Effective Explainable Artificial Intelligence using Visual Explanations in Images

By

Rami Ibrahim

Thesis submitted to the School of Information Technology in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Information Technology

Carleton University
Ottawa, Ontario

Acknowledgements

I would like to thank my advisor Professor. Omair Shafiq for his extraordinary support during my graduate study. Over the last six years, I have known Prof. Omair to be extremely helpful, caring, and resourceful. His exceptional mentorship allowed me to complete my master's degree and make remarkable progress in my Ph.D. studies.

Because of Prof. Omair's extraordinary dedication and support, I could present my research, and publish 10 research papers in top conferences and journals like Springer Journal of Big Data, IEEE BigData, IEEE HPCC, and many more conferences. With his exceptional guidance, I was able to make excellent progress in my research and was able to receive several research scholarships and awards, including Ontario Graduate Scholarship (OGS) multiple times, The Queen Elizabeth II Scholarship in Science and Technology (QEII), Natural Sciences and Engineering Research Council of Canada (NSERC) Canada Graduate Scholarship for Master's (CGS-M), and finally Alexander Graham Bell Canada Graduate Scholarship (NSERC-CGS).

Professor. Omair has truly inspired me to stay motivated toward my academic goals and bring out the best of my work. His encouragement helped me keep my progress even during the tough times of the COVID-19 pandemic. His optimism relaxed me and made me feel the light at the end of the tunnel. Without his exceptional guidance, commitment, determination, passion, and care, I could not have finished my Ph.D. research.

Also, I want to present my gratitude to my deceased parents for their great sacrifices. Many thanks to my brothers, Hamid and Adham, for being true brothers when I needed them. Finally, I would like to express my deep thankfulness to my wife, Marwa, for her patience over the last six years. I could not be able to finish my research without her support.

Author Declaration

I am the first author of the publications that have contributed to this thesis. The word 'we' is used in this document to indicate the collaboration with my supervisor on finishing these projects. I want to thank my supervisor, Dr. Omair Shafiq, for his support in achieving this work.

Publications related to this thesis include:

- Augmented Score-CAM: High resolution visual interpretations for deep neural networks. Elsevier Knowledge-Based Systems Journal. **Published**.
- Explainable Convolutional Neural Networks: A Taxonomy, Review, and Future Directions. ACM Computing Surveys (CSUR) Journal. **Under review.**
- A User Study on Measuring the Trust for Explainable Artificial Intelligence. **To be submitted**.
- Augmented Score-CAM: Improved interpretations for CNN predictions. T**o be submitted**.

More papers not directly related to this thesis:

- Rami Ibrahim, M. Omair Shafiq, Extended results from the measurement and analysis of safety in a large city. International Journal of Big Data Intelligence 6(2): 86-101 (2019).
- Rami Ibrahim, M. Omair Shafiq, Detecting taxi movements using Random Swap clustering and sequential pattern mining. Springer Journal of Big Data 6: 39 (2019).
- Rami Ibrahim, M. Omair Shafiq, On Predicting Taxi Movements Modes in Porto City Using Classification and Periodic Pattern Mining. IEEE HPCC/SmartCity/DSS 2019: 1197-1204 2018.
- Rami Ibrahim, M. Omair Shafiq, Towards a New Approach to Empower Periodic Pattern Mining for Massive Data using Map-Reduce. IEEE BigData 2018: 2206-2215.

Abstract

Convolutional neural networks (CNNs) outperformed machine learning in image classification. Their human-brain alike 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 prediction. Explainable AI (XAI) proposed collaboration between technology and humans to provide more insights into CNNs. This study presents a novel explainable model called Augmented Score-CAM, built on top of the existing Score-CAM and the existing image augmentation techniques. This model adopts the image augmentation approach by producing augmented class activation maps and merging them into one activation map. In addition, we introduce a novel taxonomy analysis for XAI models that interpret CNNs. The taxonomy categorizes the models into architecture modification, architecture simplification, feature relevance, and visual interpretations. After that, we review XAI evaluation metrics, application areas, and tasks. In the end, we discuss XAI challenges and address some concerns, and provide suggestions to improve their performance. This study improves AI systems interpretation by adding Augmented Score-CAM visual explanations. Furthermore, we highlight the importance of incorporating visual explanations in AI systems to improve user trust in decision-making.

Keywords: XAI; Convolutional Neural Networks; Augmented Score-CAM; Visual Explanations.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Meaning
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
ASC	Augmented Score-CAM
CAM	Class Activation Map
CNN	Convolutional Neural Network
GAN	Generative Adversarial Networks
GDPR	General Data Protection Regulation
GNN	Graph Neural Network
GPU	Graphical Processing Unit
HCI	Human Computer Interaction
IDE	Integrated Development Environment
IoU	Intersection over Union
KNN	K-nearest neighbors
MSE	Mean Squared Error
NLP	Natural Language Processing
SPANOVA	Split-Plot ANOVA
SSC	Score-CAM
SVM	Support Victor Machine
XAI	Explainable AI

1. *Introduction*

1.1 Background

1.1.1 CNN Intuition

CNNs are neural networks that process features for a given image to perform computer vision classification tasks like face detection and object recognition [1]. CNNs are supervised neural networks; trained using a set of classified images (i.e., tagged images). They extract the image features by using feature detectors (i.e., kernels). After that, they use simple features to learn more complicated features in the consequent stages [2]. Each image is a two-dimensional array of pixels, where each pixel has a value between 0 and 255. Grey-colored images have one 2D array; Meanwhile, colored images have three 2D arrays representing red, blue, and green channels. The CNN training process goes through four steps, building convolutions, max pooling, flattening, and the full connection.

1.1.2 CNN Architecture

First, input images are fed to the CNN. Any given image is a 2D array of pixels. In addition to the input image, the neural network has a feature detector (i.e., kernel). Each feature detector is a 2D array representing a specific feature the CNN is searching for (e.g., eye, nose, ears, lips). The size of the feature detector is smaller than the input image; it can have a size of 3X3, 5X5, or 7X7. In the first step of the training process, we take the feature detector and apply it to the top left corner of the input image. After that, we multiply corresponding pixels (i.e., zeros and ones) from the input image and feature detector arrays before summing them. We keep sliding the feature detector on the input image from left to right and top to bottom. The output of this step is a feature map that has the result of the feature detector sliding process (i.e., the sums of all pixel's multiplications). The process of producing a feature map is called convolution. A convolution is a

mathematical product of the input image and the feature detector [3]. Suppose we have an input image *a*, and the feature detector *w*; the convolution *c* is the sum of the element-wise multiplication of each pixel in *a* by each pixel in *w*. Equation 1 shows the calculated convolution *c* as follows:

$$c = \sum_{k} a(i,j) * w(m,n) \quad (1)$$

Where *k* is the output shape, *i* and *j* represent the pixel index of the input image *a*, *m* and *n* represent the pixel index of the feature detector *w*. The purpose of creating feature maps is to detect significant features and reduce the size of the input image, which speeds up the processing inside the neural network. Moreover, CNN builds multiple feature maps as it applies more than one feature detector. These feature maps represent the convolutional layer of the CNN [1]. Since images have a nonlinear structure, CNN applies the rectifier function (ReLU) $\phi(x)=\max(x,0)$ to reduce the linearity of the neural network [4]. After building the convolutional layers, CNN applies max pooling in the second step. Max pooling helps CNN to improve its spatial invariance. This feature ensures that the network can find features even if they were tilted or distorted. This step takes the feature map and places a box in its top-left corner. After that, it finds the maximum value of the box and stores it in a 2D array (i.e., pooled feature map). It keeps sliding the box from the left to right and top to bottom until all maximum values are stored. By applying this step, CNN eliminates 75% of insignificant features while preserving the significant ones, which reduces the neural network overfitting [5]. The third step is to apply the flattening. In each pooled feature map, pixel values are taken row by row and stored in one column (i.e., vector). For multiple pooled feature maps, each map is flattened into an input node (i.e., vector), which will be a part of the ANN input layer. The last step is to add an artificial neural network (ANN) to the input layer produced from CNN. The hidden layers inside the ANN are fully connected with the input and output layers. The purpose of the ANN is to take the image features produced from CNN and

predict the images classes (e.g., cat or dog). Furthermore, ANN computes the cost function after the class prediction; then, it backpropagates through both CNN and ANN networks. In CNN, the feature detectors are adjusted, while in ANN, the input weights are adjusted. Figure 1 shows the architecture of CNN with a fully connected ANN [6].

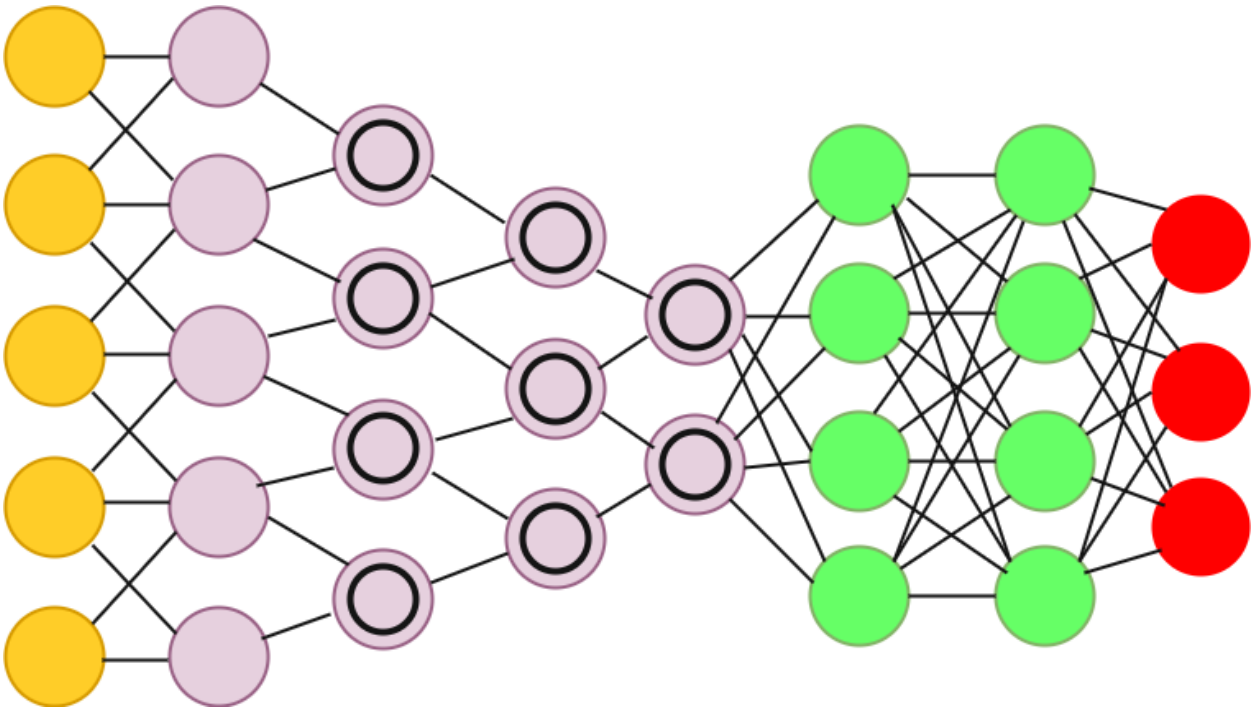


Figure 1. CNN Architecture [6]

We can notice the four steps we followed in classifying the input images. The orange nodes represent the input image, and the purple nodes represent the feature detectors (i.e., kernels). As CNN reduces the image size through its steps, the purple nodes with circles inside represent the reduced convolutions (i.e., feature maps) and pooled feature maps. These nodes are fed to a fully connected ANN as an input layer where each input node represents a pooled feature map. The ANN has hidden layers with hidden nodes in green and an output layer with output nodes in red.

Therefore, the backpropagation of CNN is performed along with the whole structure, not only the ANN. The output layer has multiple output nodes. In CNNs, the SoftMax function is applied to normalize the output probabilities and convert their values between zero and one. Additionally, the SoftMax function is applied with the cross-entropy function [7]. The cross-entropy function can be used with other functions like Classification Error and Mean Squared Error (MSE) to measure the CNN's performance.

1.1.3 CNN Achievements

Convolutional neural networks (i.e., CNNs) evolved rapidly in computer vision areas like image recognition, semantic segmentation, and object detection [8]. The CNNs noticeable performance was due to the availability of labeled datasets and recent hardware capabilities like graphics processing units. After that, multiple structures of neural networks appeared and outperformed existing AI algorithms in classification accuracy and the ability of incremental feature learning.

1.2 Problem Statement

1.2.1 Lack of Transparency in CNN

Despite the rapid success in convolutional neural networks, their decisions lack the interaction between the user and the system since they were assessed per their prediction accuracy rather than the quality of their decisions [9]. They have a black box architecture that prevents the user from understanding the reason behind their decision [10]. CNNs are made of sequential convolutional and pooling layers that incrementally learn features. After that, a fully connected layer is used to map learned features into classification output. This sophisticated structure includes complex computations that are hard to interpret [11]. Therefore, there is an increasing demand for explanations that justify the internal process of making decisions [12]. For instance, explaining

decisions like reducing speed or changing direction could be crucial for system reliability and transparency in autonomous vehicles.

1.2.2 Biased CNN

GDPR was proposed by the European Commission (EC) in 2019 to include principles like fairness and discrimination [13]. AI principles in GDPR aimed to protect personal data privacy and ensure fairness in AI systems decisions. Despite predicting images with high accuracy, CNNs biased decisions could affect people based on gender, race, and age. These decisions may lead to discrimination due to five reasons [14]. The existence of skewed data, which occurs during the data collection. Tainted data, which happens because of errors in the CNN design or parameters initialization. Another reason is the limited features of CNN. An imbalanced dataset with a majority of one label over the others can induce bias. The correlation among sensitive features is another reason for CNNs biased decisions.

1.2.3 Lack of Human Trust Evaluation

Recent XAI models were evaluated based on class discrimination, object localization, and classification accuracy. However, these models lacked the human trust evaluation. Producing high-quality heatmaps is achieved through the model without having humans in the loop. Therefore, the lack of human interaction can impact the reliability and trust in AI systems [15].

1.3 XAI Usefulness

There are various areas where XAI is important. However, we focus on transparency, fairness, and user trust in CNNs. CNNs are black-box models because they block their internal mechanism and computation from the end user [16]. XAI models can mitigate the degree of CNNs opaqueness by highlighting hidden information like correlations between features, gradients, pixels, and neurons [11]. Therefore, the existence of XAI can provide understandable justifications for CNNs

decisions. Furthermore, CNNs decisions can be biased due to model parameters and data skewness. XAI models can support the exploration of biased decisions and help build fair AI systems [17]. By propagating from the output layer back to the input image, XAI models can understand correlated features, detect bias, and identify CNNs fair decisions [18]. XAI studies addressed the collaboration between multiple disciplines like software engineering, HCI, and social science [19]. Therefore "Why a decision was made" is crucial for improving the end user trust [20]. Moreover, incorporating XAI models in high-stake applications proved to improve the user trust in the decision-making process [21].

1.4 Proposed Solution (Augmented Score-CAM)

This model is built on top of the existing Score-CAM [22] and the existing image augmentation techniques. Score-CAM is a masking XAI model that is gradient independent, it generates activation maps by using the increase of confidence instead of gradients. Then it uses each activation map to mask the input image and calculate its score accordingly. In the end, a linear combination is performed to multiply each score with its corresponding activation map and produce the final activation map. While the heatmaps generated by Score-CAM can explain the CNN predicted class with less noise than gradient-based models, they have some drawbacks. Therefore, we propose Augmented Score-CAM (ASC), a novel XAI model that produces a heatmap by combining multiple low-resolution heatmaps, then applying a super-resolution technique to sharpen the final heatmap. For simplicity, we will use the abbreviation (ASC) to represent Augmented Score-CAM and (SSC) to represent Score-CAM. We compare the activation maps for our model ASC and SSC, as shown in figure 2. Our activation map covered the eagle's head, tail, and wings, as shown in figure 2 (b). Meanwhile, in figure 2 (c), the SSC activation map only covered one wing and barely a part of the other wing. Moreover, in figure 2(e), our activation

map covered the entire ship, including the deck, sails, and top mast. However, in figure 2(f), the SSC heatmap only covered most parts of the deck and sails. It did not capture the top mast. The results show that our model enhanced SSC activation maps by adopting the image augmentation approach.

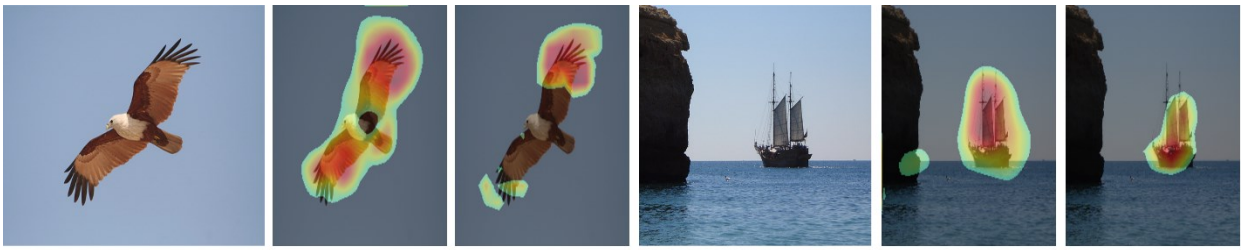


Figure 2. Class activation maps for an eagle and a ship

Augmented Score-CAM visual explanations are evaluated based on their quality. In addition, we will evaluate the human trust in these explanations. We propose a user study with the following research questions:

RQ1: What are the effects of explaining different tasks (i.e., text classification vs. image classification) on the user's trust in the AI system?

RQ2: What are the effects of explaining different AI systems (i.e., black-box, grey-box, white-box) on the user's trust in the AI system?

To address these questions, we present an experimental design for evaluating human trust in two areas, text classification and image classification. The human trust will be evaluated for each area on three levels (i.e., systems), black-box, grey-box, and white-box. The black box and white-box levels have no visual interpretations, while the grey-box level has visual interpretations. Moreover, the black box level lacks transparency, while the white box is easily interpretable by tracing its architecture.

1.5 Contributions

1.5.1 Augmented Score-CAM

- We introduce the Augmented Score-CAM (ASC) model, which integrates image augmentation to generate multiple input image versions. Consequently, each version produces a different heatmap. Since CNNs prediction varies when changing the input image rotation and translation, we believe each augmented Score-CAM heatmap carries useful spatial information.
- We apply a qualitative evaluation metric. We conduct a human study for class discrimination that helps humans select the model with the highest quality of activation maps. This study shows that humans' trust in our activation maps was greater than that generated by Score-CAM.
- We apply multiple quantitative evaluation metrics. For the faithfulness metric, we mask the input image with activation maps and measure the drop/increase in CNN confidence. This evaluation metric shows that our model was more faithful than Score-CAM. For weakly supervised localization, we apply the Intersection over Union method (IoU) to measure how much activation maps capture of the object. Results show that our model captured a higher proportion of an object than Score-CAM. For the sanity check, we prove that our activation maps are sensitive to model randomization. We produce activation maps for our model based on CNN networks with various accuracies.
- Based on our knowledge, ASC is the first XAI model that quantified the correlation between the quality activation maps and the CNN accuracy.
- Based on our knowledge, ASC is the first XAI model evaluated based on its computations by conducting a complexity analysis.

- We performed experiments to highlight the effectiveness of ASC activation maps in detecting bias, which can be useful to fix biased training data and build fair AI systems.
- We propose the use of neural style transfer as a deep learning augmentation method in Augmented Score-CAM.
- Based on our knowledge, this is the first study to evaluate user trust in activation maps.
- Based on our knowledge, this is the first study to compare user trust between two XAI models, LIME interpretation plots, and Grad-CAM activation maps.

1.5.2 XAI Survey

- We conducted a structured search method and significant terms analysis to study the trend of XAI publications over the past years.
- We introduced a novel hierarchical taxonomy for XAI models that interpreted convolutional neural networks.
- We identified the structure, scope, and dependence for each XAI model we reviewed.
- We highlighted the correlations among XAI models in CNNs by building a Sankey chart that maps XAI taxonomy with structure, scope, and dependence.
- 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.

1.6 Document Structure

This document presents two contributions to XAI, the Augmented Score-CAM model and an extensive survey that reviews XAI models that interpret CNNs. Following this introduction, Section 2 presents and discusses the trend of XAI in the last two decades, XAI terminologies, the definition of trust in XAI, a detailed analysis for the XAI models taxonomy, a correlation analysis

of XAI models, a thorough review of XAI evaluation metrics and applications, and a discussion of research gaps. Section 3 presents Augmented Score-CAM intuition, design, implementation, user study design and hypothesis. Section 4 presents evaluation and results by describing qualitative and quantitative experiments. Section 5 discusses the challenges and limitations in XAI models and presents some improvement suggestions. While Section 6 draws the expected outcomes of our study.

2. *Literature Review*

2.1 XAI Research Trend

To highlight the growing attention of explainable AI, we analyzed the number of publications in the past 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. We searched for keywords "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 in two decades. We can observe the hike in the number of 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 high demand for transparent AI systems that deliver faithful decisions, preserve users' privacy, and promote our community.

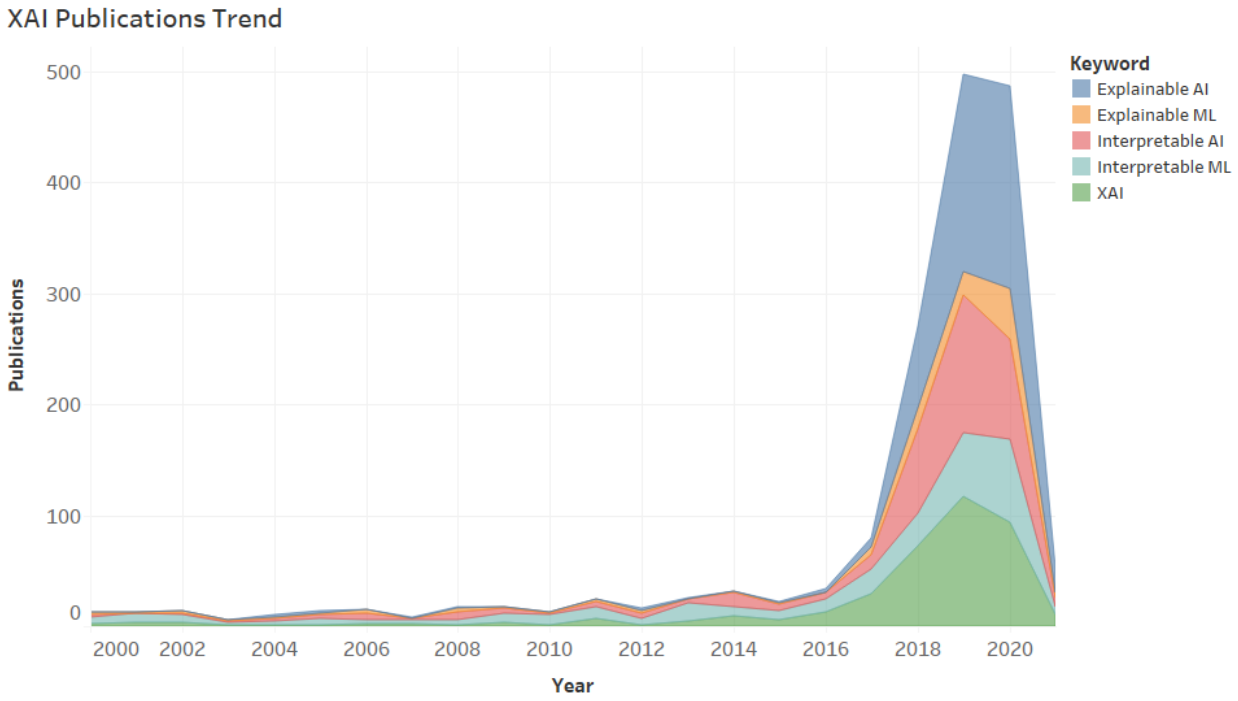


Figure 3. XAI publications trend in the last two decades

Moreover, we highlighted the significant terms in relevant publications in this area. We exported the publications abstracts from 2018 to 2021 using the Web of Science academic database. In addition, we conducted our search by identifying 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”, “interpretable convolutional neural network”. After that, we used the exported text file to build a word cloud that visualizes frequent terms in the publications’ abstracts, as shown in figure 4.

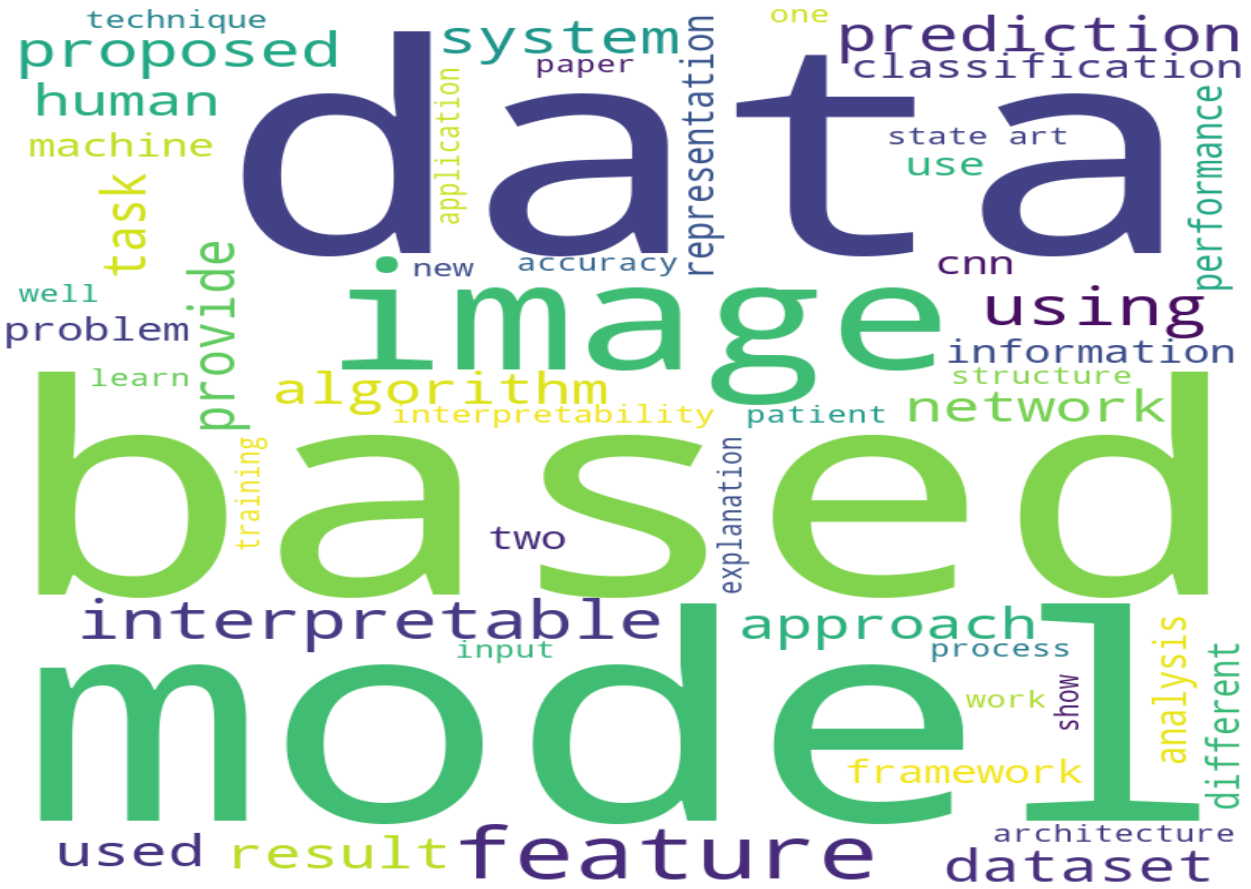


Figure 4. Word Cloud for relevant terms in papers abstracts

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 exciting 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 classification more than other applications like image captioning and visual question answering. Another interesting recurrent term is “feature”. We believe that this term represents the feature maps in CNNs. XAI models relied on features relevance to interpret their prediction or internal representations. The term “human” highlights the increasing demand for CNN interpretations that involve humans in the loop. The terms “technique”, “system”, “algorithm”, “model”, “approach” describes various synonyms for the prototype that was proposed to interpret CNNs.

2.2 XAI Concepts and Terminologies

AI systems can be explainable by nature (intrinsic) or by adding supplementary XAI models (post-hoc) [23]. Explainability was defined as the interface that provides explanations to humans. However, interpretability was defined as the cognition of these explanations [11]. Therefore, XAI models should provide explanations that are interpretable and perceivable [24]. Moreover, XAI explanations are hard to generalize due to different domains and stakeholders (e.g., end-user, AI expert). Therefore, people of various disciplines like computer scientists, HCI, and social scientists need to collaborate to generate explanations with proper levels [9], [10], [21], [11]. For instance, AI experts can receive technical explanations for the model and the training data (i.e., technology-centric). Meanwhile, end-users can receive explanations for the decision made by the model (i.e., human-centric). Despite the increasing acceptance of AI systems, users still lacked the human awareness of understanding their nature [24]. 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 [23]. These explanations can be contrastive,

conversational, selective, and counterfactual [10]. Moreover, XAI explanations should comply with AI principles described in the European General Data Protection Regulation (GDPR) [10], [11]. 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.

2.3 Trust Definition

Various definitions of trust were proposed for different domains [25]. For instance, in management, trust was defined as "The willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other part" [26]. In Sociology, trust was defined as "Subjective probability that another party will perform an action that will not hurt my interest under uncertainty and ignorance" [27]. In AI, trust was defined as "guarantee required by the trustor that the trustee will act as it is expected to do without any supervision" [28]. 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. In our experiments, the trustee is the XAI model (i.e., visual interpretations), and the trustor is the end-user. In addition, previous studies discussed trust properties such as scale, transitivity, context dependence, subjectivity, and asymmetry [28], [29]. In our study, we analyze properties like context dependence and trust strength. In terms of context dependence, trust in decisions is subjective and derived from a given context to achieve a given task (e.g., text vs. image) [30]. In terms of trust strength, we measure the degree of trust in the user decision based on a 5-point Likert scale. If the user gives more points on the scale, the trust degree of strength in the decision increases.

2.4 Trust Measurement in XAI

Measuring trust in XAI models belongs to two categories, trust against adversarial attacks and trust in decision-support systems. For the trust against adversarial attacks, models like LIME [31] indicated that explanations were beneficial in assessing untrusted predictions of a manipulated model. Thus, the user can trust the explanations regardless of the untrusted model classification. In contrast, it was shown that XAI models could provide misleading explanations which impact user trust [32]. These explanations can be reconstructed using adversarial attacks that manipulate features like race and gender. For the trust in decision-support systems, studies investigated the effect of adding explanations to clinical decision-support systems to improve trust and reliability in their decisions [33]. 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 [34]. After that, the tool shows similar images from the database along with their diagnosis. Their experiments proved that the tool increased the physicians' trust in the medical system. Moreover, trust was measured in different levels of decision-support systems like high-stake and low-stake systems [21]. Also, this study analyzed the effect of different AI system levels like black-box, grey-box, white-box on user trust.

