models visualized heatmaps for VQA answers to justify their outcome [76]. They applied guided

backpropagation, a modified saliency map that removes gradients with a negative contribution to the VQA prediction. Heatmaps could justify the answers of the VQA model. However, it was not evident if the VQA heatmaps were efficient in terms of object localization and faithfulness.

An interesting approach to visualizing the internal representations of CNNs was CNNV [77]. This model built **acyclic directed graphs (DAG)** to explain CNNs. It proposed a DAG interactive visualization to uncover the CNN internal layers. CNN layers and neurons were clustered to simplify the visualization for deep networks. This visualization could help provide features learned by neurons, explain features' evolution through layers, and debug CNN when having an issue during the training. The CNNV was evaluated by building a customized network (Base-CNN) with four convolutional layers and two fully connected layers. CNNV provided useful visualizations for low-level and high-level features in various layers. However, it was hard to generalize the model to other CNNs since it could be challenging to visualize each layer and neuron for deep networks. Moreover, the DAG visualization is specific for machine learning experts with a good background in the architecture of CNNs. Finally, the model applied multiple clustering algorithms like K-means, MeanShift, and hierarchical clustering. Initializing the optimal parameters for these algorithms requires collaboration with domain experts. Table 4 shows a detailed review of models which interpreted CNNs by visualization.

4.5 Taxonomy Correlations Analysis

To analyze the trend of XAI in convolutional neural networks, we visualized the flow among various categories. We analyzed our taxonomy in Section 4 w.r.t the XAI categories like the scope (global vs. local), structure (intrinsic vs. post-hoc), and dependence (model-agnostic vs. model specific). Our taxonomy had the following categories: architecture modification, visualization, simplification, and feature relevance. We believe correlations between taxonomies can provide useful insights into the research direction of interpreting convolutional neural networks. In Figure 5, the nodes on the left represent our taxonomy, and the nodes on the right represent the XAI categories. The thickness of the link between two nodes represents the number of models. A thicker link means more models exist in these two nodes. In terms of architecture modification, we can notice that XAI models were distributed equally between intrinsic, local, and model-agnostic categories. This means that XAI models that interpreted CNNs by modifying their architecture had to access the dataset (i.e., local) and generalize across various CNNs (i.e., model-agnostic). Moreover, all models in this criterion were intrinsic since they had to modify the CNN architecture to improve its interpretation. In the simplification, the models were distributed in terms of structure (i.e., intrinsic vs. post-hoc). However, most simplification models were local and interpreted CNNs by accessing the dataset except CNN-INTE [48] and CAR [50] models. These two models ignored the dataset and applied clustering and pruning to produce simpler CNNs. Moreover, simplification models could generalize across various CNNs (i.e., model-agnostic). In the feature relevance, most models were post-hoc and relied on features' importance to interpret CNNs without changing their architecture.

Moreover, models in this criterion accessed the dataset (i.e., local) and generalized across various CNNs (i.e., model-agnostic). In the visualization, most XAI models were post-hoc except for two models: Teacher-Student [73] and HPnet [74]. Therefore, visualization models assume the network is trained and tend to interpret CNNs by adding auxiliary parts. Moreover, visualization models were local since they accessed the dataset. Most visualization models could generalize across CNNs except the CNNV model [77], which heavily relied on a customized CNN architecture to build the acyclic directed graph. Overall, XAI models that interpreted CNNs used to be local since they accessed the dataset (i.e., input image). Additionally, these models could generalize across CNNs with various layers, neurons, and hyperparameters.

Table 4. An Overview of Models which Interpreted CNNs by Visualization

Model	Methodology	Intrinsic	Post-hoc	Model-Agnostic	Model-Specific	Global	Local
Saliency Maps [60]	Ranked the input image pixels by calculating gradients' score w.r.t the output class.	x	√	√	x	x	√
Deconv [61]	Reconstructed input from feature maps of a selected CNN layer.	x	√	√	x	x	√
CAM [62]	Added global average pooling layer to calculate feature maps contribution in the last conv. layer.	x	√	√	x	x	√
Grad-CAM [63]	Calculated positive gradients in the last conv. Layer w.r.t the output class.	x	√	√	x	x	√
Grad-CAM++ [64]	Calculated gradient weights for pixels on the last conv. layer w.r.t the output class.	x	√	√	x	x	√
Smooth Grad-CAM++ [65]	Generated multiple noised images and calculated average gradient weights.	x	√	√	x	x	√
Augmented Grad-CAM [67]	Generated augmented images and applied Grad-CAM on each augmented image before combining all activation maps.	x	√	√	x	x	√
Score-CAM [70]	Applied increase of confidence to extract activation maps and masked the input image with extracted activation maps to produce final heatmap.	x	√	√	x	x	√
Equalizer [75]	Identified bias in image captioning by adding new loss functions during the CNN training.	x	√	√	x	x	√
CNNV [77]	Visualized learned features in CNN layers by applying acyclic directed graph.	x	√	x	√	x	√
Teacher-Student [73]	Identified important regions in the image by applying an autoencoder for reconstructing the input image.	√	x	√	x	x	√
Eigen-CAM [69]	Extracted learned features by visualizing principal components.	x	√	√	x	x	√
IBD [8]	Added semantic description to generated activation maps along with their labels and ranking.	x	√	√	x	x	√
Mask [72]	Identified important regions in the input image by masking it using noising and blurring techniques.	x	√	√	x	x	√
Hpnet [74]	Extracted a hierarchy of prototypes to describe the relations between class activation maps and their class category.	√	x	√	x	x	√
U-CAM [68]	Added an attention network to LSTM to calculate uncertainty loss, minimize uncertainty, and improve the CNN interpretability.	x	√	√	x	x	√

5 REVIEW OF XAI EVALUATION METRICS

5.1 Model-centric Metrics

Model-centric metrics evaluate explanations that use a model to explain a given task, like image classification, VQA, and image captioning. Quantitative and qualitative metrics measured the explanatory power of the XAI model.

5.1.1 Visualization. Producing visual interpretations was the most frequent metric used to evaluate the XAI model's performance. This metric expressed the qualitative human trust in the CNNs interpretations. Some models visualized feature maps and filters in different layers like in NIN [33], SAD/FAD [38], and CAR [50]. For example, CAR [50] improved the interpretability of pre-trained networks by proposing CAR^c index which ranked filters based on their importance w.r.t the

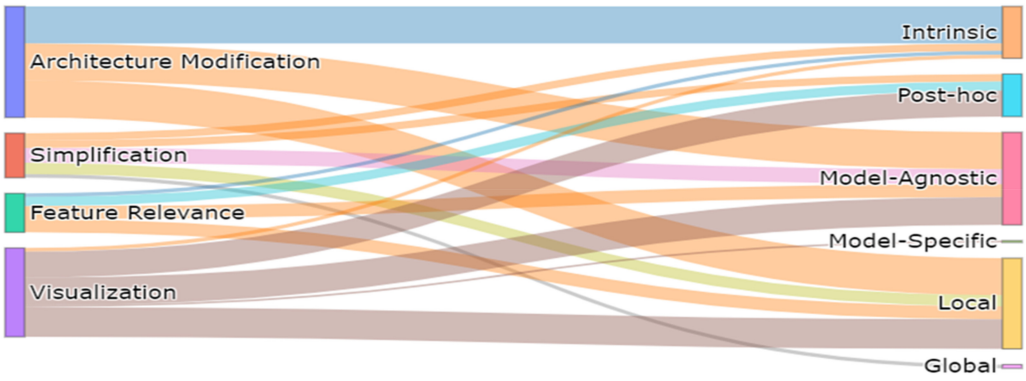

Fig. 5. Correlations of XAI models for various taxonomies.

