SAR ATR Analysis and Implications for Learning

Johannes Bauer[a], Efrain H. Gonzalez[a], William M. Severa[a], and Craig M. Vineyard[a]

[a]Sandia National Laboratories, 1515 Eubank SE, Albuquerque, NM, United States

ABSTRACT

Deep neural networks for automatic target recognition (ATR) have been shown to be highly successful for a large variety of Synthetic Aperture Radar (SAR) benchmark datasets. However, the black box nature of neural network approaches raises concerns about how models come to their decisions, especially when in high-stake scenarios. Accordingly, a variety of techniques are being pursued seeking to offer understanding of machine learning algorithms. In this paper, we first provide an overview of explainability and interpretability techniques introducing their concepts and the insights they produce. Next we summarize several methods for computing specific approaches to explainability and interpretability as well as analyzing their outputs. Finally, we demonstrate the application of several attribution map methods and apply both attribution analysis metrics as well as localization interpretability analysis to six neural network models trained on the Synthetic and Measured Paired Labeled Experiment (SAMPLE) dataset to illustrate the insights these methods offer for analyzing SAR ATR performance.

Keywords: Neural Networks, Explainability, Interpretability, ATR, SAR, SAMPLE

1. INTRODUCTION

Performing automatic target recognition (ATR) on synthetic aperture radar (SAR) imagery is a difficult algorithmic task for several reasons including signal variability due to radar physics coupled with broad operating conditions, limited target data, and a tendency for targets to be difficult to detect. In the pursuit of performant exploitation algorithms, deep neural networks are increasingly being applied for classification tasks on SAR and general satellite imagery datasets.[1–3] Training deep neural networks consists of multiple components including choosing a dataset, deciding whether to apply augmentations to data, picking a model, and testing different hyperparameters. Once trained, a model's performance is commonly evaluated by measuring accuracy on a test dataset. This process is outlined in the left side of Figure 1. Recent research related to using deep neural networks in SAR ATR has focused on improving such models by changing these basic components to improve accuracy as well as investigate model training reproducibility.[1,4] These changes could include exploring factors such as the impact of newer, larger, or multimodal datasets that impact learning (BigEarthNet, UNICORN, etc.), assessing the effects of data augmentation on datasets, or using larger or more complex machine learning algorithms.[1–3,5] Through this progress, deep neural networks have been shown to perform well at classification and segmentation tasks for SAR imagery.

However, one factor in developing neural networks that has not been extensively applied in the SAR ATR domain is *model introspection*. Model introspection, which includes explainability and interpretability methods, attempts to identify why a neural network may make certain decisions, as illustrated in the right half of Figure 1. These methods are vital to understanding the predictions of a model. Providing explanations, particularly in terms suited for a domain expert, can increase the confidence of using such neural networks in high stakes scenarios. Furthermore, explanations provide additional ways to evaluate the performance or value of a trained network. Although the accuracy of a model is an important metric for evaluation, researchers who build deep neural networks are also interested in determining if learned features of a model capture the essence of the objects in the data. By using explainability, one can better understand whether a deep neural network is using such features of an object, or rather using random or seemingly random or unimportant information.[6–8]

The goal of this paper is to provide a quick introduction of model introspection methods and show how they can be leveraged in the SAR ATR domain. First, in Section 2, this paper will briefly discuss some basic

Further author information: Send correspondence to Craig M. Vineyard (E-mail: cmviney@sandia.gov)

Automatic Target Recognition XXXIV, edited by Kenny Chen, Riad I. Hammoud, Timothy L. Overman, Proc. of SPIE Vol. 13039, 130390L · © 2024 SPIE
0277-786X · doi: 10.1117/12.3014042

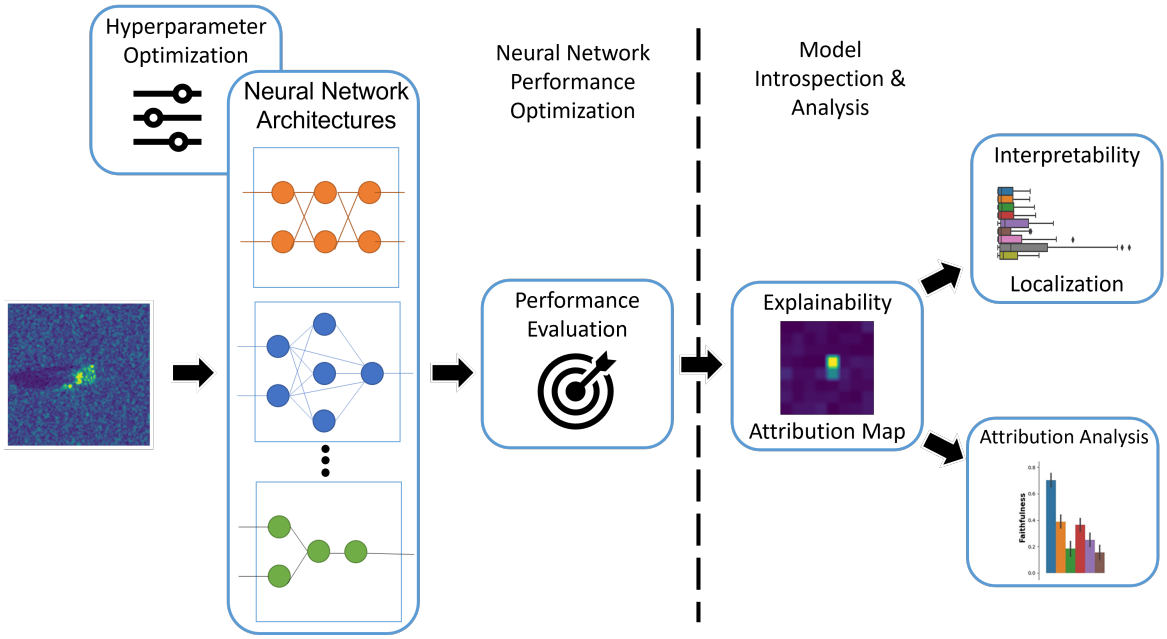

Figure 1. The typical focus of machine learning development is selecting a model and optimizing its performance via learning and hyperparameter tuning as shown on the left. Additionally, beyond just performance metrics like accuracy, the right emphasizes the role of striving to understand model performance properties (which is the focus of this paper).

definitions surrounding model explainability and interpretability. Next we also introduce metrics which provide quantitative analysis of attribution map properties, as well as localization methods to interpret attributions. In Section 3, we introduce a set of attribution map methods and metrics which we demonstrate applied to six models trained on the SAMPLE dataset. Section 4 highlights results of these experiments, showcasing variability and trends across different models, methods, and metrics. Overall, we highlight the impact explainability and interpetability can bring to understanding neural network performance for SAR ATR.

2. OVERVIEW OF EXPLAINABILITY AND INTERPRETABILITY

While literature uses the common terms explainability and interpretability to refer to the study of how machine learning algorithms make decisions, the definition of these terms varies widely from source to source. Due to the different definitions of these terms, there can be confusion in this area of research surrounding what insights a method provides and what such methods actually say about how a model makes decisions. Accordingly, next we summarize some of the connotations before establishing the meaning we use throughout this paper.

One thorough definition of explainability and interpretability comes from Arrieta et al.[9] They describe explainability as a characteristic of a model in which one can develop methods to better understand how a model makes decisions. Within this definition, what is considered a good or bad explainability method is dependent on the audience and their domain expertise. For example, attribution maps highlight what portions of an input are relevant to a model's decision making. By this explanation a SAR analyst could readily gain understanding about a model by seeing what pixels are influencing classifications. However, attribution maps may not be considered explanations in a different context, like time series analysis, where the attribution scores may not simplify insight into how a model makes decision.[9] Interpretability, in contrast, is described as a characteristic of a model in which the decision making calculation of a model is understandable to a human observer. Therefore, by Arrieta's taxonomy, algorithms that are considered black box models cannot be interpretable because interpretability is a characteristic that is inherent to the model and not one that can be merely observed once the model has been trained.

Rudin critiques explainability for deep neural networks giving definitions that are different from Arrieta, which leads to different conclusions about how one looks at explainability.[10] Namely, Rudin refers to explainability as

the ability to use a secondary model that is interpretable to explain the black box model whereas interpretability is viewed as a domain specific characteristic that is inherent to a model. Using this definition Rudin draws the conclusion that explainability methods do not provide accurate information regarding the way in which neural networks make their decisions. Rather, explainability methods give summary statistics on how predictions relate to the features.[10] Accordingly, a key implication of this definition is that few deep neural networks are interpretable. Rather, they are explainable, which means that at best it may be possible to understand the way a model makes its predictions by finding connections between features in the model and mapping them to the output of the model. Further complicating the field, in lieu of aligning with a formal notion of interpretability versus explainability, many works use them interchangeably.