2.5 XAI Taxonomy

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) [23], [35]. For example, intrinsic XAI models could embed decision trees in convolutional neural networks to interpret them [36]. For the scope criteria, the XAI model could access the data to provide explanations

(i.e., local) or analyze the network behavior (i.e., global) [10], [23], [37]. 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) [37], [23]. For the dataset criteria, XAI models could explain different data types such as images, text, and tabular data [11]. 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 [21]. 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 [11]. For instance, shallow XAI models existed in interpretable AI systems like linear and logistic regression. RuleFit [38] 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 [39], [40]. 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) [11]. The models in this area adopted approaches like architecture simplification and feature relevance. For instance, Hyper-rectangle Rule Extraction (HRE) applied clustering for generating prototypes for class samples [41]. They interpreted SVM by constructing hyperrectangle rules. For the deep XAI models, multiple models were proposed to interpret neural

networks. For the deep XAI models, studies related to explaining CNNs can be categorized as decision models and architecture models [11]. 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 mainly 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.

2.5.1 Architecture Modification

Explainable models in this category modify the CNNs architecture to improve their interpretability. The modification can replace 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 in CNNs architecture. Global-and-local attention (GALA) was integrated with neural networks like ResNet-50 to produce attention activity maps [42]. 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 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 [43] 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 [44] stacked attention modules inside Inception and ResNeXt to produce attention-aware features. Each attention module (i.e., residual unit) consisted of 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 [45] 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 [46] 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 [47] replaced max-pooling layers with increased stride convolutional layers. The size of the stride was set to 2X2 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 [48] 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 [49] 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 [50] 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. Experiments proved that Attribute Estimation improved the classification accuracy of the Inception-V3 network. However, adding extra layers and generating multiple attributes can increase the complexity of the neural network. Thus, reducing the number of attributes should be carefully considered.

A different approach was to modify the CNN loss function to improve interpretability. Interpretable CNN [51] 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 [52] 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 [53] 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 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 [54] proposed 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 [55] built 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 [56] 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 [57] was proposed to reduce the number of prototypes generated by ProtoPNet [56]. 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 the 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 [58]. The encoder received feature maps in intermediate layers and decomposed them into several object parts. After that, the decoder inverted decomposed feature maps into re-constructed 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. Therefore, there is a need for complexity analysis to analyze the Explainer model computation. XCNN [59] 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) [60] 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 the Deep Fuzzy Classifier (FCM) [61]. 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. Experiments proved that the FCM saliency maps were more interpretable than traditional CNNs. However, the classification accuracy of the FCM model was less than state-of-art networks.

Table 1 shows a detailed 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 proposed a modified CNNs architecture in the training stage 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 (i.e., image) for interpreting the CNN.

***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 [42]	Embedded attention layers in CNNs to generate attention activity maps.	√	X	√	X	X	√
DomainNet [43]	Transformed pre-trained CNN to DomainNet and apply two attention levels	√	X	√	X	X	√

	for extracting object parts and features.						
Residual Attention [44]	Stacked attention modules and integrate with CNNs to generate attention-aware features.	√	x	√	x	x	√
D-Attn [46]	Added attention layer before convolutional layer to learn local/global attentions for user reviews.	√	x	√	x	x	√
ALL-CNN [47]	Replaced max-pooling layer with convolutional layer and increased stride to reduce dimensionality.	√	x	√	x	x	√
NIN [48]	Replaced convolutional layers and linear filters with a micro network to enhance spatial invariance.	√	x	√	x	x	√
Interpretable CNN [51]	Added a loss of feature map to enforce each filter to encode distinct object parts.	√	x	√	x	x	√
AOG [55]	Added a graphical model to CNN to detect the semantic hierarchy of object representations.	√	x	√	x	x	√

CNN Explainer [58]	Added an autoencoder for each feature map to decompose object parts in the image and reconstruct feature maps.	√	x	√	x	x	√
Dynamic-K Activation [52]	Replaced stochastic gradient descent with adaptive activation thresholding to interpret the CNN.	√	x	√	x	x	√
XCNN [59]	Added Autoencoder before the CNN classifier to generate interpretable heatmaps.	√	x	√	x	x	√
Adaptive DeConv [60]	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 [49]	Combined class-specific gate with filters in CNN to assign each filter to one or more classes.	√	x	√	x	x	√

SAD/FAD [53]	Added SAD and FAD loss functions to CNNs to improve their discrimination in face recognition.	√	x	√	x	x	√
ProtoPNet [56]	Added prototype layer after last convolutional layer to assign object parts to various prototypes.	√	x	√	x	x	√
FBI [54]	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 [50]	Added fully connected layers to CNN intermediate layers for generating attributes that enable interpretability of CNN.	√	x	√	x	x	√
FCM [61]	Added a fuzzy classifier layer after the last convolutional layer. The classifier applies clustering	√	x	√	x	x	√

	and Rocchio's algorithm to classify data points.						
ProtoPShare [57]	Shared prototypes between classes to reduce the number of prototypes generated by ProtoPNet.	√	x	√	x	x	√
Loss Attention [45]	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	√

2.5.2 Architecture Simplification

Explainable models in this category rely on the rule extraction approach to extract human interpretable rules from CNNs. 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 into semantic object parts [36]. 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 process could be challenging when dealing with 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 [62]. 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 robustness 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' properties in CNN architecture. Self-Explaining Neural Networks (SENN) [63] 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 lacked the classification accuracy comparison with state-of-art networks. Another hybrid approach was embedding clustering in CNNs to improve their interpretation. CNN-INTE [64] 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 in 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 [65]. 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.

The CAR model [66] was proposed for the CNN compression to make it smaller and interpretable. The CAR model compressed pre-trained CNNs by pruning filters with insignificant contributions to the CNN prediction. Removing redundant filters reduced the number of hyperparameters and improved interpretability. The experiments proved that the CAR classification accuracy was comparable to state-of-art networks. In some networks and datasets, it outperformed state-of-art networks with improving classification accuracy by 16%-25%. However, 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 [67]. 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 the 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 [31]. 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 the CNN interpretability. In image classification, the model highlighted pixels that contributed to class prediction. A promising approach is to utilize 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.

***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 [36]	Decomposed high-level features into object parts by using a decision tree to	√	X	√	X	X	√

	calculate filters numerical contribution.						
SENN [63]	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 [62]	Combined KNN classifier with each layer to measure non-conformality of a prediction in the training stage.	√	X	√	X	X	√
CNN-INTE [64]	Applied clustering on hidden layers to generate meta-level test data and learn classifier results.	X	√	√	X	√	X
Subnetwork Extraction [65]	Extracted semantic information for CNN layers by pruning unimportant channels. Subnetworks are extracted on sample and class levels.	X	√	√	X	X	√

CAR [66]	Pruned all filters with the insignificant contribution in a greedy way to make CNN smaller and more interpretable.	√	x	√	x	√	x
Network Dissection [67]	Extracted semantics of intermediate layers by relying on Broden dataset visual concepts.	x	√	√	x	x	√
LIME [31]	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	√

2.5.3 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 [68] 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 [69]. This model trained a CNN to detect features in the first pooling layer and stored 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 [70] 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 for learning features on the neuron level. However, searching for neurons with maximum action in deep networks 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 [71] 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. In the experiments, 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 [72] 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 object recognition, diabetic retinopathy detection, and question classification. The saliency maps of integrated gradients were clearer than other gradient models. However, there was a lack of quantitative evaluation, like localization and faithfulness.

The DeepLIFT model [73] 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 high 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 [74]. 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.

***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 [68]	Clustered hyper columns in high-level layers to identify interpretable features in the image.	X	√	√	X	X	√
Feature Similarity [69]	Extracted similar features from the training database by applying cosine and Euclidean distance measures.	√	X	√	X	X	√
DGN-AM [70]	Synthesized image similar to the input image and applied backpropagation on	√	X	√	X	X	√

	it to search for neurons with maximum activations.						
Integrated Gradients [72]	Combined gradients implementation invariance with sensitivity to identify important features w.r.t the input pixels.	x	√	√	x	x	√
FER-CNN [74]	Attached a feedback CNN to the original CNN to reconstruct features in a hierarchical approach.	x	√	√	x	x	√
LRP [71]	Decomposed CNN output prediction into feature/pixel relevance scores by applying layer-wise backpropagation and Taylor-type decomposition.	x	√	√	x	x	√
DeepLIFT [73]	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	√

2.5.4 Visual Explanation

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

2.5.4.1 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 [75] 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 was challenging to localize captured objects. The deconvolutional network model (Deconv) [76] 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.

2.5.4.2 Gradient-Based Activation Maps

The Class Activation Map (CAM) [77] 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 [18] was proposed to overcome the CAM drawback. 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 variations of the Grad-CAM model were proposed to enhance object capturing and class discrimination. The Grad-CAM++ [78] 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 w.r.t the input image. The qualitative and quantitative experiments proved that Grad-CAM++ outperformed Grad-CAM in terms of faithfulness, human trust, and object localization. Furthermore, the Smooth Grad-CAM++ model [79] was proposed to improve object capturing and localization. This model combined SMOOTHGRAD [80] 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 [81]. This model adopted the image augmentation approach to generate multiple versions of the input image. 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 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 high 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 [82]. 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 feature. The gradients were used to produce the class activation map. Experiments proved that U-CAM outperformed other gradient-based models in terms of ablation analysis (i.e., uncertainty reduction) and saliency maps quality. Moreover, the U-CAM model 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 [83] 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 [84] 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 to the CNN prediction. Moreover, the human study showed that IBD visualizations were more reasonable than Grad-CAM visualizations. A drawback of IBD is the extraction of labeled object parts. Thus, finding appropriate labels for applications like medical image classification could be challenging. Also, the IBD class activation maps were not quantitively evaluated.

2.5.4.3 Masking Activation Maps

Unlike previous models that adopted the gradient approach, the Score-CAM model [22] 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 their scores. Finally, calculated scores were linearly combined with their corresponding activation maps to generate the final activation map. The qualitative and quantitative experiments showed

that Score-CAM outperformed Grad-CAM in terms of class discrimination, faithfulness, and object localization. A similar masking approach was Mask [85]. This perturbation model was proposed to interpret CNN prediction by identifying significant input regions. The model applied three techniques, replacing the input region with a fixed value, adding noise to the input image, and blurring parts of it. The quantitative experiments proved that the Mask model outperformed CAM, Grad-CAM, and Occlusion models in terms of robustness, localization error, and pointing game. Moreover, the model showed the ability to capture small areas that significantly impacted the CNN prediction. However, the model lacked the evaluation of human trust and faithfulness. Also, there was no comparative analysis for various masking techniques used in the study.

2.5.4.4 Intrinsic Visualizations

The CNN visualization models discussed earlier were post-hoc that attached an auxiliary part to interpret the CNN. However, some visualization models interpreted CNNs by modifying their architecture.

The Teacher-Student model [86] 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 [87] 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. The experiments showed that 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 [88] 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 the 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 the useful features that distinguish each group in these areas. Other models visualized heatmaps for VQA answers to justify their outcome [89]. They applied guided backpropagation, a modified saliency map that removes gradients with a negative contribution to the VQA prediction. The 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 [90]. 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. The experiments showed that 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 who have 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.

***Table 4*. An overview of models which interpreted CNNs by visualization**

Model	Methodology	Intrinsic	Post-hoc	Model-Agnostic	Model-Specific	Global	Local
Saliency Maps [75]	Ranked the input image pixels by calculating gradients' score w.r.t the output class.	X	√	√	X	X	√
Deconv [76]	Reconstructed input from feature maps of a selected CNN layer.	X	√	√	X	X	√
CAM [77]	Added global average pooling layer to calculate	X	√	√	X	X	√

	feature maps contribution in the last conv. layer.						
Grad-CAM [18]	Calculated positive gradients in the last conv. Layer w.r.t the output class.	x	√	√	x	x	√
Grad-CAM++ [78]	Calculated gradient weights for pixels on the last conv. layer w.r.t the output class.	x	√	√	x	x	√
Smooth Grad-CAM++ [79]	Generated multiple noised images and calculated average gradient weights.	x	√	√	x	x	√
Augmented Grad-CAM [81]	Generated augmented images and applied Grad-CAM on each augmented image before combining all activation maps.	x	√	√	x	x	√
Score-CAM [22]	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 [88]	Identified bias in image captioning by adding new	√	x	√	x	x	√

	loss functions during the CNN training.						
CNNV [90]	Visualized learned features in CNN layers by applying acyclic directed graph.	√	x	x	√	x	√
Teacher-Student [86]	Identified important regions in the image by applying an autoencoder for reconstructing the input image.	√	x	√	x	x	√
Eigen-CAM [83]	Extracted learned features by visualizing principal components.	x	√	√	x	x	√
IBD [84]	Added semantic description to generated activation maps along with their labels and ranking.	x	√	√	x	x	√
Mask [85]	Identified important regions in the input image by masking it using noising and blurring techniques.	x	√	√	x	x	√
Hpnet [87]	Extracted a hierarchy of prototypes to describe the relations between class	√	x	√	x	x	√

	activation maps and their class category.						
U-CAM [82]	Added an attention network to LSTM to calculate uncertainty loss, minimize uncertainty, and improve the CNN interpretability.	X	√	√	X	X	√

2.5.5 Correlation Analysis

To analyze the trend of XAI in convolutional neural networks, we visualized the flow among various categories. We analyzed our taxonomy 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, 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 connecting two nodes means more models exist in these two categories (i.e., the right and left 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 generalized 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).

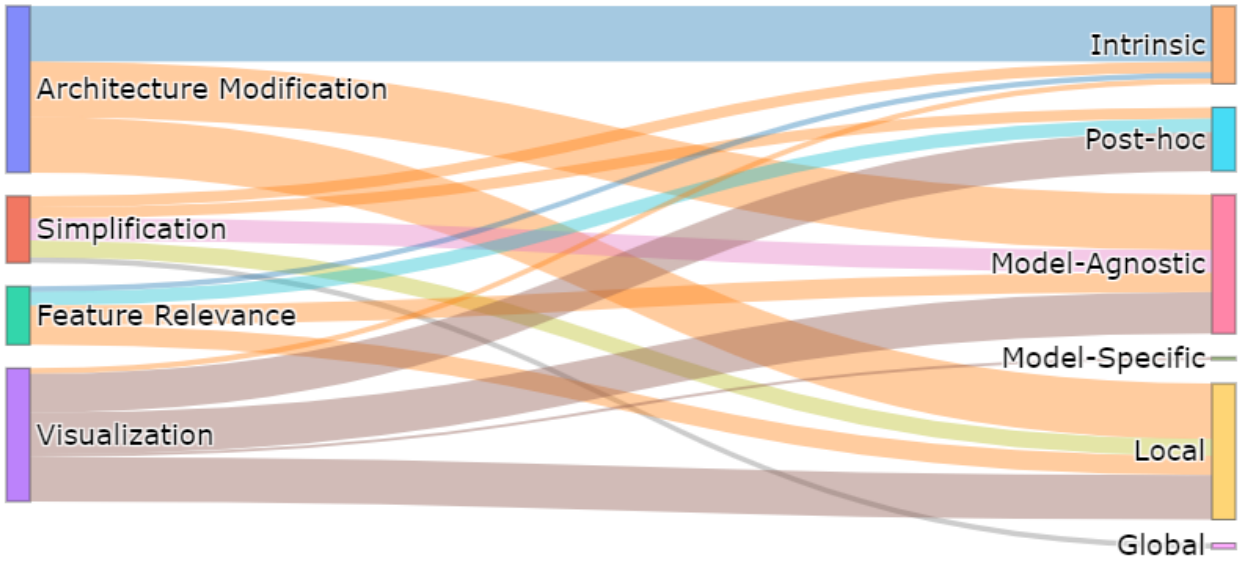


Figure 5. Correlations of XAI models for various taxonomies

However, most simplification models were local and interpreted CNNs by accessing the dataset except CNN-INTE [64] and CAR [66] 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 had to access 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 Teacher-Student [86], HPnet [87], Equalizer [88], and CNNV [90]. Therefore, most visualization models assume the network is trained and tend to interpret CNNs by adding auxiliary parts. Moreover, visualization models were local since they had to access the dataset. Most visualization models could generalize across CNNs except the CNNV model [90], 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 required access to the dataset (i.e., input image).

Additionally, these models could generalize across CNNs with various layers, neurons, and hyperparameters.

2.6 XAI Evaluation Metrics

2.6.1 Visualization

Producing visual interpretations was the most frequent metric used in literature 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 [48], SAD/FAD [53], CAR [66]. Other models visualized class activation maps to evaluate the class discrimination like Dynamic-K [52], XCNN [59], ProtoPNet [56], FBI [54], FCM [61], Loss Attention [45], SENN [63], Subnetwork Extraction [65], CAM [77], Grad-CAM [18], Grad-CAM++ [78], Smooth Grad-CAM++ [79], Augmented Grad-CAM [81], Score-CAM [22], IBD [84], Hpnet [87], U-CAM [82]. Additionally, saliency maps were visualized to reconstruct the input image based on the pixels/features influence on the CNN decision like in Integrated Gradients [72], FER-CNN [74], LRP [71], DeepLIFT [73], Saliency Maps [75], Deconv [76], Mask [85].

Figure 6 shows various saliency maps and class activation maps for single object images [22]. We can notice that the first three visualizations from the left belong to Saliency Maps [75], SMOOTHGRAD [80], and Integrated Gradients [72]. 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, as shown in figure 6. This level of noise can impact the class discrimination in the input image. Therefore, XAI models like Grad-CAM [18], Grad-CAM++ [78], Mask [85], and Score-CAM [22] were proposed to overcome the class discrimination 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.

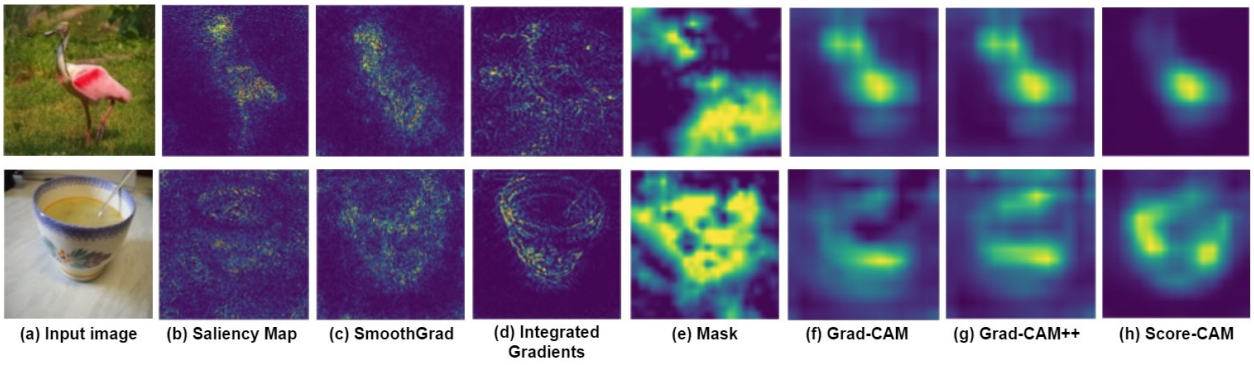
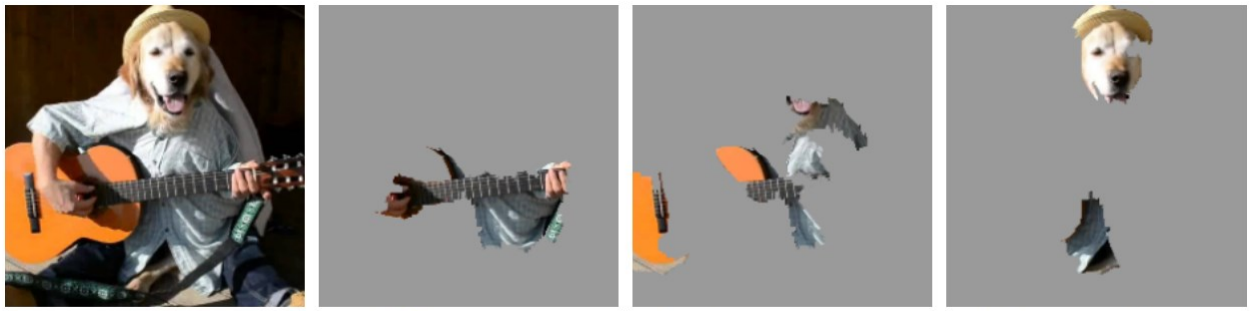


Figure 6. Various visualizations for single object classification [22]

Furthermore, other visualizations, like activation graphs plots in CNN-INTE [64], were provided to test if an instance was wrongly classified. Interpretable units at different layers in CNN Dissection [67] were visualized to check if participants could recognize high-level visual concepts. LIME [31] followed a masking approach to visualize significant pixels that contributed most to the CNN decision. Figure 7 shows the pixels which contributed to the prediction of the top three classes, “Electric Guitar”, “Acoustic Guitar”, and “Labrador” [31]. The grey areas in images represent unimportant pixels. We can notice in figure 7 (b) that LIME relied on the fretboard to decide that the input image was for an “Electric Guitar”. Meanwhile, in figure 7 (c), LIME relied on the dog’s face to decide that it was a “Labrador”. CNNV [90] provided 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 [86].



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 7. Pixels contributing to the prediction of three classes in LIME [31]

2.6.2 Object 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 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 [51], AOG [55], CNN Explainer [58], Dynamic-K Activation [52], XCNN [59], Loss Attention [45], Subnetwork Extraction [65], CAM [77], Grad-CAM [18], Augmented Grad-CAM [81], Mask [85], Eigen-CAM [83]. In figure 8, we can see the bounding boxes plotted to localize an object [91]. 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 [22] adopted an energy-based approach to measuring how much energy of 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.

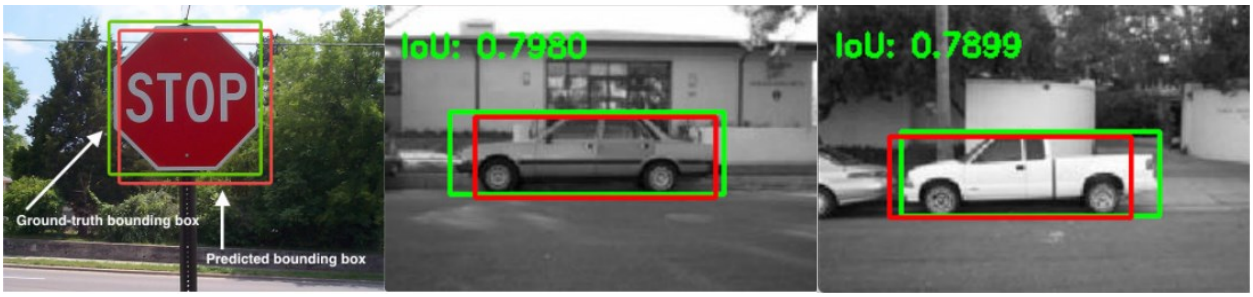


Figure 8. Intersection over Union (IoU) metric [91]

2.6.3 Robustness

Evaluating XAI models' 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 [44] classification error was compared to state-of-art networks to check if the modified CNN improved the original network accuracy. Furthermore, Grad-CAM [18] 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 [49] added a class-specific gate in the last convolutional layer that made the CNN more robust against white-box adversarial examples. Deep KNN [62] 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 [65] proved to be robust against adversarial examples. Eigen-CAM [83] was more robust against DeepFool attacks as it relied on feature extraction, not the CNN architecture. Moreover, the Mask model [85] could detect the difference of learned masks between clean and adversarial images.

2.6.4 Classification Accuracy

Classification accuracy is a quantitative metric that was extensively applied to intrinsic XAI models. For example, DomainNet [43], Residual Attention [44], NIN [48], AOG [55], Dynamic-K Activation [52], CSG [49], ProtoPNet [56], Attribute Estimation [50], Loss Attention [45], Subnetwork Extraction [65], CAR [66], and FER-CNN [74] 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 [61], CNN Explainer [58], ProtoPShare [57], Decision Trees [36], and Hpnet [87].

2.6.5 Other Metrics

Some XAI models applied other metrics for evaluation. For example, Interpretable CNN [51] and CNN Explainer [58] 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 [49] applied mutual information score (MIS) to calculate the correspondence between filter activations and the class prediction. In face recognition, the SAD/FAD [53] model applied verification and identification quantitative metrics to evaluate the performance on face occlusion datasets. Despite using multiple factors to evaluate interpretability, Network Dissection [67] model quantified the measurement of interpretability by aligning the individual hidden units with human interpretable concepts. Similarly, LIME [31]

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 [68] 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 [70] 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 [18], Grad-CAM++ [78], and Score-CAM [22] 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 [88] applied gender ratio error metric in image captioning to calculate the ratio of sentences that belong to "woman" or "man". Teacher-Student [86] applied Area Over perturbation curve (AOPC) metric to measure the CNN classification drop while erasing important pixels from the input image.

2.7 XAI Applications and Tasks

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

2.7.1 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.

2.7.2 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 [46] 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 [31] was applied in sentiment analysis to interpret positive and negative reviews. Unlike images, LIME interpreted textual reviews by visualizing features that mainly contributed to the CNN decision. Figure 9 shows a LIME plot for an insincere Quora question [92].

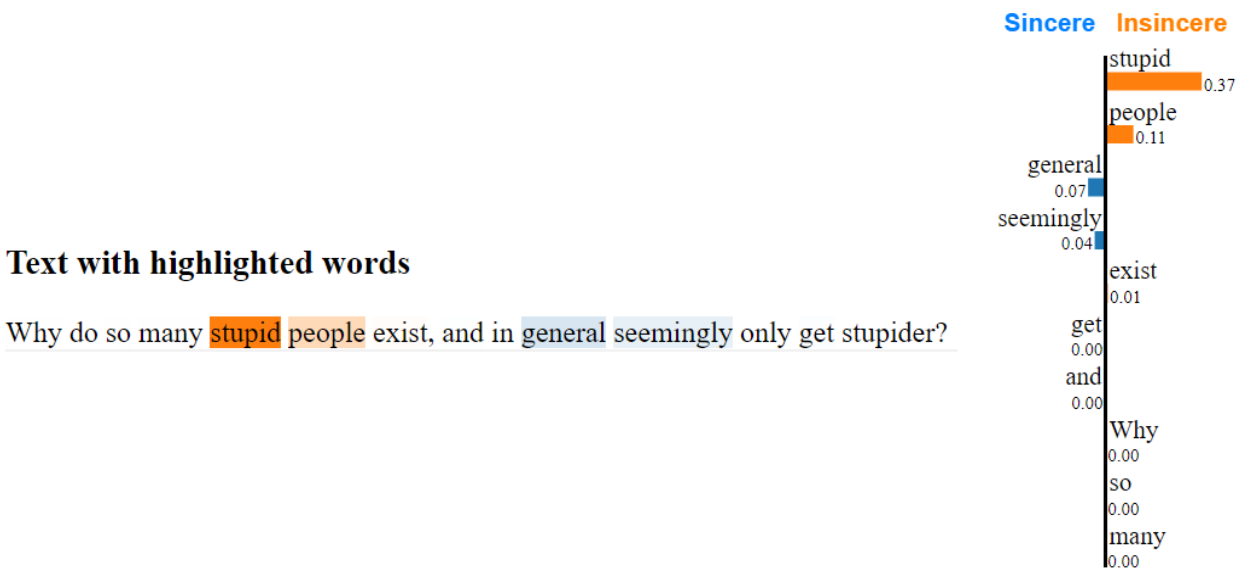


Figure 9. LIME plot for an insincere Quora question [92]

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.

2.7.3 Visual Question Answering (VQA)

Integrated Gradients [72] was used in question classification, neural machine translation, and chemistry models. In question classification, the XAI model identified the type of answer for a given question. Questions could have yes/no, numeric, string, and date answers. Furthermore, visualization XAI models like CAM [77], Grad-CAM [18], and U-CAM [82] were applied in visual question answering. These models generated class activation maps to explain the answer to the question. The activation maps captured the image regions that were most relevant to the answer. Figure 10 shows an example of using Grad-CAM activation maps with VQA applications [18].

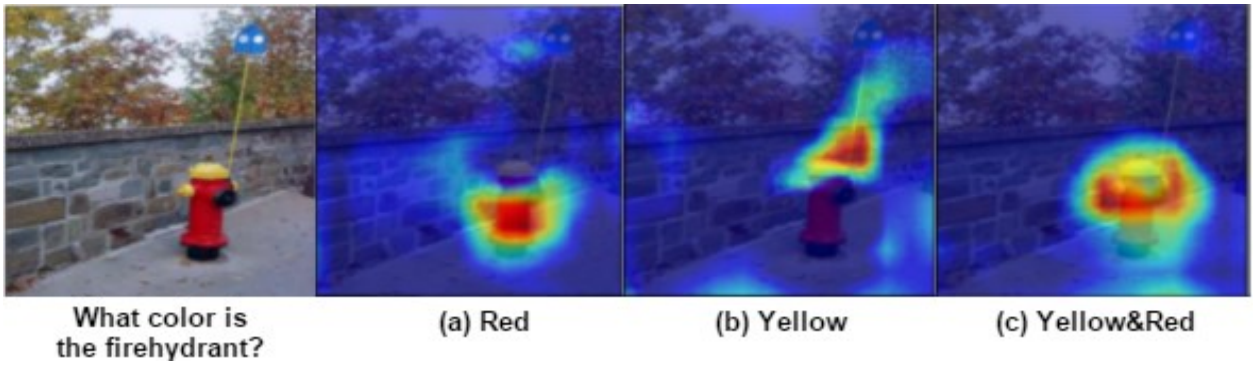


Figure 10. Class activation maps in VQA [18]

The first image is associated with the question, “What color is the fire hydrant?”. We can observe that the activation maps in figures 10 (a), 10 (b), and 10 (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.

2.7.4 Bias Detection

Grad-CAM [18] and Score-CAM [22] 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 [18], as shown in figure 11. 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.

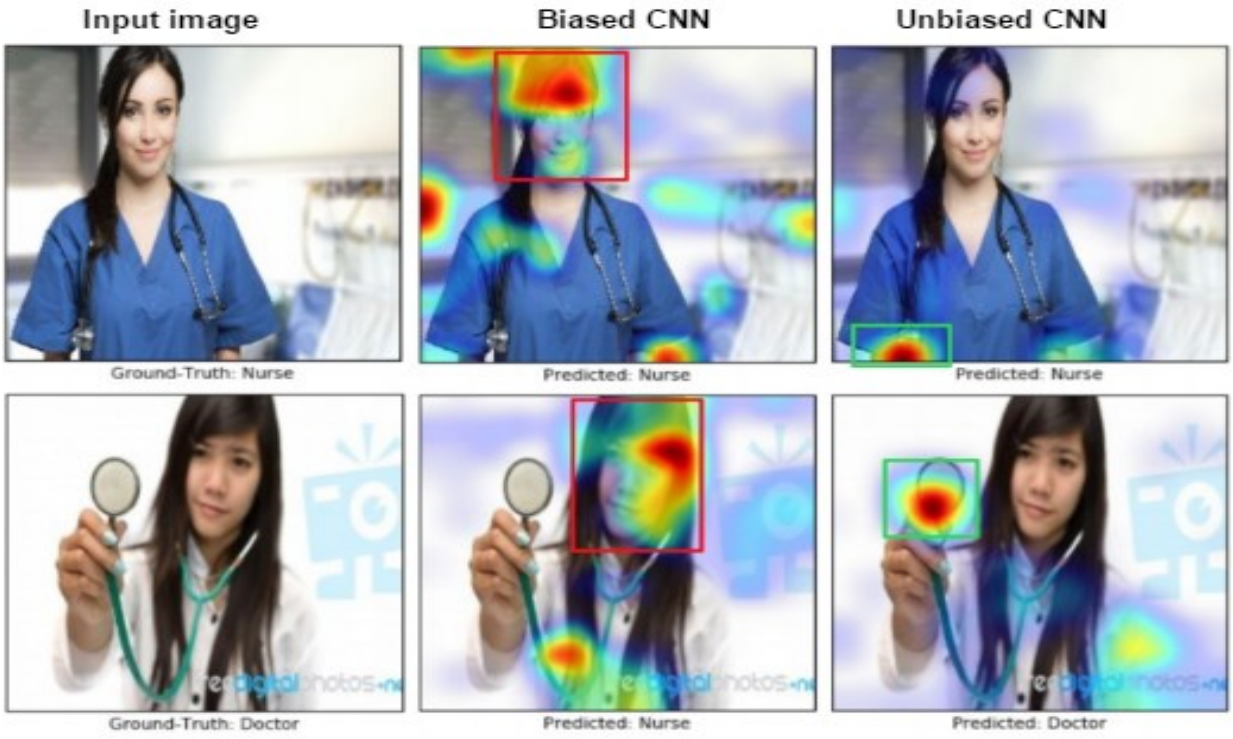


Figure 11. Gender bias detection [18]

2.7.5 Image Captioning

In image captioning, XAI models like Grad-CAM [18], Grad-CAM++ [78], and Equalizer [88] 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 12 shows how Equalizer activation maps detected the gender bias in captions generated for a given image [88]. Figure 12 (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 12 (b) proved that CNN generated the correct caption by looking at the person. Additionally, Grad-CAM++ [78] 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.