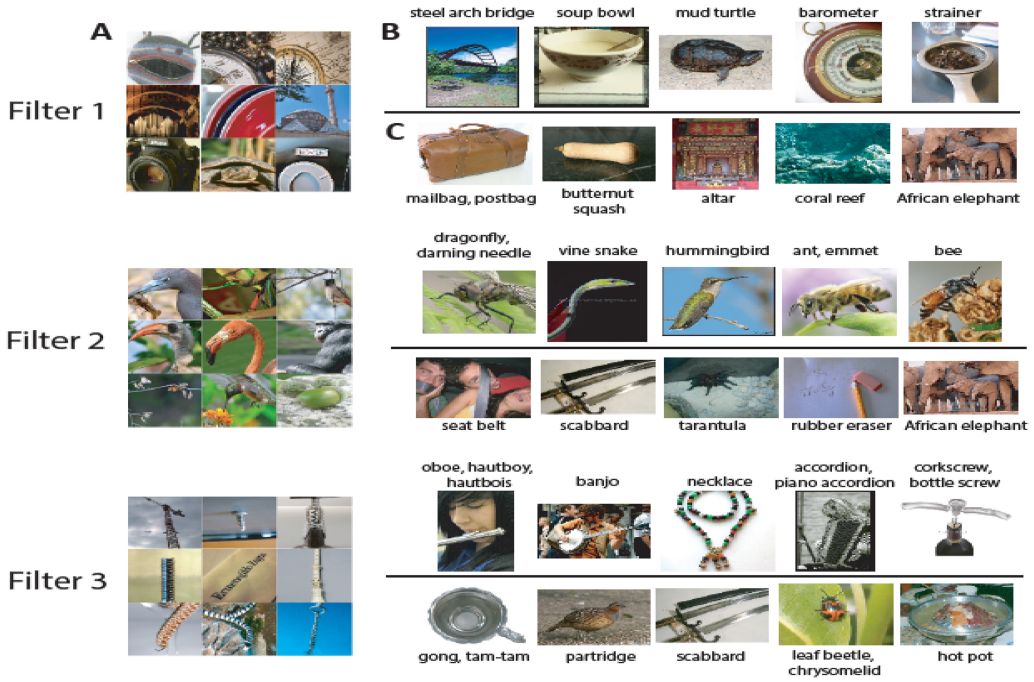

Fig. 6. AlexNet Layer 5 filters with highest CAR^c index [50].

predicted class. Figure 6(A) shows image patches of three filters with highest CAR^c index. Figure 6(B) shows the top five classes, while Figure 6(C) shows the bottom five classes. We can observe that curvature in filter 1 appeared in top classes like steep bridge and soup bowl. Meanwhile, curvature appeared less in classes like an altar and coral reef. Filter 2 appeared more in classes with bird and insect heads. Filter 3 appeared more in classes with long tools like banjo and oboe.

Other models visualized class activation maps to evaluate the class discrimination like Dynamic-K [37], XCNN [20], ProtoPNet [41], FBI [39], FCM [44], Loss Attention [30], SENN [47], Subnetwork [49], CAM [62], Grad-CAM [63], Grad-CAM++ [64], Smooth Grad-CAM++ [65], Augmented Grad-CAM [67], Score-CAM [70], IBD [8], HPnet [74], and U-CAM [68]. Additionally, saliency

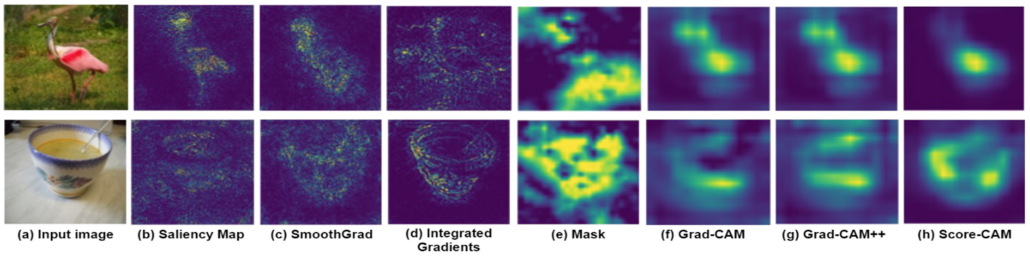

Fig. 7. Various visualizations for single object classification [70].

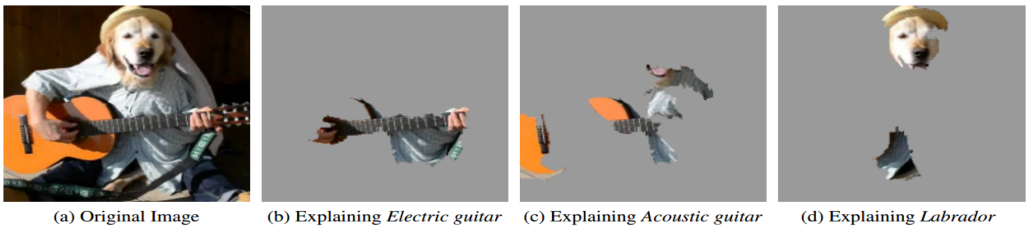

Fig. 8. Pixels contributing to the prediction of three classes in LIME [52].

maps were visualized to reconstruct the input image based on the pixels/features influence on the CNN decision like in Integrated Gradients [57], FER-CNN [59], LRP [56], DeepLIFT [58], Saliency maps [60], Deconv [61], and Mask [72]. Figure 7 shows various saliency maps and class activation maps for single object images. We can notice that the first three visualizations from the left belong to saliency maps [60], SMOOTHGRAD [66], and Integrated Gradients [57]. These XAI models rank the important pixels in the input image by applying backpropagation and building sensitivity maps. However, an apparent noise level appears in their sensitivity maps. This level of noise can impact class discrimination in the input image. Therefore, XAI models like Grad-CAM [63], Grad-CAM++ [64], Mask [72], and Score-CAM [70] were proposed to overcome this issue and enhance object localization. These XAI models shown in the last four images utilize feature maps to highlight important regions that contribute significantly to the network decision. Despite adopting different approaches to generate activation maps (i.e., gradients vs. masking), it is evident that these XAI models improved the object localization and were class discriminative. Furthermore, activation graph plots in CNN-INTE [48] were provided to test if an instance was wrongly classified. Interpretable units at different layers in CNN Dissection [51] were visualized to check if participants could recognize high-level visual concepts.

LIME [52] followed a masking approach to visualize significant pixels that contributed the most to the CNN decision. Figure 8 shows the pixels which contributed to the prediction of the top three classes, "Electric Guitar", "Acoustic Guitar", and "Labrador". The grey areas in images represent unimportant pixels. We can notice in Figure 8(b) that LIME relied on the fretboard to decide that the input image was for an "Electric Guitar". Meanwhile, in Figure 8(c), LIME relied on the dog's face to decide that it was a "Labrador". CNNV [77] provided a visual design for CNN learned features in low-level and high-level layers. Moreover, binary threshold visualization was generated to detect regions that could be a symptom of plant disease in the Teacher-Student model [73].

5.1.2 Localization. This metric was used to evaluate the ability of XAI models to capture most parts of the classified object. Most models applied the **Intersection over Union (IoU)** metric, which compared the object captured proportion with the ground truth label. The larger the IoU

Fig. 9. Intersection over Union (IoU) metric [78].

value was, the better localization the model could achieve. The bounding box was calculated using a threshold of 15% and drawing a rectangle around the largest segment of the binarized mask. The IoU metric was used to evaluate captured objects and parts in studies like in Interpretable CNN [36], AOG [40], CNN Explainer [19], Dynamic-K Activation [37], XCNN [20], Loss Attention [30], Subnetwork Extraction [49], CAM [62], Grad-CAM [63], Augmented Grad-CAM [67], Mask [72], and Eigen-CAM [69]. In Figure 9, we can see the bounding boxes plotted to localize an object. The left image in the figure shows two bounding boxes, a ground-truth bounding box (green) and a prediction bounding box (red). The ground truth bounding box is manual labeling that correctly locates the object (i.e., stop sign). However, the prediction bounding box is generated by the XAI model. The IoU metric is applied to calculate the difference between the two bounding boxes by identifying the area of overlap and the area of union. The high value of IoU, shown in the vehicle images (i.e., IoU of 0.7980 and 0.7899), proves that the two bounding boxes overlap significantly. Thus, the XAI model captured a high portion of the vehicles in both images. Some models like Score-CAM [70] adopted an energy-based approach to measuring how much energy of the saliency map lies within the bounding box. The image was binarized with 0 and 1 values based on the region (i.e., inside vs. outside the bounding box). After that, the binarized image was multiplied with the saliency map to extract the amount of energy.

5.1.3 Robustness. Evaluating robustness in the literature can fall under two categories: resistance against noised models or data, and resistance against adversarial attacks. In resistance against noise, the intrinsic Residual Attention [29] classification error was compared to state-of-art networks to check if the modified CNN improved the original network accuracy. Furthermore, Grad-CAM [63] could still localize the object when perturbing the input image. Consequently, this XAI model was robust against adversarial noise. Grad-CAM could generate an activation map that localized the object despite the misclassification of the input image. In adversarial resistance, adversarial examples such as FGSM, BIM, and C&W were applied to perturb the input image. For example, CSG [34] added a class-specific gate in the last convolutional layer that made the CNN more robust against white-box adversarial examples. Deep KNN [46] model combined the K-NN classifier with each layer in the CNN. This simplification allowed the XAI model to detect perturbed images and provide insights into the adversarial attack. Subnetwork Extraction [49] proved to be robust against multiple adversarial examples. Eigen-CAM [69] was more robust against DeepFool attacks since it relied on feature extraction, not the CNN architecture. Mask [72] could detect the difference between learned masks in clean and adversarial images.

