A Metric to Compare Pixel-Wise Interpretation Methods for Neural Networks

JAY HOON JUNG AND YOUNGMIN KWON
Computer Science Department, The State University of New York at Korea, Incheon 21985, South Korea
Corresponding author: Jay Hoon Jung (jay.jung@stonybrook.edu)

This work was supported in part by the Ministry of Science and ICT (MSIT) through the IT Consilience Creative Program (ITCCP), South Korea, under Grant IITP-2019-2011-1-00783, in part by the National Research Foundation (NRF) of MSIT, South Korea, under Grant 2019R1F1A1058770, and in part by the Korea Evaluation Institute of Industrial Technology (KEIT) through the Global Advanced Technology Center (GATC) Program under Grant 10077300.

ABSTRACT
There are various pixel-based interpretation methods such as saliency map, gradient ∗ input, DeepLIFT, integrated-gradient- n, etc. However, it is difficult to compare their performance as it involves human cognitive processes. We propose a metric that can quantify the distance from the importance scores of the interpretation methods to human intuition. We create a new dataset by adding a simple and small image, named as a stamp, to the original images. The importance scores for the deep neural networks to classify the stamped and regular images are calculated. Ideally, the pixel-based interpretation has to successfully select the stamps. Previous methods to compare different interpretation methods are useful only when the scale of the importance scores is the same. Whereas, we standardize the importance scores and define the measure to ideal scores. Our proposed method can quantitatively measure how the interpretation methods are close to human intuition.

INDEX TERMS
Interpretation of neural networks, explanation of neural networks, causality of neural networks.

I. INTRODUCTION
When the size of an input to a function is small and its input-to-output relation is simple, we can easily capture the significant part of the input that caused certain output. For example, the second Newton’s laws of motion, $F = m \cdot a$, states that the net force acting on an object causes an acceleration of the object. We do not need any explanations when we understand the meaning of the function $acc(F) = F / m$ since two variables, $F$ and $a$, are directly proportional to each other. However, when the dimension of input is large, and when the input-to-output relation is complex, it is not straightforward to find the main components of the input that caused certain output. For example, suppose that an autonomous vehicle, controlled by a deep neural network, made a lane change. For the lane change decision, it is difficult to tell whether the decision is made due to a car in front or an airplane that happens to be in the vehicle camera’s viewport.

The causality between the acceleration and the force is straightforward: observing an acceleration of 1 ($m/\text{sec}^2$) of a 1 ($t$) vehicle, we know that it is caused by a force of 1000 ($N$). Unlike this Single-Input-Single-Output (SISO) linear relation, deep neural networks have Multiple-Input-Multiple-Output (MIMO) nonlinear relations in general. The nonlinearity makes it very difficult to comprehend the input to output relation and the complexity itself induces a new problem of explaining or interpreting a neural network. Moreover, not all elements of an input are equally important. Large changes in some parts of an input may not affect the output at all, but small changes in some parts of the input can widely swing the output. This different sensitivity to input elements incurs another problem of identifying which parts of an input mainly contributed to an observed output of a neural network.

Therefore, researchers have been studying the importance of each element of the input. The importance scores are defined as how each element of the input influence the outputs or predictions. Even though the measurement methods are different for different researchers, the basic assumptions are the same. Zeiler and Fergus et. al introduced the ‘deconvolution’ approach showing the part of the input images that is most strongly activating units in the convolutional neural network with the max-pooling [1]. Springenberg et al. expanded deconvolution approaches to a broader range of network structures [2]. Another approach is related to the variation of the inputs [3]–[5]. Sundararajan et al. suggested Integrated Gradients, which accumulate the gradient over $n$
evenly-spaced intervals using mid-points of the input vector [3]. Lundberg and Lee introduced the Shapley Value regression of game theory to compute the importance scores [4]. Ribeiro et al. presented LIME which explains the model by systematic perturbation of the images [5]. There are also approaches to compute importance scores by attaching a special explaining tool to the networks [6], [7]. Chen et al. suggested the explaining neural networks, known as Learn to eXplain (L2X), to find the important pixel or input of other neural networks [6]. Jung et al. introduced explainable components inside of the neural networks and computed the importance of each component to explain the model [7].

