Improving transparency of deep neural inference process

Hiroshi Kuwajima1,2 · Masayuki Tanaka1,3 · Masatoshi Okutomi1

Received: 6 September 2018 / Accepted: 26 February 2019 / Published online: 12 April 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract
Deep learning techniques are rapidly advanced recently and becoming a necessity component for widespread systems. However, the inference process of deep learning is black box and is not very suitable to safety-critical systems which must exhibit high transparency. In this paper, to address this black-box limitation, we develop a simple analysis method which consists of (1) structural feature analysis: lists of the features contributing to inference process, (2) linguistic feature analysis: lists of the natural language labels describing the visual attributes for each feature contributing to inference process, and (3) consistency analysis: measuring consistency among input data, inference (label), and the result of our structural and linguistic feature analysis. Our analysis is simplified to reflect the actual inference process for high transparency, whereas it does not include any additional black-box mechanisms such as LSTM for highly human readable results. We conduct experiments and discuss the results of our analysis qualitatively and quantitatively and come to believe that our work improves the transparency of neural networks. Evaluated through 12,800 human tasks, 75% workers answer that input data and result of our feature analysis are consistent, and 70% workers answer that inference (label) and result of our feature analysis are consistent. In addition to the evaluation of the proposed analysis, we find that our analysis also provides suggestions, or possible next actions such as expanding neural network complexity or collecting training data to improve a neural network.

Keywords Transparency · Deep neural network · Black box · Explainable AI · Visualization · Visual attribute

1 Introduction

Machine learning techniques such as deep neural networks has led to the widespread application of systems that assign advanced environmental perception and decision-making to computer logics learned from big data, instead of manually built rule-based logics [13,25,30,33,38,41]. Deep learning especially achieves unprecedented performance on several tasks. For example, in the visual object recognition task outperformed humans [16].

Machine learning models are becoming indispensable components even in systems that require safety-critical environmental perception and decision-making, such as automated driving systems [4]. To build high credibility for machine learning models, both high performance and transparency are important. In particular, safety-critical systems must exhibit transparency [22]. However, inference processes of machine learning models such as neural networks are considered as black boxes. In this paper, a black box refers to a situation, where, although feature activation can be observed, the actual phenomenon cannot be understood. In other words, machine learning models show high performance but low transparency. Thus, it is difficult for black-box deep learning to be applied to safety-critical systems such as automated driving in which the results of deep learning models can directly cause hazard [19].

Explainable AI (XAI) is a related research area which is focused and rapidly advanced recently [6,9]. There are studies in XAI that inference networks give human-understandable explanations, as well as inference (label). For example, image caption generation and visual explanation are problems to provide highly human-understandable
Caption generation is a verbalization method, which describes the objects and the circumstances happening in the input image by natural language sentences [39,40]. Visual explanations are generated by black-box explaining models such as LSTM, to explain rationales for classification decisions [17]. They generate highly human readable explanation, however, by using mechanisms which do not reflect the actual inference process, because explanation generation and classification are done by different neural networks possibly sharing features, inference results (labels), etc. Even sharing features, explanation generation is done by black-box models (neural networks), and we cannot know they reflect the actual inference process. Inference networks which generate explanations have high performance but low transparency.

In this paper, to address the black-box property of deep learning, we develop a simple analysis method which improves the transparency of inference processes of convolutional neural networks, hereinafter referred to as CNN [11, 20], as an example of deep learning models. We assume three types of analysis for inference process: (1) structural feature analysis, (2) linguistic feature analysis, and (3) consistency analysis. Results of structural feature analysis are lists of the features contributing to inference process. The feature numbers are not human readable, but are useful when systems programmatically manage the inference process at testing time. Results of linguistic feature analysis are lists of the natural language labels describing the visual attributes for each feature obtained through the structural feature analysis. It is useful for humans to understand inference processes. Figure 1 is an example result of our feature analysis. The left and right columns in Fig. 1c are structural feature analysis and linguistic feature analysis, respectively. Consistency analysis is to measure the consistency among input data, inference (label), and result of feature analysis. It is useful for discussion, such as identifying the cause of incorrect inference (label) and possible next actions to fix problems, etc. Figure 2 shows the concept of consistency analysis. To show the usefulness of our proposed method, we conduct experiments including human evaluation, and have corresponding discussion on the experimental results.

This paper is an extended version of our previous workshop paper presented in Transparent and interpretable Machine Learning in Safety Critical Environments, NIPS2017 Workshop [21].

2 Related works

DARPA started Explainable Artificial Intelligence program in 2017 [15]. It defines three approaches: deep explanation, interpretable models, and model induction. The first and second are before-training approaches which design explainable features and explainable causal models in advance to training. The third is an after-training approach which automatically derives new explainable models after training.