Figure 12. Gender bias detection in image captioning [88]

2.8 Research Gaps and Motivation

2.8.1 Object Localization

Score-CAM relies on the increase of confidence to produce the activation maps. After that, it upsamples each activation map to the size of the input image and multiplies each one with the input image. Then, multiplied activation maps are passed through CNN to compute their prediction scores. After that, the scores are combined with their corresponding activation maps to produce the final class activation map (heatmap). However, we notice from figure 2 that Score-CAM captured a portion of the object (e.g., eagle and ship). Therefore, we propose Augmented Score-CAM (ASC), a novel XAI model that enhances the existing Score-CAM [22] in object localization and class discrimination. We design experiments to evaluate the competence of ASC quantitatively and qualitatively. The experiments compare our model's performance with Score-CAM (SSC) since it has outperformed gradient-based models like Grad-CAM and Grad-CAM++ in object localization, class discrimination, and faithfulness.

2.8.2 User Trust in XAI

Previous studies measured user trust for fields like healthcare and aircraft turbine maintenance. However, these studies focused on textual and feature relevance explanations such as LIME [31]. We conclude that trust evaluation has been inadequately investigated in image classification. Visual XAI interpretations like heatmaps were not evaluated. It is essential to check if humans' trust in image explanations is higher than other types. Therefore, by incorporating heatmaps in our user study, we can analyze the effects of feature relevance (i.e., LIME [31]) and heatmap explanations (i.e., Grad-CAM [18]) on user trust. Thus, selecting various XAI models is essential to study trust from multiple perspectives like text and image classification. Besides measuring the trust in two tasks (i.e., text classification vs. image classification), we adopt the same AI-system

level approach [21]. In each task, we analyze the effect of black-box, grey-box, white-box levels on user trust.

2.8.3 Lack of detailed XAI analysis in CNNs

Most of the 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 the literature review we discussed earlier was the first specialized survey that analyzed XAI in CNNs and addressed critical gaps in this area. Moreover, it proposed a novel correlation analysis that provided insights into the taxonomy of XAI models in CNNs. Furthermore, the study will conduct a detailed discussion in section 5. This discussion will explain some challenges that face XAI models and propose future directions to overcome these challenges.

3. Proposed Solution (Augmented Score-CAM)

3.1 Research Design

3.1.1 Image Augmentation Intuition

Our study adopts the image augmentation approach, which improves the CNNs training process and avoids issues like overfitting. The proposed solution and its evaluation are built upon the existing techniques. Image augmentations aim to increase the training set size by using geometric or deep learning augmentation methods. Geometric methods such as color transformation, rotations, and translations, while deep methods include GANs [93] and neural style transfer [94]. In the proposed Augmented Score-CAM (This work was published in Elsevier Knowledge-Based Systems [95]), we apply geometric augmentations such as image rotation and translation. This approach generates 99 augmented images besides the input image. Each augmented image is produced by applying a slight rotation followed by a slight translation. In the rotation, we rotate the input image around the origin clockwise or counter-clockwise by a degree between -28.6° and 28.6°. For the translation, we shift the input image over the *x-axis* or the *y-axis* with a degree between -30° and 30°. The remaining space is filled with black (i.e., a pixel value of zero).

We selected this range of degrees for rotation and translation based on previous augmentation studies. For instance, rotating by small angles can hardly make a difference, while rotating by large angles can impact the image context and neglect some valuable details in the corners of the image. Similarly, translating the image with a large degree along the *x-axis* or the *y-axis* can displace the image and move it out of the frame. Therefore, we adopted a range of degrees between -30° and 30°. Figure 13 shows a sample of our augmented images. The first image is the input image, while the remaining images are rotated and translated images with various degrees. Each saliency map at the bottom of the figure corresponds to the image above. We can notice the change in saliency

maps when there is a change in the object's position (i.e., rotation and translation). Moreover, we can observe that the Score-CAM saliency map CAM1 does not correspond to rotated/translated saliency maps of CAM2, CAM3, and CAM4. Therefore, each augmented image carries unique information and improves the generalization of CAM1 by applying simple transformations to the input image. Our model combines CAM1, CAM2, CAM3, and CAM4 to generate the final saliency map ASC CAM. Moreover, the final saliency map is less noisy than individual saliency maps due to the super-resolution that enhances the saliency map resolution. Previous studies proved that image augmentation methods (e.g., rotating, flipping, cropping) improved the CNN accuracy on various datasets [96]. Similarly, we believe that the image augmentation ASC adopts will improve the quality of activation maps in terms of class discrimination, faithfulness, and object localization.

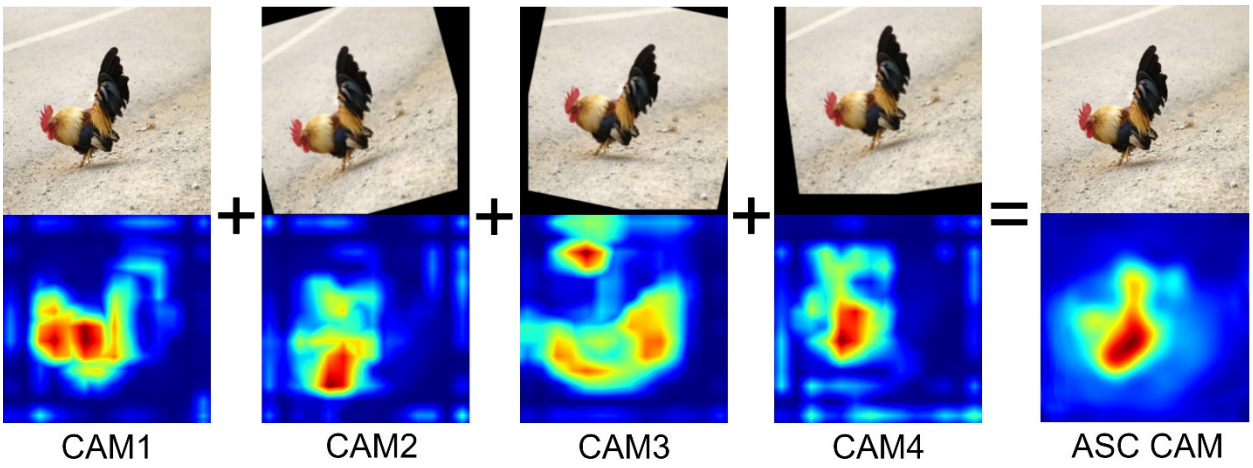


Figure 13. Example of augmented images produced by ASC

3.1.2 Rigid Image Transformations

In our study, we apply image augmentation to the input image. Rigid transformations like rotation and translation displace each pixel in the image from an original location *(x, y)* to a new location (x', y'). Rigid transformations can be rotation, translation, or a combination of both. We apply

transformations to the *XY-plane* since the input image is a two-dimensional array of pixels. For translation, we shift a pixel to a new location by adding a vector $< h, k >$. Assume that the pixel is at the original location *(x, y)*, and the new location after shifting is (x', y'). We add the vector to the original location to get the new location as follows: $x' = x + h, y' = y + k$. After applying these equations, the original point becomes a column vector with the third component of 1 as follows: $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$.

To formalize the relationship between (x, y) and (x', y'), we derive the translation matrix as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & h \\ 0 & 1 & k \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2)$$

For rotation w.r.t the origin, the rotation angle is positive for counter-clockwise direction and negative for clockwise direction. Therefore, a rotation by an angle θ around the origin in the *XY-plane* is shown as follows: $Ro_\theta: \mathbb{R}^2$, where $\mathbb{R}^2$ is the set of real numbers in the cartesian plane. To formalize the rotation of (x, y) by angle θ to become a new point (x', y'), we derive the rotation matrix as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3)$$

Figure 14 explains equation 3 derivation [97]. First, we assume that we are rotating a vector *x* (blue vector) with θ angle. The output of this rotation is the red vector x' (*cos* θ*, sin* θ). Moreover, when rotating a vector *y* with the same angle, the output is the red vector y' (-*sin* θ*, cos* θ). After rotating the two vectors, we get the rotation matrix shown in equation 3. Moreover, rotations and

translations can be combined in one process (i.e., rotation followed by translation). The combination matrix can be derived as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & h \\ \sin\theta & \cos\theta & k \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4)$$

This equation rotates the point (x, y) with an angle θ, then it translates the rotated point in the direction of $< h, k >$. However, if the translation is applied before rotation, the combination matrix will be as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & h\cos\theta - k\sin\theta \\ \sin\theta & \cos\theta & h\sin\theta + k\cos\theta \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (5)$$

Therefore, we can say that rotation and translation are not commutative (i.e., their order is important) since the combination matrix changes when changing their order [98]. In our study, we apply rotation followed by the translation, as shown in equation 4.

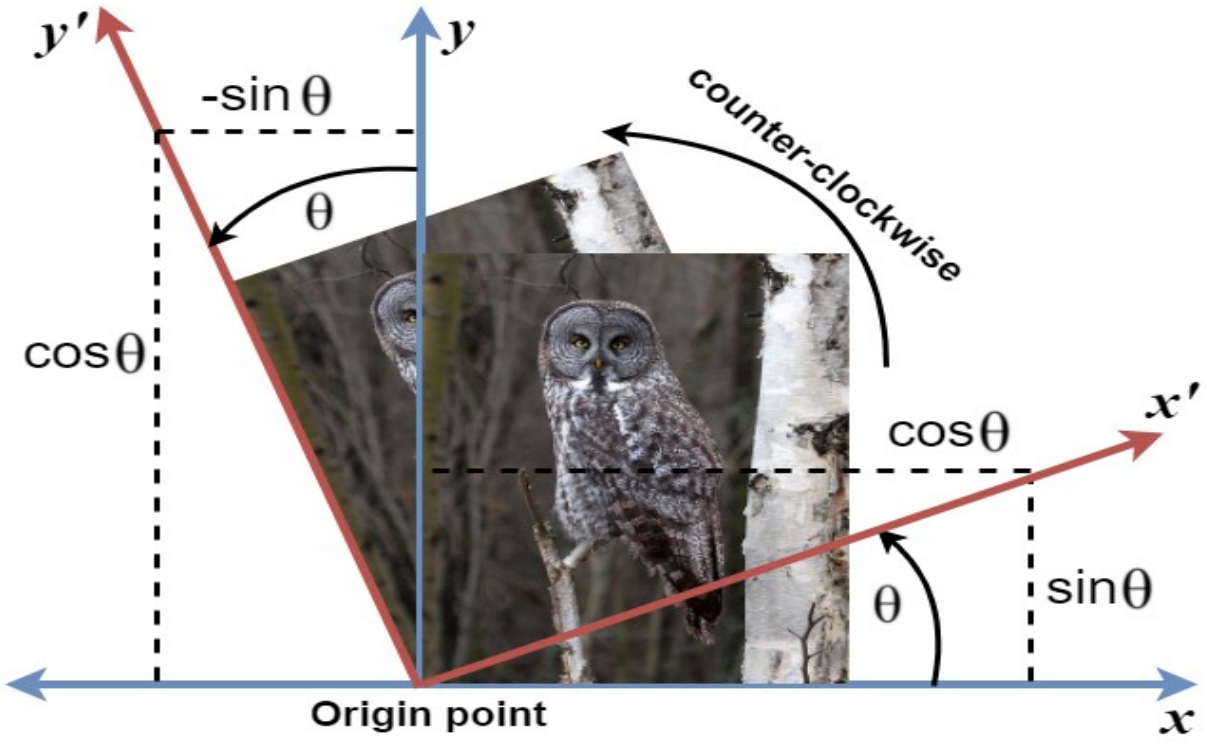


Figure 14. Image counter-clockwise rotation with θ angle [97]

3.1.3 Pipeline

This section describes how we built our approach using the existing Score-CAM [22], image augmentation based on the matrix computation techniques [98], and Augmented Grad-CAM [81]. We utilize the image augmentation feature to produce copies of the input image. Suppose we have an input image with *i* and *j* dimensions. The rotation and translation in equation 4 will be applied to each pixel in the image. Thus, the combination function for the image will iterate through every point (i.e., pixel). Therefore, equation 4 will be applied *i x j* times (i.e., number of pixels in the image). Let us denote *c* as the index of each pixel. The image combination function will be as follows:

$$A = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{c=1}, \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{c=2}, \dots, \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{c=ixj} \quad (6)$$

Therefore, if we have an input image *x*, we generate *p* augmented images with various rotations and translations using equation 6 as follows:

$$x_p = A_p(x), p = 1 \dots, n \quad (7)$$

The augmentation operator A_p performs rotations and translations to all pixels in the input image *x* to generate an augmented image x_p. After that, we generate a separate activation map for each augmented image x_p. The activation maps are computed using the Score-CAM model. Let us assume that we generate 99 augmented images in addition to the input image (n =100). In this case, we execute the Score-CAM model 100 times and generate 100 activation maps accordingly. Based on Augmented Grad-CAM intuition, we believe that each augmented image will have useful spatial information. After having the 100 activation maps, we combine them and apply a super-resolution algorithm to enhance the resolution of the final activation map.

For each iteration, we follow the Score-CAM approach, which first calculates the increase of confidence to get the activation maps as follows:

$$A_l^k = f\left(X \; o \; H_l^k\right) - f(X_b) \tag{8}$$

Where A_l^k is the activation map of the convolutional layer *l* and *k-th* channel. *f* is the model, and *X* is the input. *o* stands for Hadamard Product and H_l^k stands for a vector with the same shape of the baseline input X_b. After we get activation maps, we upsample them to the same size as the size of the input image as follows:

$$H_l^k = Up\left(A_l^k\right) \tag{9}$$

Where *Up* denotes the upsampling process. After that, pixels in activation maps are smoothed and normalized as follows:

$$s\left(A_l^k\right) = \frac{A_l^k - \min A_l^k}{\max A_l^k - \min A_l^k} \tag{10}$$

Then the activation maps are multiplied using Hadamard Product with the input image, and the α score of each activation map is calculated as follows:

$$\propto_k^c = C\left(A_l^k\right) \tag{11}$$

Where $\propto_l^c$ is the score for class *c* and activation map A_l^k. Finally, each score is combined with the corresponding activation map (linear combination) as follows:

$$L_{Score-CAM}^c = ReLU\left(\sum_k \propto_k^c A_l^k\right) \tag{12}$$

We can notice from Score-CAM calculations that this XAI model is a masking algorithm that relies on the increase of confidence rather than gradients. In the end, we will apply equations (8) to (12) for *n* times as mentioned in equation (7). The last step is to combine all Score-CAM activation maps generated from equation (12) for *n* augmented images as follows:

$$L_{ASC}^{c} = \sum_{n} ReLU\left(\sum_{k} \propto_{k}^{c} A_{l}^{k}\right) \tag{13}$$

The pipeline of the proposed Augmented Score-CAM (ASC) is shown in figure 15. We can notice that ASC starts with augmenting the input image by producing additional *n* images using geometric augmentation methods (i.e., rotations and translations). Then, each augmented image is fed into the Score-CAM (i.e., the blue-shaded box). Accordingly, each Score-CAM iteration generates a different activation map with a total of *n* activation maps. Then, activation maps are combined to generate the final activation map, which is passed to the super-resolution to be enhanced and sharpened. The detailed steps of our approach are described in figure 16.

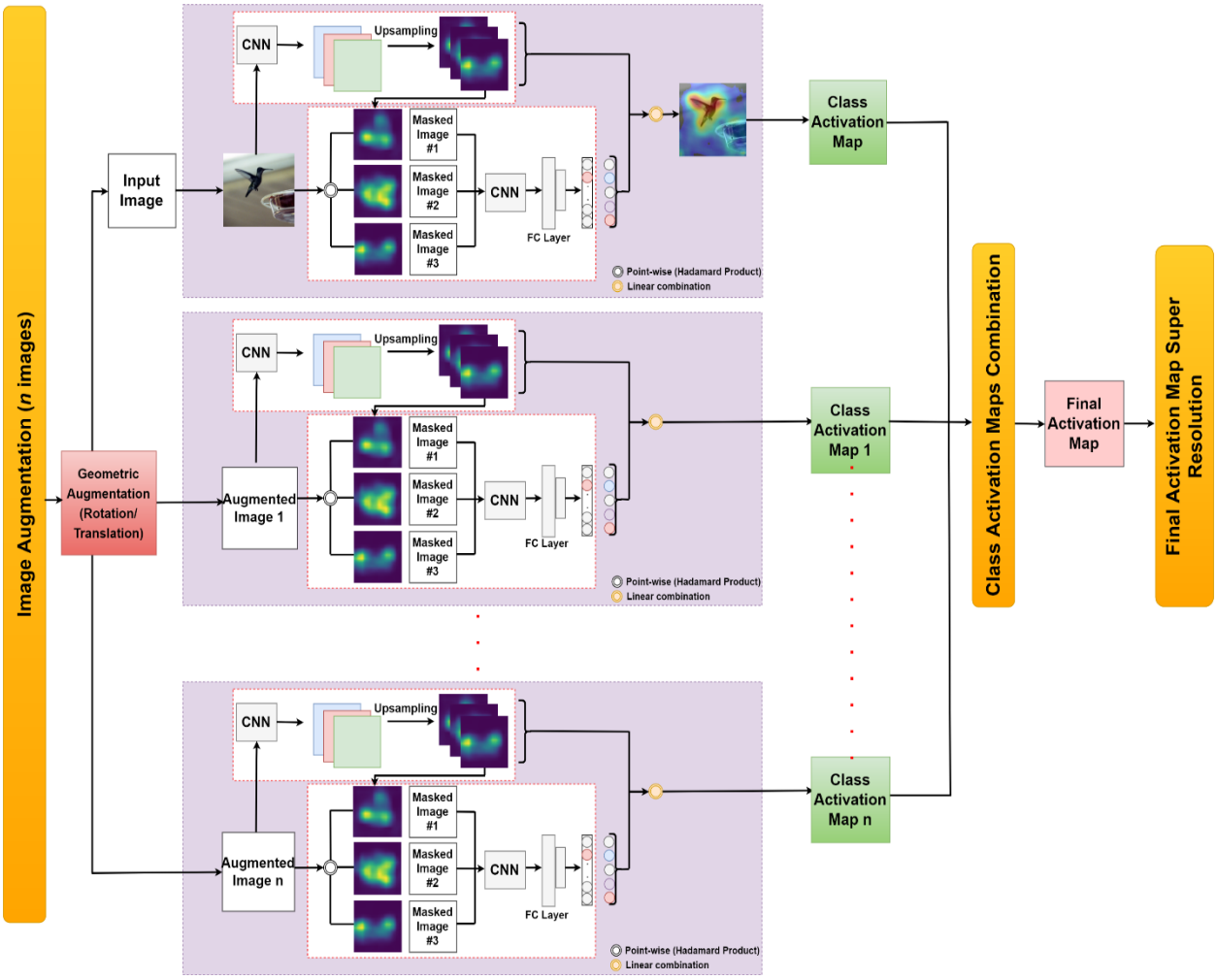


Figure 15. The pipeline of proposed Augmented Score-CAM (ASC)

Algorithm: Augmented Score-CAM (ASC)

Input: Input Image X_b,
Model *f(X)*,
Class index *c*,
Layer *l,*
Number of augmented images *n,*
Minimum rotation *angle-min*,
Maximum rotation *angle-max*,
Minimum translation *shift-min*,
Maximum translation *shift-max*

Output: L^c_{ASC} (final activation map)

1- Initialize *n,* Initialize *l* as last convolutional layer

2- Initialize *angle-min, angle-max, shift-min, shift-max*

3- Apply combination matrix (rotation followed by translation) with various degrees to generate *n* images

for *i in* [1,..,*n*] **do**

//get activation map for image X_i, layer *l*, channel *k*
$A^k_l \leftarrow f(X\ o\ H^k_l) - f(X_i)$

// upsample A^k_l
$H^k_l \leftarrow Up(A^k_l)$

// normalize the activation map
$H^k_l \leftarrow s(H^k_l)$

// calculate score of each activation map
$\propto^c_k \leftarrow C(H^k_l)$

// calculate $L^c_{Score-CAM}$ for image X_i
$L^c_{Score-CAM} \leftarrow ReLU(\sum_k \propto^c_k A^k_l)$
end

4- Calculate the final activation map for *n* images, class *c*
$L^c_{ASC} \leftarrow \sum_n ReLU(\sum_k \propto^c_k\ A^k_l)$

Figure 16. Pseudo code for Augmented Score-CAM (ASC)

3.1.4 Implementation

For building the Augmented Score-CAM, we used Keras [99] implementations from Augmented Grad-CAM [81], image augmentation based on the matrix computation techniques [98], and

Score-CAM [22] codes available on the GitHub repository [100], [101]. For the integrated development environment (IDE), we used Google Colab [102], which offers a Jupyter notebook environment and powerful graphical processing units (GPUs). We executed and evaluated our model by creating Python scripts and notebooks [103]. The Python scripts stored the Augmented Score-CAM and Score-CAM code. The Python notebook passed the input image and class index to both algorithms. Moreover, we used the well-known ImageNet (ILSVRC2012) validation set in our experiments [104]. The ImageNet is an open-source dataset that is extensively used in the field of computer vision. It contains 10,000,000 labeled images with 10,000 classes. The validation set has 50,000 images with 1000 classes. For the neural networks (CNNs), we used three pre-trained models, VGG-16 [105], ResNet-50 [106], and AlexNet [107]. VGG-16 neural network was proposed in 2014 and had 13 convolutional layers and 3 fully connected layers. The network used 2X2 or 3X3 filters. Additionally, it is twice deeper as AlexNet. ResNet-50 was proposed in 2015 and had up to 152 layers. It was the first CNN to use batch normalization. AlexNet was proposed in 2012 and had five convolutional layers and three fully connected layers. It was the first CNN to apply Rectified Linear Units (ReLUs) as activation functions. In our study, we use a combination of the three pre-trained neural networks, VGG-16, ResNet-50, and AlexNet.

3.2 User Study Design

3.2.1 Proposed Models

3.2.1.1 LIME

LIME is a post-hoc XAI model which can be applied in different tasks like text and image classification [31]. In terms of text classification, it accesses the dataset to explain the individual predictions. Once the useful features are selected, LIME plots the features that positively and negatively contribute to the model prediction. In our study, we apply LIME to explain the

prediction of Quora questions (i.e., sincere vs. insincere) [108] by plotting the features (i.e., unigrams) that positively/negatively affect the prediction of each question. Figure 9 shows a LIME plot for an insincere Quora question [92]. The logistic regression model classified the question, and LIME plots the features (i.e., unigrams) that contributed to the prediction. We can observe the term “stupid” showed a high negative score (i.e., insincere). In addition, terms like “general” and “seemingly” had positive scores. Overall, the LIME plot justifies the model prediction by showing that the negativity score was higher than the positivity score.

3.2.1.2 Grad-CAM

The Grad-CAM is a post-hoc XAI model proposed to interpret the prediction of CNNs [18]. The model calculated the gradients of feature maps in the last convolutional layer. The features that positively influenced the class prediction were captured. The calculated gradients are used to build the class activation map, which is upsampled and overlayed on the top of the input image. This overlay highlights the regions (i.e., pixels) where the CNN focused on to make its prediction. Figure 17 shows the Grad-CAM heatmap [109]. Figure 17 (b) represents a defective metal nut heatmap for the input image shown in figure 17 (a). We can notice that the heatmap could visualize the scratch in the metal nut upper side. Therefore, Grad-CAM can explain the CNN prediction by looking at the regions with a strong influence.

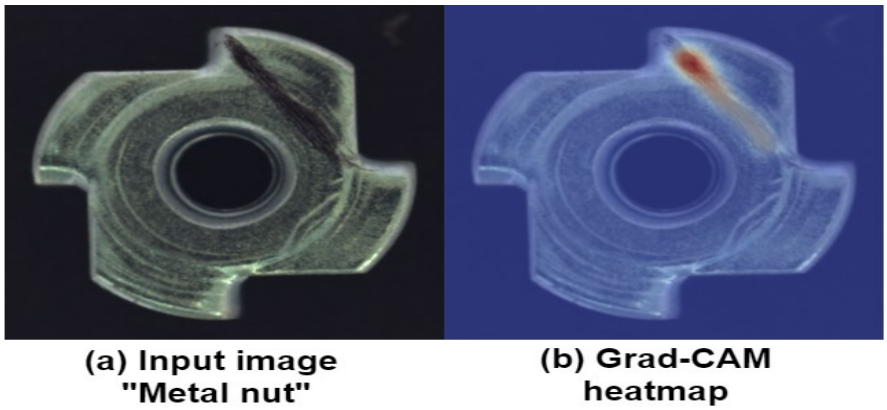


Figure 17. Grad-CAM heatmap for a defective Metal Nut [109]

3.2.1.3 Decision Trees

A decision tree is a famous approach used in classification and regression [110]. The hierarchical structure of decision trees helps to decompose the feature space into smaller subspaces. It has a rule-based structure that consists of lines and nodes. Each line represents a pathway from one node to the other. In addition, the tree has three types of nodes, root, branch, and leaf node. The root node stands for the main feature space, while the branch node is a parent to two or more nodes. The leaf node is the lowest in the tree, which carries the classification result as it has no children nodes. After selecting features, decision trees are fed with rules (i.e., selected feature weights). Then, these rules traverse across the nodes and lines to extract the final prediction result.

3.2.2 Implementation

3.2.2.1 Model Implementation

For the text classification task, we applied LIME to explain individual predictions of Quora questions. Quora is a platform that enables people to ask questions and connect. The Quora dataset [108] had three attributes, qid, question_text, and target. The qid attribute was unique and identified each question in the dataset; the question_text attribute had the textual description of the question. The target attribute had a flag that stored a value of 1 for insincere and 0 for sincere. The dataset was divided into a training set with 1306122 records and a test set with 375806 records. The logistic regression classifier was trained on the dataset to predict if the question was sincere or insincere. We used a Python notebook on Github [92] to explain the classifier predictions applying LIME to plot the features contributions. As shown in figure 9, the insincere features were on the right side, and the sincere features were on the left side of the plot. For the question part shown on the left side of figure 9, we can observe that the darker the feature color was, the higher

the score. For example, feature “stupid” appeared in darker orange since it had a larger score than other features (i.e., score of 0.37).

For the image classification, we applied Grad-CAM to visualize the explained prediction in a defective/good image. The dataset for this experiment was a set of metal nut images that had a mix of defective and good nuts. The MVTec AD dataset [111] had 3629 high-resolution images for training and 1725 images for testing. The dataset comprised 15 objects such as carpet, leather, toothbrush, and wood. For each object, the dataset stored a combination of defect-free and defective images. We selected the metal nut object for our experiment. This object had six classes, good, bent, color, flip, and scratch. Furthermore, the test data was labeled by applying a pre-trained ResNet-50 neural network [106]. In this experiment, we stored a sample of the labeled test data of metal nut images. We used a Python notebook on Github [109] that applied Grad-CAM to visualize explanations of the metal nut predicted label (i.e., good vs. defective). The Grad-CAM heatmap had a JET colormap with colors ranging from blue to red, as shown in figure 17. The red-colored regions were important as they significantly influenced the CNN prediction. In contrast, the blue-colored regions were less important as they had less influence on the CNN prediction.

3.2.2.2 Experiment Implementation

After executing LIME [31] and Grad-CAM [18] scripts and extracting the explanation plots for the text and image classification tasks, we focused on designing the survey interface for both tasks. Each survey had four sections; the first section was for the demographics, while the other sections were for the black-box, grey-box, and white-box levels. For the text classification survey, the black-box section included the question without any explanations, the grey-box section included the question with the LIME plot, and the white-box section included the decision tree with the question features table. For the image classification survey, the black-box section included the

metal nut image without any explanation. The grey-box section included the metal nut image with the Grad-CAM heatmap. The white-box section included the decision tree with the metal nut features table.

Figure 18 shows a sample of the text classification survey for black-box, grey-box, and white-box levels. At each level, the participant will decide if the question is sincere or insincere. After that, the participant provides the degree of trust in his decision on a scale of 5. Figure 19 shows a sample of the image classification survey for black-box, grey-box, and white-box sections. At each level, the participant will decide if the question is sincere or insincere. After that, the participant provides the degree of trust in his decision on a scale of 5. As we can observe from figures 18 and 19, LIME plots and Grad-CAM heatmaps have different mechanisms in explaining the model prediction. LIME plot the features that contribute to the prediction, while Grad-CAM highlights the image regions that contributed significantly to the CNN prediction.

Additionally, for each participant, either the text or image survey will be presented at a time (i.e., between subjects). The purpose is to familiarize each participant with the task he is exposed to (i.e., text or image). For the model explanation levels (i.e., black-box, grey-box, white-box), each participant will be exposed to all levels to measure the effect of the explanation level on the user trust (i.e., within-subjects).

Therefore, our experiment is a mixed-design study with two tasks (text and image) and three different levels of explanations (black-box, grey-box, white-box). For our study, we recruit 36 participants using the Upwork crowdsourcing platform. Eighteen participants are randomly assigned to the text classification survey, and eighteen are randomly assigned to the image classification survey. We target participants who have a basic background in decision trees. This background is required in the white-box level as each participant will read the features table and

use them to traverse the decision tree and make the decision. Participants who agree to the study will receive a link to the survey.