Accordingly, it is vital to define these terms in this paper to give a grounded foundation on what is meant when we use these terms. In this context, having the capability to **extract informative factors** about how a model effectively does its task is referred to as *explainability*. We contrast this concept with that of *interpretability*, which we define as a **synthesis of low-level explainability factors** into high-level understandable terms.

Using this perspective, attribution map methods are considered explainability methods, but any qualitative or quantitative methods that answer questions about whether an attribution map highlights certain definable features in an input would be considered interpretability methods. For the reasons stated below, we have chosen to deviate from Rudin's and Arrietta's definitions. First, the phrasing used in our definitions allow for a clear distinction between explainability and interpretability. Second, our definition of explainability adheres to Rudin's claim regarding the majority of explainability methods. The claim is that explainability methods do not always mimic what a model is doing, but may at times be consistent with how a model is making decisions. Therefore, when using attribution map methods, it is important to approach them with hesitancy and use metrics to evaluate the degree to which the explanation should be trusted. The metrics we studied are defined in Section 3.4. Third, the chosen phrasing allows for an explainability method to be defined independently of the domain and rather shifts this dependence onto interpretability methods.

Figure 2 provides a conceptual depiction of explainability and interpretability for explanatory purposes. As illustrated, for a cattle ranch, branding offers a means of attributing which livestock belong to a ranch. This mechanism of explaining all of the cattle a ranch possesses across various pastures can then further facilitate additional interpretation such as an understanding of how many yearlings a ranch may have or the average weight of their mature cows, and other insights which bring general understanding regarding the state of a ranch.

In addition to the definitions of these terms, there are different categories of granularity that offer insight into the explainability methods that are employed. Typically explainability methods are differentiated into two key categories: feature-based explainability and instance-based explainability. Feature-based explainability refers to methods that attempt to give explanations of the model based on what features have the greatest influence in determining the output of a model. Instance-based explainability, which Bae et al. defines concisely, is a class of techniques that explain a model's predictions in terms of the examples on which the model was trained.[11] In other words, these methods can give an approximation of how much a given test example influences the training of a model. Listed below are a few properties that can be used to characterize the methods.

Model Agnostic/Specific: Model agnostic explainaibility methods are methods that can be applied to any model architecture. This would include methods like attribution maps which give information on what were the most important features of an input to a model when making decisions. In contrast, model specific methods are methods which can only be applied to specific model architectures.

Intrinsic/Post-hoc: Intrinsic methods are those that explain the model while the model is being trained. In contrast, post-hoc methods are explainability methods which are done after the model is trained.

Global/Local: Global explainability methods are methods that make broader assertions on what a model is using from the data in order to complete a task. Local explainability methods are those which provide information on the features that are relevant to an individual data point.

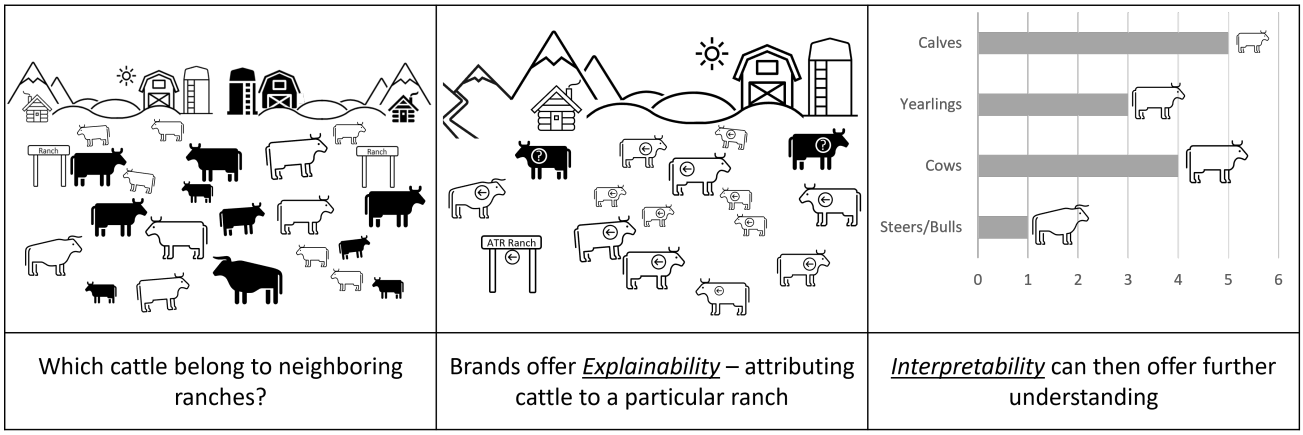

Figure 2. Conceptual illustration of explainability and interpretability. With two neighboring ranches how do you discern cattle ownership if their livestock overlap (left)? As shown in the middle, brands as an attribution method enable explaining where they reside across the ranch (as well as separating errant other cattle in the herd). This attribution then allows interpretability methods to convey further insights. For example, beyond which cattle belong to the ranch, further analysis can assess the state of the ranch based upon the distribution of cattle by age/size categories as shown on the right.

2.1 Explainability Via Attribution Maps

One of the most popular explainability methods is attribution maps: a generally post-hoc, model-agnostic method that assigns importance to characteristics in an instance of your dataset that are believed to contribute most to the classification of said instance. There are two different types of attribution map methods: gradient-based and perturbation-based. Gradient-based methods use information from the gradient in order to measure how important a characteristic or feature is to the determination of the output of the model or alternatively an individual neuron or layer. There are two common ways to measure such values. First, gradient-only methods use a backward pass of a model to calculate the importance for a specific characteristic of the input. The other is path attribution which calculates total feature importance by establishing a baseline image and using a measure of the distance between the baseline image and the original image to create an attribution map.[12]

Perturbation based attribution methods calculate the influence of features by altering specific sections of an input and calculating importance by measuring how much the perturbation affects the classification. There are at least two broad categories of perturbation methods. One of which can be seen as pure perturbation methods which measures feature importance by perturbing the input images. This includes Shapley Additive values and Occlusion.[13,14] In contrast, there are surrogate model type perturbation methods which do not interpolate a model based solely on perturbation, but rather create a surrogate model trained on such perturbations which attempts to explain the inner-workings of the larger model.

2.2 Analysis of Attribution Properties

We label the metrics described in this section attribution evaluation metrics instead of explainability methods or interpretability metrics because their role is not to offer further understanding of attribution maps, but rather to examine characteristics of attribution methodologies. These metrics should be seen as a way to test whether or not the information from attribution maps is trustworthy and as an important step in explaining neural networks. Although attribution maps are being utilized to support understanding neural networks in critical decision making such as medical imaging[15] and time series analysis for recurrent neural networks,[16] there is research that shows that they can be untrustworthy. Notably, Adebayo et al. concludes in their research that *"some widely deployed saliency methods are independent of both the data the model was trained on, and the model parameters."*[17] Importantly, this illustrates that if there are cases where the parameters of the model do not affect the attribution map, then it can be concluded that some attribution maps do not contain information about the model. Furthermore, Rudin et al.[10] identified several challenges related to trust and attribution maps. Chief among them is the potential for misleading qualitative assessment. For example, attribution maps

of correctly classified images appear to support the assertion a model is behaving in a desired manner. However, similar images that are not classified correctly may have relatively similar attribution maps. Therefore, although attribution maps are used to highlight sections of images that are important for classification, those highlighted sections do not always identify the influential parts of an image.[10]

In response to these issues, several metrics have been proposed to measure the efficacy of an attribution method with a particular model. Four key categories for the evaluation of explainability methods are:[18]

Faithfulness: a measure of the extent to which the input features highlighted by an attribution map correlate with model performance.[18]

Robustness: measures the change in the attribution map when the input to a model is perturbed. The hypothesis is that similar images should have similar attribution maps. Therefore, if small perturbations to an initial image drastically changes the attribution map, then the attribution map is not a reliable way of determining the important features that are used by the model.[18]

Complexity: measures the extent to which an explanation of a model may be considered concise.[18]

Randomization: measures the dependence of the attribution map on the parameters of the model. It is hypothesized that if the attribution map remains unchanged when the parameters of the model are changed, then the attribution map is not beneficial for understanding the model.[18]

Figure 3 provides a conceptual illustration of these concepts. In part a), marking everything from cattle to cows to ranch vehicles and buildings, while accurate, does not faithfully capture the amount of cattle on the ranch. And so more than just a marketing mechanisms, a faithful attribution method needs to capture useful details. Just a model's parameters should matter for attributions to convey meaningful information, shown in b) swapping out cattle for chickens does not make sense just because the ranch can raise one animal. Part c) of the figure portrays that having an abundance of brands or an extra large brand does not offer added attribution value. And lastly, attributions need to be robust to inconsequential changes such as in d) where a different breed of cattle are still attributed to the ranch.