We focus on the after-training approaches in this paper, because they can be applied to wide variety of existing models without changing them. LIME [32] is one of the most well-known after-training explanation methods. It derives a simple interpretable model to explain the behavior of the original model around a given data point. It prepares interpretable simplified data representations, e.g., for an image, contiguous similar pixels and RGB features are converted into a super-pixel and a binary variable indicating “presence” (original RGB value) or “absence” (average value of neighboring pixels). Then, a simplified interpretable model, such as a linear model and a decision tree, is trained to fit the neighbors around the given data point while regularizing its complexity. They can analyze the simplified model for explanation, e.g., visualize its interpretable decision boundary and feature importance, etc. It is a model-agnostic method, because the simplified model is trained without the information inside the original model. However, it is an indirect method where
they can analyze the newly trained simplified model, not the original model. Our proposed method is an after-training approach which provides direct analysis on the actual activation observed on the original model.

Visualization of deep neural networks is an active study area in the after-training approaches [14, 18, 28]. Earlier studies identify attention (focus) areas of input data in saliency maps, receptive fields, or heat maps [3, 27, 35, 37]. Deconvolution [42] is one of the earliest methods for visualization of deep neural networks. It back-propagates the activations in a feature space of a CNN to the input space through a deconvolutional neural network. Saliency map [37] is a method similar to deconvolution, which generates the input image which maximizes the desired class score. They can systematically occlude input images to identify receptive fields [43], because occluding not backgrounds but foreground objects causes large discrepancy in neural network accuracy. There are several gradient-based methods for visualization. Grad-CAM (Gradient-weighted Class Activation Mapping) [34] uses the gradient at the last convolved feature to generate heat maps. It can generate heat maps with respect to each class by taking the importance of feature maps for the target class into account. More recent visualization techniques DeepLIFT (Deep Learning Important FeaTures) [36], LRP (Layer-wise Relevance Propagation) [1], and the aforementioned LIME are identified to be included in the new class of additive feature importance methods through a unified framework for interpreting predictions, SHAP (SHapley Additive exPlanations) [24]. These visualization methods can indicate the areas in the input data which the model is looking at [5] during testing time. Attention areas of an input image revealed by visualization methods are the very beginning part of CNN inference process and are useful information. In this paper, we would like to provide analysis not only for input data, but also for inference process of neural networks. We exploit receptive fields as side information in addition to intermediate feature analysis, to indicate the locations of the visual attributes in input data. Although there are aforementioned several sophisticated methods, we used a simple backpropagation-based method to generated receptive fields, because the side information is not the main focus in this paper.

There is another type of works focusing on the visual attributes and intermediate features, i.e., activation of neural network nodes. One of past works analyzed the visual attributes for each node, and it was revealed that low-level attributes, such as black, brown, and furry are associated with neural network nodes [10]. Another work interprets receptive fields as with visual attributes of neural networks, and quantified the interpretability by using the number of human interpretable visual semantic concepts learned at each hidden layer [2]. Among visualization techniques, in this paper, we use visual attributes for our transparent analysis.

Pointing and Justification-based Explanation (PJ-X) is one of the latest explanation methods in XAI [29]. It can provide highly human-understandable explanation, by attention areas of input data space as introspective explanations (true explanation) and justification explanations at the same time. The former provides explanations of the input space, but does not provide analysis for the inference processes. The latter does not address the black-box property of target models, because it uses another black-box method LSTM to generate the explanation. PJ-X does not analyze the relationship between the inference results and features of neural networks, and introducing an additional black-box model for explanation cannot address transparency. Therefore, the purpose of PJ-X is not an analysis for improving transparency of deep neural inference process.

### 3 Observation of feature contribution

We observed conv5 feature of CaffeNet [8, 20] on selected ImageNet training data, to understand the behavior of features. Although ImageNet has approximately 1300 training images per class, for simplicity, we selected 100 examples for each class, with top-100 softmax probability on the ground truth classes.

We first make a natural assumption that inference (label) is based not on inactive features, but on highly activated features, and derived the following assumption.

![Fig. 2 Consistency analysis concept](image)
Assumption 1. Features highly activated in the inference process have contributions to inference (label).

This assumption applies especially for ReLU, which CaffeNet uses as activation functions, because ReLU is a half-linear positive monotonic function.

3.1 Magnitude of feature activation

Then we look into activation on each feature map in \texttt{conv5}, and found that the magnitude of activation changes for different features. Therefore, definition of high activation varies depending on features. Figure 3 shows the histograms of activation on example feature maps 94 and 22, which have the smallest and largest mean values, respectively. The modes of activation magnitude are different each other. These distributions are not Gaussian, because negative values are canceled by ReLU activation function. It is clear that the distributions of activation on feature maps 94 and 22 are different. By this analysis, we derived the following assumption.