5.1.4 Classification Accuracy. Classification accuracy is a quantitative metric that was extensively applied to intrinsic XAI models. For example, DomainNet [28], Residual Attention [29], NIN [33], AOG [40], Dynamic-K Activation [37], CSG [34], ProtoPNet [41], Attribute Estimation [35], Loss Attention [30], Subnetwork Extraction [49], CAR [50], and FER-CNN [59] proved that the existence of the XAI models lowered the test and validation error compared to traditional neural networks like VGG-16, VGG-11, AlexNet, ResNet-50, Inception-V3. In contrast, some XAI models

sacrificed the CNN accuracy for improving the interpretation like in FCM [44], CNN Explainer [19], ProtoPNet [41], ProtoPShare [42], Decision Trees [45], and Hpnet [74].

5.1.5 Other Metrics. Some XAI models applied other metrics for evaluation. For example, Interpretable CNN [36] and CNN Explainer [19] used location instability metrics to evaluate convolution filter interpretability. This metric supposed that the distance between inferred object part and a given landmark should not change across images. In addition, CSG [34] applied **mutual information score (MIS)** to calculate the correspondence between filter activations and class prediction. In face recognition, the SAD/FAD [38] model applied verification and identification quantitative metrics to evaluate the performance on face occlusion datasets. Despite using multiple factors to evaluate interpretability, Network Dissection [51] model quantified the measurement of interpretability by aligning the individual hidden units with human interpretable concepts. Similarly, LIME [52] quantified trust by calculating precision/recall for the model's human selection. In addition, LIME measured usefulness by providing insights into detecting CNN biased decisions. The EBANO [53] model applied IR and IRP indices. The IR index calculated the probability of real class in the original image w.r.t the perturbed image. In comparison, the IRP index measured the influence of each feature on all classes. DGN-AM [55] applied dataset generalization metric to prove that the model can analyze learned features and synthesize images similar to the input image on different datasets. Furthermore, other metrics were quantified, like faithfulness. Grad-CAM [63], Grad-CAM++ [64], and Score-CAM [70] measured the visualization faithfulness by calculating the classification drop/increase when the input image was masked with the activation maps. The sanity check metric was applied to ensure that the class activation map is sensitive to the model and data randomizations. Equalizer [75] applied the gender ratio error metric in image captioning to calculate the ratio of sentences that belong to "woman" or "man". Teacher-Student [73] applied **Area Over perturbation curve (AOPC)** metric to measure the CNN classification drop while erasing important pixels from the input image.

5.2 Human-centric Metrics

Human-centric metrics involve human subject experiments where lay humans evaluate explanations' quality. Two types of human-centric metrics are discussed, subjective and objective metrics. Subjective metrics measured human trust in explanations, while objective metrics measured the behavioral state of humans and task performance.

5.2.1 Subjective Metrics. Trust was defined as "guarantee required by the trustor that the trustee will act as it is expected to do without any supervision" [79]. Apparently, previous definitions imply the interaction between two parties, a trustor, and a trustee. The trustee is the party who is trusted, and the trustor is the party who trusts. Previous studies investigated the effect of adding explanations to clinical decision-support systems to improve trust and reliability in their decisions [80]. Their experiments stated the demand for a proper balance between comprehensive and selective explanations. Similarly, a deep learning tool called SMILY was proposed to allow pathologists to search for a medical image in the query box [81]. After that, the tool shows similar images from the database along with their diagnosis. Their experiments proved that the tool increased physicians' trust in the medical system. Moreover, trust was measured in different levels of decision-support systems like high-stake and low-stake systems [3]. Each level had three types of explanations, a black-box where no explanations were provided, a grey box where post-hoc explanations were provided, and a white-box where decision trees were used as a self-explanatory explanation. For a given level, the user had to decide and express the degree of trust based on a 5-point Likert scale.

5.2.2 Objective Metrics. The response time metric was investigated when users performed problem-solving tasks using models like decision trees and propositional rules [82]. Experiments proved that decision tables' response time was the least as users found them the easiest to perform the task. Decision sets were proposed like sets of if-then rules [83]. The interpretability of decision sets was assessed by conducting a user study where people answered multiple-choice questions. Experiments analyzed the responses and evaluated the accuracy and response time. Moreover, the stability metric was derived from prediction, where the algorithm was said to be stable if a small perturbation results in slight prediction changes [84]. In XAI, explanations are stable if similar objects return similar explanations [85]. Therefore, explanations with good quality may suffer from low stability and provide different results when repeated with the same instances and parameters. For the separability metric, XAI explanations are separable if two different objects with variant features produce different explanations [86].

6 REVIEW OF XAI APPLICATIONS

This section presents a review of different Explainable AI (XAI) applications such as image classification, recommendation systems, visual question answering, bias detection, and image captioning.

6.1 Image Classification

Most of the XAI models were applied in image classification and object recognition. The image classification involved using computer vision datasets like ImageNet ILSVRC2012, ImageNet ILSVRC2013, Caltech, CIFAR-10, CIFAR-100, CUB200-2011, PASCAL-VOC, Place365, Tiny ImageNet, MNIST, Stanford Cars, SVHN, COMPAS, GTSRB, Broden, ModelNet, COCO, VQA, and PlantVillage. Moreover, those datasets were interpreted using state-of-art pre-trained neural networks like Inception-V3, AlexNet, VGG-16, VGG-S, VGG-M, ResNet, DenseNet, VGG-11, LeNet, CaffeNet, and GoogleNet.

6.2 Recommendation Systems

Despite focusing on interpreting CNNs in image classification, some XAI models improved the interpretation in other applications like text classification, face recognition, visual question answering, image captioning, and bias detection. For example, D-Attn [31] was applied in recommendation systems to learn user and item features and predict a review rating by adding attention layers to the CNN. Similarly, LIME [52] was applied in sentiment analysis to interpret positive and negative reviews. Unlike images, LIME interpreted textual reviews by visualizing features that mostly contributed to the CNN decision. Figure 10 shows a LIME plot for an insincere Quora question [87]. First, the logistic regression algorithm classified the question as "insincere"; then LIME plots the features (i.e., unigrams) that contributed to this prediction. We can notice that the term "stupid" showed a high negative score (i.e., insincere with a 0.37 score), followed by the "people" term with a negative score of 0.11. In addition, terms like "general" and "seemingly" had positive scores of 0.07 and 0.04, respectively. Overall, the LIME plot justifies the model prediction by showing that the average negativity score was higher than the average positivity score.

6.3 Visual Question Answering

Integrated Gradients [57] was used in question classification, neural machine translation, and chemistry models. In question classification, it identified the type of answer for a given question. Questions could have yes/no, numeric, string, and date answers. Visualization models like CAM [62], Grad-CAM [63], and U-CAM [68] were applied in visual question answering. These models generated activation maps to explain the answer to the question. The activation maps captured the

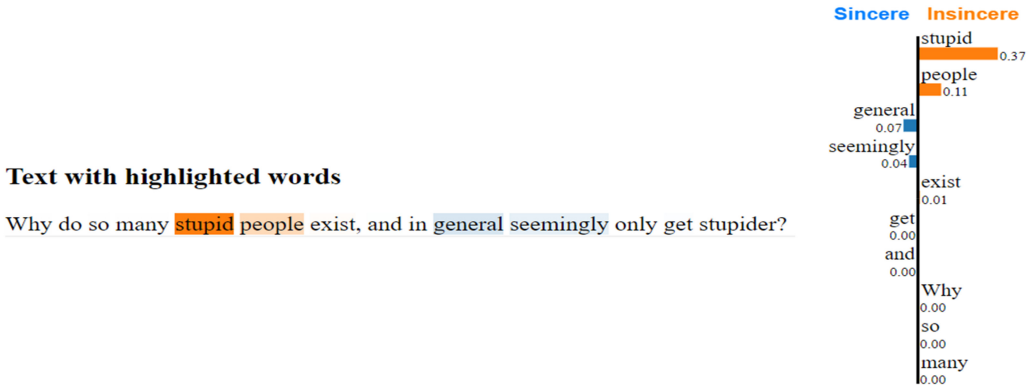

Fig. 10. LIME plot for an insincere Quora question [87].

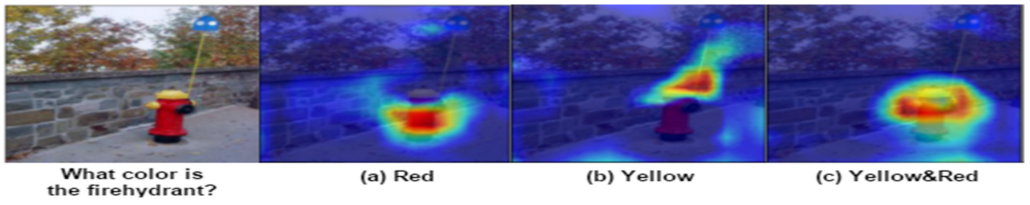

Fig. 11. Class activation maps in VQA [63].

image regions that were relevant to the answer. Figure 11 shows an example of using Grad-CAM in VQA applications.