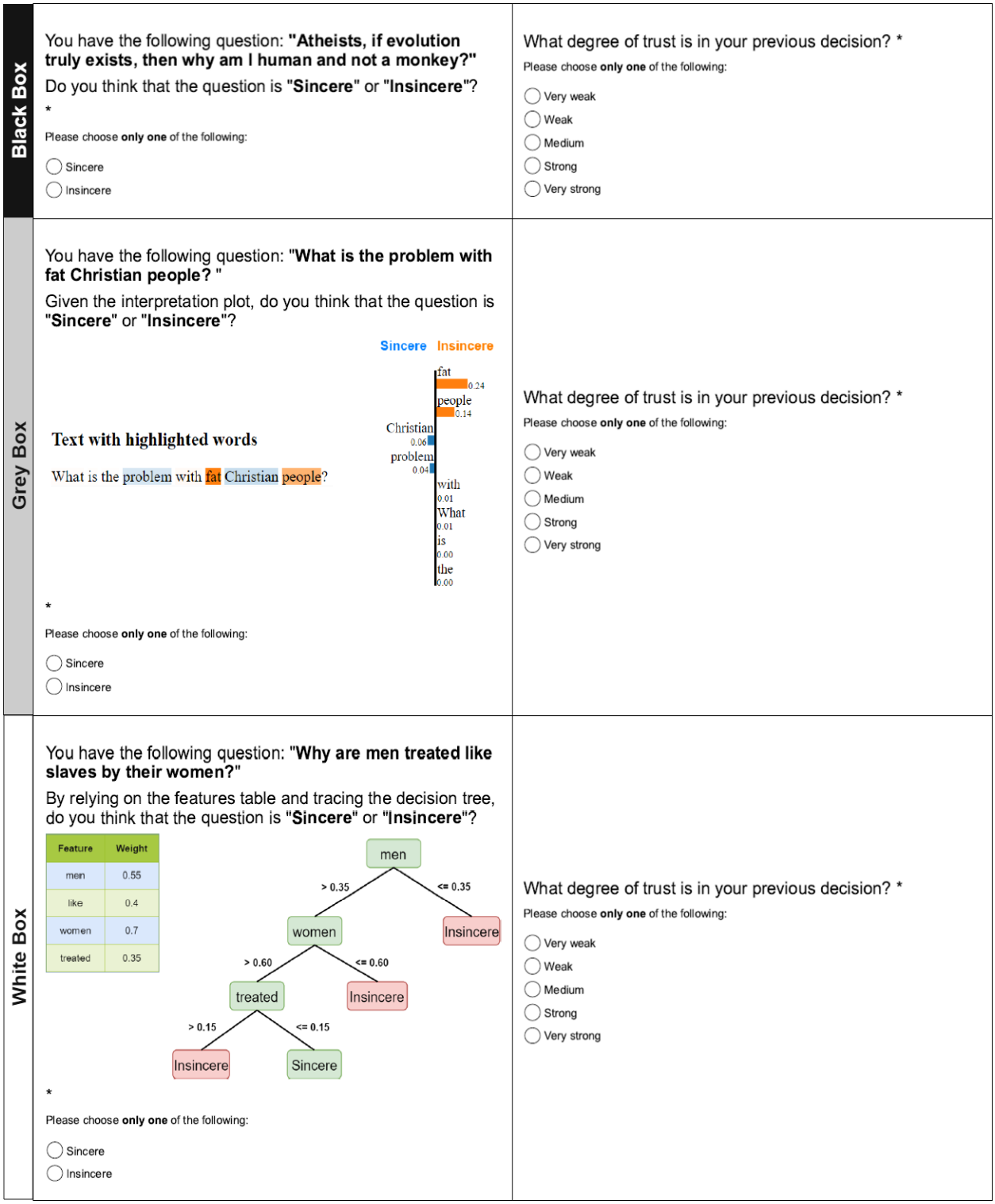


Figure 18. Explanation levels for measuring trust in text classification

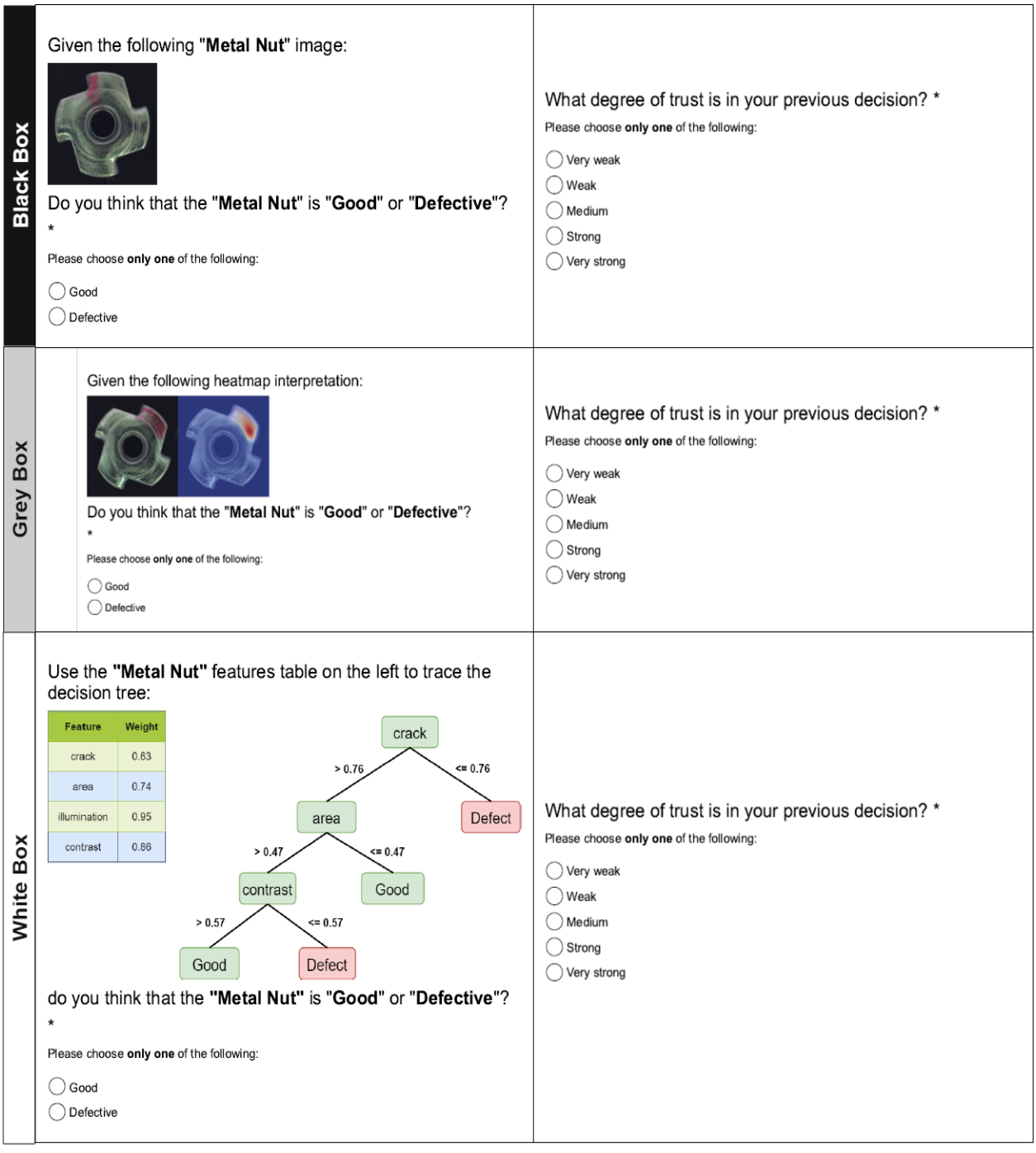


Figure 19. Explanation levels for measuring trust in image classification

We conduct our experiments with the following steps:

- First, we assign one of the tasks (i.e., text survey and image survey) to the participant to

familiarize him with the environment.

- We ask each participant to fill in five questions related to their demographic background. We can find some correlations between decisions made and demographics.
- After that, each participant will answer questions in three levels, black-box, grey-box, and white-box. Each question asks the participant to select a decision, then express the degree of trust in this decision, as shown in figures 18 and 19. Additionally, we reduce the order effect among the three explanation levels by applying counterbalance with Latin square, as shown in table 5. Since we have six unique combinations (task vs. level) and 36 participants, we repeat each order six times (i.e., 36 participants/6 combinations).

***Table 5*. Latin square counterbalance for tasks and levels**

Task	Explanation level		
Text classification	Back-box	Grey-box	White-box
Text classification	Grey-box	White-box	Black-box
Text classification	White-box	Black-box	Grey-box
Image classification	Back-box	Grey-box	White-box
Image classification	Grey-box	White-box	Black-box
Image classification	White-box	Black-box	Grey-box

3.2.3 Hypothesis

In this study, we assume that adding explanations to AI systems can improve user trust in their decisions. Moreover, we assume that black-box models lack transparency and provide no explanations [11]. Thus, we suppose that the trust level in grey-box and white-box models will be higher. Therefore, we derive our first hypothesis:

H1: The use of explanations (i.e., grey box) improves the user trust in the decision-making process more than models with no explanations (i.e., black box).

We believe that participants will adjust their trust level positively or negatively based on the explanation level. Therefore, we assume that simple explanations generated by white-box models will be more acceptable than grey-box models' explanations. Meanwhile, grey-box explanations will be more acceptable than the absence of explanations in black-box models. Accordingly, we derive the following hypothesis:

H2: The effect of the explanation level on user trust will be significant.

Previous studies highlighted the correlation between the explanation quality and the trust level [112]. We rely on this factor to test if the user trust level varies through different tasks of explanations (i.e., text classification vs. image classification). The purpose is to test which explanations have a higher trust degree (i.e., LIME vs. Grad-CAM). Therefore, we derive the following hypothesis:

H3: The effect of the classification tasks on user trust will be significant.

3.2.4 Evaluation Criteria

In table 6, we show the independent and dependent variables in our study. We have two independent variables, task and explanation level. The task variable has two levels, text and image, while the explanation level variable has three levels, black-box, grey-box, and white-box. Moreover, the table shows two metrics used to measure user trust, effectiveness, and trust level. The effectiveness is measured by calculating the error rate in the user decision (i.e., mismatch score). The decision made by the user is compared to the ground truth to detect if it was correct or incorrect. After that, the average of incorrect answers is calculated per task and explanation level. For the trust level measurement, we use a five-level Likert scale in which the user expresses the

degree of trust with five responses, “Very weak”, “Weak”, “Medium”, “Strong”, and “Very strong”. Since the Likert scale has a rank order of degrees, we calculate the average mean for the 15 questions in both surveys (i.e., text and image). Also, we calculate the frequency for each response per task and explanation level.

***Table 6*. Evaluation criteria and metrics**

Independent Variables	➢ Task (Text, Image) ➢ Explanation Level (Black-box, Grey-box, White-box)
Dependent Variables	User trust
Decision effectiveness	Error rate
Trust level	5-point Likert scale

4. Evaluation and Results

4.1 Class Discrimination

In our experiments, we denote Augmented Score-CAM as ASC and Score-CAM as SSC. We conducted a user study to measure the performance of ASC against SSC in terms of class discrimination. This user study entitled "Augmented Score-CAM: High-resolution visual interpretations for deep neural networks" was granted clearance by the Carleton University Research Ethics Board-B (CUREB-B) with a clearance ID of 114825. Moreover, we completed the Tri-Council Policy Statement (TCPS2) to conduct this user study. We randomly selected the images from ImageNet (ILSVRC2012) validation set. We used VGG-16 in this evaluation. We generated two saliency maps for each annotated image, one using ASC and the other using SSC. We show these saliency maps to 18 participants and ask them, "Which explanation better describes the input image?". Figure 20 shows a sample of the survey for an input image of a dragonfly. The participant viewed the input image then selected one of the visualizations (ASC vs. SSC). The participant could also select "Both images are the same" if it were believed that there is no significant difference between the two saliency maps. Moreover, both saliency maps were anonymized and shuffled in each question to ensure non-bias in their responses. In this survey, the better visualization that describes the input image corresponds to the algorithm (ASC or SSC) that better discriminates the class index (i.e., the object being classified). In the survey, we selected 158 images, half of animals and half of miscellaneous objects. Since we had 18 participants (i.e., responses) per image, we normalized the answers of each image to 1. For example, among the 18 responses, if 11 responses chose ASC, two responses chose SSC, and five responses chose "Both images are the same", the scores would be 0.61, 0.12, and 0.27, respectively. After that, the normalized scores are added w.r.t the total score, which is 158. ASC achieved a score of 97.4

(61.67%) compared to 32.2 (20.39%) of SSC. The score for “Both images are the same” was 28.3 (17.93%), as shown in table 7. Therefore, these results prove that ASC improved SSC in class discrimination, thus, improving the human trust in the CNN model.

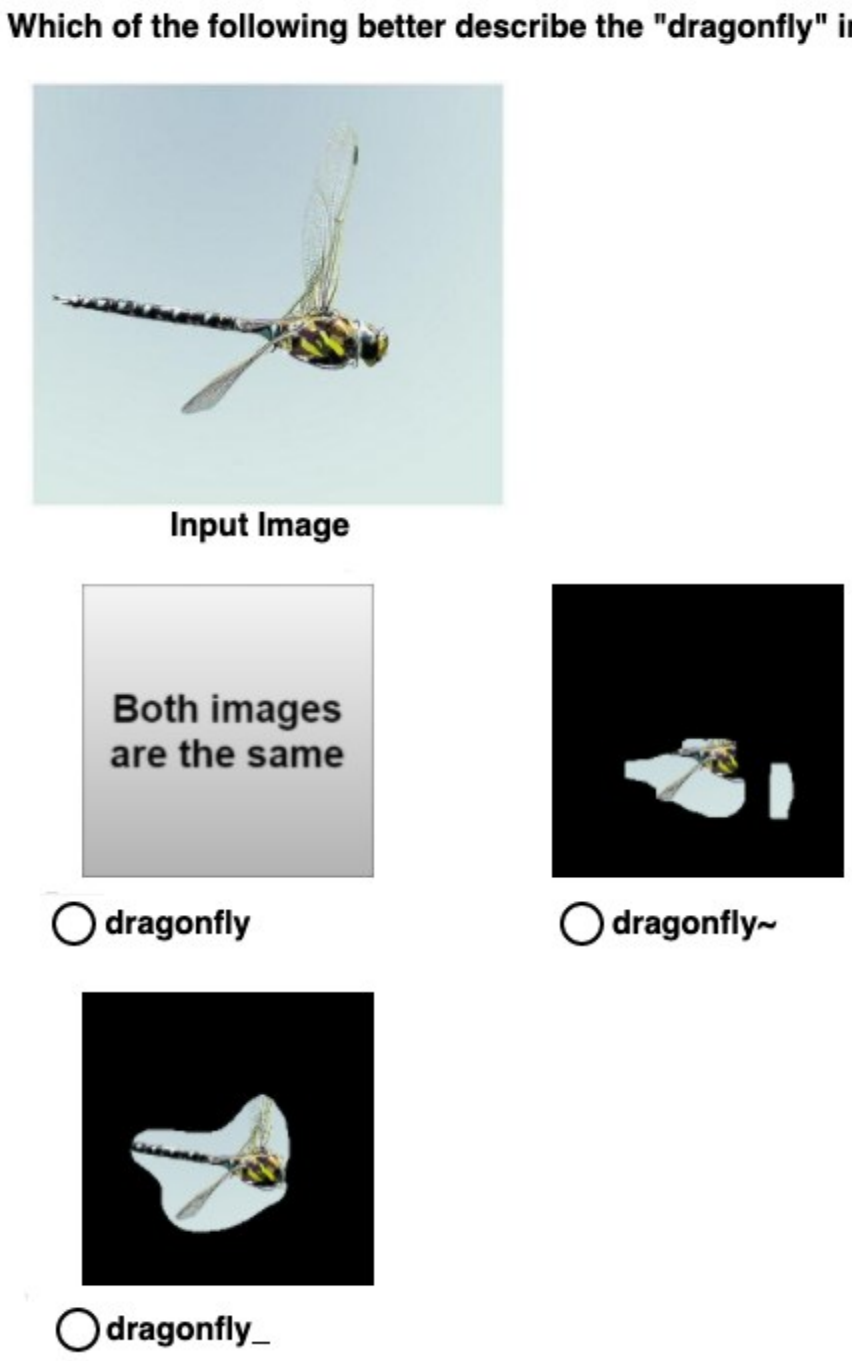


Figure 20. Survey interface for evaluating two saliency maps

Moreover, it was evident from the survey visualizations that ASC was better than SSC in discriminating multiple objects of the same class. Figure 21 shows two examples from the survey.

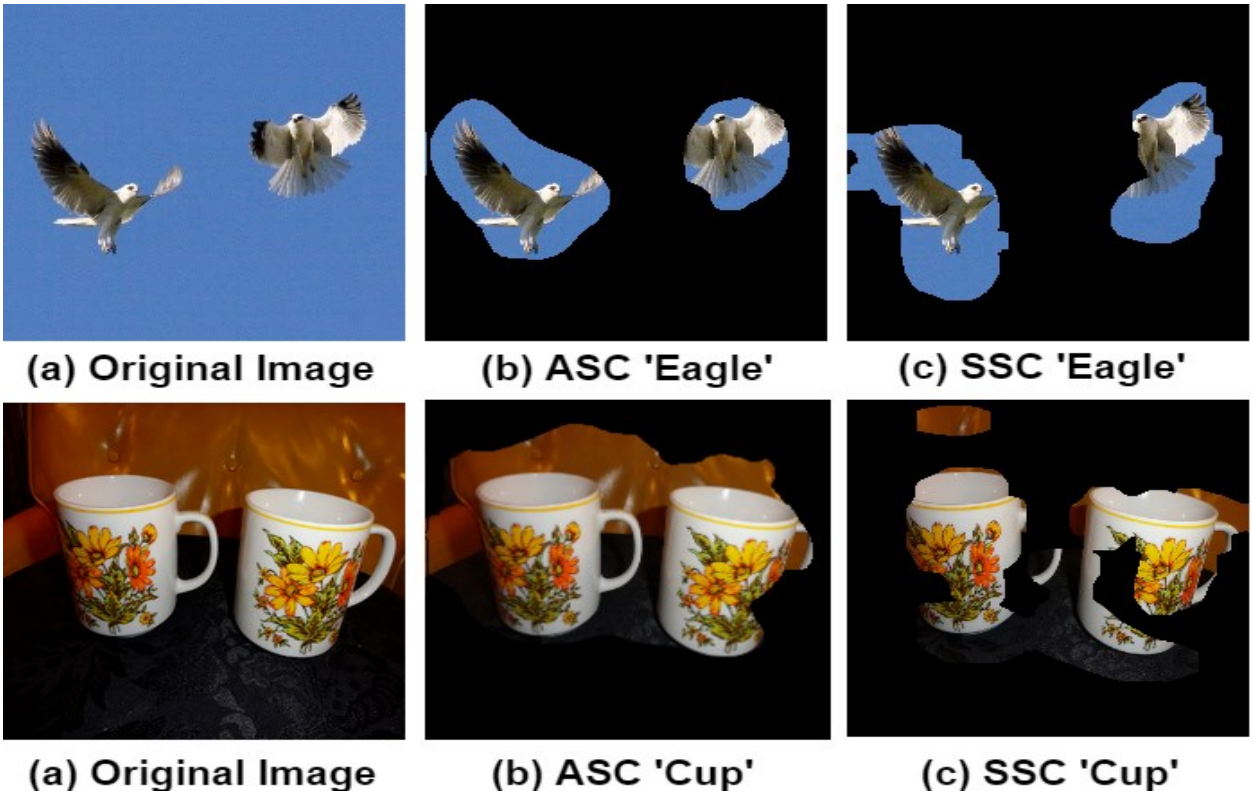


Figure 21. Multiple objects class discrimination for ASC and SSC

The first example is an image of two eagles. It is observed that ASC could capture a large portion of both eagles with most of their bodies. While for SSC, half the body of the eagle on the right is not captured. Moreover, SSC did not capture the left eagle's wing. The second example is for an image of two cups. ASC could capture both cups, while SSC did not capture some areas.

***Table 7*. User study results for ASC and SSC**

Metric	ASC	SSC	Same
Average score percentage *(Higher is better)*	**61.67%**	20.39%	17.93%

4.2 Faithfulness

We generated masked images shown in figures 20 and 21 by multiplying the input image with the saliency map. Given an input image *X*, the masked image E^c is calculated as follows:

$$E^c = L^c \circ X \quad (14)$$

Where c is the class index and L^c is the generated saliency map. Figure 22 shows an example of this process. In figure 22 (b), ASC generates the saliency map, then less important pixels are removed to produce a more focused saliency map, as shown in figure 22 (c). After applying equation (14) on figure 22 (c), we have figure 22 (d), which is the masked image.

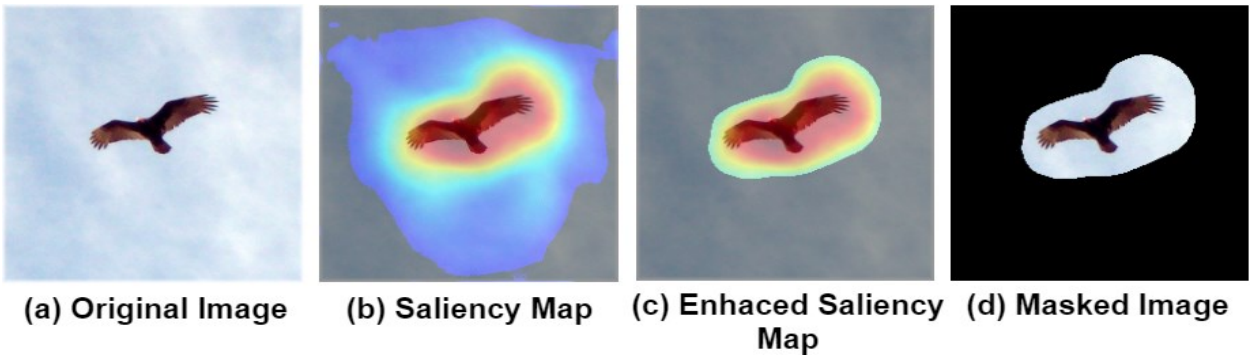


Figure 22. Process of masking the input image with saliency map

To measure the faithfulness of our model, we use three pre-trained CNNs, VGG-16, AlexNet, and ResNet. We feed three versions of a selected image for each neural network, the input image, the ASC masked image, and the SSC masked image. For this experiment, we use 336 images from ImageNet (ILSVRC2012) validation set. We feed these images along with their masked versions (i.e., 336 X 3) into the three pre-trained CNNs. The prediction confidence is calculated for each image. After that, we use three metrics to evaluate faithfulness, average drop, average increase, and win percentage. The average drop is the CNN confidence drop cause by removing parts of an image while masking it. This drop is expected since there could be a change in confidence between

the input and the masked images. Normally, masking will remove some of the input image contexts. If the average drop of ASC was less than SSC, this means that the ASC saliency map provides more relevant parts of the object. The average drop is calculated as follows:

$$Average\ Drop = \sum_{i=1}^{N} \frac{\max(0, Y_i^c - O_i^c)}{Y_i^c}\ X\ 100 \tag{15}$$

Where Y_i^c is the confidence score for class *c* and input image *i*. O_i^c is the confidence score for class *c* and the masked image. The average drop is calculated for each image and averaged over the dataset.

Meanwhile, the average increase often happens when the input image has some distractions, which are mitigated by the masked image. This metric calculates the frequency of images with increased confidence. The average increase is calculated as follows:

$$Average\ Increase = \sum_{i=1}^{N} \frac{1\ Y_i^c < O_i^c}{N}\ X100 \tag{16}$$

Where 1 is returned if the condition is true (i.e., the confidence of the input image is less than the masked image). Finally, the win percentage metric is a complement of previous metrics. It is the number of times the ASC confidence drop was less than the SSC confidence drop. The faithfulness experiment results for three pre-trained CNNs are shown in tables 8, 9, and 10. The tables show that the ASC average drop was less than the SSC average drop in all pre-trained CNNs. Moreover, our model's average increase was higher than the average increase of SSC in all pre-trained CNNs. In addition, the ASC win percentage was higher than the SSC win percentage in all pre-trained CNNs. Among the pre-trained CNNs, the ASC performed better on ResNet in terms of average drop and win percentage. Overall, the results support the ASC outperformance and prove that our model effectively captures the most relevant context for the given image.

***Table 8*. Faithfulness results for ASC and SSC on VGG-16**

Metric	ASC (Augmented Score-CAM)	SSC (Score-CAM)
Average Drop % *(Lower is better)*	**57.02**	**69.37**
Average Increase % *(Higher is better)*	**13.39**	**11.0**
Win % *(Higher is better)*	**64.88**	**35.12**

***Table 9*. Faithfulness results for ASC and SSC on AlexNet**

Metric	ASC (Augmented Score-CAM)	SSC (Score-CAM)
Average Drop % *(Lower is better)*	**82.96**	**86.24**
Average Increase % *(Higher is better)*	**9.82**	**6.84**
Win % *(Higher is better)*	**60.42**	**39.58**

***Table 10*. Faithfulness results for ASC and SSC on ResNet**

Metric	ASC (Augmented Score-CAM)	SSC (Score-CAM)
Average Drop % *(Lower is better)*	**54.82**	**68.65**
Average Increase % *(Higher is better)*	**11.3**	**6.84**
Win % *(Higher is better)*	**66.66**	**33.33**

4.3 Object Localization

In this evaluation, we show the effectiveness of our model in localizing objects. We use 373 images from ImageNet (ILSVRC2012) validation set since it has labeled bounding boxes for each image. We generate bounding boxes using pre-trained VGG-16 CNN. To draw a bounding box around the class, we follow the same approach of Grad-CAM [18]. We binarize the saliency maps with a threshold value of 15%. After that, we draw a rectangle around the largest segment. Moreover, our model was never trained on labeled bounding box images. We have two saliency maps for each

image, the ASC saliency map, and the SSC saliency map. We apply the Intersection over Union (IoU) metric on each saliency map (i.e., 373 X 2). The IoU metric is calculated as follows:

$$IoU_I^c(\delta) = \frac{S(internalpixels)}{I(bounding\ box) + S(externalpixels)} \quad (17)$$

Where *c* is the class, *I* is the input image, δ is the threshold value of 15%, *S* is the saliency map. *S (internal pixels)* is the number of non-zero pixels that lie inside the bounding box. *S (external pixels)* is the number of non-zero pixels that lie outside the bounding box. *I (bounding box)* is the ground truth bounding box for a given image *I*. Moreover, we selected images with one object and one bounding box in this evaluation, as shown in figure 23.

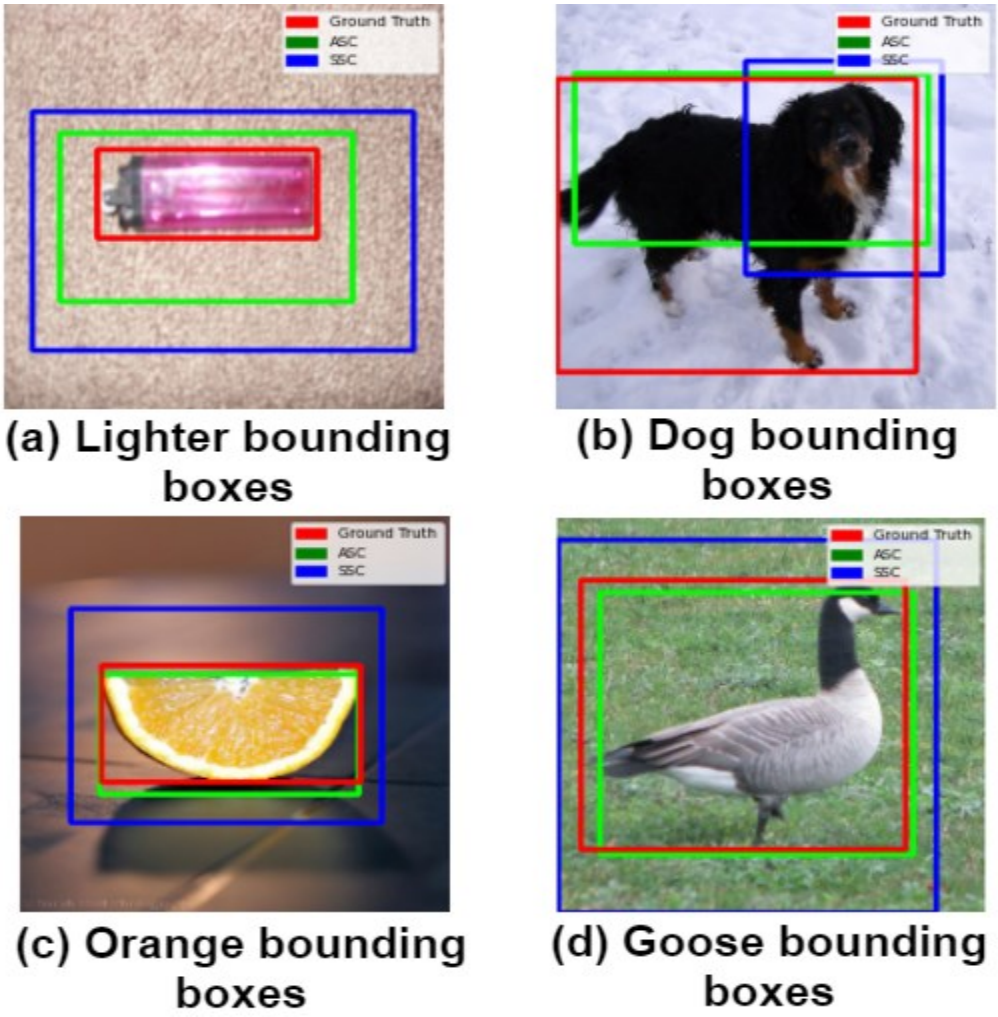


Figure 23. Object localization for ASC and SSC

The red rectangle stands for the ground truth bounding box. The green rectangle stands for the ASC bounding box, while the blue rectangle stands for the SSC. The object localization evaluation was measured by comparing IoU values for both ASC and SSC algorithms, as shown in table 11. The IoU value of the ASC algorithm was 0.56, while the IoU value of the SSC algorithm was 0.49. The higher value of IoU makes it evident that our model has improved the object localization, as shown in figure 23. Also, we calculated the number of times the IoU value of ASC was higher. Our model expressed better localization in 66% of the images, while SSC was better in 34%.

Table 11. **IoU results for ASC and SSC**

Metric	ASC	SSC
Average IoU value *(Higher is better)*	**0.56**	0.49
IoU percentage *(Higher is better)*	**66%**	34%

4.4 Sanity Check

We perform a sanity check evaluation to prove that ASC saliency maps are dependent on the CNN model and the dataset (i.e., input image). This evaluation aims to check if the ASC saliency maps are sensitive to the randomization of model parameters and the input image. We believe that the quality of the saliency maps could be correlated with the accuracy of the CNN model.

4.4.1 Model Randomization

For model randomization, we use five pre-trained VGG-16 models with various prediction accuracies. First, we create five versions of the pre-trained VGG-16 model. After that, we randomly permute the weights of the model. For each model, we shuffle the weights with a higher level of noise. Table 12 shows the five VGG-16 models with their noise levels. Additionally, we use the ImageNet (ILSVRC2012) validation set to calculate the accuracies of the five models. We

can notice from table 12 that the accuracy decreases when adding more noise to the model weights. The next step was to generate ASC saliency maps for each model in the table using the same image every time. Interestingly, we observed that the variation in the model's architecture impacted the quality of saliency maps. Saliency maps became less focused when the model accuracy decreased.

***Table 12*. Five VGG-16 models with their accuracies**

Model	Noise level	Accuracy
VGG Model 1	0 (no noise)	86%
VGG Model 2	0.15	82%
VGG Model 3	0.2	59%
VGG Model 4	0.25	46%
VGG Model 5	0.35	6%

In figure 24, we can notice that the ASC saliency map successfully captured the dog's face, as shown in figure 24 (b). However, the saliency map did not correctly capture the dog when the model accuracy decreased, as shown in figure 24 (f). The same approach applies to the cat image where the saliency maps were successful in figure 24 (h) and were less successful in figure 24 (l).