Assumption 2. Activation in different features have different dynamic ranges.

3.2 Features and visual attributes

Figure 4 includes three visual attributes: furry, rubber tires, and fine cell patterns, but only two feature maps 226 and 230. These features share the visual attribute furry, while at the same time, have the other visual attributes different each other. This observation implies the following assumption.

Assumption 3. Visual attributes and features are in a many-to-many relationship.

4 Proposed analysis

In this section, we propose a transparent analysis method to improve the transparency of deep neural inference processes based on assumptions. We carry out both training time and testing time feature analysis to obtain three types of features, as described in Sect. 4.1. We performed manual feature annotation to associate features with visual attributes, as described in Sect. 4.2. Then, three types of consistency ratios among input image, result of our proposed feature analysis, and inference (label) are measured through human tasks, as described in Sect. 4.3.

4.1 Structural feature analysis

We propose three concepts of features: (1) activated feature, (2) class frequent feature, and (3) inference feature as depicted in Figs. 5, 6, and 7, respectively. Activated feature and inference feature are defined for each inference, whereas class frequent feature is defined for each class.

To analyze inference process, we focus on the activation of an intermediate feature called \texttt{conv5} which is the final
convolved feature in CaffeNet. It is reported that \texttt{conv5} of AlexNet, which is also the final convolution layer, learns high-level visual concepts such as objects and parts, and they are interpretable for humans [2]. Let \( x \) and \( y \) be input and the output, i.e., inference (label), of CaffeNet.

**Activated feature** \( a \) in Fig. 5 is the binarized feature vector generated from \texttt{conv5}. Activated feature \( a \) is a binary feature vector; however, CNN feature maps have spatial dimension. We decided to ignore the location of activation for simplicity, and applied global max pooling [23] to contract \texttt{conv5}, which is originally \( 13 \times 13 \times 256 \) tensor, into 256-dimensional feature vector \( z \), as it is the simplest way to obtain a vector from a tensor. Therefore, we consider a feature map, with spatial feature elements, as a single feature. An element of the vector \( a \) is one if the associated feature, i.e., a feature map in \texttt{conv5}, is activated. Based on Assumption 2, in order to determine whether the feature is activated or not, it is necessary to use statistical information such as mean, variance, or higher moment to capture the differences among features. In this paper, we decided to use mean normalization and thresholding. We compute a mean-normalized feature vector \( \hat{z} \) from a feature vector \( z \), as each element of \( z \) has varying dynamic range, and normalization makes them comparable each other. Thresholding \( \hat{z} \) at a scalar value \( \gamma \) gives a binarized feature vector \( a \) corresponding to \( x \).

We clarify the definition of activated feature by mathematical formal presentation. The activated feature is calculated for each input data \( x \). First, we apply the global max pooling for the output of \texttt{conv5} layer

\[
\begin{align*}
z_c^e &= \max_{h,w} F_{c,h,w}, \\
F_{c,h,w} &= z_c^i,
\end{align*}
\]

where \( F_{c,h,w} \) is the output of \texttt{conv5} layer of the CaffeNet, superscripts of \( c, h, w \) specify the position (channel, height, width) of feature, and \( z_c^i \) is the element of the global max pooling results. Then, \( c \)th element of the binary activated feature \( a \) can be expressed as

\[
a_c^e = \begin{cases} 1 & (\hat{z}_c^e / \mu_c^e - \gamma) \\ 0 & (\hat{z}_c^e / \mu_c^e - \gamma) < 0 \end{cases},
\]

where \( a_c^e \) is \( c \)th element of the binary activated feature for the input data \( x \), \( \gamma \) is a parameter, \( \mu_c^e \) is the mean of \( z_c^i \) over training data \( x_i^{\text{train}} \), and \( U(\xi) \) is a unit step function. The mean \( \mu_c^e \) can be calculated by

\[
\mu_c^e = \frac{1}{N} \sum_{i=1}^{N} z_c^i,
\]

where \( z_c^i \) is the global max pooling result for the \( i \)th training data \( x_i^{\text{train}} \), \( N \) is the number of data. The unit step function \( U(\xi) \) is defined by

\[
U(\xi) = \begin{cases} 1 & (\xi \geq 0) \\ 0 & (\xi < 0) \end{cases}.
\]

**Class frequent feature** \( q \) in Fig. 6 is binary vectors indicating the frequently activated features for each class. We hypothe-
size that each class has a different frequent activation pattern which is obtained by the following procedure. A class frequent feature is built with the training dataset. We assume the training dataset is annotated. In other words, every input data belong to a certain class. Figure 6 shows how to compute the class frequent feature for an example class, dog. The training data $x_{i}^{\text{train}}$ of the dog class is binarized into $a_{i}^{\text{train}}$, and their summation over $i$, i.e., histogram $g_{\text{dog}}$, counts how many times each feature is activated for the dog class in the training data. After summation, we select the top-$k$ frequent features which consist the class frequent features for the dog class, where $k = 3$ in the case of Fig. 6. Class frequent features are computed for each class at the training time and stored in a lookup table $q_{m}$ for $m$th class. They will be used in the test- 

ting process is based on these key parts, and we derived the following assumption.