The first image is associated with the question, "What color is the fire hydrant?". We can observe that the activation maps in Figures 11(a)–11(c) were synchronized with the top answers. For instance, when the RNN-CNN network processed visual (i.e., image) and textual (i.e., question) information to provide an answer of "Red", the Grad-CAM activation map captured the lower red part of the fire hydrant. In addition, when an answer of "Yellow" was provided, the class activation map captured the upper yellow part of the fire hydrant. Moreover, the class activation map captured all parts of the fire hydrant when the "Yellow and Red" answer was provided.

6.4 Bias Detection

Grad-CAM [63] and Score-CAM [70] experiments proved that these XAI models effectively detected biased decisions. For instance, Grad-CAM activation maps revealed that CNN was looking at the person's face/hairstyle to decide if the image was a doctor or nurse, as shown in Figure 12. This gender-biased CNN was because of the biased training data. They analyzed the training data and found that 78% of doctor's images were men, while 93% of nurse images were women. Therefore, the unbalanced training data forced CNN to learn a gender stereotype.

6.5 Image Captioning

In image captioning, XAI models like Grad-CAM [63], Grad-CAM++ [64], and Equalizer [75] used the generated caption to capture the occurrence of every object in the caption. Besides highlighting the important regions in the input image, XAI models could detect gender-biased captions. The models found that CNN was not looking at the person but at other visual cues when generating captions like "man" and "woman". For instance, Figure 13 shows how Equalizer activation

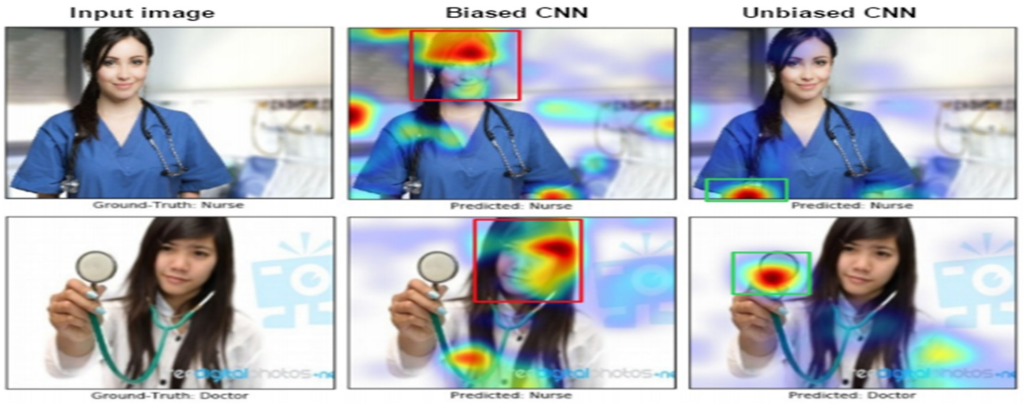

Fig. 12. Gender bias detection [63].

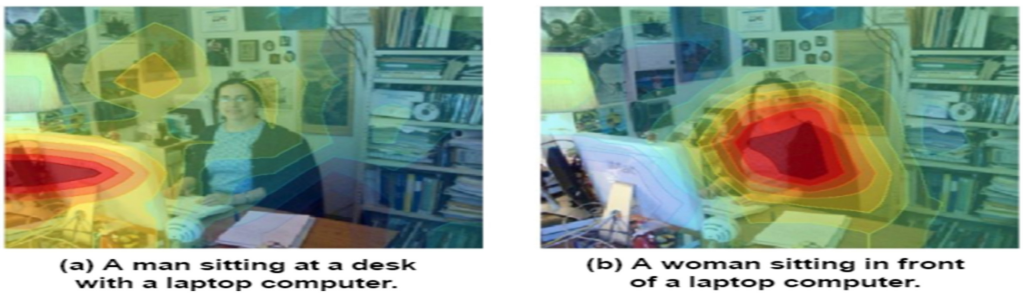

Fig. 13. Gender bias detection in image captioning [75].

maps detected the gender bias in captions generated for a given image. Figure 13(a) shows an incorrect caption of "A man sitting at a desk with a laptop computer". Moreover, it was evident that CNN was looking at the computer, not the person. In contrast, the Equalizer activation map in Figure 13(b) proved that CNN generated the correct caption by looking at the person. Additionally, Grad-CAM++ [64] was applied in 3D video action recognition. Therefore, visualizations were generated for each frame in the original video. The XAI model could classify human activities like soccer, tennis, and baseball by highlighting important regions in each frame.

7 DISCUSSION AND FUTURE DIRECTIONS

This section presents discussion and future directions, such as model generalization, the need for unified evaluation criteria, model or parameters selection framework, interpreting adversarial attacks, and semantic interpretation.

7.1 Model Generalization

A major concern of XAI models is their ability to generalize across different applications and neural networks.

7.1.1 Post-hoc vs. Intrinsic (Generalization across Applications/Tasks). In this section, we conduct a comparative analysis between post-hoc and intrinsic models in terms of generalization. Most intrinsic and post-hoc XAI models were applied in image classification and object recognition. However, some post-hoc models were applied in other applications like image captioning,

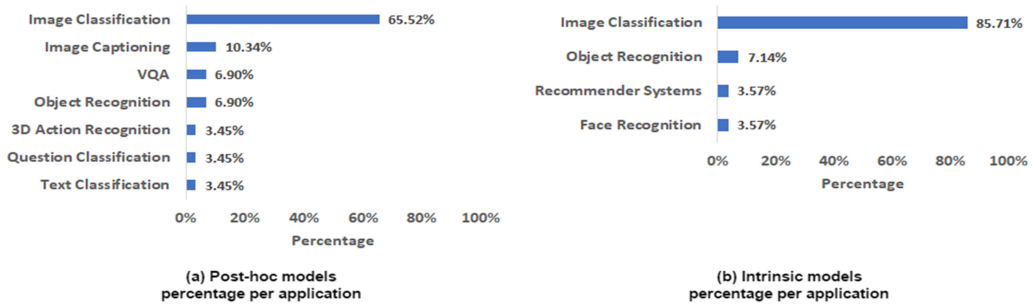

Fig. 14. XAI models per application.

question classification, and VQA, such as Integrated Gradients [57], Grad-CAM [63], Grad-CAM++ [64], Equalizer [75], and U-CAM [68]. This indicates that post-hoc models had a higher ability to generalize across different tasks.

Moreover, intrinsic models modified parts in CNNs like layers, features, and loss functions to improve the network interpretability. Most intrinsic models expressed a higher classification accuracy compared to state-of-art neural networks. However, some models decreased the accuracy of state-of-art neural networks like NIN [33], CNN Explainer [19], XCNN [20], and FCM [44]. The accuracy of these models should be considered when they are applied in crucial applications such as autonomous vehicles and medical imaging. Therefore, the generalization of intrinsic models that cause a drop in accuracy is limited to other applications where interpretability is preferred over accuracy.

Some post-hoc models proved to be useful in detecting bias like Grad-CAM [63], Score-CAM [70], Equalizer [75], and LIME [52]. However, the heatmaps they generated were passive. For instance, the gender bias detection example in Figure 12 clarified the existence of the bias but could not identify the defect in the training data. Additionally, it could not provide suggestions to mitigate or avoid this bias, like fixing the training data or tuning the model's hyperparameters. Therefore, we believe that providing more suggestions with the heatmaps can improve the user trust in CNNs and encourage the adoption of post-hoc models in more applications.

7.1.2 Model-specific vs. Model-agnostic (Generalization across Neural Networks). In this study, we consider a model to be agnostic if it was applied to different state-of-art CNNs. Also, we consider the model to be specific if it was applied only to a customized CNN. Therefore, most XAI models we reviewed in the literature were model-agnostic as they could generalize across state-of-art neural networks. For instance, Interpretable CNN [36] and CNN Explainer [19] modified neural networks like AlexNet, VGG-M, VGG-S, and VGG-16. CAR [50] compressed neural networks like LeNet, AlexNet, and ResNet-50. Grad-CAM [63], Grad-CAM++ [64], and Score-CAM [70] heatmaps were generated by neural networks like AlexNet, VGG-16, ResNet-50, and GoogleNet. In addition, only the CNNV model [77] failed to generalize since it produced an acyclic directed graph built on a customized neural network, with four convolutional layers and two fully connected layers.