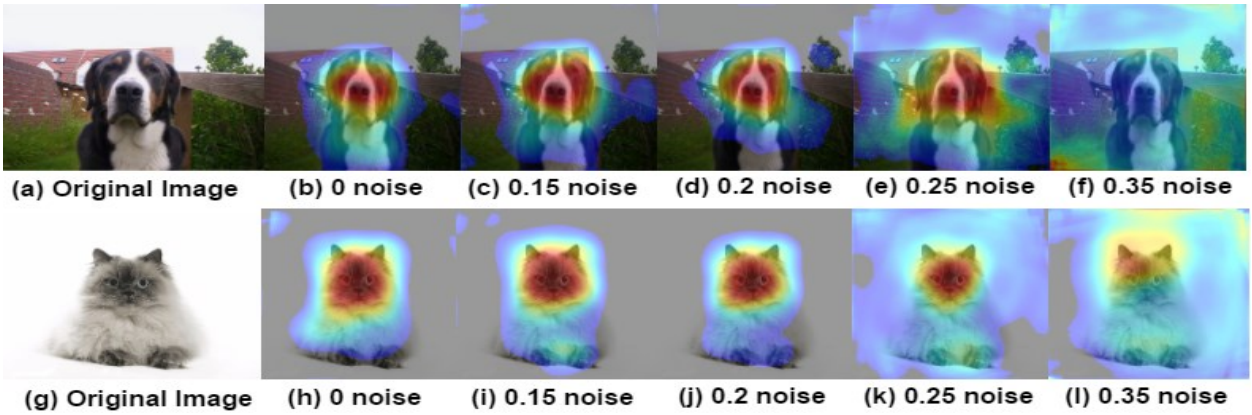


Figure 24. Sanity check results by model weights randomization

4.4.2 Data Randomization

For data randomization, we pass an input image but with the wrong class index. We use a pre-trained VGG-16 model to predict the class. We expect that passing a wrong class to the model will impact the ASC saliency map since the model prediction accuracy will degrade. We generate two ASC saliency maps, one for a correct class index and the second for a wrong class index. We chose a cock rooster image from ImageNet (ILSVRC2012) validation set. This image has a class index of 7. Therefore, we pass a class index of 7; then, we pass a different class index. For example, we pass a class index of a tabby cat that is 282. We can observe from figure 25 that the correct class index produced a successful saliency map that captured the cock rooster, as shown in figure 25 (b). Interestingly, despite the absence of the correct class index (i.e., cock rooster), the ASC algorithm could correctly localize the object but with more noise in the background, as shown in figure 25 (c). Moreover, the saliency map with the wrong class (i.e., class index 282) had less quality than the correct class index (i.e., class index 7).

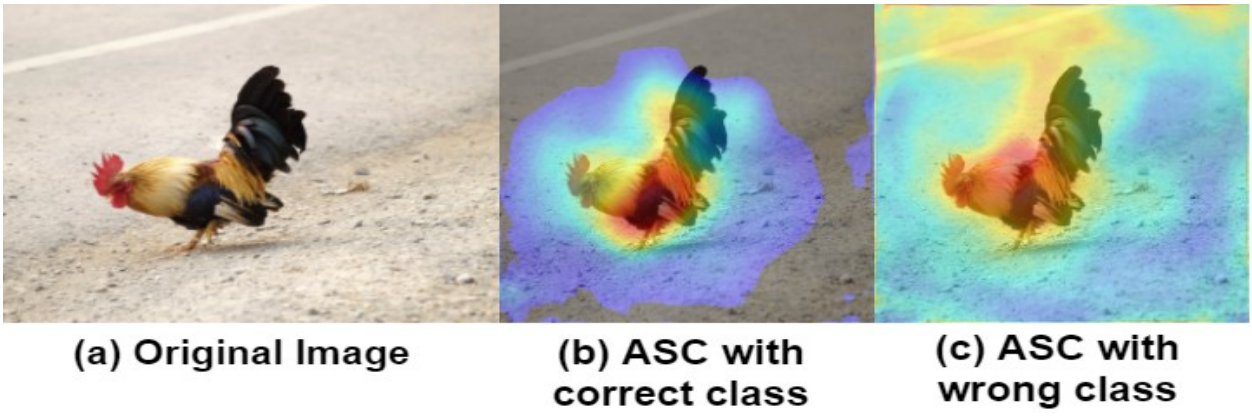


Figure 25. Sanity check results by dataset randomization

4.5 Bias Detection

We highlight the importance of using Augmented Score-CAM in detecting biased training data. Neural networks can achieve high accuracy in areas like image classification. However, their

prediction is sensitive to the nature of the given dataset. The bias in datasets can be hidden and hard to uncover [113]. Therefore, XAI models like ASC generate saliency maps that identify which regions the neural network focuses on when classifying an object (e.g., wolf vs. dog). We built a neural network with two Conv. Layers and one FC layer. The neural network performed binary classification for wolf and dog images. Therefore, the output layer had two nodes with a SoftMax function to calculate the probabilities for each class (i.e., wolf vs. dog). We trained the network on a balanced dataset with 270 images for each class. The images were scraped from the internet and various sources [114]. Moreover, the neural network achieved an accuracy of 90% on the training data. When testing images, Augmented Score-CAM revealed that the neural network had learned to focus on the background to distinguish wolves from dogs. The network was misclassifying dogs with a snowy background as wolves. This performance was due to the biased training data. Most of the wolves' images in training data were captured with snow in the background. Therefore, the network classification was biased and focused on the background rather than the animal features. Figure 26 shows three images with their predictions and corresponding saliency maps. We conclude from the top row that the dog playing with the snowball was misclassified as a wolf. Our saliency map justifies this prediction by highlighting the region of interest. Interestingly, CNN looked at the snow in the background, including the snowball. The middle row shows a dog running on snow. This dog was misclassified as a wolf, and the saliency map highlighted the region of interest, which was the snowy ground underneath the dog. The last row shows a wolf that was classified correctly. However, it was apparent that the network was looking at the snowy ground to verify its prediction. Therefore, its prediction was misleading despite correctly classifying the wolf. This experiment demonstrates the importance of applying Augmented Score-CAM to help

neural networks in detecting bias. Thus, helping to modify training data, reduce bias, and make ethical decisions.

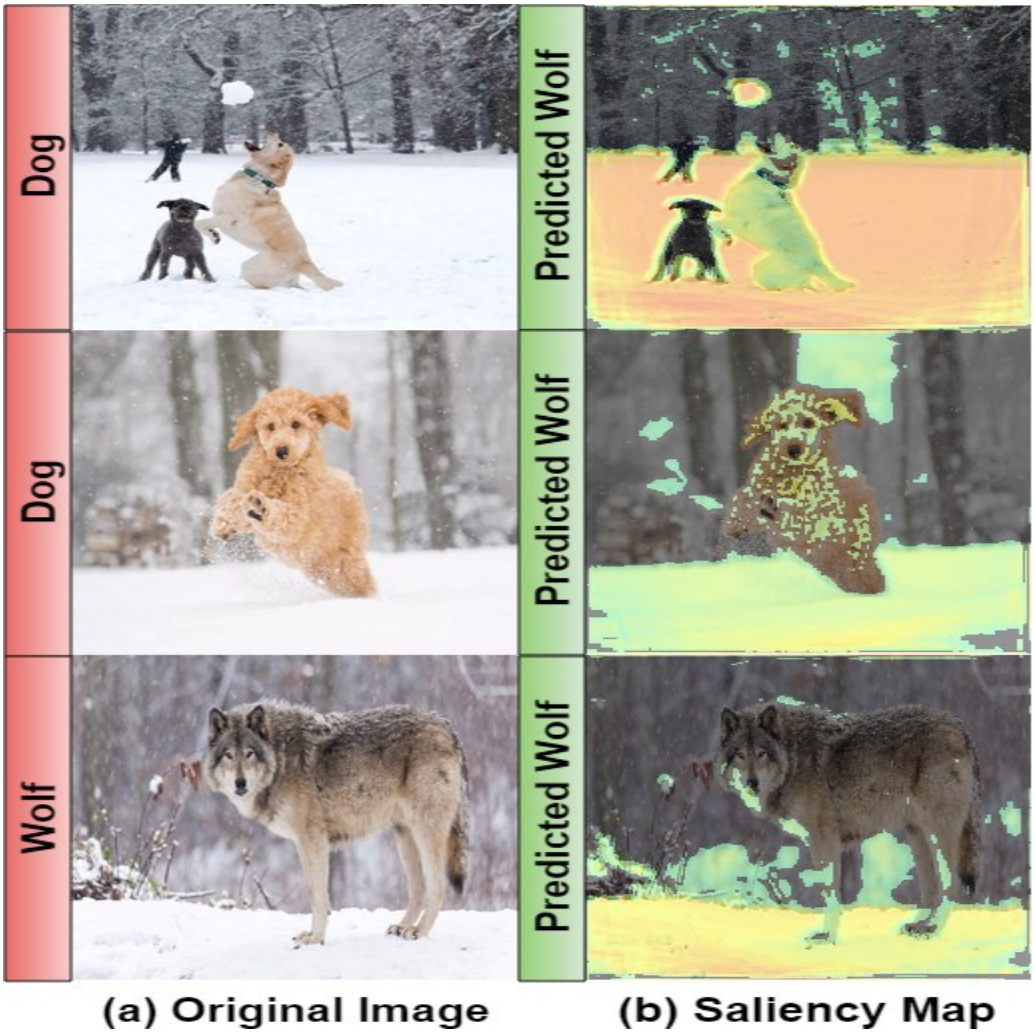


Figure 26. Detection of biased predictions by ASC

4.6 CNN Convergence

In this section, we analyze CNN convergence as the correlation between CNN accuracy and saliency maps quantitatively. We build four pre-trained VGG-16 models with various prediction accuracies. We followed the same approach we did in the sanity check and randomly permuted the weights of each model. Table 13 shows the four VGG-16 models with their noise levels and

corresponding accuracies. In this analysis, we use 40 images from the ImageNet (ILSVRC2012) validation set. We can notice from table 13 that the accuracy decreases when adding more noise to the model weights.

***Table 13*. Four VGG-16 models with their accuracies**

Model	Noise level	Accuracy
VGG Model 1	0 (no noise)	87%
VGG Model 2	0.25	59%
VGG Model 3	0.5	18%
VGG Model 4	0.75	8%

After that, we used the Intersection over Union (IoU) metric to calculate the object localization for each model and image. The purpose of the analysis was to monitor the object localization of saliency maps in each CNN. Interestingly, we found a high correlation between the IoU value (i.e., object localization) and the CNN's accuracy. As shown in figure 27, there is an inverse linear relationship between the IoU value and the CNN accuracy. When we add more noise to CNN, and the accuracy degrades, the IoU value decreases linearly. Therefore, we proved that there is a high correlation between object localization and CNN accuracy. This conclusion could be useful as an indication of the CNN performance. Hence, high-quality saliency maps mean that the CNN model has a good prediction performance. Moreover, figure 28 shows the bounding boxes of a dog and a starfish. Following the convention from the related works, the red rectangle represents the ground truth, while the green rectangle represents the ASC saliency map. In the dog image, we notice that the ASC object localization was better in figure 28 (b) as it surrounded the dog's body. No noise was added to this CNN.

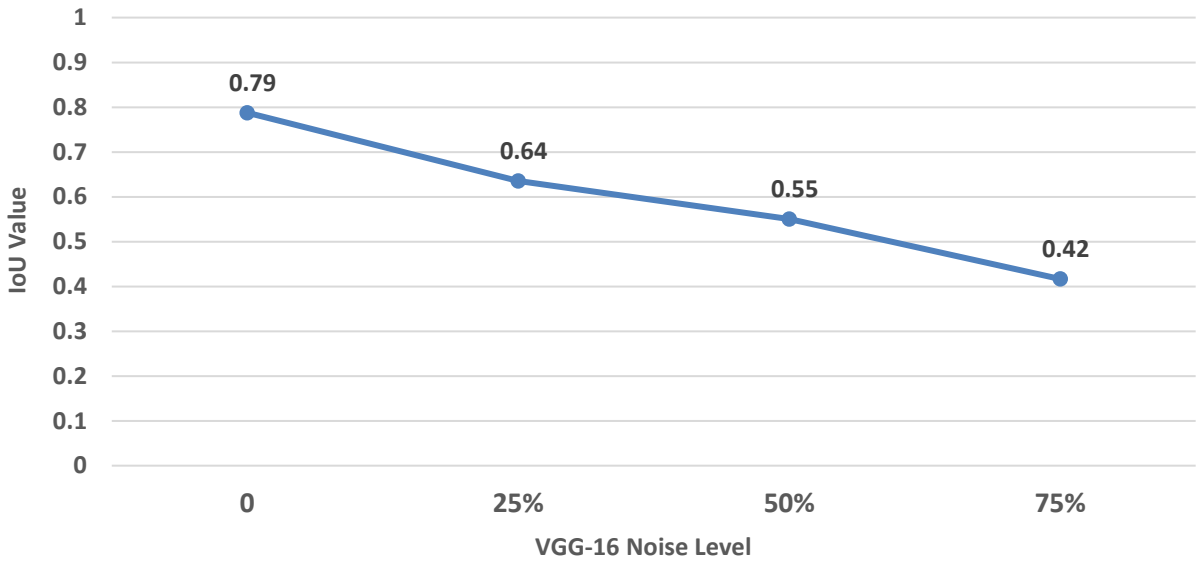


Figure 27. The IoU value per CNN noise level (accuracy)

The object localization was still acceptable in figure 28 (c) with 75% CNN accuracy. However, the bounding box started to slip outside the dog's body as the CNN accuracy decreased, as shown in figure 28 (d) with 50% CNN accuracy. Figure 28 (e) showed that the CNN accuracy was the lowest among the four CNN models with 25%. Therefore, we can notice that the ASC object localization failed to capture the dog and was completely outside its body.

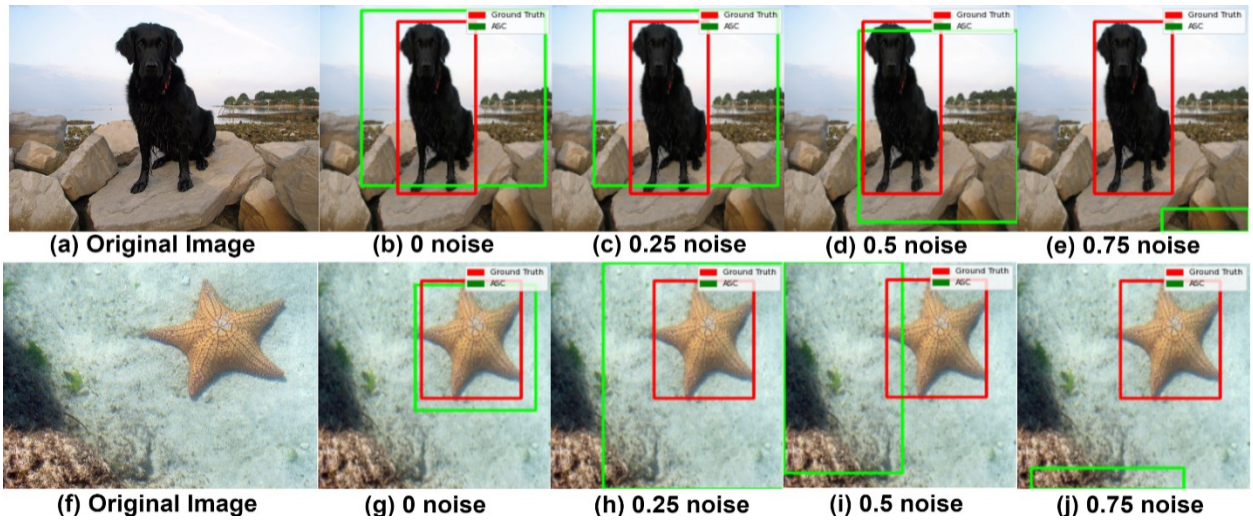


Figure 28. Object localization per noised CNN models

For the starfish image, we can notice that the ASC object localization in figure 28 (g) was close to the ground truth. After that, the bounding box slipped outside the starfish body as the CNN's accuracy decreased. Therefore, the consequent images were impacted by the accuracy degradation and had poor object localization, as shown in figures 28 (h), 28 (i), and 28 (j), respectively.

4.7 Model Scalability

For evaluating model scalability like in Grad-CAM [18] and Score-CAM [22], we measure the effect of a various number of augmented images (n in equation 7) on the model execution time. Moreover, we explore the impact of different image resolutions on the saliency maps' quality. We passed three versions of the "hummingbird" image to the VGG-16 network. Low resolution, moderate resolution, and high resolution, as shown in figure 30. We run the ASC model with three different parameters, 50 augmented images, 100 augmented images, and 150 augmented images. In previous experiments, we used the same parameters values of Augmented Grad-CAM [81], and we initialized the number of augmented images to 100 (n = 100). For the model scalability, we measured the execution time for our model while generating 50, 100, and 150 augmented images, as shown in table 14.

***Table 14*. Execution time per number of augmented images**

Number of augmented images (n)	Execution time (in seconds)
50	191
100	366
150	541

We can notice that the change in the number of augmented images had a significant effect on the model execution time. The model spent more time generating the saliency map when more

augmented images n were generated. When 50 augmented images were generated, the execution time was around 191 seconds, while 100 augmented images required around 366 seconds, and 150 augmented images required about 541 seconds to generate the final saliency map. Figure 29 shows the linear increase of the model execution time w.r.t the number of augmented images. Therefore, the scalability analysis indicated that Augmented Score-CAM required more time when the number of augmented images *n* was higher.

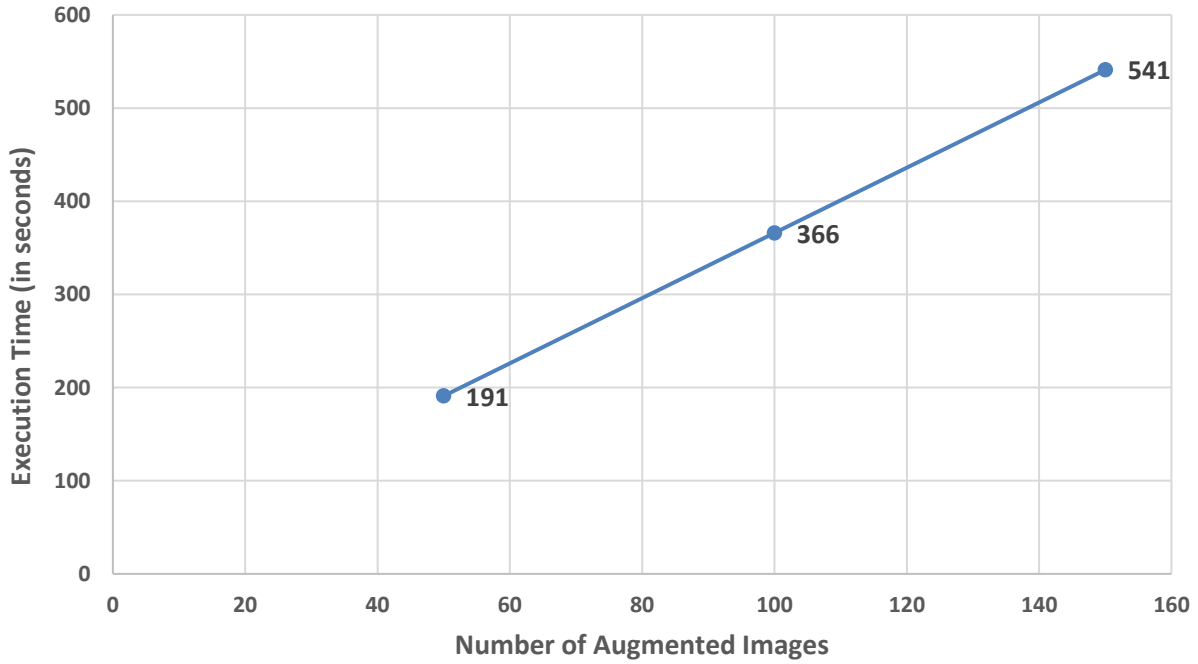


Figure 29. Model scalability per number of augmented images

Furthermore, we analyzed the effect of the image resolution on the saliency map quality. Figure 30 shows three ASC saliency maps generated for low, moderate, and high resolution "hummingbird" images. We generated 100 augmented images (n = 100). It can be observed that our model generated a good saliency map for the high-resolution image. However, when the

resolution dropped in figures 30 (b) and 30 (c), the saliency map quality was impacted and could not capture the bird adequately, as shown in figures 30 (e) and 30 (f).

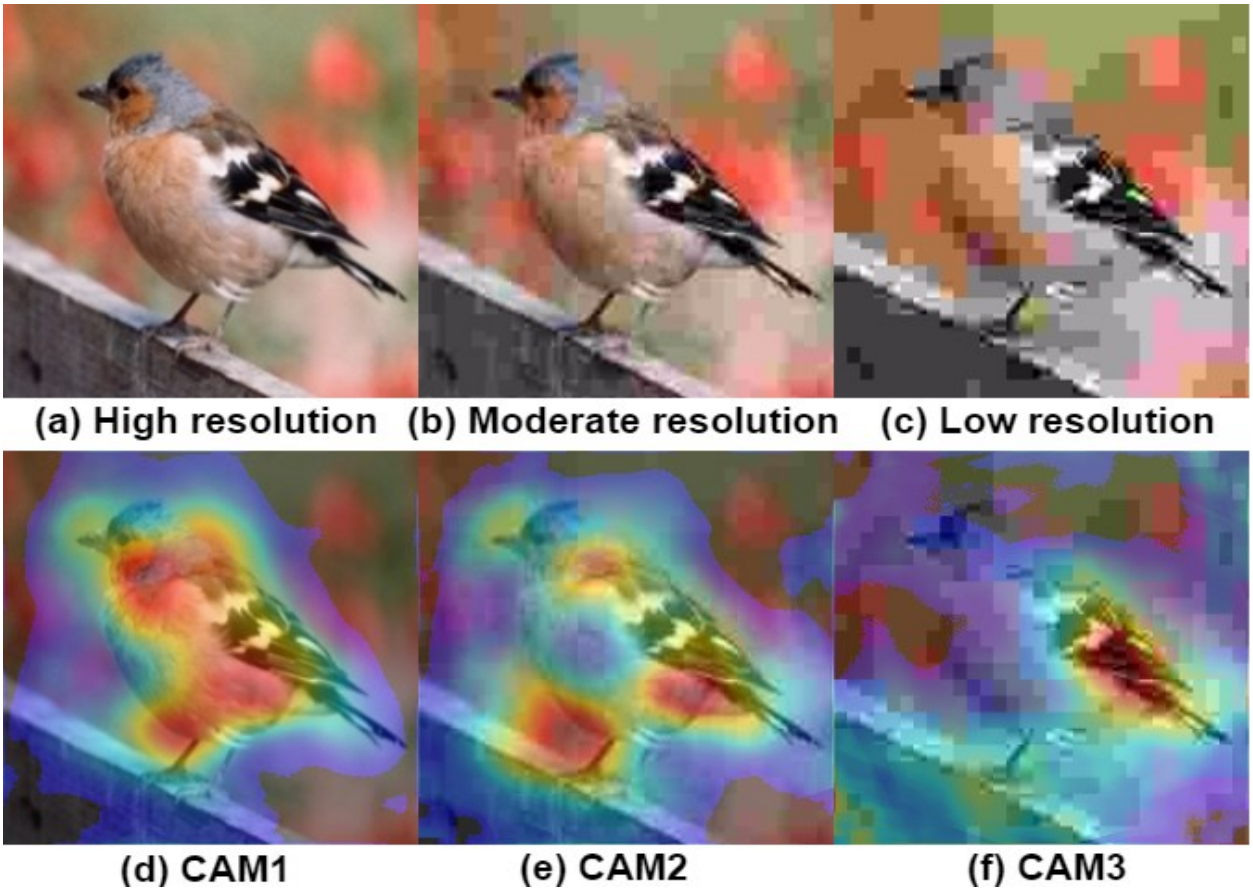


Figure 30. Saliency maps for various image resolutions

4.8 Complexity Analysis

Unlike the Score-CAM model, where a single image is used to produce the final saliency map, our model iterates through a set of augmented images to extract the final saliency map. The ASC model keeps iterating by the initialized number of images *n* (e.g., 100 images). To quantify the complexity analysis, we use Big-O-Notation to study the runtime growth of our model. The first step is to analyze the Big-O-Notation for the Score-CAM model. The Score-CAM takes a single image and generates a set of activation maps A_l^k. Then, it iterates through activation maps to calculate their

corresponding scores $\propto_k^c$. In the end, it generates the final saliency map by linearly combining each activation map in A_l^k with its score in $\propto_k^c$. The time complexity for the Score-CAM model is $O(A_l^k * \propto_k^c)$. Where $\propto_k^c$ is a constant value of the activation maps scores, and A_l^k denote the activation maps generated through *n* iterations. Therefore, the time complexity for Score-CAM can be represented by $O(n)$. This means that the complexity of Score-CAM increases linearly in proportion to the number of activation maps A_l^k.

For the ASC time complexity, we rely on figure 16, which describes the pseudo-code for the ASC model. We can notice that our model starts with generating *n* augmented images. After that, it has a for loop that iterates through the augmented images and applies Score-CAM on each one. Therefore, the difference between Augmented Score-CAM and Score-CAM is the additional loop that generates the set of augmented images. Since Score-CAM is applied on each augmented image, the time complexity for our model is $O(n * A_l^k * \propto_k^c)$. Where *n* is the set of augmented images generated through iteration, the time complexity for ASC can be represented by $O(n^2)$. This means that the complexity of ASC is proportional to the square of the number of activation maps A_l^k.

4.9 Neural Style Transfer

In Augmented Score-CAM, we proposed geometric methods for augmenting the input image. As shown in figure 15, we applied rotation and translation with slight degrees to generate various saliency maps before combining them. Moreover, our experiments proved that image augmentation could be adopted in XAI models to produce enhanced saliency maps. However, geometric augmentation methods are applied to the input image space. We believe that it is worth exploring feature space augmentation methods like deep neural networks (i.e., deep image augmentation). Deep image augmentation applies convolutional neural networks to produce

synthetic variations of the input image for enlarging the training set [115]. For example, GANs [93] can be used in image augmentation to create synthetic images like the features of the training set. Therefore, they can enrich the training data, improve CNN's performance, and reduce overfitting. GANs [93] have two components, the generator, and the discriminator. The generator takes a noise signal to produce an image and tries to fool the discriminator into accepting it as a real image. Meanwhile, the discriminator compares the fake image with a training set to decide if it is real. The generator improves the generated image quality through backpropagation until the discriminator accepts it as a real image. Therefore, replacing geometric image augmentation in our model with GANs is an improper approach. GANs [93] require a large number of images in the training process to produce high-resolution images [115]. Additionally, we generate augmentations derived from the input image, but GANs generate images from a noise signal. Thus, GANs [93] create a new synthetic image similar to the training data but unique in its contexts.

A promising approach for replacing our geometric augmentations is to apply neural style transfer [94]. This method is popular as an artistic tool, but it can be used as an image augmentation technique. It requires a content image (i.e., input image) and a style image to produce a new stylized image, which adds the style of the style image while preserving the content image. Multiple variations of neural style transfer were proposed to minimize content loss and style loss. Furthermore, selecting the style image can be fruitful for applications like autonomous cars. For example, using styles like time (i.e., night vs. daytime) and weather (i.e., rainy vs. sunny) can help AI systems make decisions in various environments. Previous studies proved that using neural style transfer in image augmentation can improve computer vision tasks like object localization [116]. Moreover, neural style transfer improved image classification accuracy on images from Caltech101 and Caltech256 datasets [117]. Experiments showed that neural style transfer

augmentation improved the VGG16 accuracy by 2%. This section builds a new variation of the Augmented Score-CAM model by replacing geometric augmentation with neural style transfer. The purpose is to check if saliency maps quality can be improved when using other image augmentation methods.

First, we select eight style images from the STaDA experiments [117]. After that, we use a pre-trained neural style transfer model [118]. This pre-trained model applies real-time stylization by merging artistic style with fast style transfer networks. Then, we select a sample of three images from the ImageNet (ILSVRC2012) validation set and feed them into the pre-trained model to generate the corresponding stylized images. After producing stylized images, we get nine input images for the ASC model, the original image, and the eight stylized images. Therefore, instead of generating augmented images with rotations and translations (i.e., geometric augmentation), we use the eight stylized images as augmented images (i.e., neural style transfer augmentation). After that, an array of the input image and stylized images is fed to the Augmented Score-CAM. The ASC model will receive each stylized image and produce a corresponding saliency map using the Score-CAM model as shown in the blue-shaded box in figure 15. After that, saliency maps are combined before a super-resolution algorithm is applied to produce the final saliency map. Figure 31 shows the ASC saliency maps for a shark object. The figure has two versions of saliency maps, geometric augmentation saliency map in figure 31 (b), and neural style transfer augmentation saliency map in figure 31 (c). We can notice that both augmentation methods were successful in highlighting the body of the shark. However, the resolution of geometric augmentation was better than neural style transfer augmentation. Moreover, the neural style transfer augmentation noise is higher despite applying the super-resolution algorithm. A possible reason for the noise in figure 31 (c) is the low number of augmented images in the neural style transfer method.

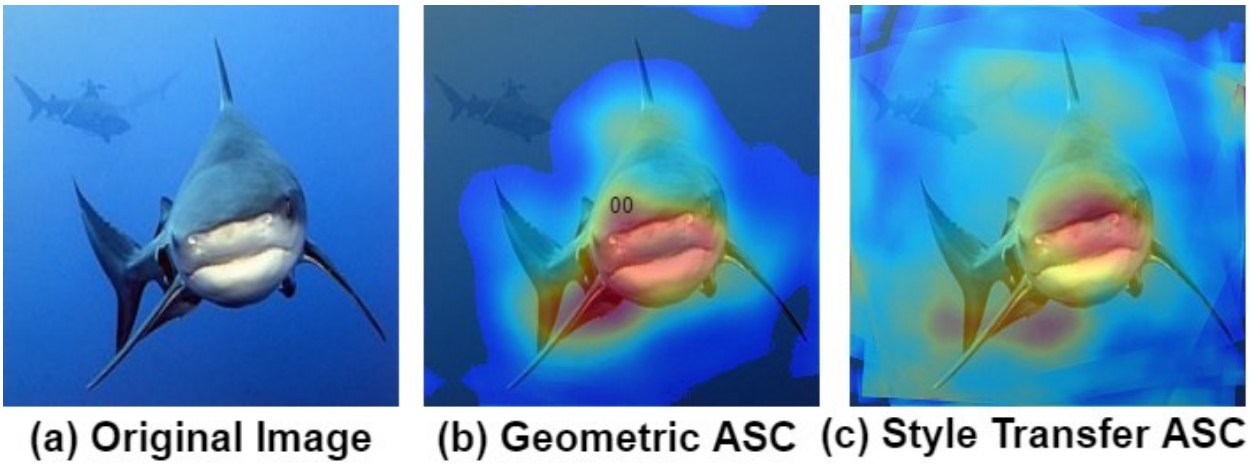


Figure 31. Saliency maps for various augmentation methods

We used eight augmented images (i.e., stylized images) to generate the saliency map in figure 31 (c). In comparison, we used 99 augmented images (i.e., various rotations and translations) to generate the saliency map in figure 31 (b). Another possible reason for the lower quality in figure 31 (c) is the selection of style images. A drawback of neural style transfer augmentation is the effort required to select the best combination of style images for producing a high-quality saliency map [115]. To explore the quality of style images, we analyze the stylized images we used in our experiment by feeding each image to the Score-CAM algorithm. Accordingly, we get a saliency map for each stylized image, as shown in figure 31. Interestingly, we can notice that some stylized images produce better saliency maps than others. For example, styles like "la_muse" and "udnie" distracted the CNN from the shark and produced poor saliency maps. Some styles like "rain_princess" and "sunflower" produced saliency maps but with some noise. However, styles like "the_scream", "the_shipwreck", "wave", and "your_name" produced high-quality saliency maps that captured most parts of the shark. Therefore, styles that have too many colors and shapes, like in figures 32 (a), 32 (b), and 32 (f), could distract the CNN and impact its performance. Thus, produce saliency maps with poor quality [117]. By relying on these results, we believe that neural

style transfer augmentation could be useful for XAI research in computer vision. In the future, we also plan to study this area by selecting more styles and applying the same qualitative and quantitative evaluation metrics we did for the original ASC model.