- **Assumption 4.** Features frequently activated for the class of inference (label) have high contributions to inference (label).

Inference feature $e_{\text{test}}$ in Fig. 7 is the overlap between the activated feature and the class frequent feature. An inference feature is binary vector calculated with each test data. Let the inference (label) for a test data $x_{\text{test}}$ be $\hat{y}_{\text{test}}$. We define inference feature $e_{\text{test}}$ as

$$e_{\text{test}} = a_{\text{test}} \otimes q_{\hat{y}_{\text{test}}}$$

where $a_{\text{test}}$ is the activated feature associated with the test data $x_{\text{test}}$. $q_{\hat{y}_{\text{test}}}$ is the class frequent feature for $\hat{y}_{\text{test}}$ class, and $\otimes$ represents the element-wise product.
Based on Assumption 1 and Assumption 4, features contributing to inference process should be a part of both activated feature and class frequent feature. The dotted box in Fig. 7 is the conventional inference without feature analysis. $ \hat{a}_{test} $ is computed based on $ x_{test} $, whereas the class frequent feature $ g_{y_{test}} $ is just lookup by the inference (label) $ y_{test} $ given by CaffeNet, because the ground truth is unknown in the testing time. $ e_{test} $ is the result of structural feature analysis. The number of positive feature vector elements in an inference feature $ e_{test} $ is generally variable for each inference. Due to the human readability, in an inference feature $ e_{test} $, we show at most top-\( \ell \) feature vector elements with the maximum mean-normalized activation $ \hat{z} $.

### 4.2 Linguistic feature analysis

To generate human readable analysis, we annotate visual attributes for each feature by looking at the input samples on which it is activated in the focused network. Although there are many ways to achieve human readable visual attributes, we decided to conduct human annotation, because it is the most simple method.

**Annotation data** are prepared by using the training data set. At first, we select a subset of training data suitable to feature annotation. And then, for each feature, we sample images so that their inference features (identified by the ground truth labels) include it. We generated receptive fields for the human annotator to understand the part of the image where visual attributes appear, as shown in Fig. 10.

**Annotation process** is an iterative way to annotate the combination of multiple features representing a single visual attribute, and vice versa. In order to annotate this many-to-many relationship based on Assumption 3, starting from free description, feature annotation is repetitively refined. We defined a process which consists of three steps: (1) open annotation, (2) label organization, and (3) closed annotation.

The open annotation is the first step where a human annotator annotates all features by free description. For each feature, a human annotator looks at the images associated with the feature and writes down what he or she can see in the receptive fields. He or she can come up with as many visual attributes as possible and write them down, because of the many-to-many relationship. The number of annotated visual attributes varies. Some features represent few visual attributes; others represent many visual attributes. At the end of the first step, we obtain the list of variable numbers of visual attributes for each feature, where we can have redundant and/or inconsistent visual attributes. Human annotation may include fluctuation at this step, because it is done by his or her intuition at that time which could be shifting as time goes by. Generally annotation is using the fix set of labels, e.g., 1000 labels of ImageNet dataset; however, we need to conduct free description annotation at this step, because there is no available list of visual attributes which the original model captured.

Second in the label organization step, similar visual attributes are integrated, different visual attributes with the same label are divided, new visual attributes are introduced, etc., so that the fluctuated labels of feature annotation are well organized. For example, if the human annotator gave different names “furry” and “fluffy” on the almost similar visual attribute in the step one, they can be integrated into ”furry” in the step two. At the end of the second step, we obtain the list of visual attributes whose redundancy and inconsistency are eliminated. We assume that it is the current best list of all visual attributes which the original model captured. The list is updated through iterating this step two and the following step three.

Then, the human annotator works on the closed annotation to classify features into sets of visual attributes. This step is similar to the first step, except that we use the list of visual attributes defined in the second step. It is allowed that a human annotator selects multiple visual attributes for each feature, because of the many-to-many relationship. At the end of the third step, we obtain the normalized list of variable numbers of visual attributes for each feature, where we do not have redundant and/or inconsistent visual attributes.