There are approaches related to the gradients of the output with respect to the input vector [8]–[11]. Grad-CAM computes coarse-grained important scores by using the gradients of any target class flowing into the final convolutional layers [11]. Simonyan et al. suggested the absolute values of gradients of the predicted class scores functions w.r.t. the input pixels, also known as the saliency map, as importance scores [9]. The saliency map represents the gradient-backpropagation through the model. As a result, it is possible to have relatively high importance scores even though the input pixel’s values are equal to 0. Baehrens et al. computed the gradient using Pazen window approximations [12]. Bach et al. proposed a method whose importance scores are calculated by propagating scores through the layers called Layerwise Relevance Propagation (LRP) [8]. Shrikumar et al. proved that the importance scores based on LRP are equivalent to the element-wise product between the gradients and the input [13]. Consequently, the influence of the 0-valued pixels is removed in the LRP importance scores and the max-valued pixels are emphasized compared to the saliency map. In addition, Shrikumar et al. suggested a method called DeepLIFT which assigns importance scores according to the activation differences between the input and some reference input [14].

Unfortunately, human interpretation is different from how individual elements of the input influence the outputs. Humans need an abstract concept to interpret or explain. In the case of classification, we classify images or objects into several classes based on our abstract definition of the class. For example, we may describe the number 1 as a straight vertical line from top to bottom. We actively search for these features in an image to recognize it. Montavon et al. also defined an interpretation as “the mapping of an abstract concept into a domain that the human can make sense of” [15]. Therefore, the importance scores cannot be directly used to interpret the results by themselves, but have to be processed one more time by humans. In other words, we have to find the domain from the importance score that makes sense. Hence, quantifying the usefulness and correctness of several methods is difficult since there is no quantifying standard to measure interpretability.

In this article, we develop a metric to compare several different interpretation methods of the neural networks quantitatively. To acquire objectiveness, we have to minimize humans’ intervention as much as possible by adding small but carefully designed shapes to the images. The deep neural networks are trained to search these shapes. Every pixel of the shapes should be important because of its size. It is also too obvious for humans to interpret differently. Therefore, using these shapes, we standardize the importance scores computed by different interpretation methods to compare them.

We organize this article in the following way. First, we will state the problem formally in Section II. The problem statement is composed of interpretation methods, comparing method, stamps, and a quantitative metric and we will discuss each part of the statements in the following sections. We will list the different pixel-wise interpretation methods in Section III and their comparison methods in Section IV. We propose stamps and our metric in Section V and demonstrate the experiments based on the proposed method in the subsequent sections.

II. PROBLEM STATEMENTS

Suppose \(i_1, i_2, \ldots, i_n\) are the importance scores computed by the various methods such as gradient, gradient x input, and etc. of the model to search the stamps \(S\) which are the small unique patterns attached to the image. The stamp is responsible for the classification of the images. Therefore, our goal is to develop a quantitative metric for comparing the importance scores, \(i_1, i_2, \ldots, i_n\), to indicate which score successfully selects the stamp \(S\).

In other words, we will show which interpretation methods are close to human intuition as we discuss in the Introduction.

III. PIXEL-WISE INTERPRETATION METHODS

In this section, we will briefly discuss several pixel-wise methods mathematically by defining a trained neural network model, which leads to get intuitive insight on how these interpretation methods work. To simplify the analysis we used Rectified Linear Units (ReLU) functions whose steep edges are linear and whose shallow edges are zero. That is, ReLU(x) = a · x for \(x \geq 0\) and ReLU(x) = 0 for \(x < 0\), where a is a positive number.