2.3 Interpretability Via Localization Metrics

Localization metrics are ways to quantify the degree to which an attribution map is centered around a region of interest.[18] While Hedstrom et al.[18] and many papers use localization metrics as an attribution evaluation metric, in this paper it is proposed that these metrics should actually be viewed as interpretability methods for attribution maps. Using localization as an attribution evaluation metric is problematic because of the unpredictable nature of deep neural networks. Even if an attribution map does not conform to the expectations of a domain expert, the attribution map method may still be valid. Localization metrics should then be considered interpretability methods because these metrics are ways of quantifying how localized or dispersed an attribution is in identifying the influential portions of an input.

3. METHODS

Within this section we introduce the neural network models, datasets, basic attribution methods and metrics that we demonstrate in this paper. All of the attribution map implementation code comes from Captum which is an open-source library dedicated for explainability of deep neural networks.[19] The attribution evaluation metrics as well as some of the interpretability localization metrics were implemented via a library named Quantus [*]. This library consists of modules that evaluate the trustworthiness of attribution maps specifically for convolutional neural networks.[18]

[*]Custom implementations were developed for AUC-Judd based upon,[20] NSS based upon ,[21] and IG based upon [20]

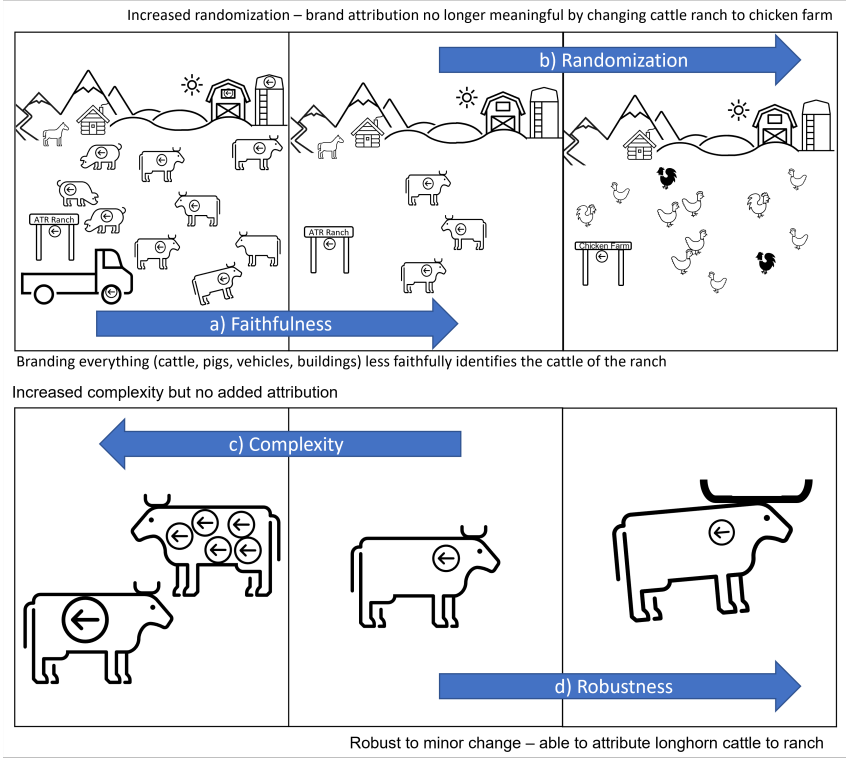

Figure 3. Brands when properly employed offer attribution of cattle to a ranch. Here we showcase attribution evaluation metric concepts via this analogy. a) Faithful attribution marks what is necessary (shown here branding everything does not offer attribution benefit) b) Random changes may not make sense c) Complex attributions are not necessarily more informative, and d) Robust attribution overcomes variability.

Both of these frameworks offer a variety of methods and metrics, each of which may have a set of parameters specific to the particular technique. Accordingly, we have used the defaults parameters for our illustrative purpose. Investigating the optimal settings for SAR ATR requires further research investigation, and is potentially dependent upon the particular dataset and model.

3.1 Dataset

The dataset we have used for training the models as well as for producing attribution maps is the Synthetic and Measured Paired Label Experiment (SAMPLE) dataset which consists of 10 different classes of military ground vehicles.[22] This dataset consists of synthetic (predicted) data for training and real (measured) SAR data for testing. This small dataset consists of a total of 1366 SAR chips across all the classes. In this paper, for illustration purposes rather than exhaustive analysis, we use a selection of 20 images per class (400 of the 1,346 total images) to produce attribution maps, conduct attribution evaluation, and localization analysis.

3.2 Models

The neural network models used in this study are VGG13, VGG16, EfficientNet-B0, EfficientNetV2 small, MNASNet, and ShuffleNet.[23–27] Each model was trained on synthetic non-augmented SAR data from the SAMPLE data and tested on the corresponding set of real data (denoted SAMPLE Real). In Table 3.1 basic information is shown about each model. The models have been chosen to represent a range of sizes as well as complexities. Each model performs at a high accuracy. More information about the tradespace of model computational structure and performance can be found in Melzer et al.[1,2]

Table 1. List of Models

Model	Parameter Count	SAMPLE Accuracy
VGG16	134,308,810	93.82%
VGG13	128,996,554	93.98%
EfficientNetv2_s	11,178,378	93.01%
EfficientNet-b0	7,166,938	87.29%
MNASNet 0.35x	836,150	89.51%
ShuffleNet 0.5x	352,666	91.15%

3.3 Summary of Applied Attribution Map Methods

Deconvolution Deconvolution is a method that produces layer-wise visualizations that demonstrate the importance of each feature. In Zeiler and Fergus, deconvolutional layers are constructed out of an unpooling layer, a rectification, and a transposed filter layer.[28] Unpooling is accomplished through the use of "switches." The "switches" record information regarding the original location of the maximum values during the max pooling step of a convolutional layer.[28] The rectification step takes the feature maps obtained after unpooling and applies a ReLU activation function which removes any negative values. Filters within a deconvolutional layer are created by taking the transpose of the filters of their corresponding convolutional layer in the original network. In the last step of a deconvolutional layer, these filters are applied to the rectified feature maps. An attribution map based on deconvolution is created for a particular image by first taking the image as an input in a convolutional neural network. Then the resulting feature map is passed through the deconvolutional layers. The output of the deconvolutional network then serves as the attribution map.

Integrated Gradients The general concept of Integrated Gradients is to calculate importance of a pixel on an image by calculating a path integral of images between a baseline image and the target image. Formally, given a neural network denoted $\mathbf{f} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{1 \times k}$ where m and n are the dimensions of the image and k is the number of classes in your dataset, the integrated gradient is:

$$\text{Integrated Gradients}(X)_i = (X_i - \acute{X}_i) \int_0^1 \frac{\partial \mathbf{f}(\acute{X} - \alpha(X - \acute{X}))}{\partial X_i} \partial\alpha \tag{1}$$

where X is the image, $\acute{X}$ is the baseline image, and the i denotes the i^{th} pixel in both images.[12] One of the main motivations for developing this method was to construct a method of calculating influence that satisfied desirable axiomatic properties. Specifically, the method satisfies the axioms of sensitivity, implementation invariance, and completeness. The sensitivity property states that, if a difference between an input image and the baseline image causes a difference in prediction, then the differing features impart non-zero attribution. The implementation invariance property states that if the outputs for two networks are equal for all inputs, then their corresponding attribution maps should be identical. The completeness property states that the sum of the attributions is equal to the difference between the output of $\mathbf{f}$ when evaluated at the input image and the baseline image.