7.1.3 Summary. Figure 14 shows the summary of applications and tasks for intrinsic and post-hoc XAI models. We can notice that post-hoc models were used in seven applications, while intrinsic models were used in four applications. Furthermore, image classification was the most used application to test post-hoc and intrinsic models with a percentage of 65.52% and 85.71%, respectively. This result indicates that XAI studies focused on the impact of adding an auxiliary component (i.e., post-hoc) or modifying the neural network (i.e., intrinsic) on the network classification accuracy.

7.2 Unified Evaluation Criteria

Most XAI models were evaluated based on classification accuracy, class discrimination, object localization, and robustness. There was a lack of a complexity analysis to measure the computation of XAI models. Applying XAI models in real-time applications like surveillance systems, autonomous vehicles, and sports analysis requires efficiency along with efficacy. Therefore, XAI models should consider three factors, accuracy, explainability, and complexity.

7.2.1 Post-hoc vs. Intrinsic (XAI Model Evaluation).

7.2.1.1 Classification Accuracy. Intrinsic XAI models like Interpretable CNN [36], CNN Explainer [19], XCNN [20], and FCM [44] improved CNN interpretation by adding loss functions, autoencoders, clustering algorithms, and decision trees. These extra components impacted the CNN classification and degraded the accuracy. Therefore, embedding these XAI models in crucial applications like autonomous vehicles and medical imaging can have dangerous consequences. In addition, the training of modified neural networks can be challenging and time-consuming, like in Deep KNN [46]. This intrinsic model applied a KNN classifier on the top of each CNN layer which impacts the computational cost of CNN. Therefore, previous XAI models ignored the additional cost caused by combining and adding these extra components. Meanwhile, post-hoc models like Grad-CAM [63], Score-CAM [70], Eigen-CAM [69], and U-CAM [68] were added as auxiliary components to pre-trained CNNs. Therefore, these models ignored the classification accuracy and maintained CNNs architecture.

7.2.1.2 Class Discrimination. For the class discrimination evaluation, saliency maps and heatmaps were evaluated qualitatively. XAI models in this criterion relied on human judgment to decide that the model visualization was more interpretable. Some intrinsic models like CNN Explainer [19], Dynamic-K Activation [37], ProtoPNet [41], and Loss Attention [30] used Grad-CAM [63] heatmaps to prove that their modified CNNs produced better interpretation. Meanwhile, other intrinsic models like XCNN [20], FCM [44], and SENN [47] used various heatmaps like Saliency Maps [60], LRP [56], LIME [52], and SHAP [11] for class discrimination evaluation. Furthermore, post-hoc models were qualitatively evaluated by comparing their heatmaps with other post-hoc models. For instance, DeepLIFT [58] compared its saliency maps with other models like Integrated Gradients [57]. Additionally, Score-CAM [70] compared its heatmap with the heatmap of Grad-CAM [63].

7.2.1.3 Object Localization. For object localization, most XAI models used the Intersection over Union (IoU) metric. These models followed the approach described in Section 5.2, which plots bounding boxes for the model and ground-truth. After that, it uses the IoU metric to calculate the overlapping area between the two bounding boxes. Intrinsic models like Interpretable CNN [36], AOG [40], and Dynamic-K Activation [37] applied the IoU metric to evaluate object localization. Other intrinsic models like CNN Explainer [19] used the location instability metric that measured the localization of an object part generated by a specific feature map and filter. Similarly, post-hoc models like Grad-CAM [63], Grad-CAM++ [64], Eigen-CAM [69], and Mask [72] used the IoU metric to evaluate the object localization. Score-CAM [70] followed an energy-based approach to evaluate object localization. This approach binarized the image based on the bounding box; then, it multiplied the binarized image with the saliency map to calculate the energy inside the bounding box.

7.2.1.4 Robustness. Furthermore, CNNs proved to be vulnerable to adversarial examples. Unnoticeable perturbations could cause the CNN to misclassify the image. Therefore, it is important to evaluate the robustness of XAI models against adversarial examples. Although most XAI models ignored the robustness evaluation metric, intrinsic models like Residual Attention [29] proved that

Table 5. Intrinsic Models that Impacted CNN Classification Accuracy

Model	Description	Added components
Interpretable CNN [36]	AlexNet, VGG-M. VGG-S, VGG-16 outperformed Interpretable CNN in classification accuracy for single class	Added loss of feature map to each filter in high conv. layers
CNN Explainer [19]	AlexNet, VGG-M. VGG-S, VGG-16 outperformed CNN Explainer in classification accuracy	Added Autoencoder to each feature map in middle layers
XCNN [20]	VGG-16 outperformed XCNN in classification accuracy	Attached Autoencoder to CNN
FCM [44]	Base CNN outperformed FCM in classification accuracy	Added fuzzy classifier to the last conv. layer

Table 6. Robustness Evaluation in XAI Models

Model	Description	Evaluation
Residual Attention [29]	Evaluated against noised labels and proved to outperform ResNet in classification accuracy	Perturbed images from CIFAR dataset
ProtoPShare [42]	Evaluated against perturbed images and proved to outperform ProtoPNet [41] in classification accuracy	Perturbed images from CUB-200-2011 dataset
SENN [47]	Evaluated against perturbed images and proved to outperform LIME [52] and SHAP [11] in interpretation	Perturbed images from MNIST and COMPAS datasets
Deep KNN [46]	Evaluated against FGSM, BIM, and SW adversarial attacks and proved to outperform traditional CNN in prediction	Perturbed images from MNIST, SVHN, and GTSRB datasets
Grad-CAM [63]	Evaluated against perturbed images and proved to localize the object correctly	Perturbed images from ImageNet dataset
Mask [72]	Evaluated against perturbed images and proved to outperform Grad-CAM [63] in classification accuracy	Perturbed images
Subnetwork Extraction [49]	Evaluated against FGSM, BIM, DeepFool, and SW adversarial attacks and proved to outperform LID and Mahalanobis in adversarial example detection (AUROC)	Perturbed images from CIFAR-10, CIFAR-100, and SVHN datasets
Eigen-CAM [69]	Evaluated against DeepFool adversarial attacks and proved to outperform CAM [62] and Grad-CAM [63] in prediction	Perturbed images from ImageNet dataset

the model was resistant to noised labels when comparing its accuracy with ResNet. In addition, ProtoPShare [42] proved to be robust against image perturbations like contrast and brightness. SENN [47] proved to be more robust than LIME [52] and SHAP [11] against adversarial examples on various datasets. Moreover, Deep KNN [46] proved to be more robust than traditional CNN against adversarial examples like FGSM, BIM, and C&W. Meanwhile, post-hoc models like Grad-CAM [63] and Mask [72] were evaluated against local image perturbations. Subnetwork Extraction [49] and Eigen-CAM [69] evaluated their robustness against effective attacks such as FGSM, BIM, DeepFool, and C&W. However, the literature lacked a comparative analysis for analyzing the impact of adversarial attacks on various XAI models, neural networks, and datasets. Also, there was no quantitative and qualitative evaluation of the XAI models' performance after being attacked. Overall, there is a need for a unified evaluation framework that provides guidelines for selecting evaluation metrics that fit with the XAI model, dataset, network, and application area.

7.2.2 Summary. Table 5 shows four intrinsic models that caused a drop in the CNN accuracy when adding components to improve the network interpretability. Meanwhile, Table 6 shows XAI models evaluated against various adversarial attacks.

7.3 Model or Parameters Selection Framework

Various modifications were applied to CNN architecture to improve their interpretation. However, there was a lack of analysis in selecting features and layers to be modified. XAI models did not

Table 7. XAI Models with Parametric Algorithms

Model	Algorithm	Parameters
Deep KNN [46]	KNN classification that applied cosine similarity to find nearest neighbors	Number of neighbors k
Subnetwork Extraction [49]	Agglomerative hierarchical clustering to categorize specific subnetwork representations	Number of clusters k
CAR [50]	Structural compression of CNN filters	Compression ratio r_{target}
EBANO [53]	K-means clustering of hyper columns to identify interpretable features	Number of clusters n, *Set of centroids* k

propose a criterion for identifying the most informative features/layers (i.e., low vs. intermediate vs. high layers). For example, what features/layers can provide more insights when adding KNN classifiers or attention layers. Therefore, we believe that selecting informative features/layers requires further analysis. Moreover, some XAI models integrated parametric machine learning algorithms to interpret CNNs like Deep KNN [46], Subnetwork Extraction [49], CAR [50], and EBANO [53]. However, the main drawback of these algorithms is the initialization of their parameters. Selecting the appropriate number of clusters for large networks and datasets could be challenging. Table 7 shows XAI models that integrated parametric algorithms. We notice from the table that algorithms like KNN, hierarchical clustering, K-means, and compression require initializing some parameters. For example, KNN proved to be sensitive to the value of k, and selecting its value can be challenging for datasets with various sizes [88]. Moreover, the prediction stability relies on the value of k. If the k value is low (i.e., equal to 1), the prediction becomes less stable. Meanwhile, if the k value increases to a certain point, the prediction will produce more errors. In K-means [89], the algorithms select a random set of centroids k as initial seeds. However, this random seeding could generate poor results since some clusters are merged early and are hard to split later [90]. Therefore, initializing parameters in XAI models that use machine learning algorithms should be considered carefully.