Label organization and closed annotation steps are repeated to refine the feature annotation, as depicted in Fig. 11. A human annotator may notice that the whole set of visual attributes are too large or small after seeing the actual lists of visual attributes allocated to features. In that case, he or she can modify the list of all visual attributes by repeating the step two, and subsequently update the alloca-
tion of visual attributes by the step three. One possible simple stopping criterion of iteration is that there is no update in the steps two and three.

4.3 Consistency analysis

To gain further insight, we measure the consistency among input data, inference (label), and result of our proposed feature analysis, i.e., inference feature. This measurement is for discussion, checking whether our analysis method or the target neural network are incorrect when we get incorrect analysis, identifying possible next actions to fix problems, etc.

Precision and recall [31] are commonly used measures of relevance for binary or multi-class classification problems [12] where ground truth labels or true conditions are known. However, feature analysis does not have ground truth labels, and therefore, we cannot calculate precision and recall for our problem. Thus, we propose physical consistency ratio (PCR) and logical consistency ratio (LCR), which are the consistency between inference feature and input data, and consistency between inference feature and inference (label), respectively (Fig. 2). These two ratios are measured through human tasks. Specification of human tasks in this paper will be described in 5. In addition to two measures for consistency, we use the softmax probability corresponding to the class of inference (label), i.e., the maximum softmax probability, as inference consistency ratio (ICR), the consistency between input data and inference (label). All the ratios are in the range of 0.0 to 1.0.

5 Experiments

In this section, we conducted experiments to test our proposed analysis method. We try to analyze the inference processes of the publicly available CaffeNet with the weights pre-trained on ImageNet. These feature vectors were binarized by the method introduced above, with the binarization threshold $\gamma = 2$. We chose $k = 5$ the number of feature vector elements in a class frequent feature, and $\ell = 3$ the maximum number of feature vector elements in an inference feature. Receptive fields for each feature vector elements in inference features are accompanied with the result of feature analysis as informative clue for human feature annotation, and side information to support the analysis.

As with the feature analysis, selected 100 training images per class are used for computing mean values for each feature map in conv5, class frequent features, and annotating visual attributes by human. On the other hand, we reduced the 1000 object categories of ImageNet to 32 for testing, because it is difficult for human to distinguish 1000 categories and understand the corresponding analysis precisely. The 32 classes are a subset of ImageNet 1000 classes, which are programmatically selected according to the WordNet [26] hierarchy, such that each new class has approximately the same number of WordNet synsets.

Human evaluation was done on Amazon Mechanical Turk. For each input image, we made two questions for the physical consistency and logical consistency on our feature analysis, shown in Figs. 12 and 13. The first question was asked without showing inference (label), and the second one was asked without showing the input image. The list of response alternatives shown to workers were strongly agree, agree, disagree, and strongly disagree.

After obtaining the results from workers, we merged the former two and the latter two into agree and disagree, respectively. The results of the questions are used to evaluate physical consistency ratio and logical consistency ratio, respectively. For an input image, each question is redundantly
Fig. 12 Q1: Question to measure physical consistency. We ask whether the inference features are relevant to the whole or parts of the input image, without using the term inference feature. We do not show the inference (label) in this question.

- It has animal legs.
- It has human legs, animal legs or beige.
- It has animals, furs or brown.

Fig. 13 Q2: Question to measure logical consistency. We ask that if an object satisfies the inference feature, then it is an object in the class of inference (label), without using the term inference feature. We do not show the image in this question.

- It has animal legs.
- It has human legs, animal legs or beige.
- It has animals, furs or brown.

Where \( \#(X) \) represents the number of responses \( X \). To evaluate these ratios, we conducted 12,800 human tasks for total (Table 1). We also recorded the softmax probability of the class of inference (label) as ICR.

Figure 14a, b shows the joint discrete probability distribution between physical consistency ratio and logical consistency ratio on correct inference and incorrect inference, respectively.

When inference is incorrect, the peak is low although it is located in \((0.8, 1.0)\), and the both physical consistency ratio and logical consistency ratio spread throughout from low consistency to high consistency. On the other hand, both ratios tend to be high when inference is correct. The distribution on correct inference is clearly high contrast compared to that on incorrect inference. Therefore, our method provides better analysis for correct inference than that for incorrect inference. The mean values of physical consistency ratio, logical consistency ratio, and inference consistency ratio, i.e., softmax probability, over entire experimental data even including incorrect inference were 0.75, 0.70, and 0.48, respectively. According to these results, our method gained consensus on humans, overall distribution of consistency ratios are reasonable.

Table 1 Number of human tasks to evaluate consistency measures

| Measure | Class | Sample | Worker | Total tasks |
|---------|-------|--------|--------|-------------|
| PCR     | 32    | 10     | 20     | 6400        |
| LCR     | 32    | 10     | 20     | 6400        |

Redundancy added by discrete samples and workers is to eliminate individual biases. Inference consistency ratio (maximum softmax probability) is automatically computed.