The choice of baseline is an aspect of this method that requires special consideration. In many situations, it is common to choose a black image as the baseline. However, in some domains it may make sense to change the baseline in order to attain better results. Sundararajan et al. advocate for the use of black images as baselines because these images best represent "missingness" within an image.[12] On the other hand, Sturmfels et al. describe the use of alternative baselines, some of which include using the target image with added Gaussian noise or using an image with a uniform distribution of pixel intensities.[29] The experiments conducted in this paper use basic black images as the baseline.

DeepLIFT Similar to Integrated Gradients, DeepLIFT requires the use of a baseline image in order to establish the contribution of a feature.[30] The contribution of a feature is defined as the amount of change in the output of the model that can be associated to the difference between the input feature and a baseline feature. The DeepLIFT algorithm uses multipliers which are analogous to partial derivatives, defines a chain rule for the multipliers, and establishes rules for handling the linear and nonlinear layers of a neural network in order to backpropagate the "contributions of all neurons in the network to every feature of the input.[31]" However, the RevealCancel rule created by Shrikumar et al. is not currently implemented in Captum and so the results presented in this paper are not representative of the full DeepLIFT algorithm.[19]

Feature Ablation The feature ablation method is a perturbation-based method in which an image is split into different subsections, and, for each forward pass of the image through the model, a subsection of the image is chosen to be replaced with a baseline value. The influence of each perturbed subsection of an image is determined by the difference between the output from a forward pass of the original image and the output of the perturbed image. This method was implemented using Captum,[19] where each 128 by 128 image is split into subsections with dimensions 3 pixels high and 3 pixels long.

Occlusion Similar to feature ablation, occlusion is a perturbation-based attribution map method which calculates the influence of each subsection of an input image by looking at the change in the class output of the model when that subsection is perturbed. For this method, an input image is first split into rectangular subsections, and, at every iteration, one subsection is replaced with a baseline and the resulting perturbed image is run through the model to obtain an output. The attribution score associated with the subsection is determined by the difference in the output of the model for the original input image and the output for the perturbed image. Unlike feature ablation, occlusion allows for features to lie in multiple rectangular regions therefore the attribution scores for those features are averaged.[19] The use of rectangular regions potentially makes occlusion better suited for handling image data because it incorporates the local dependence between the pixels within the region.

GradientSHAP GradientSHAP, which is also referred to as *expected gradients*,[19] is a method which aims to reduce the uncertainty in selecting a baseline image by averaging over multiple baseline images.[29] The process for GradientSHAP requires that several baseline images be created for a given input image. Noise is sampled several times from a Gaussian distribution and added to the input image to create several different baseline images. The integrated gradients method is used to calculate an attribution map for the input image and each baseline image. Lastly, the attribution maps are averaged in order to obtain the final attribution map. Under certain assumptions the method can be used to approximate SHAP values.

KernelSHAP KernelSHAP is an extension of LIME[32] which leverages the properties of Shapley values in order to approximate the influence of features on the classification of images.[13] The model that approximates these values is of the form:

$$g(\acute{z}) = \phi_0 + \sum_{k=1}^{M} \phi_k \acute{z}_k, \tag{2}$$

where ϕ_k represents the influence associated with feature k, $\acute{z}_k$ represents a "simplified input feature" that has a value of 1 or 0 which indicates the presence or absence of the feature, M represents the number of such features, and ϕ_0 is the y-intercept.[13] KernelSHAP takes advantage of linear regression techniques in order to calculate attribution maps; accordingly it is more efficient than calculating Shapley values.[13]

Saliency The concept of Saliency revolves around the notion that gradients hold information on how individual pixels affect the classification of an image. Simonyan et al. describes their reasoning by stating that the linear models of interest consist of weighting an input and adding bias in order to gain some desired output.[33] By this reasoning, the final weights of a fully trained model can represent the importance of each component of the input in obtaining the desired output. Under a linear approximation for a deep neural network, the weights may be used as a measure of influence. Thus, formally Saliency is calculated in the following way:

$$\text{Saliency}(X) = \left| \frac{\partial \mathbf{f}\left(X_i\right)}{\partial X_i} \right|, \tag{3}$$

where $\mathbf{f}$ represents the neural network model, and X_i represent the i^{th} pixel in the image X. The derivatives used in this approximation are calculated by back-propagation. For grayscale images, the magnitude of the derivatives will be directly used to create the attribution map for the input image.

Input×Gradient The Input×Gradient method generates saliency maps as previously described and multiplies them by their respective input images. Scaling the attribution map by the image is intuitively appealing because this resembles the output that would be expected from a linear model and therefore results in attribution maps that are potentially more faithful to the model.[34]

3.4 Summary of Applied Attribution Evaluation Metrics

Faithfulness Correlation (Faithfulness): Faithfulness assumes that the important parts of attribution maps correspond to meaningful characteristics that impact classification. A ground truth for what can be considered meaningful for a network does not generally exist, therefore the idea of what is considered meaningful is tuned to the extent of how much the classification score changes when a given portion of an image is perturbed. It is assumed that by occluding portions of an image that were highlighted by an attribution map, the classification of the image will be affected more dramatically than if seemingly unimportant pixels were occluded. Thus, faithfulness correlation measures the correlation between scores and the change in the classification of an image when a portion of the image is perturbed. Formally, this is measured in the following way:

$$\mu(\mathbf{f}, \mathbf{g}; X) = \underset{S \in \binom{[d]}{|S|}}{corr} \left(\sum_{i \in S} \mathbf{g}(\mathbf{f}, X)_i,\ \mathbf{f}(X) - \mathbf{f}(X_p) \right), \tag{4}$$

where S denotes a subset of pixels, d denotes all pixels in an input image, X_p denotes the image with the pixels in S perturbed, and $\binom{[d]}{|S|}$ denotes the set of all sets of size $|S|$.[35] Within this equation, $\mathbf{g}$ is a function that takes in an image X and a function $\mathbf{f}$ which represents a neural network in order to output an attribution map. Note that if the attribution map method requires a significant amount of computation time, then this evaluation metric quickly becomes too computationally expensive. Additionally, it may be too computationally expensive to calculate this metric over all $\binom{[d]}{|S|}$ combinations of pixels.

Average sensitivity (Robustness): Robustness measures the change in the attribution map for a source image given that some pixels in the source image have been perturbed. Average sensitivity, developed by Yeh et al., is an example of a robustness metric which was developed out of the relationship between gradients and sensitivity.[36] In order to calculate average sensitivity one must first obtain an attribution map for the original image as well as attribution maps for several perturbed versions of the original image. For every perturbed image the difference between the original image and the perturbed image is calculated by using a chosen distance metric. Lastly, one must calculate the average of the differences. Therefore, higher values for this metric indicate that the attribution map method is more susceptible to small changes in the input.

Sparseness (Complexity): One way to calculate complexity is by using sparseness demonstrated in Chalasani et al.[37] They explain that the Gini index is a method of calculating sparseness. Given a neural network model $\mathbf{f}$, an image $X \in \mathbb{R}^{n \times m}$ and an attribution map method $\mathbf{g}(\mathbf{f}, X)$, the Gini index is defined as:

$$G(\mathbf{v}) = 1 - 2 \sum_{k=1}^{m \times n} \frac{\mathbf{v}_k}{||\mathbf{v}||_1} \left(\frac{(m \times n) - k + 0.5}{(m \times n)} \right) \tag{5}$$

where $\mathbf{v}$ is the flattened and sorted version of the attribution map produced by $\mathbf{g}(\mathbf{f}, X)$.[37] The value of $G(\mathbf{v})$ is bounded between 0 and 1, where higher values signify that the attribution map is more sparse. Note that it is desirable for the attribution map to be more sparse because it indicates that the attribution map is less complex and more readable.

Random Logits (Randomization): Random Logits is a measure of randomness for an attribution method. Sixt et al. note that attribution maps should be sensitive to the class that they belong to. The random logits metric allows for the comparison between the attribution map given one image and its corresponding label and the attribution map of that same image but with a randomly chosen label. If the attribution method is not random for a given neural network, the random logits value should be low. The paper quantifies such differences between images by using the structural similarity index (SIMM).[38]

3.5 Summary of Applied Localization Metrics

In order to apply localization metrics, a segmentation map is needed to act as a ground truth specifying precise locations of objects in an image. For SAR images, generally there are three characteristics of interest: the shadow, the target, and the background. Given the high complexity of the pixel distributions for SAR images, such segmentation of regions can be difficult without using advanced techniques like deep neural networks. However, there have been recent progress in creating these segmentation maps for the SAMPLE dataset specifically. This includes methods which utilize statistical tests[39] or clustering algorithms.[40] These methods generally consist of two phases. There is first the image processing stage which attempts to remove noise from the image. This makes boundaries of objects more defined within a SAR image. Afterwards, there is a machine learning step which attempts to create the segmentation maps for each of these models.