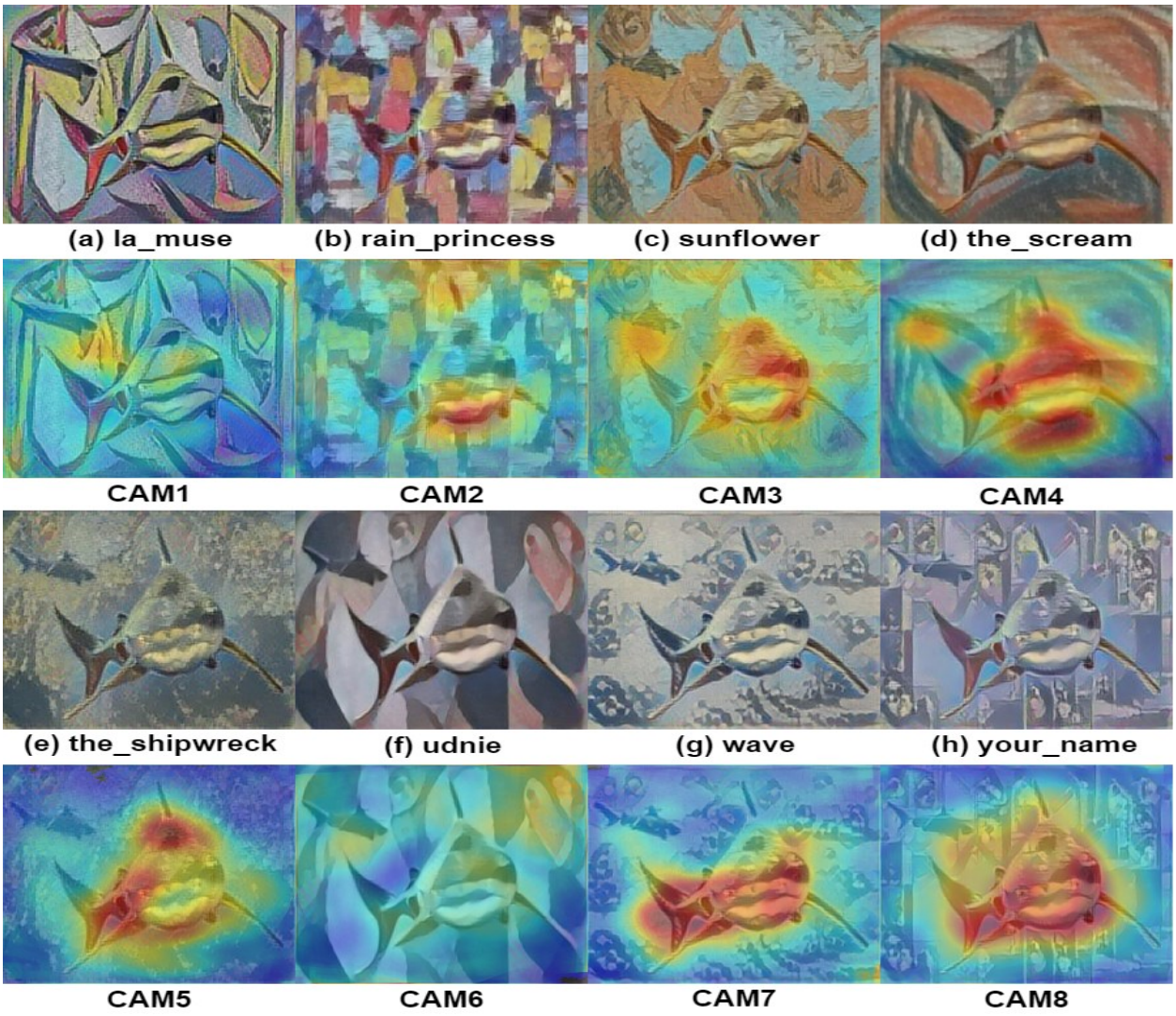


Figure 32. Saliency maps for different stylized images

4.10 User Trust

This user study entitled “Measuring the trust for explainable deep models” was granted clearance by the Carleton University Research Ethics Board-B (CUREB-B) with a clearance ID of 115686. Moreover, we completed the Tri-Council Policy Statement (TCPS2) to conduct this user study.

4.10.1 Demographics

In the user study, we recruited 36 participants through Upwork, a popular crowdsourcing marketplace. The average completion time for the survey was 5.62 minutes. Participants consisted of 28 males and 8 females with an average age of 25.7 years (range: 20-31 years). 17 participants said that the English language was not their first language. Finally, all participants indicated that they were familiar with decision trees. Table 15 summarizes the demographics of recruited participants.

***Table 15*. Participants Demographics**

Variable		No. of Participants	Percentage (%)
Gender	Female	28	77.7%
	Male	8	22.3%
Age	<= 25	18	50%
	> 25 && <= 30	17	47.2%
	> 30 && <= 35	1	2.8%
Occupation	Software Engineer	12	33.3%
	Data Scientist	9	25%
	Student	15	41.7%
English	First Language	19	52.7%
	Second Language	17	47.3%

4.10.2 Decision Effectiveness Analysis

To measure the effectiveness of a participant decision, we compared it with the ground truth decisions of our questions (i.e., the correct answers). If the participant made the wrong decision,

we counted the answer as a mismatch. All mismatches were counted per task (i.e., text and image) and explanation level (i.e., black-box, grey-box, white-box). After that, the average mismatch score was calculated to measure the effectiveness of the decisions they made. Figure 33 shows the mismatch scores for different criteria.

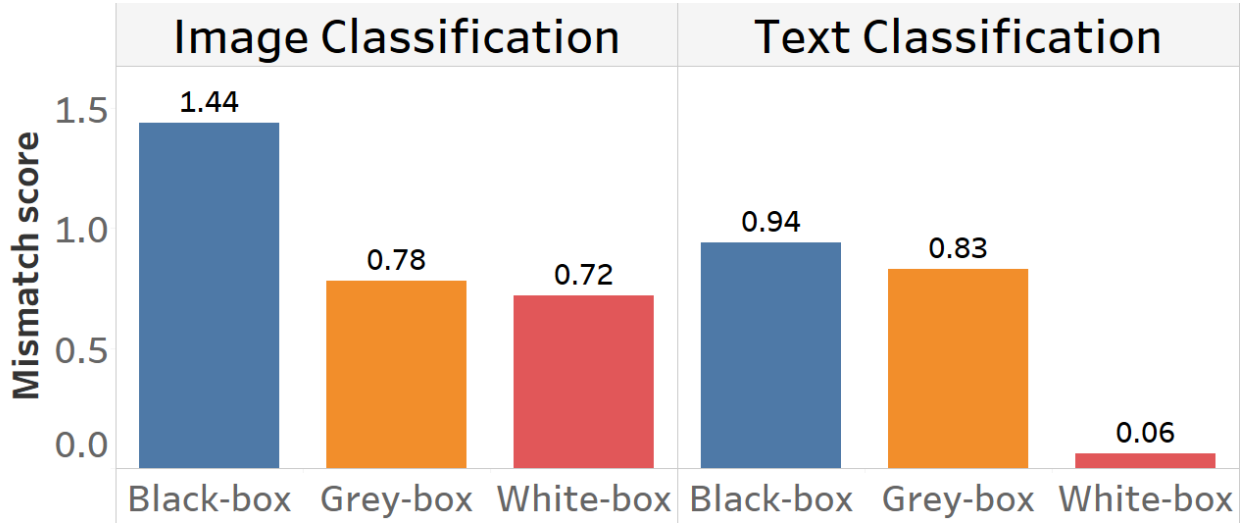


Figure 33. Mismatch scores (task vs. explanation level)

We can notice from the figure that the highest mismatch score was in the image classification task and black-box level with an average score of 1.44, while the lowest mismatch score was in the text classification task and white-box level with an average score of 0.06. We can observe that the mismatch score decreased when moving from black-box to grey-box and white-box levels in both tasks. A possible reason is that participants felt more confident when various explanations for textual or visual data were provided. Moreover, they mostly trusted their decisions at the white-box level, where they could trace the decision tree and conclude the result. In terms of the task, the text classification average mismatch score was 1.83, while the average mismatch score was 2.94 for the image classification. This analysis highlights the importance of embedding explanations in AI systems to maximize decision effectiveness.

4.10.3 Trust Degree Analysis

For analyzing the trust degree, we calculate the frequencies of each response in the 5-point Likert scale. In addition, we measure the trust degree for each task and explanation level. The purpose is to explore the effect of the task and explanation level on the trust degree. Figure 34 shows the response frequencies for all tasks and levels. We can observe that "Very strong" was the most frequent response with 181 responses, while "Very weak" was the least frequent response with 58 responses.

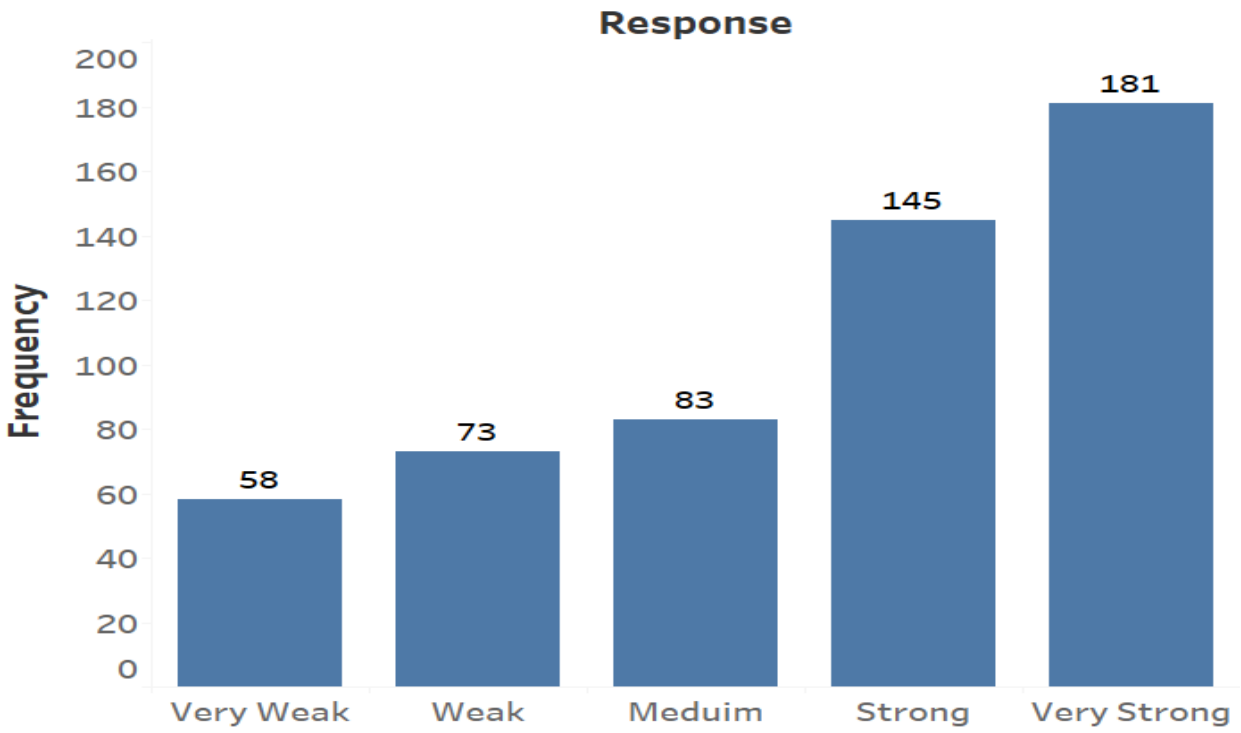


Figure 34. Overall trust degree

Moreover, we analyzed the trust degree per explanation level (i.e., black-box, grey-box, white-box). Figure 35 shows that black-box responses were mostly distributed over "Weak" and "Very weak". Responses of grey-box were distributed over "Medium", "Strong", and "Very strong". While white-box responses were mostly distributed over "Very strong". In the black-box level, the

highest number of responses was for “Weak”, with 67 responses. The lowest number of responses was for “Very Strong”, with four responses. This indicated that participants’ degree of trust was low when they made decisions without explanations. For the grey-box section, “Strong” was the most frequent response, with 82 responses. This result expressed the improvement in trust degree when the user had some explanations like interpretation plots and heatmaps. For the white-box section, the most frequent response was “Very strong”, with 135 responses. This result indicates a high degree of trust when the user relies on decision trees.

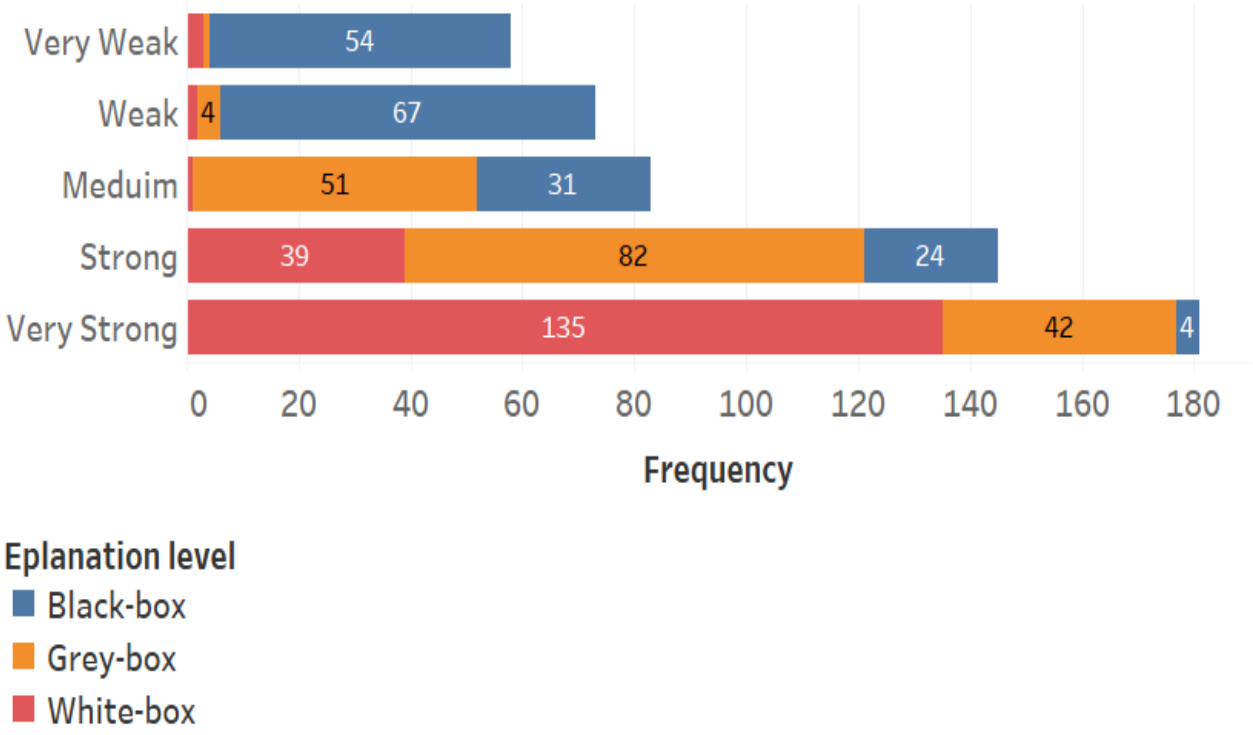


Figure 35. Trust degree per explanation level

After that, we analyzed the trust degree for each task (i.e., text classification, image classification). As shown in figure 36, in the text classification task, the highest number of responses was for “Very strong”, with 96 responses. The lowest number of responses was for “Very weak”, with 14 responses. This result expressed the high degree of trust for making decisions in text-based

applications. The highest number of responses for the image classification task was for “Very strong” with 85 responses, while the lowest was for “Medium” with 25 responses. When comparing text and image classification tasks, the analysis indicated that users were more confident when making decisions based on textual data rather than images.

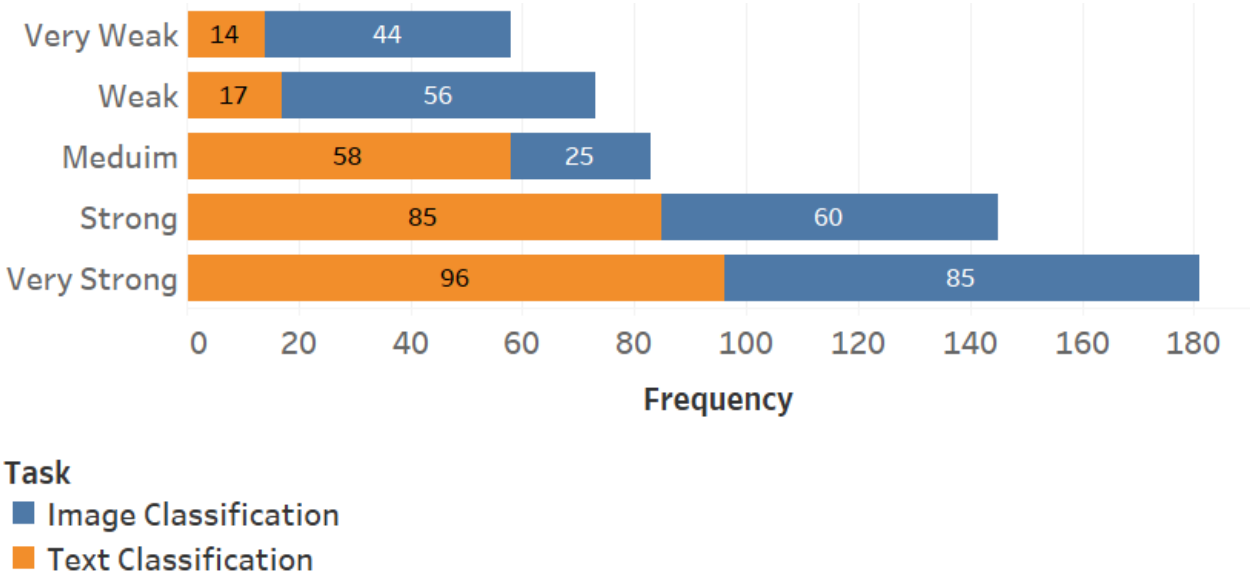


Figure 36. Trust degree per task

Next, we analyzed the trust degree per task and explanation level. As shown in figure 37, the highest number of responses among all explanation levels and tasks was for the “Very strong” response in text classification and white-box sections with 70 responses. Interestingly, in image classification, participants did not trust their decisions when images were provided without supporting explanations (i.e., black-box). Responses were either “Very weak” or “Weak”. In contrast, participants expressed a high degree of trust when they made their decision by relying on decision trees (i.e., white-box). However, participants still expressed higher trust at the grey-box level when we showed interpretation plots and heatmaps. After that, we calculated the average mean of responses for the 15 questions.

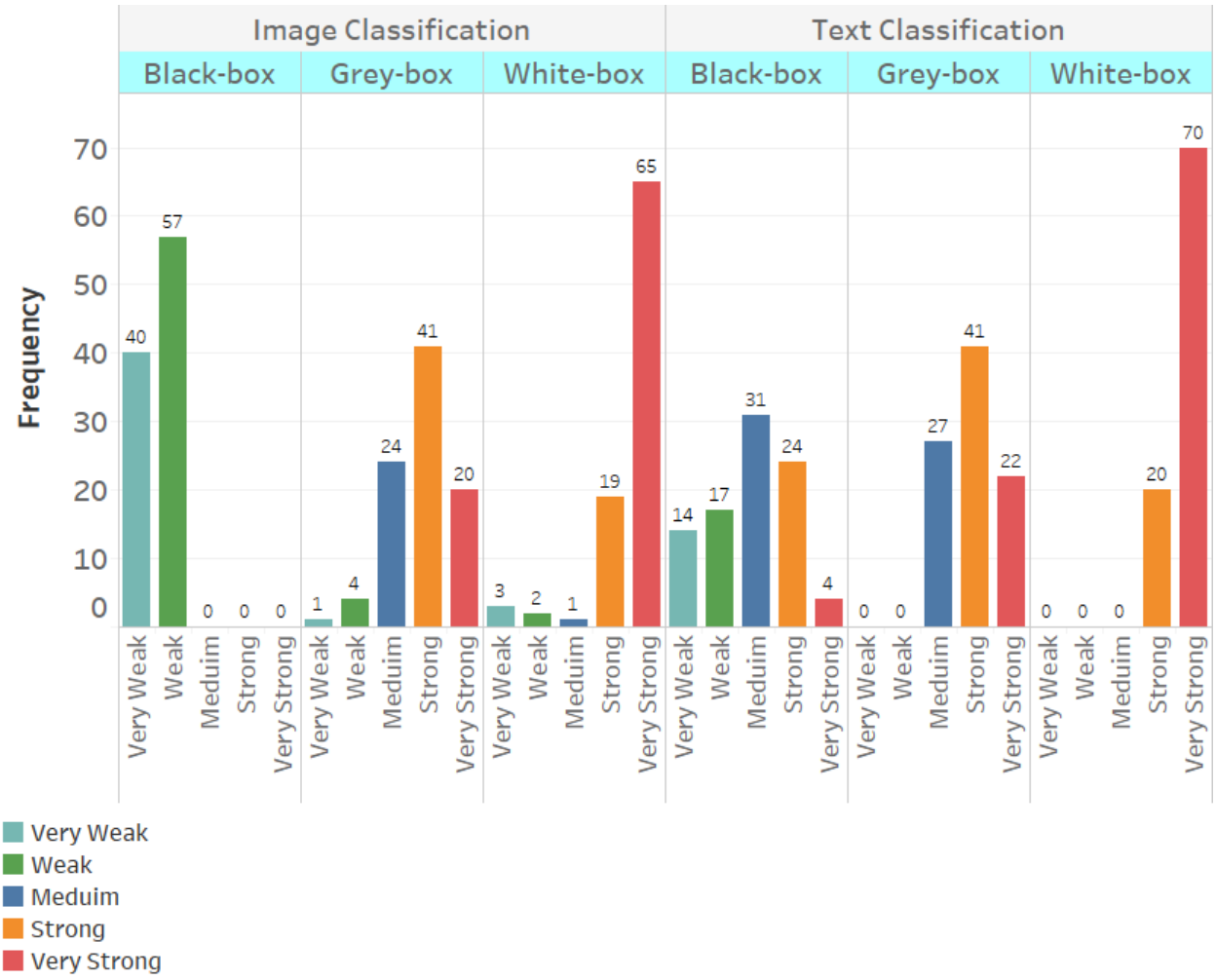


Figure 37. Trust degree per task and explanation level

However, before the calculation, we assigned a weight for each response as follows, “Very Weak” = 1, “Weak” = 2, “Medium” = 3, “Strong” = 4, “Very Strong” = 5. Then, we multiplied the frequency of each response with its weight. A higher value indicated a higher degree of trust. In the text classification, the average weight of responses was 3.86. This result indicated that most responses were between “Medium” and “Strong” responses. In the image classification, the average weight of responses was 3.32. This result indicated that image classification had slightly less weight of responses than text classification. Next, we calculated the percentage of responses for each question in each survey. In the survey, questions Q1 to Q5 represent the black-box level,

questions Q6 to Q10 represent the grey-box level, and questions Q11 to Q15 represent the white-box level. Figure 38 shows the percentage of the responses for each question in the text classification survey. We can observe that responses were distributed over different trust degrees in Q1 to Q5 (i.e., black-box). However, some questions had a majority of "Strong" trust degree like Q4, "Is Kenya safe for residents?". For questions Q6 to Q10 (i.e., grey-box), most participants turned to have a "Strong" trust degree in their decisions. For questions Q11 to Q15, most participants expressed "Strong" or "Very strong" trust degrees in their decisions.

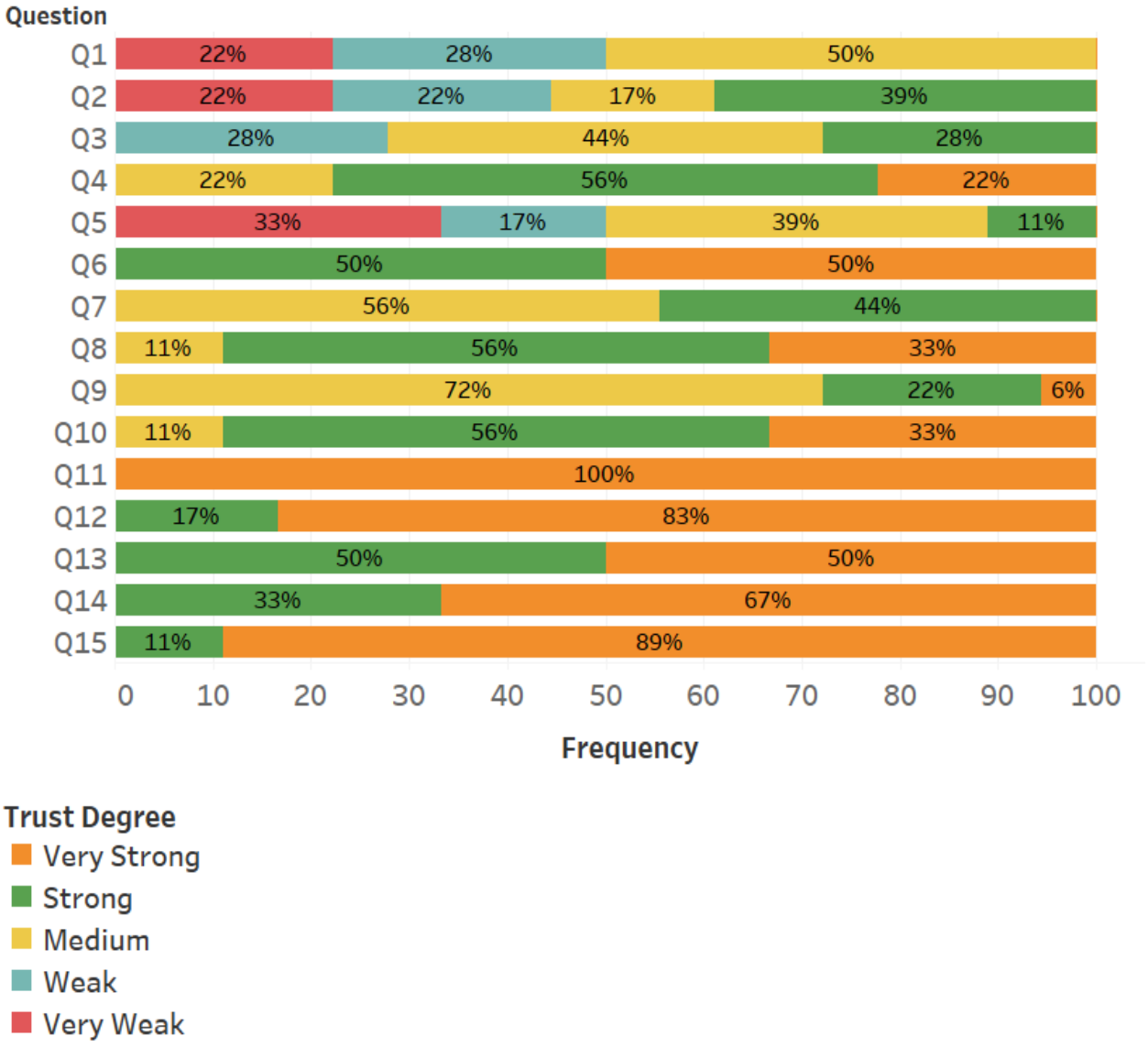


Figure 38. Responses percentage in text classification

Figure 39 shows the percentage of the responses for each question in the image classification survey. For questions Q1 to Q5 (i.e., black-box), participants had either “Weak” or “Very weak” trust degrees. In questions Q6 to Q10 (i.e., grey-box), trust degrees were distributed over “Medium”, “Strong”, and “Very Strong”. In questions Q11 to Q15 (i.e., white-box), most participants expressed a “Very strong” trust degree. In terms of image and text classification, the analysis indicated when an image was shown alone, the trust degree was lower than when a text was shown alone.

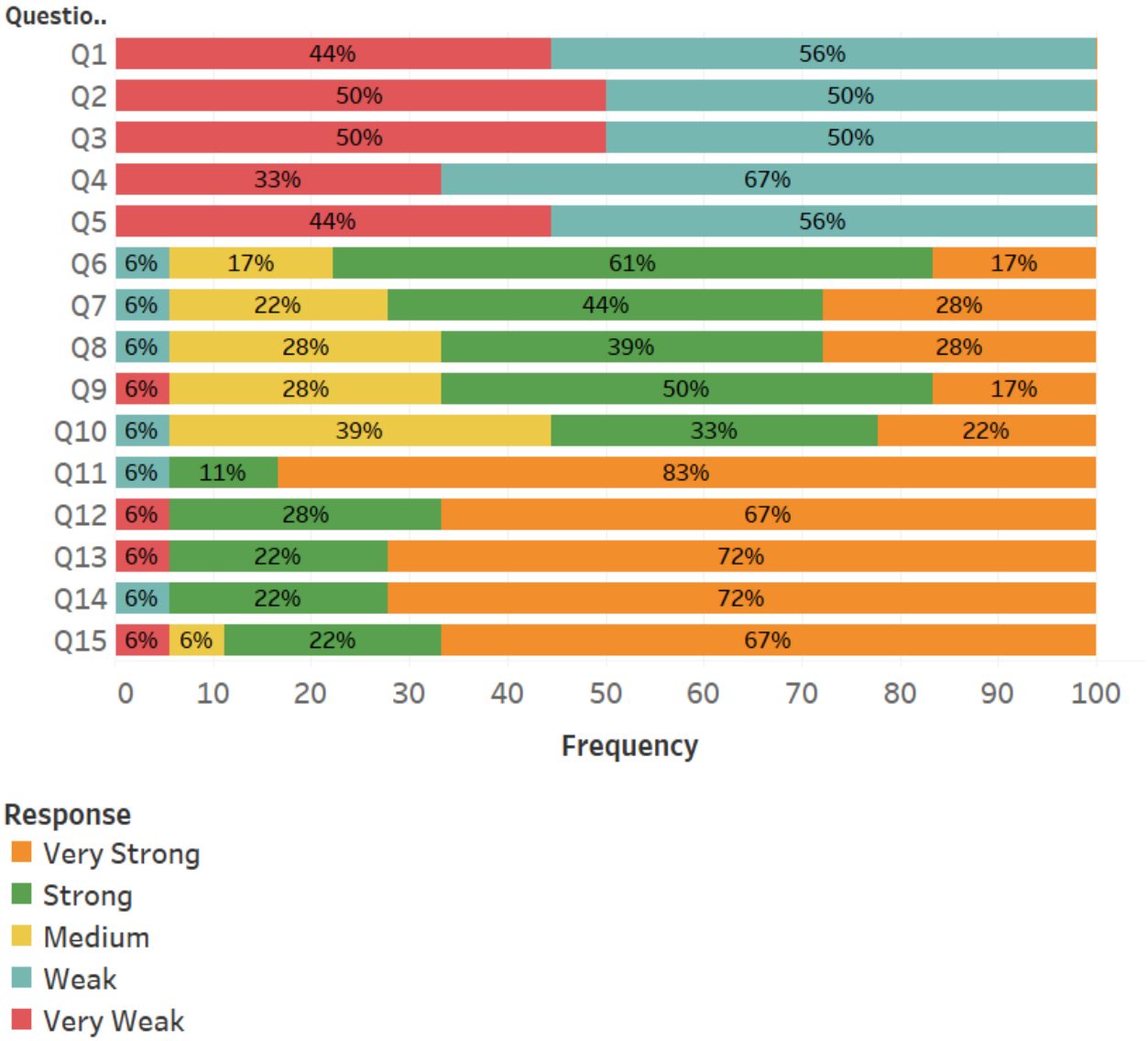


Figure 39. Responses percentage in image classification

Moreover, when we showed the heatmaps (i.e., Grad-CAM), the trust degree was higher than the interpretation plots (i.e., LIME). Therefore, providing heatmaps (i.e., Grad-CAM) was more effective in improving the trust degree.