In addition, selecting the appropriate layer and network to be compressed or clustered requires collaboration with domain experts. For instance, the features clustering in EBANO [53] lacked the importance of features ranking and the analysis of features interaction (i.e., interconnection) in CNN. In addition, visualization models did not mention which activation maps fit a specific level of users or applications, for example, which output is more interpretable to end-users in medical imaging, activation maps, saliency maps, or masked images. Overall, there is a need for a unified framework for selecting the XAI model that provides optimal interpretation for a given dataset, neural network, and application.

7.4 Interpreting Adversarial Attacks

Despite the lack of robustness evaluation, some efforts were made to clarify the vulnerability of XAI models against adversarial examples. It was proven that ADV^2 [91] attack succeeded in fooling CNN and the post-hoc XAI model. The reason for the post-hoc model vulnerability was the gap between prediction and interpretation. They argued that the gap was due to partial independence between CNN and XAI models since they partially described the prediction. Moreover, the **adversarial interpretation distillation (AID)** framework was proposed to reduce this gap by adding a loss function to the XAI model to minimize interpretation loss. A promising approach is embedding defense strategies in XAI models to empower robustness. Defense strategies can be categorized into three types: modifying input image, modifying neural network, and adding an auxiliary network. Table 8 shows defense strategies that can be applied in XAI models.

Table 8. Defense Strategies Models

Defense Model	Defense Strategy	Modifying Input Image	Modifying Neural Network	Network add-on
Defensive Distillation [92]	Network Distillation		Yes	
Noise-GAN [93]	Adversarial Training	Yes		
Defense-Net [94]	Adversarial Detection			Yes
Image Super-Resolution [95]	Input Reconstruction	Yes		
Spartan [96]	Feature Reduction		Yes	
FN [97]	Gradient Masking		Yes	

7.5 Semantic Interpretation

Combining semantic details with CNN filters or class activation maps is an interesting approach for improving human cognition. For example, Subnetwork Extraction [49] applied hierarchical clustering to measure the semantic similarity among a set of samples. Each cluster could represent a unique semantic label (e.g., eagle heads, car wheels). However, these clusters were extracted but not assigned to annotations. Network Dissection [51] extracted the semantics of CNN intermediate layers by using the Broden dataset. This dataset contains a set of labeled visual cues. Convolutional units were binary segmented and compared to the Broden dataset to predict the semantic label. Moreover, IBD [8] used the Broden dataset to extract decomposed semantic labels for the CNN prediction. The model generated class activation maps and associated each map with a semantic label and a rank. A potential limitation of semantic interpretation is that XAI models relied heavily on the Broden dataset. Therefore, the quality of visual cues in this dataset could impact the interpretations (i.e., semantic labels). Another limitation for semantic interpretation is extracting semantic labels in applications like medical imaging. For instance, finding labeled radiology image datasets could be challenging. Also, the extraction of semantic labels and identifying the important ones require collaboration with domain experts.

8 CONCLUSION

We conducted an extensive review of XAI models that improved the interpretation of convolutional neural networks. We started by describing our search methodology. First, we used Google Scholar to retrieve papers related to “explainable”, “interpretable”, and “convolutional neural networks” keywords. After that, we excluded non-relevant ones. In addition, we analyzed the latest trend of XAI papers in the last two decades. The trend showed that explainability and interpretability were attracting more researchers. Furthermore, we identified frequent terms in XAI papers in the last three years. It was evident that terms like “image”, “classification”, “feature”, and “human” were closely related to the interpretation of CNNs. After that, we explained XAI terms and definitions, like explainability and interpretability. We highlighted the importance of collaboration between HCI and software engineering to generate perceivable explanations for different users. Also, we discussed how explanations should be provided to build responsible neural networks.

We discussed XAI taxonomies such as scope (global vs. local), structure (intrinsic vs. post-hoc), dependency (model-specific vs. model-agnostic), and dataset (image vs. text). Then, we categorized XAI models that interpreted CNNs into four categories: architecture modification, architecture simplification, features relevance, and visualization. In each category, we discussed each model and described its approach and drawbacks. Furthermore, we summarized models in each category and clarified each model’s scope, structure, and dependency. After that, we conducted a correlation analysis to recognize the behavior of XAI models. We found that models in

the architecture modification category were intrinsic and local. While in the architecture simplification, models were model-agnostic and local. In feature relevance, most models were post-hoc and local. In the visualization category, most models were model-agnostic, post-hoc, and local. In addition, we studied the evaluation metrics in XAI models. This analysis showed that most interpretations were evaluated by visualization, localization, robustness, and classification accuracy metrics. In visualization metric, we added use cases to describe visualizations like saliency maps [60, 57, 66], class activation maps [63, 64, 70], and pixels visualization [52]. We showed how class activation maps outperformed saliency maps in class discrimination. In the localization metric, we added a use case to discuss the IoU metric. We showed that the higher value of IoU reflects better object localization. In robustness metric, we discussed models that proved to be resistant against noised images [63, 72] or adversarial attacks [34, 46, 49, 69]. Moreover, we showed that intrinsic models [28, 29, 33, 44, 19, 41] relied on the accuracy metric to evaluate the neural network after modifying or adding some components.

Furthermore, we studied the trend of the applications and tasks in XAI models. This analysis showed that most models were applied to image classification, recommendation systems, visual question answering (VQA), bias detection, and image captioning. We added a use case to describe how class activation maps [63] could capture fire hydrant parts that represent the answer in VQA. In bias detection, we showed how class activation maps [63] helped uncover the gender stereo bias in convolutional neural networks. Interestingly, the activation maps revealed that the CNN was looking at the hairstyle and face of the person, not at the dress or tools the person was wearing. In image captioning, we showed how class activation maps [63, 64] could capture each object mentioned in the generated caption. Moreover, we added a use case to prove how activation maps [75] could detect the gender bias in captions generated.

Finally, we summarized our reflections on the gaps and future directions of CNN interpretation models. In terms of generalization, we conducted a comparative analysis between post-hoc and intrinsic models. It was apparent that post-hoc models could generalize across more applications like image captioning and VQA. Moreover, post-hoc and intrinsic models were mostly applied to the image classification task. For the evaluation criteria, we conducted a comparative analysis between post-hoc and intrinsic models in terms of classification accuracy, class discrimination, object localization, and robustness. Intrinsic models relied more on the classification accuracy metric to measure the performance of modified CNN. In addition, intrinsic models have used post-hoc saliency maps and heatmaps to evaluate their visualizations qualitatively. For object localization, the IoU metric was mostly used by intrinsic and post-hoc models. Furthermore, four intrinsic models were evaluated against perturbed images and adversarial attacks [29, 42, 47, 46], and four post-hoc models were evaluated against perturbed images and adversarial attacks [63, 72, 49, 69]. In terms of parameters selection, some models used machine learning algorithms like clustering and compression to improve CNN interpretation [46, 49, 50, 53]. However, there is a demand for collaboration with domain experts to initialize parameters. We showed that post-hoc models were vulnerable to adversarial attacks due to their partial independence from CNNs [91]. We proposed defense strategies to improve the robustness of the interpretation model. Also, we highlighted the importance of adding semantics to activation maps and discussed some limitations in this area.

This survey aims to provide researchers and practitioners with a wide range of interpretation models they can use in different tasks and application areas.

LIST OF ABBREVIATIONS

XAI Explainable Artificial Intelligence
CNN Convolutional Neural Network
CAM Class Activation Map

REFERENCES

[1] X.-H. Li et al. 2020. A survey of data-driven and knowledge-aware explainable AI. *IEEE Trans. Knowl. Data Eng.* (2020), 1–1. DOI: 10.1109/TKDE.2020.2983930

[2] J. J. Ferreira and M. S. Monteiro. 2020. What are people doing about XAI user experience? A survey on AI explainability research and practice. In *Design, User Experience, and Usability. Design for Contemporary Interactive Environments 12201*, A. Marcus and E. Rosenzweig (Eds.). Cham: Springer International Publishing (2020), 56–73. DOI: 10.1007/978-3-030-49760-6_4

[3] J. Wanner, L.-V. Herm, K. Heinrich, C. Janiesch, and P. Zschech. 2020. White, grey, black: Effects of XAI augmentation on the confidence in AI-based decision support systems 9, (2020).