For the purpose of creating segmentation masks for localization metric analysis on the SAR SAMPLE dataset, this paper will use a form of the wavelet decomposition constant false alarm rate segmentation algorithm in Huang et al.[41] This method uses Wavelet decomposition in order to clean noise away from SAR images as well as make borders of the shadow and the target more well defined. Afterwards, a CFAR algorithm is used to pull out the target and shadow. We add additional image processing techniques in order to pull out the shadow as well as the target. The result of basic image processing led to more refined masks. However, it limits the segmentation model only to process SAR images from the SAMPLE or MSTAR dataset.

Area Under the Curve (AUC) One way to evaluate localization for attribution maps is by using the Area Under the Curve (AUC) metric.[20] Let A represent the set of all pixels within an attribution map for an individual image. Also, let K represent the segmentation map of the image and let S be the set of indices for the pixels in K. Let X represent the set of indices of pixels that belong to the set $\{a > t : a \in A, t \in \mathbb{R}\}$, where t is a chosen threshold. A true positive occurs when the index $x \in X$ is in S. A false positive occurs when the index $x \in X$ is not in S. With this information, one can create the receiver operating characteristics (ROC) curve and measure localization based on the area under the curve. The higher the AUC value, the better the attribution map highlights the desired section of the image. The variant of AUC described above is also referred to as AUC-Judd.[20] In our work, we are interested in assessing two particular sets of pixels $S \in K$ and $T \in K$ which represent the pixels of the shadow and the target respectively.

Normalized Scanpath Saliency (NSS) Given an attribution map A and a segmentation map K for some image, the Normalized Scanpath Saliency (NSS) metric is defined as:

$$\mathrm{NSS}(A, K) = \frac{1}{N} \sum_{i=1}^{N} \bar{A}_i \times K_i \tag{6}$$

where $\bar{A}$ is a normalized attribution map, i represents the i^{th} pixel, and N is the total number of pixels in the segmentation map that have values greater than 0. The higher the value is for NSS, the better the attribution map highlights the desired area.[20]

Relevance Rank Accuracy (RRA) Relevance rank accuracy was introduced by Arras et al. and measures the proportion of the pixels with the largest intensity, as determined by an attribution map, which lie within a chosen portion of an image.[42] Let P_I represent the set of pixels in the selected portion of an image and let A_I represent an attribution map for the image, then the metric may be calculated as follows:

$$\mathrm{RRA}(P_I, A_I) = \frac{|P_I \cap H_{A_I}|}{n} \tag{7}$$

where $n = |P_I|$ represents the number of pixels in P_I, H_{A_I} represents the set of n pixels with the largest attribution map scores, and the numerator is the number of pixels in the intersection of both sets. Higher values of relevance rank accuracy indicate that the model is relying more on the selected portion of the image.

Information Gain (IG) Information gain is an information theoretic method of calculating localization of segmentation maps. This metric measures the amount of saliency that is predicted by an attribution map for a given image relative to the amount predicted by an attribution map for a baseline image.[20] Larger values of information gain indicate that the saliency map is better at predicting relevant pixels than the baseline. Formally, information is measured in bits and the equation for information gain is defined as:

$$\mathrm{IG}(A, K) = \frac{1}{N} \sum_{i=1}^{N} K_i \times [\log_2(\epsilon + A_i) - \log_2(\epsilon + B_i)] \tag{8}$$

where i represents the index of the i^{th} pixel, K represents the segmentation map, A represents an attribution map, B represents a baseline map, and epsilon is a regularization term. Furthermore, N is the total number of pixels over which the metric is to be evaluated. Baseline maps can be varied according to the different types of attribution being tested. For this paper, we will be using a baseline map whose pixel intensity is drawn from a uniform distribution between 0 and 1.

4. RESULTS

4.1 Attribution Maps

Each of the attribution methods introduced in Section 3.3 produces an output attribution map for every inference. Furthermore, this is replicated across every neural network resulting in an abundance of results across the attribution map and model combinations we examine here. Figure 4 presents exemplar outputs where the input is a 2s1 tank from the SAMPLE real data. For each attribution map and model combination, input images were normalized and lightly blurred. The columns of Figure 4 are associated with a specific attribution map type while each row is associated with the model that was used to produce attribution maps.

As examples of insights that the attribution maps can provide for explaining model behavior we note the following. Overall, high intensity pixels seem to localize near the target (the 2s1 tank in this example) rather than other parts of the image like the shadow or background. There is some indication of a relationship between attribution maps of larger models like the EfficientNetv2_s and VGG16 models and high intensity pixels being

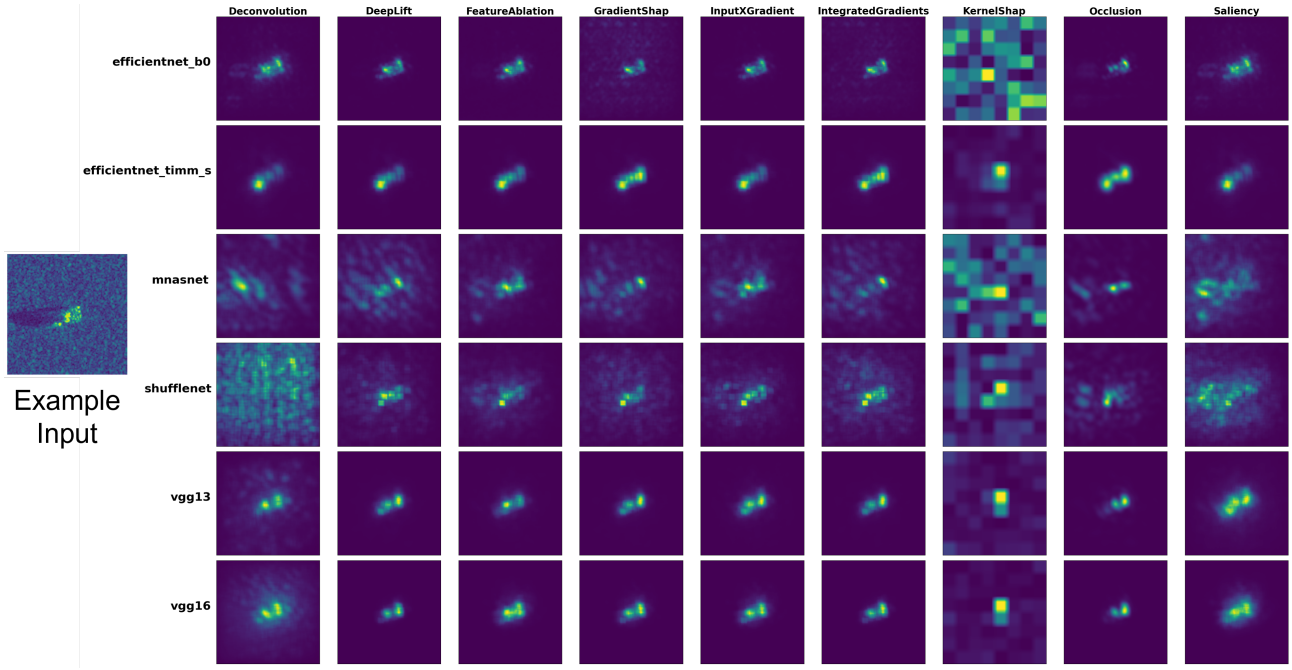

Figure 4. Exemplar attribution maps for the nine attribution methods we introduced, applied across six different neural networks. While the differences may be minor between some methods (columns) or models (rows), one can also readily visualize there are differences in these outputs. And furthermore, this is one example (2s1 tank) out of 1345 SAR chips in the SAMPLE dataset.

more localized. Additionally, Figure 4 seems to indicate that occlusion attribution maps are far less noisy and localize specifically to the target. In contrast, KernelSHAP and Deconvolutional attribution maps tend to be quite noisy and therefore may be harder to interpret. KernelSHAP, as it is applied here, produces noisy images as a result of partitioning images into super pixels in order to train a linear model to calculate influence. The relationship between pixel resolution and the size of the objects a classifier is identifying impact how precisely this partitioning occurs. In the SAR ATR scenario here, the resulting partitions of the attribution maps are squares spanning both portions of the target but also nearby pixels such as the background. Consequently, the attribution maps are less localized on the target due to this processing technique. Conversely, the reason that other perturbation techniques such as Occlussion and Feature Ablation appear sharper in comparison to KernelSHAP is due to the fact that one pixel in each image represents the basic subsection one is applying perturbations to rather than the large superpixels in KernelSHAP.