4.10.4 Hypothesis Testing

To verify hypothesis H1, we created a line chart to analyze the trend of the trust degree over different explanation levels. As we can notice from figure 40, the black-box level had an absence of explanations which caused a gradual decrease in trust degree. The grey-box trust degree increased, which proves that providing explanations improved the participants' trust. The white-box trust degree significantly increased, proving that participants were more confident when making decisions.

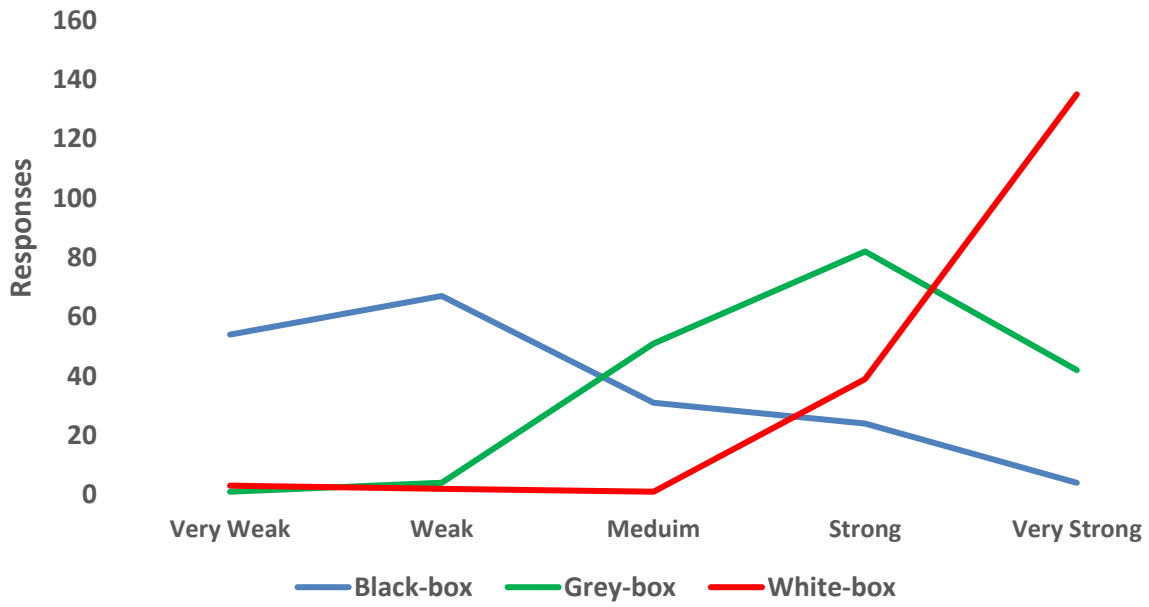


Figure 40. Trust degree for various explanation levels

We calculated the weighted frequency of responses for each explanation level to quantify the trust degree. Table 16 shows that the white-box and grey-box weighted responses were significantly

higher than the black-box level. The black-box level had a weighted response of 397. Meanwhile, the grey-box and white-box levels were comparable with 700 and 841 responses, respectively. Therefore, we could verify hypothesis H1 since the trust degree of models with explanations (i.e., grey-box and white-box) were higher than the trust degree of models without explanations (i.e., black-box).

***Table 16*. Weighted responses for each explanation level**

Explanation level	Weighted responses
Black-box	397
Grey-box	700
White-box	841

To verify hypotheses H2 and H3, we used the split-plot ANOVA (SPANOVA) since we have a mixed design study with a between subject's variable (i.e., task) and a within-subjects variable (i.e., explanation level). The analyzed dataset had a two-level task variable (i.e., text vs. image) and a three-level explanation level variable (i.e., black-box vs. grey-box vs. white-box). In addition, we stored the average trust degree for each participant, task, and explanation level. The average trust degree was calculated based on the Likert-scale five responses as follows, "Very Weak" = 1, "Weak" = 2, "Medium" = 3, "Strong" = 4, "Very Strong" = 5. In the end, the table had 36 participants with their average trust degrees. For H2, we test if there is a significant difference in the test degree between the explanation levels. As shown in table 17, the results prove the statistically significant differences between the explanation levels. There was a significant effect for explanation level on the trust, (F = 280.999, P < .05). Therefore, we could verify hypothesis H2 since there is a difference in the trust degree between black-box, grey-box, and white-box

levels. Moreover, a significant interaction effect (Level x Task) was obtained (F = 19.463, P < .05), as shown in table 17. For verifying H3, we test if there is a significant effect on the trust degree between text classification and image classification tasks. As shown in table 18, the result shows the statistically significant differences between the two tasks. There was a significant effect for task on the trust, (F = 17.895, P < .05). Therefore, we could verify hypothesis H3 since there is a difference in the trust degree between text and image classification tasks. Moreover, we analyzed the trust degree change across different levels and tasks. Figure 41 shows a significant increase in the trust degree means in the image classification task from the black-box level (M = 1.6) to the white-box level (M = 4.6). Meanwhile, the trust degree means the change was less significant in the text classification from the black-box level (M = 2.9) to the white-box level (M = 4.8).

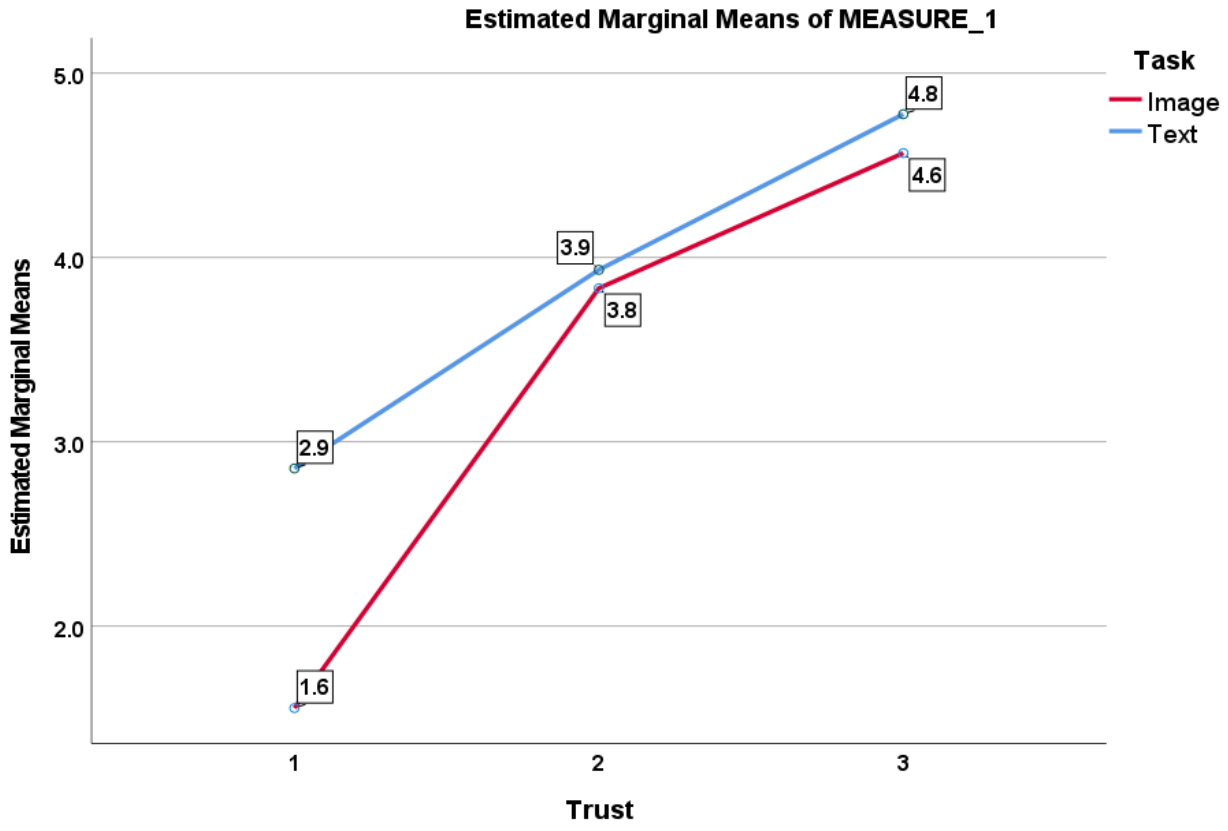


Figure 41. Trust degree means per tasks and explanation levels

***Table 17*. Tests of within-subjects**

Within-Subjects Effects (Level)	*F-value*	*P-value*
	280.999	0.000
Intersection (Level*Task)	19.463	0.000

***Table 18*. Tests of between-subjects**

Between-Subjects Effects (Task)	*F-value*	*P-value*
	17.895	0.000

5. Discussion and Future Directions

5.1 Augmented Score-CAM Complexity

Our model proved to outperform Score-CAM in terms of localization, class discrimination, and faithfulness. However, the complexity analysis showed that the ASC had a Big-O-Notation of $O(n^2)$, while the Score-CAM had a Big-O- Notation of $O(n)$. Therefore, the model execution time of ASC was significantly higher than SSC. The average time for generating saliency maps for our model was around 5 minutes. The generation of saliency maps for each image was a time-consuming process because of the augmentation iterations. Unlike Score-CAM (SSC), where a single pass was performed to generate saliency maps, our model performed one pass (i.e., iteration) for each augmented image before combining corresponding saliency maps. A possible approach to reduce the model complexity is to utilize frameworks like Hadoop Map-Reduce. This framework can parallelize our model and process each augmented image on a separate machine [119], significantly minimizing the algorithm computation. In the future, we plan to set up a parallel distributing environment on the cloud with multiple clusters. We will generate *n* augmented images, divide them into several subsets and store each subset on a different node. After that, we will utilize Hadoop MapReduce by writing a map function to run Augmented Score-CAM on each node. The last step will be the reduce function which merges all activation maps into a final activation map. The MapReduce approach allows Augmented Score-CAM to be executed on a small portion of the augmented images instead of using all of them. The time complexity for Augmented Score-CAM was $O(n^2)$. However, it will be divided by the number of nodes on the cloud [119]. Accordingly, the complexity of our algorithm will be $O(\frac{n^2}{k})$, where *k* is the number of nodes and *n* is the number of augmented images, such that $k << n$.

In addition, Augmented Score-CAM was evaluated using a pre-trained VGG-16 network. In the CNN convergence subsection, we explored the correlation between object localization and network prediction accuracy, shown in figure 27. Since our model is correlated to the CNN prediction and performance, it is crucial to investigate the fine-tuned CNNs effect on our saliency maps. Further experiments should provide valuable insights into the effect of various hyperparameters values and training/test ratios on the performance of Augmented Score-CAM.

Moreover, it is a good approach to compare Augmented Score-CAM (ASC) with other recent variations of Score-CAM like Smoothed Score-CAM (SS-CAM) [120] and Integrated Score-CAM (IS-CAM) [121]. A promising approach is to combine these variations toward building a robust XAI model that improves Score-CAM in different dimensions. In addition, recent XAI models are passive; they provide explanations without any suggestions or possible solutions. A good potential is to leverage XAI models to a higher level to provide counterfactual explanations and suggestions. This level of explanation provides a set of changes to be made by end-users, data scientists, and domain experts for solving misclassification and biased decisions.

5.2 User Trust

In the user trust evaluation, we targeted trust degree in making decisions. However, the decisions in our experiment were not critical, like clinical and air traffic applications. The first task was deciding if a Quora question was sincere or insincere. The second task was deciding if a metal nut was good or defective. Moreover, we used post-hoc models like LIME [31] and Grad-CAM [18]. These models were attached to the classifier to provide explanations. Furthermore, we used decision trees as white-box XAI models in our experiments. By verifying our hypotheses, we proved that incorporating XAI interpretations and visualizations improved user trust in the decision-making systems.

Augmented Score-CAM produces a class activation map highlighting image regions that significantly contributed to the CNN prediction. Therefore, it can be used as a post-hoc model to provide explanations and improve user trust in critical applications like autonomous vehicles. For instance, Augmented Score-CAM visualization in semi-automated vehicles can help drivers decide if the vehicle should reduce its speed or not [122]. Furthermore, our activation maps can be incorporated into clinical decision-support systems to improve clinicians' trust in diagnosing COVID-19 in medical images [123].

5.3 Fairness and Discrimination

Fairness and discrimination were addressed in the AI principles published by the European Commission (EC) in 2019 [13]. These AI principles should exist in any AI system. Interestingly, XAI models can analyze the output w.r.t the input image and uncover hidden correlations that cause discrimination in the CNNs decision. Models like Augmented Score-CAM can visualize the areas CNN was looking at when classifying an image. CNNs bias could affect some groups like gender, race, or age. Despite predicting images with high accuracy, CNNs tend to consider some factors to make unfair decisions. These decisions may lead to discrimination due to reasons like limited data and imbalanced datasets. Therefore, XAI models can detect this bias and mitigate discrimination against a particular group of people.

In addition, previous studies mentioned five reasons for biased CNNs [14]. The existence of skewed data, which occurs during the data collection. Tainted data, which happens because of errors in the CNN design or parameters initialization. Another reason is the limited features of CNN. An imbalanced dataset with a majority of one label over the others can induce bias. The correlation among sensitive features is another reason for CNNs biased decisions. Moreover, different approaches were presented to mitigate the effect of bias, such as pre-processing, in-

processing, and post-processing. The pre-processing approach is a set of techniques that are applied before the CNN training process. In-processing is the set of techniques applied during the CNN training to optimize the fairness constraints in the model. Post-processing techniques are applied after the CNN training. They propose to adjust thresholds in the classification and minimize the difference between true positive (TP) and false positive (FP).

Another approach for mitigating the bias issue focuses on CNN's diverse results more than the prediction accuracy [124]. This diversity is accomplished by ensuring that all objects are represented in CNN's prediction and outputs. XAI models can help to identify the ability of the CNN to preserve diversity in input data. Therefore, we can observe that XAI models within the literature detected biased CNN decisions but could not justify or mitigate the bias. The process of bias detection and mitigation is complementary between the AI system and the XAI model. The responsibility of the AI system is to use XAI explanations to report and justify the bias. Moreover, the AI system should give feedback on why a user was treated unfairly based on the dataset. Simultaneously, the XAI model's role is to analyze correlations between input data and the CNN prediction to generate high-quality explanations that help the AI system detect bias.

5.4 CNN Accuracy Analysis

In the CNN convergence section, we showed that the generated saliency maps could help the user recognize the prediction accuracy. Harnessing the model visualizations can contribute to analyzing the performance of CNN. Therefore, there is a strong correlation between noise level in saliency maps and the prediction accuracy, as shown in figure 27. Some studies analyzed the effect of removing/keeping saliency maps pixels on the model accuracy [125]. Their experiments showed that the model accuracy and percentage of removed pixels had a linear relationship. The model accuracy kept decreasing when more salient pixels were removed from the saliency map. They

used gradient and non-gradient XAI models in their study. Accordingly, we believe that there is no need for a trade-off between explainability and accuracy as they are highly correlated. The existence of XAI models demonstrates the CNN's level of accuracy. However, there is a high potential in this area for future research.

5.5 Privacy

Recent AI systems in areas like finance, marketing, and banking use personal information in their tasks. Privacy of data should be ensured in all stages of the AI system. For example, users could be unaware of privacy risks when their data is collected by Internet of Things (IoT) sensors without any consent [126]. In addition, people had little understanding of the sensor-enabled AI systems and their process of collecting data. Similarly, the privacy of XAI models should be considered since these models need to access CNN architecture to generate saliency maps (e.g., backpropagation to the last convolutional layer). Therefore, privacy in XAI models has some concerns that require further study by the research community.

Another concern of XAI models is explanations' level of details. High-level explanations of the AI system's functionality can be overwhelming for end-users. For example, social media users are not interested in understanding how the system provides recommendations. Previous studies proposed multiple methods for providing explanations in XAI models. For example, simulation of the model is encouraged for providing simple textual and visualized explanations [127]. This method succeeds with interpretable sparse linear models. The Decomposability method explains separate parts of the model like input and parameters [128]. However, the drawback of this method is that humans should understand every part of the model without any supplementary tools. Overall, XAI explanations should be adjusted in their levels of complexity to fit various stakeholders' needs.

5.6 XAI in Decision Support Systems

Our results showed that providing explanations had a significant effect on user trust. Moreover, we proved that without explanations (i.e., black-box), the user relied on himself to make a decision that impacted his trust in the system. In decision-support systems, the lack of trust in the AI system can lead to making false decisions. In contrast, providing explanations and improving user trust can help in making critical decisions in clinical, industrial, and air traffic control systems. Moreover, previous studies showed that automated decision-support systems like neural networks could make wrong decisions and predictions [33]. Furthermore, it was proven that some users were insensitive to these false decisions as they believed that these systems were reliable [129]. This insensitivity toward the AI systems' decisions was known as over-reliance [130]. Therefore, we believe there is an increasing demand for explanation models (XAI) that interpret the decision-support systems and justify their decisions.

5.7 Trustworthy AI

The lack of explanations in AI systems can impact user trust in their decisions. Reasonable explanations can be provided by integrating XAI with the AI system. For instance, if a recommender system shows an ad to the user, the XAI model should explain why this ad was sent to the user. Moreover, in image classification, the XAI model can highlight the pixels contributing to the CNN decision. Thus, it can improve the user trust in the neural network. In addition, models like Grad-CAM [18] proved the correlation between saliency maps and CNN accuracy. They suggested that saliency maps' quality can express the CNN performance and improve the user trust accordingly. Therefore, user trust should be addressed along with the model performance when developing and building AI systems [23]. Various model interpretations can be included in the design stage, like visualizations, semantic representations, and features relevance.

5.8 Towards Democratic AI Systems

Despite the significant advancements and wide use of AI systems, there is a demand for democratic governance of decision-making systems that impacts people's lives [131]. Explainable AI (XAI) aims to improve ethical guidelines and protect people from possible risks. Keeping humans-in-the loop will reduce the reliance on fully automated AI systems. In 2021, the European Commission proposed the Artificial Intelligence Act to protect humans from possible risks related to safety, regulations compliance, transparency, and trustworthiness [132]. This Act mentioned explainable AI as an approach to improve AI systems transparency and allow humans to interpret decisions made by these systems. This indicates the high demand for incorporating XAI models in AI systems. However, the AI Act did not identify the level of transparency or mention what explanations could be provided to the user. Therefore, we believe our contributions can fulfill regulations compliance and mitigate risks mentioned in the AI Act. For instance, Augmented Score-CAM can transform CNNs from black-box to gray-box and make them more transparent by providing activation maps that help the user interpret decisions. Moreover, the user trust was improved when visual explanations were associated with the input image.

5.9 XAI Generalization

5.9.1 Post-hoc vs. Intrinsic (Application/Task generalization)

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, question classification, and VQA such as Integrated Gradients [72], Grad-CAM [18], Grad-CAM++ [78], Equalizer [88], and U-CAM [82]. This wide range of applications 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 [48], CNN Explainer [58], XCNN [59], FCM [61], and Decision Trees [36]. 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 the accuracy is limited to other applications where interpretability is preferred more than accuracy.

Some Post-hoc models proved to be useful in detecting bias like Grad-CAM [18], Score-CAM [22], Equalizer [88], and LIME [31]. However, the heatmaps they generated were passive. For instance, the gender bias detection example in figure 11 clarified the existence of the bias but could not identify the defect in the training data [18]. 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.

5.9.2 Model-Agnostic vs. Model-Specific (Networks generalization)

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 [51] and CNN Explainer [58] modified neural networks like AlexNet, VGG-M, VGG-S, and VGG-16. CAR [66] compressed neural networks like LeNet, AlexNet, and ResNet-50. Grad-CAM [18], Grad-CAM++ [78], and Score-CAM [22] heatmaps were generated by neural networks like AlexNet, VGG-16, ResNet-

50, and GoogleNet . In addition, only the CNNV model [90] 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.

5.9.3 Summary

Figure 42 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. Both models shared image classification and object recognition applications. Furthermore, image classification was the most application used 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.

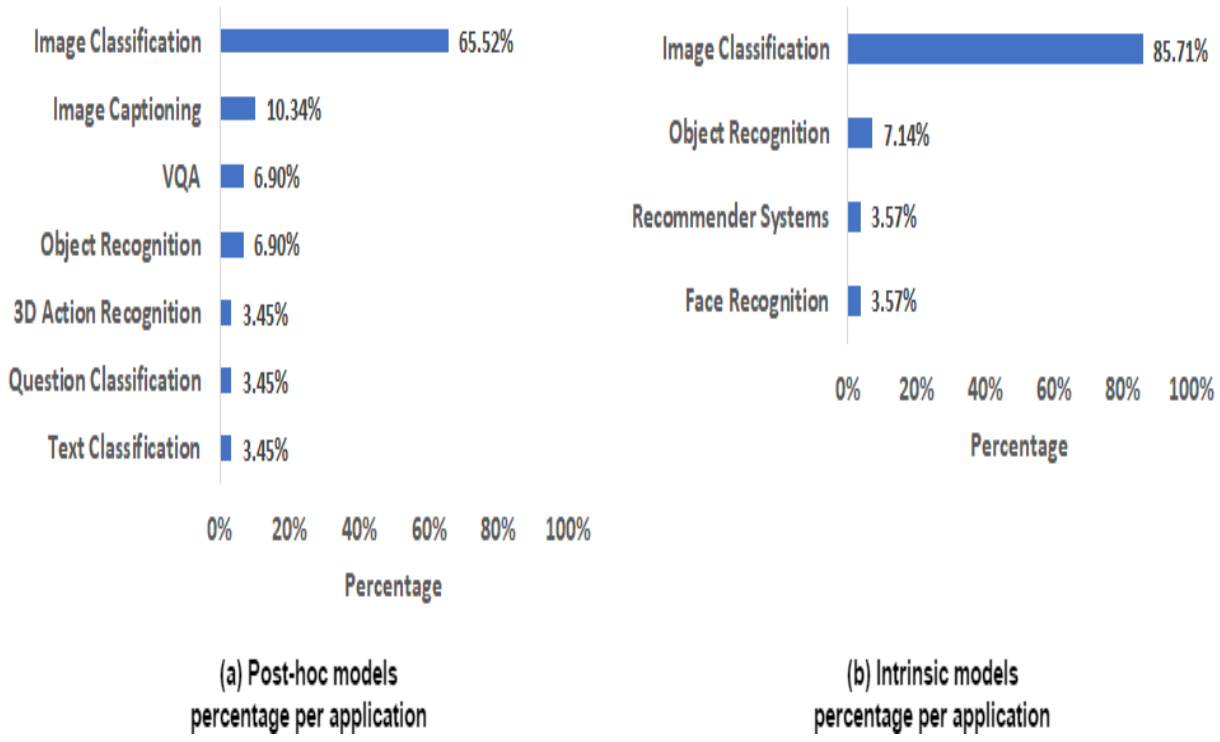


Figure 42. XAI models per application

5.10 XAI 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, we believe XAI models should consider three factors, accuracy, explainability, and complexity.

5.10.1 Post-hoc vs. Intrinsic

5.10.1.1 Classification Accuracy

Intrinsic XAI models like Interpretable CNN [51], CNN Explainer [58], XCNN [59], FCM [61], and Decision Trees [36] 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 [62]. 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 [18], Score-CAM [22], Eigen-CAM [83], and U-CAM [82] were added as auxiliary components to pre-trained CNNs. Therefore, these models ignored the evaluation of the classification accuracy as they maintained CNNs architecture.

5.10.1.2 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 [58], Dynamic-K Activation [52], ProtoPNet [56], and Loss Attention [45] used Grad-CAM [18] heatmaps to prove that their modified CNNs produced better interpretation. Meanwhile, other intrinsic models like XCNN [59], FCM [61], and SENN [63] used various heatmaps like Saliency Maps [75], LRP [71], LIME [31] and SHAP [133] for class discrimination evaluation. Furthermore, post-hoc models were qualitatively evaluated by comparing their heatmaps with other post-hoc models. For instance, DeepLIFT [73] compared its saliency maps with other models like Integrated Gradients [72]. Additionally, Score-CAM [22] compared its heatmap with the heatmap of Grad-CAM [18].

5.10.1.3 Object Localization

For the object localization, most XAI models used the Intersection over Union (IoU) metric. These models followed the approach described in section 5.2 that 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 [51], AOG [55], and Dynamic-K Activation [52] applied the IoU metric to evaluate object localization. Other intrinsic models like CNN Explainer [58] 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 [18], Grad-CAM++ [78], Eigen-CAM [83], and Mask [85] used the IoU metric to evaluate the object localization. Score-CAM [22] 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.

5.10.1.4 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 [44] proved that the model was resistant to noised labels when comparing its accuracy with ResNet. In addition, ProtoPShare [57] proved to be robust against image perturbations like contrast and brightness. SENN [63] proved to be more robust than LIME [31] and SHAP [133] against adversarial examples on various datasets. Moreover, Deep KNN [62] proved to be more robust than traditional CNN against adversarial examples like FGSM, BIM, and C&W. Meanwhile, Post-hoc models like Grad-CAM [18] and Mask [85] were evaluated against local image perturbations. Subnetwork Extraction [65] and Eigen-CAM [83] evaluated their robustness against effective attacks such as FGSM, BIM, DeepFool, and C&W. However, 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 for 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.

5.10.2 Summary

Table 19 shows five intrinsic models that caused a drop in the CNN accuracy when components were added to improve the network interpretability. Meanwhile, Table 20 shows XAI models evaluated against various adversarial attacks.

***Table 19*. intrinsic models that impacted CNN classification accuracy**

Model	Description	Added components

Interpretable CNN [51]	State-of-art networks like AlexNet, VGG-M. VGG-S, and VGG-16 outperformed Interpretable CNN in classification accuracy for single class images	Added loss of feature map to each filter in high conv. layers
CNN Explainer [58]	State-of-art networks like AlexNet, VGG-M. VGG-S, and VGG-16 outperformed CNN Explainer in classification accuracy	Added Autoencoder to each feature map in middle layers
XCNN [59]	VGG-16 outperformed XCNN in classification accuracy	Attached Autoencoder to CNN classifier
FCM [61]	Base CNN outperformed FCM in classification accuracy	Added fuzzy classifier to the last conv. layer
Decision Trees [36]	State-of-art networks like AlexNet, VGG-M. VGG-S, and VGG-16 outperformed Decision Trees in classification accuracy	Added decision trees to high-level features

***Table 20*. Robustness evaluation in XAI models**

Model	Description	Evaluation
Residual Attention [44]	Evaluated against noised labels and proved to outperform ResNet in classification accuracy	Perturbed images from CIFAR dataset
ProtoPShare [57]	Evaluated against perturbed images and proved to outperform ProtoPNet [56] in classification accuracy	Perturbed images from CUB-200-2011 dataset

SENN [63]	Evaluated against perturbed images and proved to outperform LIME [31] and SHAP [133] in interpretation	Perturbed images from MNIST and COMPAS datasets
Deep KNN [62]	Evaluated against FGSM, BIM, and SW adversarial attacks and proved to outperform traditional CNN in prediction confidence	Perturbed images from MNIST, SVHN, and GTSRB datasets
Grad-CAM [18]	Evaluated against perturbed images and proved to localize the object correctly	Perturbed images from ImageNet dataset
Mask [85]	Evaluated against perturbed images and proved to outperform Grad-CAM [18] in classification accuracy	Perturbed images
Subnetwork Extraction [65]	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 [83]	Evaluated against DeepFool adversarial attacks and proved to outperform CAM [77] and Grad-CAM [18] in recognition prediction	Perturbed images from ImageNet dataset

5.11 XAI and Parameters Selection

Various modifications were applied to CNN architecture to improve their interpretation. However, there was a lack of analysis for their approach in selecting features and layers to be modified. XAI models did not propose a criterion for identifying the most informative features/layers (i.e., low vs. intermediate vs. high layers). For example, which features/layers can provide more insights

when adding KNN classifiers or attention layers. Therefore, we believe that the selection of informative features/layers requires further analysis. Moreover, some XAI models integrated parametric machine learning algorithms to simplify and interpret CNNs like Deep KNN [62], Subnetwork Extraction [65], CAR [66], and EBANO [68]. 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. Another example of selecting appropriate parameters is the proposed Augmented Score-CAM. To run our model, we had to initialize rotation/translation degrees and the number of augmented images *n*. In our experiments, we adopted Augmented Grad-CAM [81] approach and generated 100 augmented images ($n = 100$) to get the final class activation map. Before running the experiments, we tested various *n* values. However, choosing a smaller *n* value impacted the activation map quality, while choosing a larger *n* value increased the execution time. Therefore, it is worth exploring the impact of the *n* value on the quality of the activation map and finding an approach to select the optimal *n* value.

Table 21 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 [134]. 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 [135], 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 [136]. Therefore, initializing parameters in XAI models that use machine learning algorithms should be considered carefully.

***Table 21*. XAI models with Parametric algorithms**

Model	Algorithm	Parameters
Deep KNN [62]	KNN classification that applied cosine similarity to find nearest neighbors	Number of neighbors k
Subnetwork Extraction [65]	Agglomerative hierarchical clustering to categorize specific subnetwork representations	Number of clusters k
CAR [66]	Structural compression of CNN filters	Compression ratio r_{target}
EBANO [68]	K-means clustering of hyper columns to identify interpretable features	Number of clusters *n, Set of centroids k*

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 [68] lacked the importance ranking of features and the analysis of features interaction (i.e., interconnection) in the 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.

5.12 Adversarial Attacks and XAI Robustness

Despite the lack of robustness evaluation, as discussed earlier, some efforts were made to clarify the vulnerability of XAI models against adversarial examples. It was proven that ADV^2 [137] attack succeeded in fooling both the 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 to embed various defense strategies in XAI models to empower their robustness. Defense strategies can be categorized into three types, modifying input image, modifying neural network, and adding an auxiliary network. Table 22 shows some defense strategies that can be applied in neural networks and XAI models.

***Table 22*. Defense strategies models**

Defense Model	Defense Strategy	Modifying Input Image	Modifying Neural Network	Network add-on
Defensive Distillation [138]	Network Distillation		Yes	
Noise-GAN [139]	Adversarial Training	Yes		
Defense-Net [140]	Adversarial Detection			Yes
Image Super-Resolution [141]	Input Reconstruction	Yes		
Spartan [142]	Feature Reduction		Yes	
FN [143]	Gradient Masking		Yes	

5.13 Knowledge-Driven Systems

Augmented Score-CAM generates an activation map that highlights the pixels which contributed to the network decision. However, the activation map can be more human cognitive by adding the semantic description. Object parts can be recognized and labeled by analyzing the network representations. The activation map can be formalized as a set of interpretable concepts.

For example, Subnetwork Extraction [65] 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 [67] extracted 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 [84] 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, this dataset's quality of visual cues 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. Therefore, Augmented Score-CAM can rely on these semantics to improve the human perception of its activation maps. A promising research direction is incorporating semantic description using knowledge graphs with more informative explanations for building knowledge-driven XAI models.