[4] R. Hoffman, S. T. Mueller, G. Klein, and J. Litman. 2018. Metrics for explainable AI: Challenges and prospects. *ArXiv* (2018), abs/1812.04608.

[5] A. Barredo Arrieta et al. 2020. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115. DOI: 10.1016/j.inffus.2019.12.012

[6] A. Rai. 2020. Explainable AI: From black box to glass box. *J. of the Acad. Mark. Sci.* 48, 1 (2020), 137–141. DOI: 10.1007/s11747-019-00710-5

[7] Y. Chung, C.-A. Chou, and C.-Y. Li. 2021. Central attention and a dual path convolutional neural network in real-world tree species recognition. *IJERPH* 18, 3 (2021), 961. DOI: 10.3390/ijerph18030961

[8] B. Zhou, Y. Sun, D. Bau, and A. Torralba. 2018. Interpretable basis decomposition for visual explanation. In *Computer Vision – ECCV 2018*, Cham, 122–138

[9] R. McGrath et al. 2018. Interpretable credit application predictions with counterfactual explanations. *CoRR*, vol. abs/1811.05245, 2018, [Online]. Available http://arxiv.org/abs/1811.05245.

[10] S. Wachter, B. Mittelstadt, and C. Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *SSRN Journal* 2017. DOI: 10.2139/ssrn.3063289

[11] S. Lundberg and S.-I. Lee. 2017. A unified approach to interpreting model predictions 2017.

[12] M. Du, N. Liu, and X. Hu. 2019. Techniques for interpretable machine learning. *Association for Computing Machinery* 63 (2019). DOI: 10.1145/3359786

[13] P. K. Sharma and P. Bhattacharyya. Survey of explainable AI: Interpretability and causal reasoning, 16.

[14] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. 2020. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis.* 128, 2 (2020), 336–359. DOI: 10.1007/s11263-019-01228-7

[15] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. 2018. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2018), 839–847. DOI: 10.1109/WACV.2018.00097

[16] Alizadeh Fatemeh, Esau Margarita, Stevens Gunnar, and Cassens Lena. 2020. eXplainable AI: Take one step back, move two steps forward (2020). DOI: 10.18420/MUC2020-WS111-369

[17] M. Chromik and M. Schuessler. 2020. A taxonomy for human subject evaluation of black-box explanations in XAI, 7 (2020).

[18] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu. 2019. Interpretable machine learning: Definitions, methods, and applications. *Proc. Nat'l. Acad. Sci. USA* 116, 44 (2019), 22071–22080. DOI: 10.1073/pnas.1900654116

[19] Q. Zhang, Y. Yang, Y. Liu, Y. N. Wu, and S.-C. Zhu. 2021. Unsupervised learning of neural networks to explain neural networks. *arXiv:1805.07468 [cs]* Accessed: Feb. 20, 2021. [Online]. Available http://arxiv.org/abs/1805.07468.

[20] A. Tavanaei. 2021. Embedded encoder-decoder in convolutional networks towards explainable AI. *arXiv:2007.06712 [cs] 2020*, Accessed: Feb. 20, 2021. [Online]. Available http://arxiv.org/abs/2007.06712.

[21] J. H. Friedman and B. E. Popescu. 2008. Predictive learning via rule ensembles. *Ann. Appl. Stat* 2, 3 (2008), 916–954, DOI: 10.1214/07-AOAS148

[22] T. Hastie, R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media.

[23] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin. 2015. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics* 24, 1 (2015), 44–65, DOI: 10.1080/10618600.2014.907095

[24] Y. Zhang, H. Su, T. Jia, and J. Chu. 2005. Rule extraction from trained support vector machines. In *Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg, 61–70.

[25] Y. Liu, Y. Cheng, and W. Wang. 2018. A survey of the application of deep learning in computer vision. In *Global Intelligence Industry Conference (GIIC 2018)*. Beijing, China, 68. DOI: 10.1117/12.2505431

[26] Y. LeCun and Y. Bengio. 1998. Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 255–258.

[27] D. Linsley, D. Shiebler, S. Eberhardt, and T. Serre. 2021. Learning what and where to attend. *arXiv:1805.08819 [cs]*, Accessed: Feb. 19, 2021. [Online]. Available http://arxiv.org/abs/1805.08819.

[28] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Z. Zhang. 2015. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA, 842–850. DOI: 10.1109/CVPR.2015.7298685

[29] F. Wang et al. 2017. Residual attention network for image classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, 6450–6458. DOI: 10.1109/CVPR.2017.683

[30] X. Shi et al. 2021. Loss-based attention for interpreting image-level prediction of convolutional neural networks. *IEEE Trans. on Image Process* 30, (2021), 1662–1675. DOI: 10.1109/TIP.2020.3046875

[31] S. Seo, J. Huang, H. Yang, and Y. Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. Como, Italy (2017), 297–305. DOI: 10.1145/3109859.3109890

[32] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. 2015. Striving for simplicity: The all convolutional net, 2015. [Online]. Available http://lmb.informatik.uni-freiburg.de/Publications/2015/DB15a.

[33] M. Lin, Q. Chen, and S. Yan. 2021. Network in network. *arXiv:1312.4400 [cs]*, Mar. 2014, Accessed: Feb. 19, 2021. [Online]. Available http://arxiv.org/abs/1312.4400.

[34] H. Liang et al. 2020. Training interpretable convolutional neural networks by differentiating class-specific filters. In *Computer Vision – ECCV 2020*, 12347, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, (Eds.). Cham: Springer International Publishing, 622–638. DOI: 10.1007/978-3-030-58536-5_37

[35] K. Horii, K. Maeda, T. Ogawa, and M. Haseyama. 2020. Paper interpretable convolutional neural network including attribute estimation for image classification 8, 2 (2020), 14.

[36] Q. Zhang, X. Wang, Y. N. Wu, H. Zhou, and S.-C. Zhu. 2020. Interpretable CNNs for object classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (2020). DOI: 10.1109/TPAMI.2020.2982882

[37] Y. Sun, S. Ravi, and V. Singh. 2019. Adaptive activation thresholding: Dynamic routing type behavior for interpretability in convolutional neural networks. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 4937–4946. DOI: 10.1109/ICCV.2019.00504

[38] B. Yin, L. Tran, H. Li, X. Shen, and X. Liu. 2019. Towards interpretable face recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* Seoul, Korea (South), 9347–9356. DOI: 10.1109/ICCV.2019.00944

[39] C. Hwa Yoo, N. Kim, and J.-W. Kang. 2019. Relevance regularization of convolutional neural network for interpretable classification 2019.

[40] Q. Zhang, R. Cao, Y. N. Wu, and S.-C. Zhu. 2017. Growing interpretable part graphs on ConvNets via multi-shot learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. San Francisco, California, USA, 2898–2906.

[41] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su. 2019. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems* 2019, 32. [Online]. Available https://proceedings.neurips.cc/paper/2019/file/adf7ee2dcf142b0e11888e72b43fcb75-Paper.pdf.

[42] D. Rymarczyk, Ł. Struski, J. Tabor, and B. Zieliński. 2020. ProtoPShare: Prototype sharing for interpretable image classification and similarity discovery. *Association for Computing Machinery* (2020), 1420–1430. DOI: 10.1145/3447548.3467245

[43] M. D. Zeiler, G. W. Taylor, and R. Fergus. 2011. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*. Barcelona, Spain, 2018–2025. DOI: 10.1109/ICCV.2011.6126474

[44] M. Yeganejou, S. Dick, and J. Miller. 2019. Interpretable deep convolutional fuzzy classifier. *IEEE Trans. Fuzzy Syst.* 1–1 (2019). DOI: 10.1109/TFUZZ.2019.2946520

[45] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu. 2019. Interpreting CNNs via decision trees. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, 6254–6263. DOI: 10.1109/CVPR.2019.00642

[46] N. Papernot and P. McDaniel. 2018. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning, *ArXiv*, vol. abs/1803.04765.

[47] D. Alvarez-Melis and T. Jaakkola. 2018. Towards robust interpretability with self-explaining neural networks, 2018.

[48] X. Liu, X. Wang, and S. Matwin. 2018. Interpretable deep convolutional neural networks via meta-learning. *2018 International Joint Conference on Neural Networks (IJCNN)* (2018), 1–9.

[49] Y. Wang, H. Su, B. Zhang, and X. Hu. 2020. Interpret neural networks by extracting critical subnetworks. *IEEE Transactions on Image Processing* 29 (2020), 6707–6720. DOI: 10.1109/TIP.2020.2993098

[50] R. Abbasi-Asl and B. Yu. 2017. Structural compression of convolutional neural networks based on greedy filter pruning. *ArXiv*, abs/1705.07356.