Visual inspection of the attribution maps for additional input examples and across different target classes yields similar general trends. This includes observations such as larger models are producing cleaner attribution maps than smaller models (e.g. VGG16 compared with MnasNet). Furthermore, the attribution maps seem to indicate that specific regions of the target are being used more than other regions. An example of this would be shown in Figure 5 with the M35 target in which a specific high intense dot is shown in all attribution maps. This part seems to correspond to the tip of the vehicle. However, there are many limitations to anecdotal, visual inspection, and these limitations motivate the application of quantitative analysis to attribution maps in the next section.

4.2 Attribution Evaluation

We applied each of the 4 attribution evaluation metrics to all 9 of the attribution map methods across all 6 neural network models over the 400 SAR chip image samples. Figure 6 shows the results of this analysis, where rather than producing 54 attribution maps for every inference (1 from each of the 6 neural networks across 9 different attribution methods), the attribution evaluation metrics assess the method's performance, based upon their

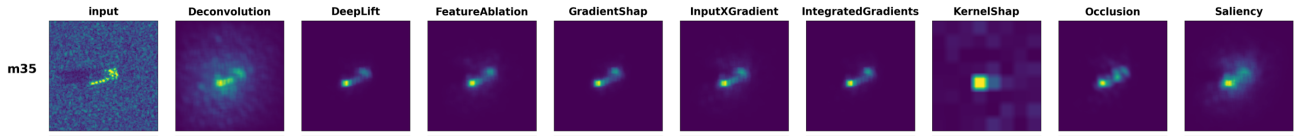

Figure 5. Attribution maps produced by VGG16 neural network for the M35 target class. The M35 is a 2.5 ton cargo carrier. As shown, a common pattern across all nine attribution map methods is the bright attribution at the tip of the vehicle.

respective property, providing a quantitative output. Visually one can see that the notion of what is considered to be a good attribution map varies across the different models (variability in the bar charts in a given row), as well as across the different attribution metrics (rows).

For example, examining Faithfulness via Faithfulness Correlation (second row), the MnasNet (third from left, green bar) is a lower quality attribution map for the majority of the methods. However, for Input×Gradient, this ordering significantly changes and MnasNet is assessed to be much more faithful.

We also use this to highlight the importance of a broad approach. If only looking at the Deconvolution method, the appearance is that only the EfficientNets are Faithful. In contrast, when using FeatureAblation, all the models appear to be faithful. Given this variability and sensitivity, we recommend practitioners to adopt a broad methodology to be characterize their neural network models.

Additionally, we can observe different takeaways by considering the implications of different metrics across different rows. The Saliency method shows low Faithfulness across all models, however, conversely it shows the desired relatively high Complexity values for all models.

4.3 Localization Evaluation

Further quantifying the attribution results, we perform localization analysis to offer interpretability. Similar to the attribution evaluation metrics, localization and other interpretability approaches provide a quantitative result rather than another output for every inference. In particular we were interested in demonstrating that localization metrics can be a very important tool when making more global claims about a neural network from the local explanations of attribution maps. The box plots shown in Figure 7 show a comparison between two models, namely MnasNet and VGG16 across three localization metrics. Since each of the localization metrics in Figure 7 requires that the user provide a region of interest, we chose two regions to study resulting in each metric being represented twice in each of the subplots. As mentioned in Section 3.5 our goal was to study the degree to which the different attribution maps were centered around the target (t) and the shadow (s). The first thing to note in the figure is that attribution maps across VGG16 tend to localize to the target of the image rather than the shadow. This conclusion is derived from the fact that the values for the localization metrics are higher for the target pixels than for the shadow pixels across all three metrics. In contrast, the values of the localization metrics for the attribution maps generated from MnasNet do not make as clear of a distinction between the importance of the target and that of the shadow. From these results, it would be reasonable to conclude that relative to MnasNET, the VGG16 model tends to focus more heavily on the use of the target in the image in order to do classification. These types of conclusions can be easily visualized when using localization metrics but would be difficult to attain by a purely qualitative analysis of the attribution maps.

We note that because interpretability provides analysis on explainability measures, the utility of the analysis depends upon the quality of the explainability data. For example, in the SAR ATR domain, care needs to be taken when making claims about how a neural network is using shadows for classification. This is due to the fact that many of the attribution map methods applied use a black image as a baseline in their calculation of influence which means that the shadow would be given no influence over the classification. With this in mind, attribution maps that would be better suited to evaluate the influence of a shadow would be Deconvolution, Saliency, and Input×Gradient.

Additionally, these results demonstrate the importance of using multiple types of localization metrics to conduct a general analysis. First off, as shown in Figure 7, there is significant variation of scores across different

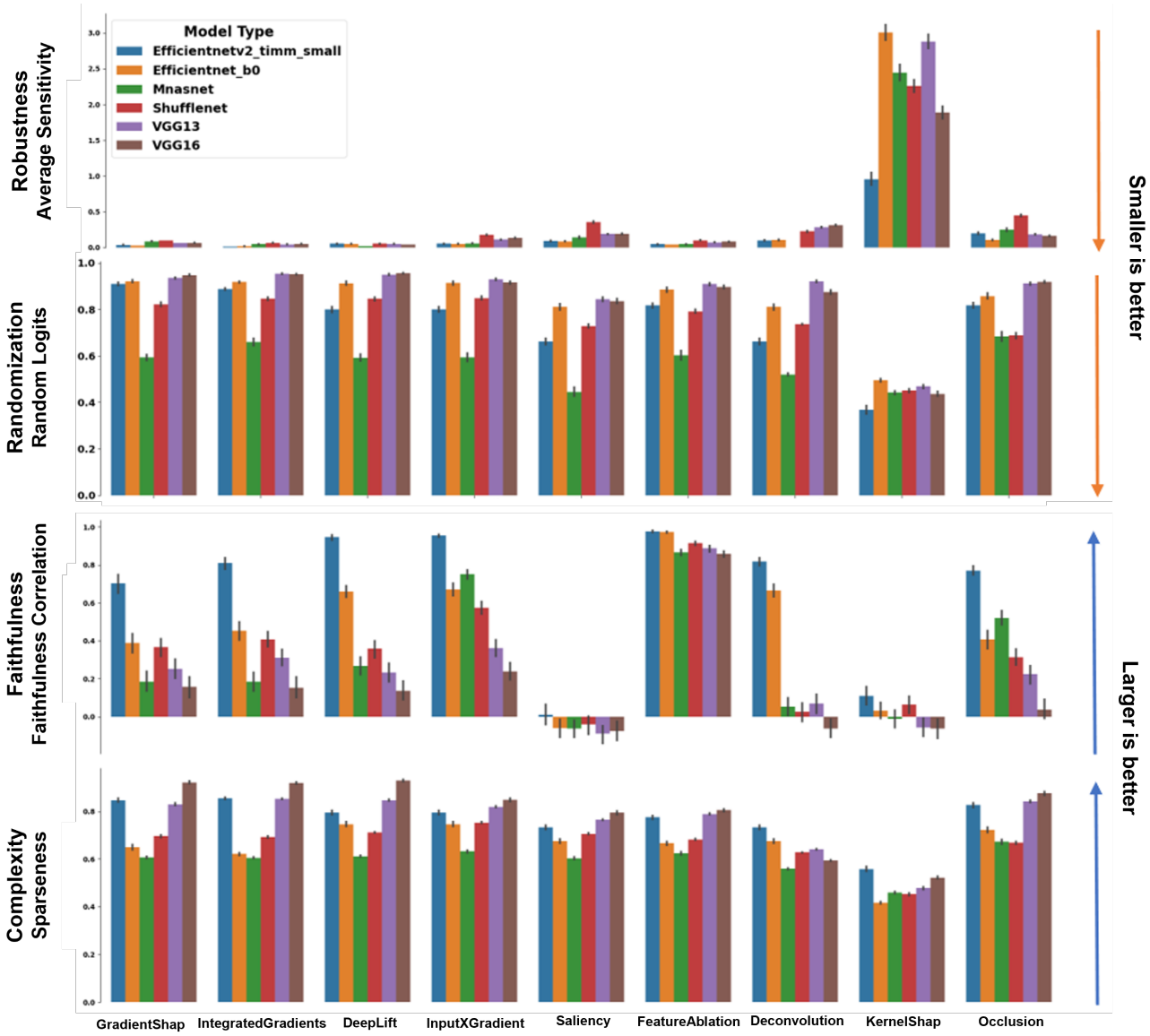

Figure 6. Results of attribution evaluation metrics applied across all nine attribution map methods for all six neural networks. Trends and variability offer insight into how these methods and metrics are performing in relationship to the models employed as well as the underlying dataset.

metrics. While, for example, attribution maps for VGG16 obtain almost perfect scores for the AUC-T localization metric, these attribution maps generally had lower NSS localization scores. These types of variations are important as each metric has its own way of defining what it means for an attribution map to highlight a given portion of an image and therefore conveys different information. The broader perspective of trends across multiple localization metrics can accordingly offer greater evidence towards interpreting the model's operation.