5.14 XAI for other CNNs

5.14.1 XAI for Transformers

Transformers have made significant contributions to natural language processing (NLP) [144] and graphs [145]. However, transformers suffer from high complexity and a large number of parameters which impact their transparency. Previous studies followed different approaches to explain transformers. Some studies proposed post-hoc models like "attention rollout" and

"attention flow" to quantify transformers' attention flow [146]. The first model rolls out the attention weights to capture the information flow from input tokens to hidden layers in a transformer. The second model calculates the flow values from hidden layers to input tokens. Moreover, in text classification tasks like movie reviews and Twitter sentiment analysis [147], gradient-based models like Saliency Maps [75] outperformed LIME [31] in explaining transformers.

5.14.2 XAI for GNNs

Graph Neural Networks (GNNs) can capture observations in graphical applications like computer vision, molecular chemistry, and molecular biology [148]. However, due to its sophisticated architecture, like a large number of nodes and edges in the input graph and stacked interaction blocks, various explanation techniques were proposed to interpret GNNs. Some studies applied Grad-CAM [18] to extract attributions of the graph nodes [149]. LRP [71] was used to explain the nodes and edges of GNNs in natural language processing (NLP) [150]. GNN-LRP [151] was proposed to explain the interaction between the input graph and the GNN model. It outperformed other XAI models like GNNExplainer [152] in terms of "Area Under the Flipping Curve" (AUFC).

6. Conclusions

6.1 Augmented Score-CAM

This study proposed Augmented Score-CAM (ASC), a novel post-hoc XAI model for producing high-quality saliency maps. Our method adopts the existing Score-CAM [22] and image augmentation approach to generate saliency maps for each augmented image. In addition to the input image, every augmented image has useful spatial features. Our method outperforms Score-CAM in terms of class discrimination. The human studies proved that our saliency maps could discriminate accurately between classes, thus, enhancing the user's trustworthiness. Moreover, we validated the faithfulness of our model using popular pre-trained CNNs and datasets. Results showed that our model was more faithful as the confidence drop was less than Score-CAM for all pre-trained CNNs. To evaluate weakly supervised localization, we adopted the Intersection over Union (IoU) metric. Experiments proved that our model's IoU value was higher than Score-CAM, indicating that our model captured more object parts than Score-CAM. Additionally, we demonstrated that our model passed the sanity check. Experiments showed that the model was sensitive to model and data randomization. Thus, the quality of saliency maps could express the accuracy of models.

Furthermore, we proved that Augmented Score-CAM could successfully identify biased datasets by highlighting the regions of interest in image classification. We believe that bias detection is crucial for fixing datasets and building fair AI systems. In addition, we detected a strong connection between the ASC object localization and the accuracy of CNN. Furthermore, we built a new version of our model by replacing geometric augmentation with neural style transfer. We showed that some styles had generated high-quality saliency maps. However, an effort is required to select the appropriate style images to enhance the quality of saliency maps. Incorporating XAI

models into existing AI systems is in its infancy stages. However, XAI experiments in applications like image classification, image captioning, and 3D action recognition seem promising.

We evaluated human trust in image and text classification. Furthermore, we evaluated the human trust on various explanation levels like black-box, grey-box, and white-box. The results showed that decisions made on the black-box level had a higher mismatch score and were less effective than the grey-box and white-box levels. This low effectiveness showed the need for explanations to improve the quality of the decisions. In terms of trust degree, it improved when the user had some explanations. The trust degree was the lowest for the black-box level and the highest for the white-box level. Additionally, the trust degree in the text classification task was higher than the image classification task. In terms of hypothesis testing, we verified hypothesis H1 by quantifying the trust degree and showing that the weighted responses for the grey-box and white-box levels were significantly higher than the black-box level. We conducted the SPANOVA test to verify hypotheses H1 and H2. The results showed that both hypotheses were verified. For hypothesis H2, there was a significant effect of explanation level on user trust. While for hypothesis H3, there was a significant effect for the task on user trust. This evaluation confirms the importance of adopting XAI models to improve human trust in decision-support systems and provide useful explanations in crucial applications like clinical, industrial, and financial systems.

6.2 XAI models in CNNs

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 had another screening to skim the papers and exclude 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 terms that frequently appeared in XAI papers in the previous three years. It was evident that some terms like “image”, “classification”, “feature”, and “human” were closely related to the interpretation of CNNs. After that, we explained terms and definitions related to XAI, like explainability and interpretability. Also, we discussed how explanations should be provided to build responsible and faithful neural networks.

We discussed previous 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 architecture and decision models. Accordingly, we took a step further to put forward our survey taxonomy and divided architecture and decision XAI models into four categories, architecture modification, architecture simplification, features relevance, and visualization. In each category, we extensively 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 gain more insights into recent XAI models’ behavior. We found that most models in the architecture modification category we intrinsic and local. While in the model simplification, most 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 the visualization metric, we added use cases to describe different visualizations like saliency maps, class activation maps, and pixels visualization. We showed how

class activation maps outperformed saliency maps in class discrimination. In the localization metric, we added a use case to discuss the approach of the IoU metric. We showed that the higher value of IoU reflects a better object localization. In the robustness metric, we discussed some models that proved to be resistant against noised images or adversarial attacks. In the classification accuracy metric, we showed that intrinsic models relied more on this 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 could capture fire hydrant parts that represent the answer in VQA. In bias detection, we showed how class activation maps 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 could capture each object mentioned in the generated caption. Moreover, we added a use case to prove how activation maps 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, and four post-hoc models were evaluated against perturbed images and adversarial attacks. In terms of parameters selection, some models used machine learning algorithms like clustering and compression to improve CNN interpretation. However, there is an increasing demand for collaboration with domain experts as an improper initialization of parameters can impact the CNN interpretation. Moreover, we showed that post-hoc models were more vulnerable to adversarial attacks due to the partial independence between CNN and post-hoc models. Therefore, we proposed some defense strategies which can be applied to improve the robustness of CNN and the interpretation model. Also, we highlighted the importance of adding semantics to activation maps and discussed some limitations in this area. We aim in this survey to provide researchers and practitioners with a wide range of interpretation models which they can use in different tasks and application areas.

7. Bibliography

[1] J. Wu, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology. Nanjing University. China*, vol. 5, no. 23, p. 495, 2017.

[2] J. Song, S. Gao, Y. Zhu, and C. Ma, "A survey of remote sensing image classification based on CNNs," *null*, vol. 3, no. 3, pp. 232–254, Jul. 2019, doi: 10.1080/20964471.2019.1657720.

[3] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.

[4] C.-C. J. Kuo, "Understanding convolutional neural networks with a mathematical model," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 406–413, Nov. 2016, doi: 10.1016/j.jvcir.2016.11.003.

[5] D. Scherer, A. Müller, and S. Behnke, "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," in *Artificial Neural Networks – ICANN 2010*, Berlin, Heidelberg, 2010, pp. 92–101.

[6] T. Andrew, "The mostly complete chart of Neural Networks, explained," *The mostly complete chart of Neural Networks, explained*, Aug. 04, 2017. https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464 (accessed Mar. 12, 2020).

[7] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-Margin Softmax Loss for Convolutional Neural Networks," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, New York, NY, USA, 2016, pp. 507–516.

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

[9] M. Chromik and M. Schuessler, "A Taxonomy for Human Subject Evaluation of Black-Box Explanations in XAI," p. 7, 2020.

[10] X.-H. Li *et al.*, "A Survey of Data-driven and Knowledge-aware eXplainable AI," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2020, doi: 10.1109/TKDE.2020.2983930.

[11] A. Barredo Arrieta *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.

[12] J. J. Ferreira and M. S. Monteiro, "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*, vol. 12201, A. Marcus and E. Rosenzweig, Eds. Cham: Springer International Publishing, 2020, pp. 56–73. doi: 10.1007/978-3-030-49760-6_4.

[13] A. Hleg, "Ethics guidelines for trustworthy AI," *B-1049 Brussels*, 2019.

[14] S. Barocas and A. D. Selbst, "Big data's disparate impact," *Calif. L. Rev.*, vol. 104, p. 671, 2016.

[15] P. K. Sharma and P. Bhattacharyya, "Survey of Explainable AI: Interpretability and Causal Reasoning," p. 16.

[16] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A Survey of Methods for Explaining Black Box Models," *ACM Comput. Surv.*, vol. 51, no. 5, Aug. 2018, doi: 10.1145/3236009.

[17] J. Zou and L. Schiebinger, “AI can be sexist and racist — it’s time to make it fair,” *Nature*, vol. 559, no. 7714, pp. 324–326, Jul. 2018, doi: 10.1038/d41586-018-05707-.
[18] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626. doi: 10.1109/ICCV.2017.74.
[19] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *Artificial Intelligence*, vol. 267, pp. 1–38, 2019, doi: https://doi.org/10.1016/j.artint.2018.07.007.
[20] F. Rossi, “Building Trust in Artificial Intelligence,” *Journal of International Affairs*, vol. 72, p. 127, 2018.
[21] J. Wanner, L.-V. Herm, K. Heinrich, C. Janiesch, and P. Zschech, “White, Grey, Black: Effects of XAI Augmentation on the Confidence in AI-based Decision Support Systems,” p. 9, 2020.
[22] H. Wang *et al.*, “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, Jun. 2020, pp. 111–119. doi: 10.1109/CVPRW50498.2020.00020.
[23] M. Du, N. Liu, and X. Hu, “Techniques for Interpretable Machine Learning,” *Association for Computing Machinery*, vol. 63, May 2019, doi: 10.1145/3359786.
[24] Alizadeh, Fatemeh, Esau, Margarita, Stevens, Gunnar, and Cassens, Lena, “eXplainable AI: Take one Step Back, Move two Steps forward,” 2020, doi: 10.18420/MUC2020-WS111-369.
[25] J.-H. Cho, K. Chan, and S. Adali, “A Survey on Trust Modeling,” *ACM Comput. Surv.*, vol. 48, no. 2, Oct. 2015, doi: 10.1145/2815595.
[26] R. C. Mayer, J. H. Davis, and F. D. Schoorman, “An integrative model of organizational trust,” *Academy of management review*, vol. 20, no. 3, pp. 709–734, 1995.
[27] D. Gambetta and others, “Can we trust trust,” *Trust: Making and breaking cooperative relations*, vol. 13, pp. 213–237, 2000.
[28] W. Sherchan, S. Nepal, and C. Paris, “A Survey of Trust in Social Networks,” *ACM Comput. Surv.*, vol. 45, no. 4, Aug. 2013, doi: 10.1145/2501654.2501661.
[29] J. Sabater and C. Sierra, “REGRET: Reputation in Gregarious Societies,” in *Proceedings of the Fifth International Conference on Autonomous Agents*, New York, NY, USA, 2001, pp. 194–195. doi: 10.1145/375735.376110.
[30] A. Jøsang and S. Pope, “Semantic Constraints for Trust Transitivity,” in *Proceedings of the 2nd Asia-Pacific Conference on Conceptual Modelling - Volume 43*, AUS, 2005, pp. 59–68.
[31] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘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, 2016, pp. 1135–1144. doi: 10.1145/2939672.2939778.
[32] H. Lakkaraju and O. Bastani, “‘How Do I Fool You?’: Manipulating User Trust via Misleading Black Box Explanations,” in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, New York, NY, USA, 2020, pp. 79–85. doi: 10.1145/3375627.3375833.

[33] A. Bussone, S. Stumpf, and D. O'Sullivan, "The Role of Explanations on Trust and Reliance in Clinical Decision Support Systems," in *2015 International Conference on Healthcare Informatics*, 2015, pp. 160–169. doi: 10.1109/ICHI.2015.26.

[34] C. J. Cai *et al.*, "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, 2019, pp. 1–14. doi: 10.1145/3290605.3300234.

[35] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, "Interpretable machine learning: definitions, methods, and applications," *Proc Natl Acad Sci USA*, vol. 116, no. 44, pp. 22071–22080, Oct. 2019, doi: 10.1073/pnas.1900654116.

[36] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu, "Interpreting CNNs via Decision Trees," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 6254–6263. doi: 10.1109/CVPR.2019.00642.

[37] A. Rai, "Explainable AI: from black box to glass box," *J. of the Acad. Mark. Sci.*, vol. 48, no. 1, pp. 137–141, Jan. 2020, doi: 10.1007/s11747-019-00710-5.

[38] J. H. Friedman and B. E. Popescu, "Predictive learning via rule ensembles," *Ann. Appl. Stat.*, vol. 2, no. 3, pp. 916–954, Sep. 2008, doi: 10.1214/07-AOAS148.

[39] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[40] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, "Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation," *Journal of Computational and Graphical Statistics*, vol. 24, no. 1, pp. 44–65, 2015, doi: 10.1080/10618600.2014.907095.

[41] Y. Zhang, H. Su, T. Jia, and J. Chu, "Rule Extraction from Trained Support Vector Machines," in *Advances in Knowledge Discovery and Data Mining*, Berlin, Heidelberg, 2005, pp. 61–70.

[42] D. Linsley, D. Shiebler, S. Eberhardt, and T. Serre, "Learning what and where to attend," *arXiv:1805.08819 [cs]*, Jun. 2019, Accessed: Feb. 19, 2021. [Online]. Available: http://arxiv.org/abs/1805.08819

[43] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Z. Zhang, "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, Jun. 2015, pp. 842–850. doi: 10.1109/CVPR.2015.7298685.

[44] F. Wang *et al.*, "Residual Attention Network for Image Classification," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, Jul. 2017, pp. 6450–6458. doi: 10.1109/CVPR.2017.683.

[45] X. Shi *et al.*, "Loss-Based Attention for Interpreting Image-Level Prediction of Convolutional Neural Networks," *IEEE Trans. on Image Process.*, vol. 30, pp. 1662–1675, 2021, doi: 10.1109/TIP.2020.3046875.

[46] S. Seo, J. Huang, H. Yang, and Y. Liu, "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, Aug. 2017, pp. 297–305. doi: 10.1145/3109859.3109890.

[47] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net," 2015. [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2015/DB15a

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

[49] H. Liang *et al.*, "Training Interpretable Convolutional Neural Networks by Differentiating Class-Specific Filters," in *Computer Vision – ECCV 2020*, vol. 12347, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 622–638. doi: 10.1007/978-3-030-58536-5_37.

[50] K. Horii, K. Maeda, T. Ogawa, and M. Haseyama, "Paper Interpretable Convolutional Neural Network Including Attribute Estimation for Image Classification," vol. 8, no. 2, p. 14, 2020.

[51] Q. Zhang, X. Wang, Y. N. Wu, H. Zhou, and S.-C. Zhu, "Interpretable CNNs for Object Classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, Mar. 2020, doi: 10.1109/TPAMI.2020.2982882.

[52] Y. Sun, S. Ravi, and V. Singh, "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), Oct. 2019, pp. 4937–4946. doi: 10.1109/ICCV.2019.00504.

[53] B. Yin, L. Tran, H. Li, X. Shen, and X. Liu, "Towards Interpretable Face Recognition," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), Oct. 2019, pp. 9347–9356. doi: 10.1109/ICCV.2019.00944.

[54] C. Hwa Yoo, N. Kim, and J.-W. Kang, "Relevance Regularization of Convolutional Neural Network for Interpretable Classification," Jun. 2019.

[55] Q. Zhang, R. Cao, Y. N. Wu, and S.-C. Zhu, "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, 2017, pp. 2898–2906.

[56] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, "This Looks Like That: Deep Learning for Interpretable Image Recognition," in *Advances in Neural Information Processing Systems*, 2019, vol. 32. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/adf7ee2dcf142b0e11888e72b43fcb75-Paper.pdf

[57] D. Rymarczyk, Ł. Struski, J. Tabor, and B. Zieliński, "ProtoPShare: Prototype Sharing for Interpretable Image Classification and Similarity Discovery," *Association for Computing Machinery*, pp. 1420–1430, Nov. 2020, doi: 10.1145/3447548.3467245.

[58] Q. Zhang, Y. Yang, Y. Liu, Y. N. Wu, and S.-C. Zhu, "Unsupervised Learning of Neural Networks to Explain Neural Networks," *arXiv:1805.07468 [cs]*, May 2018, Accessed: Feb. 20, 2021. [Online]. Available: http://arxiv.org/abs/1805.07468

[59] A. Tavanaei, "Embedded Encoder-Decoder in Convolutional Networks Towards Explainable AI," *arXiv:2007.06712 [cs]*, Jun. 2020, Accessed: Feb. 20, 2021. [Online]. Available: http://arxiv.org/abs/2007.06712

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

[61] M. Yeganejou, S. Dick, and J. Miller, "Interpretable Deep Convolutional Fuzzy Classifier," *IEEE Trans. Fuzzy Syst.*, pp. 1–1, 2019, doi: 10.1109/TFUZZ.2019.2946520.

[62] N. Papernot and P. McDaniel, "Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning," *ArXiv*, vol. abs/1803.04765, 2018.

[63] D. Alvarez-Melis and T. Jaakkola, “Towards Robust Interpretability with Self-Explaining Neural Networks,” 2018.
[64] X. Liu, X. Wang, and S. Matwin, “Interpretable Deep Convolutional Neural Networks via Meta-learning,” *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, 2018.
[65] Y. Wang, H. Su, B. Zhang, and X. Hu, “Interpret Neural Networks by Extracting Critical Subnetworks,” *IEEE Transactions on Image Processing*, vol. 29, pp. 6707–6720, 2020, doi: 10.1109/TIP.2020.2993098.
[66] R. Abbasi-Asl and B. Yu, “Structural Compression of Convolutional Neural Networks Based on Greedy Filter Pruning,” *ArXiv*, vol. abs/1705.07356, 2017.
[67] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, “Network Dissection: Quantifying Interpretability of Deep Visual Representations,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3319–3327, 2017.
[68] F. Ventura and T. Cerquitelli, “What’s in the box? Explaining the black-box model through an evaluation of its interpretable features,” *ArXiv*, vol. abs/1908.04348, 2019.
[69] M. Tamajka, W. Benesova, and M. Kompanek, “Transforming Convolutional Neural Network to an Interpretable Classifier,” in *2019 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2019, pp. 255–259. doi: 10.1109/IWSSIP.2019.8787211.
[70] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, “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, 2016, pp. 3395–3403.
[71] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, “On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation,” *PLOS ONE*, vol. 10, no. 7, pp. 1–46, 2015, doi: 10.1371/journal.pone.0130140.
[72] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic Attribution for Deep Networks,” in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, Sydney, NSW, Australia, 2017, pp. 3319–3328.
[73] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning Important Features Through Propagating Activation Differences,” in *Proceedings of the 34th International Conference on Machine Learning*, International Convention Centre, Sydney, Australia, Aug. 2017, vol. 70, pp. 3145–3153. [Online]. Available: http://proceedings.mlr.press/v70/shrikumar17a.html
[74] N. U. Islam and S. Lee, “Interpretation of Deep CNN Based on Learning Feature Reconstruction With Feedback Weights,” *IEEE Access*, vol. 7, pp. 25195–25208, 2019, doi: 10.1109/ACCESS.2019.2899901.
[75] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps,” *CoRR*, vol. abs/1312.6034, 2014.
[76] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, “Deconvolutional networks,” in *2010 IEEE Computer Society Conference on computer vision and pattern recognition*, 2010, pp. 2528–2535.
[77] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2921–2929. doi: 10.1109/CVPR.2016.319.
[78] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks,” in *2018*

IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 839–847. doi: 10.1109/WACV.2018.00097.

[79] D. Omeiza, S. Speakman, C. Cintas, and K. Weldemariam, "Smooth Grad-CAM++: An Enhanced Inference Level Visualization Technique for Deep Convolutional Neural Network Models," *ArXiv*, vol. abs/1908.01224, 2019.

[80] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "SmoothGrad: removing noise by adding noise," *ArXiv*, vol. abs/1706.03825, 2017.

[81] P. Morbidelli, D. Carrera, B. Rossi, P. Fragneto, and G. Boracchi, "Augmented Grad-CAM: Heat-Maps Super Resolution Through Augmentation," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4067–4071. doi: 10.1109/ICASSP40776.2020.9054416.

[82] B. Patro, M. Lunayach, S. Patel, and V. Namboodiri, "U-CAM: Visual Explanation Using Uncertainty Based Class Activation Maps," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 7443–7452. doi: 10.1109/ICCV.2019.00754.

[83] M. Bany Muhammad and M. Yeasin, "Eigen-CAM: Visual Explanations for Deep Convolutional Neural Networks," *SN Computer Science*, vol. 2, no. 1, p. 47, Jan. 2021, doi: 10.1007/s42979-021-00449-3.

[84] B. Zhou, Y. Sun, D. Bau, and A. Torralba, "Interpretable Basis Decomposition for Visual Explanation," in *Computer Vision – ECCV 2018*, Cham, 2018, pp. 122–138.

[85] R. C. Fong and A. Vedaldi, "Interpretable Explanations of Black Boxes by Meaningful Perturbation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3449–3457. doi: 10.1109/ICCV.2017.371.

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

[87] P. Hase, C. Chen, O. Li, and C. Rudin, "Interpretable Image Recognition with Hierarchical Prototypes," *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 7, pp. 32–40, 2019, [Online]. Available: https://ojs.aaai.org/index.php/HCOMP/article/view/5265

[88] L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach, "Women Also Snowboard: Overcoming Bias in Captioning Models," in *Computer Vision – ECCV 2018*, Cham, 2018, pp. 793–811.

[89] Y. Goyal, A. Mohapatra, D. Parikh, and D. Batra, "Towards Transparent AI Systems: Interpreting Visual Question Answering Models," *arXiv: Computer Vision and Pattern Recognition*, 2016.

[90] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards Better Analysis of Deep Convolutional Neural Networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 91–100, 2017, doi: 10.1109/TVCG.2016.2598831.

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

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

[93] I. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, 2014, vol. 27, pp. 2672–2680. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf
[94] L. Gatys, A. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," *Journal of Vision*, vol. 16, no. 12, pp. 326–326, Sep. 2016, doi: 10.1167/16.12.326.
[95] R. Ibrahim and M. O. Shafiq, "Augmented Score-CAM: High resolution visual interpretations for deep neural networks," *Knowledge-Based Systems*, vol. 252, p. 109287, 2022, doi: https://doi.org/10.1016/j.knosys.2022.109287.
[96] L. Taylor and G. Nitschke, "Improving Deep Learning with Generic Data Augmentation," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018, pp. 1542–1547. doi: 10.1109/SSCI.2018.8628742.
[97] C. Hill, *Learning scientific programming with Python*. Cambridge University Press, 2020.
[98] A. I. R. Galarza and J. Seade, *Introduction to classical geometries*. Springer Science & Business Media, 2007.
[99] F. Chollet and others, "Keras," 2015. https://github.com/fchollet/keras
[100] diegocarrera89 and sofficelli, "Augmented Grad-CAM code," Nov. 12, 2019. https://github.com/diegocarrera89/AugmentedGradCAM (accessed Oct. 01, 2020).
[101] andreysorokin, "Score-CAM code," Nov. 01, 2019. https://github.com/andreysorokin/scam-net (accessed Oct. 01, 2020).
[102] E. Bisong, "Google Colaboratory," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, Berkeley, CA: Apress, 2019, pp. 59–64. doi: 10.1007/978-1-4842-4470-8_7.
[103] F. Pérez and B. E. Granger, "IPython: a System for Interactive Scientific Computing," *Computing in Science and Engineering*, vol. 9, no. 3, pp. 21–29, May 2007, doi: 10.1109/MCSE.2007.53.
[104] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.
[105] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1409.1556
[106] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.
[107] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
[108] S. Gabbard and J. Yang, "Quora insincere question classification," presented at the Baskin Engineering, University of California, Santa Cruz. [Online]. Available: https://www.kaggle.com/c/quora-insincere-questions-classification/overview
[109] L. Kreisköther, "grad-cam-analysis," 2020. [Online]. Available: https://github.com/lkreiskoether/grad-cam-analysis

[110] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, “An introduction to decision tree modeling,” *Journal of Chemometrics*, vol. 18, no. 6, pp. 275–285, 2004, doi: https://doi.org/10.1002/cem.873.
[111] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, “MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9584–9592. doi: 10.1109/CVPR.2019.00982.
[112] M. S. Gönül, D. Önkal, and M. Lawrence, “The effects of structural characteristics of explanations on use of a DSS,” *Decision Support Systems*, vol. 42, no. 3, pp. 1481–1493, 2006, doi: https://doi.org/10.1016/j.dss.2005.12.003.
[113] S. Williams, “An Experiment in Demonstrating and Mitigating Bias in Image Classification,” 2021.
[114] H. Vutukuri and sofficelli, “Classification of Dogs and Wolves, Version 2,” Feb. 06, 2020. https://www.kaggle.com/harishvutukuri/dogs-vs-wolves (accessed Aug. 15, 2021).
[115] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.
[116] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30. doi: 10.1109/IROS.2017.8202133.
[117] X. Zheng, T. Chalasani, K. Ghosal, S. Lutz, and A. Smolic, “STaDA: Style Transfer as Data Augmentation,” 2019, pp. 107–114. doi: 10.5220/0007353401070114.
[118] G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, and J. Shlens, “Exploring the structure of a real-time, arbitrary neural artistic stylization network,” *CoRR*, vol. abs/1705.06830, 2017, [Online]. Available: http://arxiv.org/abs/1705.06830
[119] R. Ibrahim and M. O. Shafiq, “Towards a New Approach for Empowering the MR-DBSCAN Clustering for Massive Data Using Quadtree,” in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2018, pp. 91–98. doi: 10.1109/HPCC/SmartCity/DSS.2018.00044.
[120] R. Naidu and J. Michael, “SS-CAM: Smoothed Score-CAM for sharper visual feature localization,” *arXiv preprint arXiv:2006.14255*, 2020.
[121] R. Naidu, A. Ghosh, Y. Maurya, S. S. Kundu, and others, “IS-CAM: Integrated Score-CAM for axiomatic-based explanations,” *arXiv preprint arXiv:2010.03023*, 2020.
[122] J. Haspiel *et al.*, “Explanations and Expectations: Trust Building in Automated Vehicles,” in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, New York, NY, USA, 2018, pp. 119–120. doi: 10.1145/3173386.3177057.
[123] K. Goel, R. Sindhgatta, S. Kalra, R. Goel, and P. Mutreja, “The effect of machine learning explanations on user trust for automated diagnosis of COVID-19,” *Computers in Biology and Medicine*, vol. 146, p. 105587, 2022.
[124] M. Drosou, H. Jagadish, E. Pitoura, and J. Stoyanovich, “Diversity in big data: A review,” *Big data*, vol. 5, no. 2, pp. 73–84, 2017.
[125] T. Mundhenk, B. Chen, and G. Friedland, “Efficient Saliency Maps for Explainable AI,” *ArXiv*, vol. abs/1911.11293, 2019.
[126] Y. Yao, Y. Huang, and Y. Wang, “Unpacking People’s Understandings of Bluetooth Beacon Systems - A Location-Based IoT Technology,” in *Proceedings of the 52nd Hawaii*

International Conference on System Sciences, 2019, pp. 1638–1647. doi: 10.24251/HICSS.2019.198.

[127] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[128] Y. Lou, R. Caruana, and J. Gehrke, "Intelligible Models for Classification and Regression," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: Association for Computing Machinery, 2012, pp. 150–158. [Online]. Available: https://doi.org/10.1145/2339530.2339556

[129] P. Madhavan and D. A. Wiegmann, "Similarities and differences between human-human and human-automation trust: An integrative review.," *Theoretical Issues in Ergonomics Science*, vol. 8, no. 4, pp. 277–301, 2007, doi: 10.1080/14639220500337708.

[130] E. Alberdi, P. Ayton, A. A. Povyakalo, and L. Strigini, "Automation bias and system design: a case study in a medical application," in *2005 The IEE and MOD HFI DTC Symposium on People and Systems - Who Are We Designing For (Ref. No. 2005/11078)*, 2005, pp. 53–60. doi: 10.1049/ic:20050451.

[131] P. Keller and A. Drake, "Exclusivity and paternalism in the public governance of explainable AI," *Computer Law & Security Review*, vol. 40, p. 105490, 2021, doi: https://doi.org/10.1016/j.clsr.2020.105490.

[132] M. Ebers, "Regulating Explainable AI in the European Union. An Overview of the Current Legal Framework (s)," *An Overview of the Current Legal Framework (s)(August 9, 2021). Liane Colonna/Stanley Greenstein (eds.), Nordic Yearbook of Law and Informatics*, 2020.

[133] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," 2017.

[134] D. Cheng, S. Zhang, Z. Deng, Y. Zhu, and M. Zong, "kNN Algorithm with Data-Driven k Value," in *Advanced Data Mining and Applications*, vol. 8933, X. Luo, J. X. Yu, and Z. Li, Eds. Cham: Springer International Publishing, 2014, pp. 499–512. doi: 10.1007/978-3-319-14717-8_39.

[135] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979, [Online]. Available: http://www.jstor.org/stable/2346830

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

[137] X. Zhang, N. Wang, H. Shen, S. Ji, X. Luo, and T. Wang, "Interpretable Deep Learning under Fire," in *29th USENIX Security Symposium (USENIX Security 20)*, Aug. 2020, pp. 1659–1676. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/zhang-xinyang

[138] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks," in *2016 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, May 2016, pp. 582–597. doi: 10.1109/SP.2016.41.

[139] A. S. Hashemi and S. Mozaffari, "Secure deep neural networks using adversarial image generation and training with Noise-GAN," *Computers & Security*, vol. 86, pp. 372–387, Sep. 2019, doi: 10.1016/j.cose.2019.06.012.

[140] A. S. Rakin and D. Fan, "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, Jul. 2019, pp. 332–337. doi: 10.1109/ISVLSI.2019.00067.

[141] A. Mustafa, S. H. Khan, M. Hayat, J. Shen, and L. Shao, "Image Super-Resolution as a Defense Against Adversarial Attacks," *IEEE Trans. on Image Process.*, vol. 29, pp. 1711–1724, 2020, doi: 10.1109/TIP.2019.2940533.
[142] F. Menet, P. Berthier, M. Gagnon, and J. M. Fernandez, "Spartan Networks: Self-feature-squeezing neural networks for increased robustness in adversarial settings," *Computers & Security*, vol. 88, p. 101537, Jan. 2020, doi: 10.1016/j.cose.2019.05.014.
[143] K. Han, Y. Li, and J. Hang, "Adversary resistant deep neural networks via advanced feature nullification," *Knowledge-Based Systems*, vol. 179, pp. 108–116, Sep. 2019, doi: 10.1016/j.knosys.2019.05.007.
[144] A. Radford *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
[145] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," *Advances in neural information processing systems*, vol. 32, 2019.
[146] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," *arXiv preprint arXiv:2005.00928*, 2020.
[147] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein, "A Diagnostic Study of Explainability Techniques for Text Classification," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, Nov. 2020, pp. 3256–3274. doi: 10.18653/v1/2020.emnlp-main.263.
[148] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.
[149] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability Methods for Graph Convolutional Neural Networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10764–10773. doi: 10.1109/CVPR.2019.01103.
[150] R. Schwarzenberg, M. Hübner, D. Harbecke, C. Alt, and L. Hennig, "Layerwise Relevance Visualization in Convolutional Text Graph Classifiers," *ArXiv*, vol. abs/1909.10911, 2019.
[151] T. Schnake *et al.*, "Higher-Order Explanations of Graph Neural Networks via Relevant Walks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021, doi: 10.1109/TPAMI.2021.3115452.
[152] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating Explanations for Graph Neural Networks," *Advances in neural information processing systems*, vol. 32, pp. 9240–9251, 2019.