[51] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 3319–3327.

[52] M. T. Ribeiro, S. Singh, and C. Guestrin. 2016. 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*. New York, NY, USA, 1135–1144. DOI: 10.1145/2939672.2939778

[53] F. Ventura and T. Cerquitelli. 2019. What's in the box? Explaining the black-box model through an evaluation of its interpretable features. *ArXiv*, vol. abs/1908.04348.

[54] M. Tamajka, W. Benesova, and M. Kompanek. 2019. Transforming convolutional neural network to an interpretable classifier. In *2019 International Conference on Systems, Signals and Image Processing (IWSSIP)* 2019, 255–259. DOI: 10.1109/IWSSIP.2019.8787211

[55] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. 2016. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA, 3395–3403.

[56] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE* 10, 7 (2015), 1–46. DOI: 10.1371/journal.pone.0130140

[57] M. Sundararajan, A. Taly, and Q. Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 Sydney*. NSW, Australia, 3319–3328.

[58] A. Shrikumar, P. Greenside, and A. Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning, International Convention Centre*. Sydney, Australia, 70 3145–3153. [Online]. Available http://proceedings.mlr.press/v70/shrikumar17a.html.

[59] N. U. Islam and S. Lee. 2019. Interpretation of deep CNN based on learning feature reconstruction with feedback weights. *IEEE Access* 7 (2019), 25195–25208, DOI: 10.1109/ACCESS.2019.2899901

[60] K. Simonyan, A. Vedaldi, and A. Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*. abs/1312.6034, (2014).

[61] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. 2010. Deconvolutional networks. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2010), 2528–2535.

[62] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. 2016. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 2921–2929. DOI: 10.1109/CVPR.2016.319

[63] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 618–626. DOI: 10.1109/ICCV.2017.74

[64] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. 2018. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2018), 839–847. DOI: 10.1109/WACV.2018.00097

[65] D. Omeiza, S. Speakman, C. Cintas, and K. Weldemariam. 2019. Smooth grad-CAM++: An enhanced inference level visualization technique for deep convolutional neural network models. *ArXiv*, (2019), abs/1908.01224.

[66] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. 2017. SmoothGrad: Removing noise by adding noise. *ArXiv* (2017), abs/1706.03825.

[67] P. Morbidelli, D. Carrera, B. Rossi, P. Fragneto, and G. Boracchi. 2020. Augmented Grad-CAM: Heat-maps super resolution through augmentation. In *ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), 4067–4071. DOI: 10.1109/ICASSP40776.2020.9054416

[68] B. Patro, M. Lunayach, S. Patel, and V. Namboodiri. 2019. U-CAM: Visual explanation using uncertainty based class activation maps. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 7443–7452. DOI: 10.1109/ICCV.2019.00754

[69] M. Bany Muhammad and M. Yeasin. 2021. Eigen-CAM: Visual explanations for deep convolutional neural networks. *SN Computer Science* 2, 1 (2021) 47. DOI: 10.1007/s42979-021-00449-3

[70] H. Wang et al. 2020. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Seattle, WA, USA, 111–119. DOI: 10.1109/CVPRW50498.2020.00020

[71] R. Ibrahim and M. O. Shafiq. 2022. Augmented score-CAM: High resolution visual interpretations for deep neural networks. *Knowledge-Based Systems* 252, 109287, (2022). DOI: https://doi.org/10.1016/j.knosys.2022.109287

[72] R. C. Fong and A. Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 3449–3457. DOI: 10.1109/ICCV.2017.371

[73] M. Brahimi, S. Mahmoudi, K. Boukhalfa, and A. Moussaoui. 2019. Deep interpretable architecture for plant diseases classification. In *2019 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)* (2019), 111–116. DOI: 10.23919/SPA.2019.8936759

[74] P. Hase, C. Chen, O. Li, and C. Rudin. 2019. Interpretable image recognition with hierarchical prototypes. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 7, (2019), 32–40, [Online]. Available https://ojs.aaai.org/index.php/HCOMP/article/view/5265.

[75] L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach. 2018. Women also snowboard: Overcoming bias in captioning models. In *Computer Vision – ECCV 2018*, Cham, 793–811.

[76] Y. Goyal, A. Mohapatra, D. Parikh, and D. Batra. 2016. Towards transparent AI systems: Interpreting visual question answering models. *arXiv: Computer Vision and Pattern Recognition*, 2016.

[77] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. 2017. Towards better analysis of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 91–100, DOI: 10.1109/TVCG.2016.2598831

[78] A. Rosebrock. 2021. Intersection over Union (IoU) for object detection. *Intersection over Union (IoU) for Object Detection* 07, 2016. https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/#pyis-cta-modal (accessed Sep. 23, 2021).

[79] W. Sherchan, S. Nepal, and C. Paris. 2013. A survey of trust in social networks. *ACM Comput. Surv.* 45, 4 (2013), DOI: 10.1145/2501654.2501661

[80] A. Bussone, S. Stumpf, and D. O'Sullivan. 2015. The role of explanations on trust and reliance in clinical decision support systems. In *2015 International Conference on Healthcare Informatics* (2015), 160–169. DOI: 10.1109/ICHI.2015.26

[81] C. J. Cai et al. 2019. Human-centered tools for coping with imperfect algorithms during medical decision-making. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA, 1–14. DOI: 10.1145/3290605.3300234

[82] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. 2011. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems* 51, 1 (2011), 141–154. DOI: 10.1016/j.dss.2010.12.003

[83] H. Lakkaraju, S. H. Bach, and J. Leskovec. 2016. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA, 1675–1684. DOI: 10.1145/2939672.2939874

[84] J. F. Bonnans and A. Shapiro. 2013. *Perturbation Analysis of Optimization Problems*. Springer Science & Business Media.

[85] M. Honegger. 2018. Shedding light on black box machine learning algorithms: Development of an axiomatic framework to assess the quality of methods that explain individual predictions. *arXiv preprint arXiv:1808.05054*, 2018.

[86] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics* 8, 8 (2019), 832. DOI: 10.3390/electronics8080832

[87] M. Amami. 2020. Quora sincere questions 2020. [Online]. Available https://github.com/amamimaha/Explainable-Models.

[88] D. Cheng, S. Zhang, Z. Deng, Y. Zhu, and M. Zong. 2014. kNN algorithm with data-driven k value. In *Advanced Data Mining and Applications 8933*, X. Luo, J. X. Yu, and Z. Li, (Eds.). Cham: Springer International Publishing, 499–512. DOI: 10.1007/978-3-319-14717-8_39

[89] J. A. Hartigan and M. A. Wong. 1979. Algorithm AS 136: A K-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28, 1 (1979), 100–108, [Online]. Available http://www.jstor.org/stable/2346830.

[90] D. Arthur and S. Vassilvitskii. 2006. k-means++: The Advantages of Careful Seeding. *Stanford, Technical Report 2006–13*, Jun. 2006. [Online]. Available http://ilpubs.stanford.edu:8090/778/.

[91] X. Zhang, N. Wang, H. Shen, S. Ji, X. Luo, and T. Wang. 2020. Interpretable deep learning under fire. In *29th USENIX Security Symposium (USENIX Security 20)*, (2020), 1659–1676. [Online]. Available https://www.usenix.org/conference/usenixsecurity20/presentation/zhang-xinyang.

[92] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, 582–597. DOI: 10.1109/SP.2016.41

[93] A. S. Hashemi and S. Mozaffari. 2019. Secure deep neural networks using adversarial image generation and training with Noise-GAN. *Computers & Security* 86 (2019), 372–387. DOI: 10.1016/j.cose.2019.06.012

[94] A. S. Rakin and D. Fan. 2019. Defense-Net: Defend against a wide range of adversarial attacks through adversarial detector. In *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Miami, FL, USA, 332–337. DOI: 10.1109/ISVLSI.2019.00067

[95] A. Mustafa, S. H. Khan, M. Hayat, J. Shen, and L. Shao. 2020. Image super-resolution as a defense against adversarial attacks. *IEEE Trans. on Image Process* 29 (2020), 1711–1724. DOI: 10.1109/TIP.2019.2940533

[96] F. Menet, P. Berthier, M. Gagnon, and J. M. Fernandez. 2020. Spartan networks: Self-feature-squeezing neural networks for increased robustness in adversarial settings. *Computers & Security* 88, (2020), 101537. DOI: 10.1016/j.cose.2019.05.014

[97] K. Han, Y. Li, and J. Hang. 2019. Adversary resistant deep neural networks via advanced feature nullification. *Knowledge-Based Systems* 179 (2019), 108–116. DOI: 10.1016/j.knosys.2019.05.007

Received 12 April 2021; revised 10 June 2022; accepted 29 August 2022