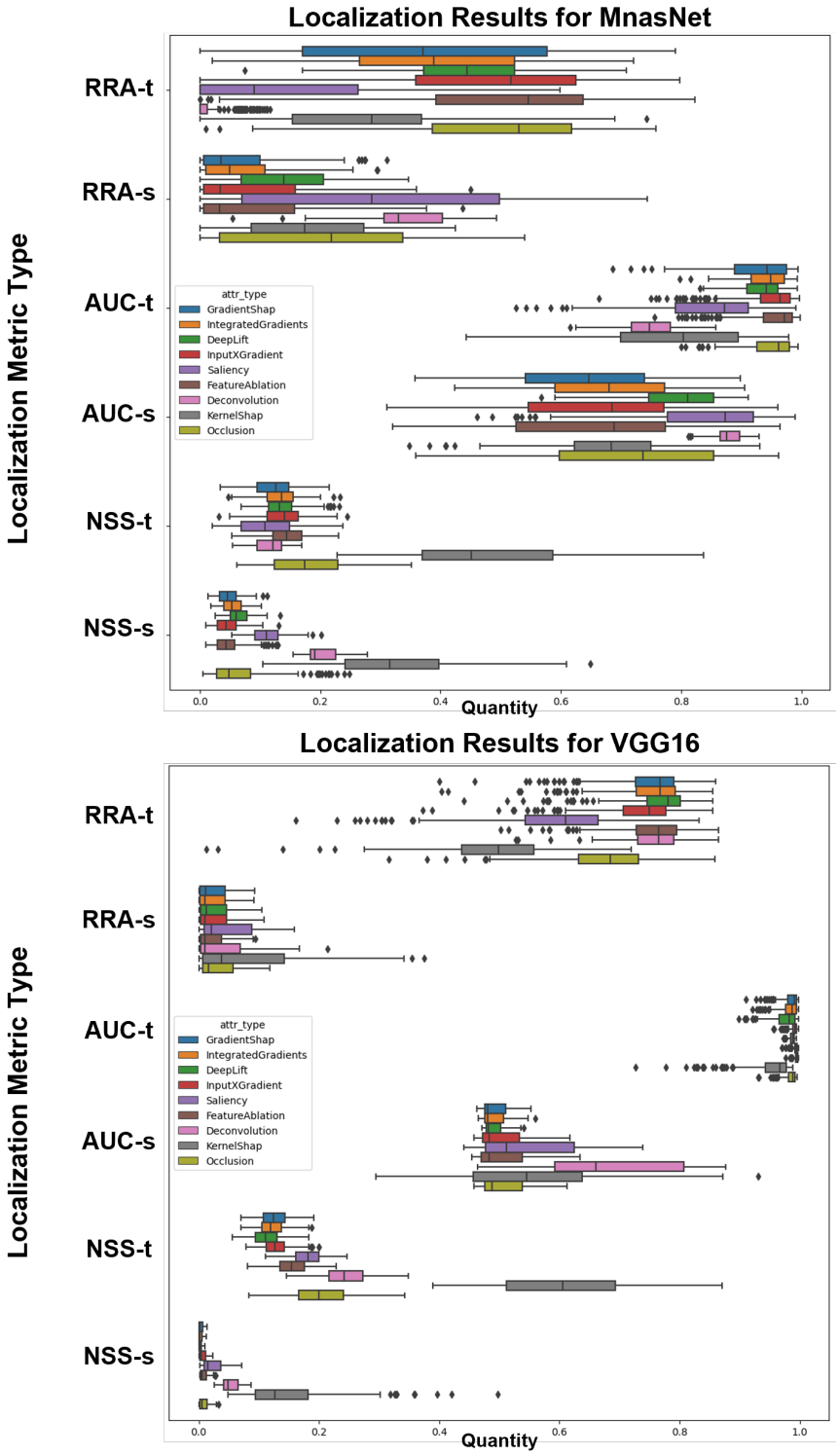

Figure 7. Comparison of localization metrics to interpret whether neural networks are using the target or shadow of the SAR chip to perform classification. Presented for comparison here are the largest neural network examined, VGG16 (bottom), and one of the smaller networks MnasNet (top). While in both cases the target is used more predominantly across all three metrics for both models, VGG16 more predominantly uses the target.

Variation across models and attribution map types can be more explicitly seen in Figure 8. In this figure, each cell of the clustergram represents the average information gain statistic for a given attribution map method (specified in rows) and a given model with the region of interest set to either the shadow or target (specified in columns). One interesting characteristic of this graph is that for all of these methods and models, columns of the cluster gram (with exception of shadow_mnasnet), are split into two distinct clusters where one is a cluster of columns associated with the target and the other is a cluster of columns associated with the shadow. This indicates that attribution maps across most of these models tend to have high localization of the target and lower localization across the shadow. Second, similar meaningful clustering occurs with the attribution map methods where methods of similar type (gradient based, occlusion based, path attribution based) are clustered together. The clustering of these methods can indicate that attribution map methods of similar types tend to produce attribution maps that are similar to each other with this set of models.

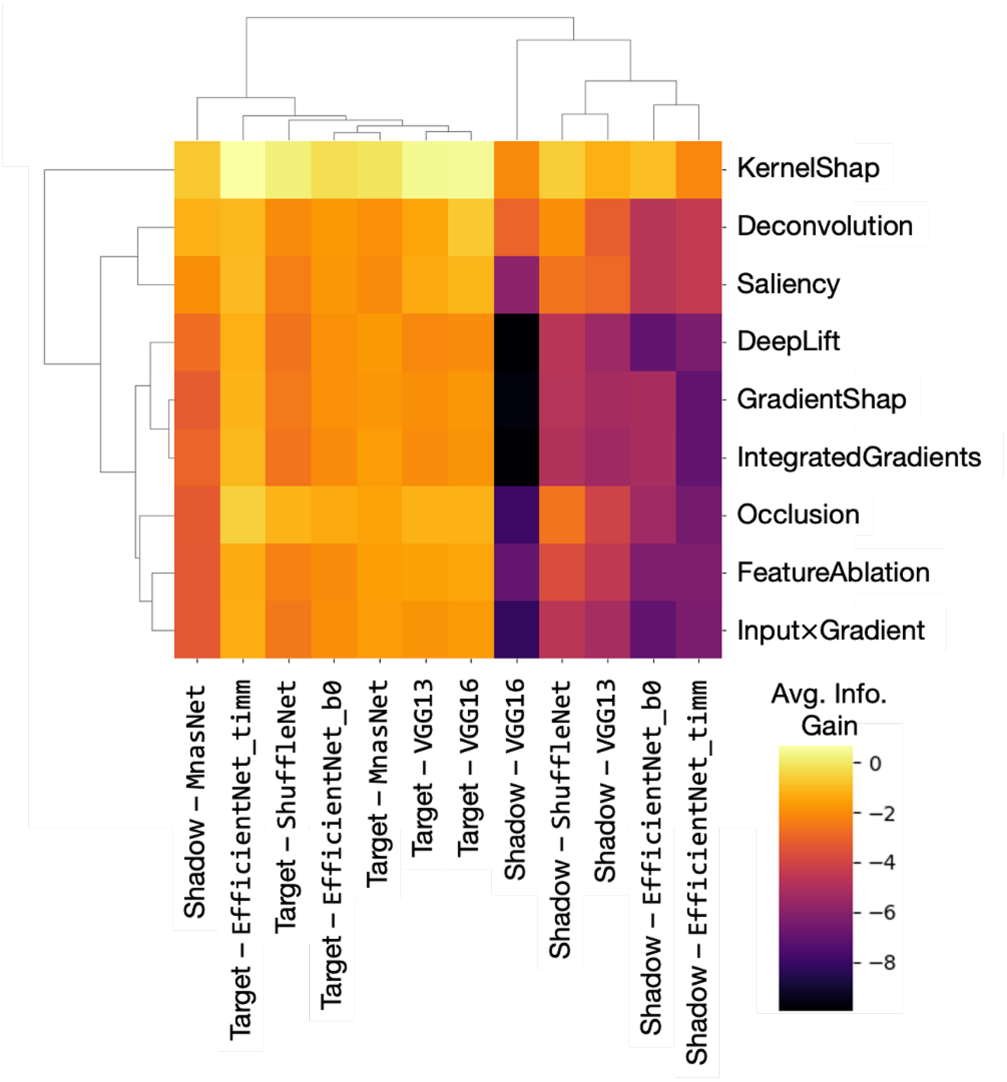

Figure 8. A clustergram for Information Gain metric results across models (columns) and attribution map types (rows) The brighter intensity left half corresponds to the target impacting model performance more than the shadow (with the exception of the MnasNet-shadow model).