6 Discussion

In this section, we show whether our proposed simple analysis improves the transparency of inference processes of convolutional neural networks, and we study what practical discussion in a machine learning training and testing process can be done on a neural network thanks to that improved accuracy.

Let us assume that we have a CNN which we are currently train and test. We have inference (label) by the currently trained model, and the results of our proposed analysis. Figures 15 and 16 show results of analysis for correct inference and incorrect inference, respectively. Images in Figs. 15 and 16 are converted into \(227 \times 227\) which is the actual size of the input image to CaffeNet, and receptive fields are omitted due to space limitation. We walk through these results of analysis to see what we can read from them.

For the images on the left column in Figure 15a which have the results of analysis in low physical consistency ratio and inference consistency ratio, the number of feature vector elements in inference features can be less than \(\ell\), the maximum number of feature vector elements in the inference features. Human workers may have evaluated these images’ physical consistency ratios low, because they saw few feature vector elements in inference features. It is interesting that inference consistency, which is the maximum softmax probability, is also low, if the number of feature vector elements in inference features is small. It is suggested that the inference process we
neural networks. The hypothesis in this work is not far from actual processes which humans cannot understand in some cases.

The images on the right column in Fig. 15a have high physical consistency ratio and low inference consistency ratio. The bottom one has low logical consistency ratio, because the labels of visual attributes are not appropriate. Cassette players should have two large speakers on the left and right, and the feature maps 196 and 171 may represent them. However, the labels (visual attributes) for these feature maps are rubber tires or rounded (shape), and human workers may not be able to associate them. There is room to improve the labels of visual attributes so that humans can easily comprehend the linguistic feature analysis.

The top right image in Fig. 15b has high physical, logical, and inference consistency. We see three types of visual attributes; (1) shape (fine lattice patterns, accumulated fine boxes/circles, leopard patterns), (2) color (two-tone red/white), and (3) concrete object (black square windows, faces of small animals) in the linguistic feature analysis, and these visual attributes are relevant to ambulance vehicles for humans, too. This is one of the best examples.

The images on the right bottom in Fig. 15b has low physical and logical consistency ratio and high inference consistency ratio. The linguistic feature analysis indicates sharp roofs/caps and accumulated fine boxes/circles or rubber tires, but humans may not find these visual attributes in the image. On top of them, even if these visual attributes are in the scene, humans cannot understand why they are associated with inference (label): barber chair. This example shows the combination of the above two situations. These three examples show the limitations of deep neural networks in terms of transparency. There must be the essential complexity of deep neural network which we cannot make transparent.

The images on the left top in Fig. 15b has high logical and inference consistency ratios, and low physical consistency ratio. We can see fur like visual attributes in the result of linguistic feature analysis, however they are not found in the input image. This example shows that there are features in the trained model which humans cannot understand.

The images on the left bottom in Fig. 15b has low physical and logical consistency ratio and high inference consistency ratio. The linguistic feature analysis indicates sharp roofs/caps and accumulated fine boxes/circles or rubber tires, but humans may not find these visual attributes in the image. On top of them, even if these visual attributes are in the scene, humans cannot understand why they are associated with inference (label): barber chair. This example shows the combination of the above two situations. These three examples show the limitations of deep neural networks in terms of transparency. There must be the essential complexity of deep neural network which we cannot make transparent.

In the second example from the left on the first row in Fig. 16, inference (label) of CNN is snake, but the correct label is brambling, a type of bird. Our analysis indicates that the inference feature includes feature vector elements for “squiggle” visual attributes. This is an example of understandable mistake of CNN. Although the inference (label) is incorrect, we see squiggle in the input image; and squiggle is likely to be a snake. We assume that the size of the bird was too small compared to the size of squiggle patterns, and CNN may put high priority on snake class. If squiggle patterns, which are made by roof tiles, are larger than the bird, there is room for discussion whether the ground truth class should be roof tile, rather than bird.
In the second example from the right on the first row in Fig. 16, the inference (label) of CNN is flat-coated retriever, but the correct label is groenendael. Both of them are black dogs. The inference features for this incorrect inference (label) produced by the CNN: flat-coated retriever are very similar to the inference features for the correct inference (label): groenendael. This is another pattern of understandable mistake of CNN that the currently learnt visual attributes are not enough to distinguish between two classes. We need to collect training data more to acquire relevant visual attributes.
If inference (label) is incorrect with correct inference features, then it suggests insufficient training data to train relevant visual attributes for these classes. Possible action for this case is to collect additional training data for these classes. If (1) inference features are correct for the input image, and (2) inference (label) is correct for the inference features; however, (3) inference (label) is incorrect for the input image, then an inaccurate ground truth label is suggested. Possible action for this case is to review and fix the ground truth label.