5. CONCLUSIONS

Throughout this paper, we have provided an introduction to basic explainability and interpretability methods in the context of ATR of SAR images. To demonstrate the power of some methods, basic attribution map explainability methods as well as a handful of interpretability methods are applied on six different trained models of various size and computational complexity. Our results demonstrate two important observations. First, there is a large amount of variation between attribution map method outputs visually as well as quantifiably (in terms of attribution evaluation metrics). These results show that it is not clear that one attribution map method is considered the best choice for any model trained for SAR ATR. Thus, attribution evaluation metrics and multiple attribution map types may be necessary to gain a seemingly trustworthy understanding of a model's operation. Furthermore, localization metrics show promise in providing a way to make global claims about how a model works from local explanations. While qualitative analysis only provides limited and sometimes overly complicated information, localization metrics can focus information from multiple attribution maps. In doing so they can try to answer potentially interesting questions such as whether a model is influenced more by the shadow or target of an image. The techniques presented here are not an exhaustive representation of the explainability and interpretability mechanisms for machine learning decision making. Furthermore, many of the methods have parameters which need to be investigated further. Nevertheless, as we have shown, explainability and interpretability techniques can further the understanding of machine learning algorithms in critical decision making tasks like automatic target recognition.

ACKNOWLEDGMENTS

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC (NTESS), a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration (DOE/NNSA) under contract DE-NA0003525. This written work is authored by an employee of NTESS. The employee, not NTESS, owns the right, title and interest in and to the written work and is responsible for its contents. Any subjective views or opinions that might be expressed in the written work do not necessarily represent the views of the U.S. Government. The publisher acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this written work or allow others to do so, for U.S. Government purposes. The DOE will provide public access to results of federally sponsored research in accordance with the DOE Public Access Plan.

REFERENCES

[1] R. Melzer, W. M. Severa, and C. M. Vineyard, "Exploring sar atr with neural networks: going beyond accuracy," in *Automatic Target Recognition XXXII*, **12096**, pp. 125–144, SPIE, 2022.

[2] R. Melzer, W. M. Severa, M. Plagge, and C. M. Vineyard, "Exploring characteristics of neural network architecture computation for enabling SAR ATR," in *Automatic Target Recognition XXXI*, R. I. Hammoud, T. L. Overman, and A. Mahalanobis, eds., **11729**, p. 1172909, International Society for Optics and Photonics, SPIE, 2021.

[3] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl, "Bigearthnet: A large-scale benchmark archive for remote sensing image understanding," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, July 2019.

[4] N. Inkawhich, M. J. Inkawhich, E. K. Davis, U. K. Majumder, E. Tripp, C. Capraro, and Y. Chen, "Bridging a gap in sar-atr: Training on fully synthetic and testing on measured data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **14**, pp. 2942–2955, 2021.

[5] C. Leong, T. Rovito, O. Mendoza-Schrock, C. Menart, J. Bowser, L. Moore, S. Scarborough, M. Minardi, and D. Hascher, "Unified coincident optical and radar for recognition (unicorn) 2008 dataset," 2019.

[6] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, *et al.*, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," in *International conference on machine learning*, pp. 2668–2677, PMLR, 2018.

[7] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill* **2**(11), p. e7, 2017.

[8] C. Molnar, *Interpretable machine learning*, Lulu. com, 2020.
[9] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," 2019.
[10] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature machine intelligence* **1**(5), pp. 206–215, 2019.
[11] J. Bae, N. Ng, A. Lo, M. Ghassemi, and R. B. Grosse, "If influence functions are the answer, then what is the question?," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds., **35**, pp. 17953–17967, Curran Associates, Inc., 2022.
[12] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," 2017.
[13] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017.
[14] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833, Springer, 2014.
[15] Z. Gandomkar, P. L. Khong, A. Punch, and S. Lewis, "Using occlusion-based saliency maps to explain an artificial intelligence tool in lung cancer screening: Agreement between radiologists, labels, and visual prompts," *Journal of Digital Imaging* **35**(5), pp. 1164–1175, 2022.
[16] J. Bento, P. Saleiro, A. F. Cruz, M. A. Figueiredo, and P. Bizarro, "TimeSHAP: Explaining recurrent models through sequence perturbations," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, ACM, aug 2021.
[17] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *Advances in neural information processing systems* **31**, 2018.
[18] A. Hedström, L. Weber, D. Krakowczyk, D. Bareeva, F. Motzkus, W. Samek, S. Lapuschkin, and M. M. M. Höhne, "Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond," *Journal of Machine Learning Research* **24**(34), pp. 1–11, 2023.
[19] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson, "Captum: A unified and generic model interpretability library for pytorch," 2020.
[20] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?," *IEEE transactions on pattern analysis and machine intelligence* **41**(3), pp. 740–757, 2018.
[21] R. J. Peters, A. Iyer, L. Itti, and C. Koch, "Components of bottom-up gaze allocation in natural images," *Vision research* **45**(18), pp. 2397–2416, 2005.
[22] B. P. Lewis, T. Scarnati, E. Sudkamp, J. W. Nehrbass, S. Rosencrantz, and E. G. Zelnio, "A sar dataset for atr development: the synthetic and measured paired labeled experiment (sample)," in *Defense + Commercial Sensing*, 2019.
[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556* , 2014.
[24] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.
[25] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International conference on machine learning*, pp. 10096–10106, PMLR, 2021.
[26] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2820–2828, 2019.
[27] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, 2018.
[28] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," 2013.
[29] P. Sturmfels, S. Lundberg, and S.-I. Lee, "Visualizing the impact of feature attribution baselines," *Distill* **5**(1), p. e22, 2020.

[30] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Towards better understanding of gradient-based attribution methods for deep neural networks," 2018.

[31] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," 2019.

[32] 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*, pp. 1135–1144, 2016.

[33] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2014.

[34] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," 2017.

[35] U. Bhatt, A. Weller, and J. M. F. Moura, "Evaluating and aggregating feature-based model explanations," 2020.

[36] C.-K. Yeh, C.-Y. Hsieh, A. S. Suggala, D. I. Inouye, and P. Ravikumar, "On the (in)fidelity and sensitivity for explanations," 2019.

[37] P. Chalasani, J. Chen, A. R. Chowdhury, S. Jha, and X. Wu, "Concise explanations of neural networks using adversarial training," 2020.

[38] L. Sixt, M. Granz, and T. Landgraf, "When explanations lie: Why many modified bp attributions fail," 2020.

[39] F. Gao, J. You, J. Wang, J. Sun, E. Yang, and H. Zhou, "A novel target detection method for sar images based on shadow proposal and saliency analysis," *Neurocomputing* **267**, pp. 220–231, 2017.

[40] S. Papson and R. Narayanan, "Modeling of target shadows for sar image classification," in *35th IEEE Applied Imagery and Pattern Recognition Workshop (AIPR'06)*, pp. 3–3, 2006.

[41] S. Huang, W. Huang, and T. Zhang, "A new sar image segmentation algorithm for the detection of target and shadow regions," *Scientific reports* **6**(1), p. 38596, 2016.

[42] L. Arras, A. Osman, and W. Samek, "Clevr-xai: A benchmark dataset for the ground truth evaluation of neural network explanations," *Information Fusion* **81**, pp. 14–40, 2022.