It is important in practice to know the actions we should take next. Low physical consistency ratio suggests that the feature extraction part of the neural network is not well trained to capture enough visual attributes. On the other hand, low logical consistency ratio suggests that the decision-making part of the neural network, such as classification or regression, is not well trained. Possible action for the former case is to increase the layers in the feature extraction part, which is considered as the layers before \texttt{conv5} for CaffeNet. Possible action for the latter case is to increase the layers in decision-making part which is considered as the layers after \texttt{conv5}.

7 Conclusion

In this paper, we developed three types of simple analysis: 1) structural feature analysis, 2) linguistic feature analysis, and 3) consistency analysis which improve the transparency of deep neural inference process, to address the black-box property of deep neural networks for safety critical applications. We then evaluated and discussed our analysis methods and the results both qualitatively and quantitatively, and introduced the usefulness of our proposed analysis by showing how to use the analysis results in the development process of deep learning models.

It is known that quantitative evaluation of the transparency of algorithms is challenging [7], and we cannot say our work solved the problem completely. However, deep neural inference process was black box until now, and the experiments and discussion in this paper shows that our work moved it forward to transparency. For example, there was no clue to improve a neural network when it produced incorrect inference (label). Now our method gives suggestions, or the possible next actions, such as expanding networks or collecting training data.

Future works include improving evaluation process. In this preliminary work, for simplicity, we conducted only simple closed questions with a small number of selections (\textit{agree/disagree}) for human evaluation. These can be extended with open written questioners and valuation questions asking high resolution, e.g., hundred level, scores to obtain more precise qualitative and quantitative feedback from experiment participants on their perceptions of transparency.

References

1. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE 10(7), e0130,140 (2015). https://doi.org/10.1371/journal.pone.0130140
2. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: quantifying interpretability of deep visual representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
3. Binder, A., Montavon, G., Lapuschkin, S., Müller, K., Samek, W.: Layer-wise relevance propagation for neural networks with local renormalization layers. In: Villa, A. E. P., Masulli, P., Rivero, A. J. P. (eds.) Artificial Neural Networks and Machine Learning—ICANN 2016—25th International Conference on Artificial Neural Networks, Barcelona, Spain, 6–9 Sept, 2016, Proceedings, Part II. Springer, Lecture Notes in Computer Science, vol 9887, pp. 63–71 (2016) https://doi.org/10.1007/978-3-319-44781-0_8
4. Bojarski, M., Testa, D.D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Montfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., Zieba, K.: End to end learning for self-driving cars. CoRR (2016) arXiv:1604.07316
5. Bojarski, M., Choromanska, A., Choromanski, K., Firner, B., Ackel, L.J., Muller, U., Yeres, P., Zieba, K.: Visualbackprop: Efficient visualization of cnns for autonomous driving. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 1–8 (2018)
6. Choo, J., Liu, S.: Visual analytics for explainable deep learning. IEEE Comput. Graph. Appl. 38(4), 84–92 (2018). https://doi.org/10.1109/MCG.2018.04273161
7. Dam, H.K., Tran, T., Ghose, A.: Explainable software analytics. In: Zisman, A., Apel, S. (eds.) Proceedings of the 40th International Conference on Software Engineering: New Ideas and Emerging Results, ICSE (NIER) 2018, Gothenburg, Sweden, May 27–June 03, 2018, pp. 53–56. ACM (2018). https://doi.org/10.1145/3183399.3183424
8. Ding, W., Wang, R., Mao, F., Taylor, G.: Themo-based large-scale visual recognition with multiple gpus (2014) arXiv preprint arXiv:1412.2302
9. Došilović, F.K., Brčić, M., Hlupić, N.: Explainable artificial intelligence: a survey. In: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 0210–0215 (2018)
10. Escorcia, V., Niebles, J.C., Ghanem, B.: On the relationship between visual attributes and convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, 7–12 June, 2015, pp. 1256–1264 (2015). https://doi.org/10.1109/CVPR.2015.7298730
11. Fukushima, K., Miyake, S.: Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position. Pattern Recognit. 15(6), 455–469 (1982). https://doi.org/10.1016/0031-3203(82)90024-3
12. Ganesan, K.: Computing precision and recall for multi-class classification problems. http://text-analytics101.rxnlp.com/2014/10/computing-precision-and-recall-for.html (2014)
13. Graves, A., Jaitly, N., Mohamed, A.: Hybrid speech recognition with deep bidirectional LSTM. In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, 8–12 Dec, 2013, pp. 273–278. IEEE (2013) https://doi.org/10.1109/ASRU.2013.6707742
14. Grüner, F., Rupprecht, C., Navah, N., Federico, T.: A taxonomy and library for visualizing learned features in convolutional neural networks. In: ICML Workshop on Visualization for Deep Learning (ICML-W) (2016)

15. Gunning, D.: (2016) Explainable artificial intelligence (xai). https://www.darpa.mil/program/explainable-artificial-intelligence

16. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), IEEE Computer Society, Washington, DC, USA, ICCV ’15, pp. 1026–1034 (2015), https://doi.org/10.1109/ICCV.2015.123

17. Hendricks, L.A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., Darrell, T.: Generating visual explanations. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision—ECCV 2016—14th European Conference, Amsterdam, The Netherlands, 11–14 Oct, 2016. Proceedings, Part IV, Springer, Lecture Notes in Computer Science, vol. 9908, pp. 3–19 (2016), https://doi.org/10.1007/978-3-319-46493-0_1

18. Hohman, F.M., Kahng, M., Pienta, R., Chau, D.H.: Visual analytics in deep learning: An interrogative survey for the next frontiers. In: IEEE Transactions on Visualization & Computer Graphics (2018), https://doi.org/10.1109/TVCG.2018.2843369

19. Koopman, P., Wagner, M.: Challenges in autonomous vehicle testing and validation. SAE Int J Transf Saf 4(2016—01—0128), 15–24 (2016)

20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105. Curran Associates, Inc. (2012)

21. Kuwajima, H., Tanaka, M.: Network analysis for explanation. In: Transparent and interpretable Machine Learning in Safety Critical Environments (NIPS2017 Workshop) (2017)

22. Kuwajima, H., Yasouka, H., Nakae, T.: Open problems in engineering and quality assurance of safety critical machine learning systems. In: Joint Workshop Between ICML, AAMAS and IJCAI on Deep (or Machine) Learning for Safety-Critical Applications in Engineering (2018)

23. Lin, M., Chen, Q., Yan, S.: Network in network. CoRR (2013). arXiv:1312.4400

24. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds) Advances in Neural Information Processing Systems, vol. 30, pp. 4765–4774. Curran Associates, Inc. (2017)

25. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 26, pp. 3111–3119. Curran Associates, Inc. (2013)

26. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J.: Introduction to WordNet: an on-line lexical database. Int. J. Lexicogr. 3(4), 235–244 (1990)

27. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.: Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recognit. 65, 211–222 (2017). https://doi.org/10.1016/j.patcog.2016.11.008

28. Montavon, G., Samek, W., Müller, K.: Methods for interpreting and understanding deep neural networks. Digital Signal Process. 73, 1–15 (2018)

29. Park, D.H., Hendricks, L.A., Akata, Z., Schiele, B., Darrell, T., Rohrbach, M.: Attentional explanations: justifying decisions and pointing to the evidence. CoRR (2016). arXiv:1612.04757

30. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)

31. Powers, D.M.W.: Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. J. Mach. Learn. Technol. 2(1), 37–63 (2011)

32. Ribeiro, M.T., Singh, S., Guestrin, C.: “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, San Francisco, CA, USA, 13–17 Aug, 2016, pp. 1135–1144 (2016)

33. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV) 115(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y

34. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: The IEEE International Conference on Computer Vision (ICCV) (2017)

35. Shrikumar, A., Greenside, P., Shcherbina, A., Kundaje, A.: Not just a black box: learning important features through propagating activation differences. CoRR (2013). arXiv:1605.01713

36. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, PMLR, International Convention Centre, Sydney, Australia, Proceedings of Machine Learning Research, vol. 70, pp. 3145–3153 (2017)

37. Simonoyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. In: Proceedings of the International Conference on Learning Representations (ICLR) (2014)

38. Uchida, K., Tanaka, M., Okutomi, M.: Coupled convolution layer for convolutional neural network. Neural Netw. 105, 197–205 (2018)

39. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, 7–12 June, 2015, pp. 3156–3164 (2015). https://doi.org/10.1109/CVPR.2015.7298935

40. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A.C., Salakhutdinov, R., Zemel, R.S., Bengio, Y.: Show, attend and tell: neural image caption generation with visual attention. In: Bach, F.R., Blei, D.M. (eds) Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015, JMLR.org, JMLR Workshop and Conference Proceedings, vol. 37, pp. 2048–2057 (2015)

41. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. IEEE Comput. Intell. Mag. 13, 55–75 (2018). https://doi.org/10.1109/CLI.2018.2840738

42. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision—ECCV 2014, pp. 818–833. Springer Science, vol. 9908, pp. 3–319-46493-0_1

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.