A. PIECE-WISE LINEAR NEURAL NETWORKS

For a deep feedforward network model \(M\) with \((n_l + 1)\) layers, \(w_{lj}^{(l)}\) denotes the weight from the \(i^{th}\) unit in the \(l^{th}\) layer to the \(j^{th}\) unit in the \((l + 1)^{th}\) layer, and \(L_i^{(l)}\) and \(B_i^{(l)}\) denote the \(i^{th}\) unit (or its value) and the \(i^{th}\) bias in the \(l^{th}\) layer respectively. Let \(L^{(l)}\) be the set of units in the \(l^{th}\) layer and let \([L^{(0)}]\) be the set of unit indexes in \(L^{(0)}\). Especially, \(L^{(0)}\) and \(L^{(n)}\) are the input layer and the output layer with the widths of \(n_{in}\) and \(n_{out}\) respectively.

If a Rectified Linear Units function, ReLU(x), is the activation function for an NN model \(M\), then the value of a unit \(L_j^{(l+1)}\) is defined as

\[
L_j^{(l+1)} = \text{ReLU} \left( \sum_{i \in [L^{(l)}]} w_{ij}^{(l)} \cdot L_i^{(l)} + B_j^{(l+1)} \right),
\]

(1)
for all \( j \in L^{(l+1)} \). Observe that even though the weights \( w_{i,j}^{(l)} \) are a constant for a trained model \( M \), \( L_i^{(l)} \) is a variable that depends on an input \( I \).

**B. OUTPUT LAYERS WITH RESPECT TO INPUT LAYER**

We showed that for any feedforward network model \( M \) with only \( \text{ReLU} \)-activation, the value of the output units \( O_j \) for each \( j \in [L^n] \) is given as

\[
O_j = \sum_{i \in [L^0]} (\alpha_{i,j} \cdot I_i) + c_j, \tag{2}
\]

where \( \alpha_{i,j} \) and \( c_j \) are constants valid within a polytope-shape linear region defined by the combinations of the ReLU functions of each unit in each layer [16]–[18].

Equation (2) shows that the output layer of an NN is a piece-wise linear function of the input layer. Moreover, each argument to ReLU function defines a half-space in the input space \( I \) and they define a polytope. Specifically, the equation \( \sum_{i \in [L^0]} (\alpha_{i,j} \cdot I_i) = c_j \) defines a hyperplane in the input space \( I \) and the set \( \{ I : \sum_{i \in [L^0]} (\alpha_{i,j} \cdot I_i) + c_j \geq 0 \} \) is a half-space. Hence, the intersections of the half-spaces define a polytope in the input space. Because \( \alpha_{i,j} \) and \( c_j \) are constants within each polytope, the output layer of an NN is a piece-wise linear function of the input layer.

**C. RELATIONS TO INTERPRETATION METHODS**

According to Equation (2), the derivative of the class score \( O_c \) with respect to the input \( I \) is given by

\[
\alpha_{i,c} = \frac{\partial O_c}{\partial I_i},
\]

where \( c \) is an index of the predicted class. Simonyan et al. showed that the absolute value of coefficients can be interpreted as the image-specific saliency since they are the gradients w.r.t. the input pixels. The \( i \)th importance score \( S_i \) of saliency map is given by

\[
S_i = |\alpha_{i,c}|
\]

Similarly, multiplication of coefficients and pixels of an input are the same as the scores calculated by LRP or the gradient\times\text{input} [8]. The \( i \)th importance score \( S_i \) of the integrated gradient\(-n \) method is given by

\[
S_i = \alpha_{i,c} \cdot I_i
\]

Sundararajan et al. suggested a method that numerically integrates the gradient over \( n \) evenly-spaced intervals using the midpoints [3]. In other words, the \( i \)th importance score \( S_i \) of the integrated gradient\(-n \) is given by

\[
S_i = \frac{1}{n} \sum_k \alpha_{i,c}^k
\]

where \( \alpha_{i,c}^k \) are coefficients of the gradient when the input is equal to \( k/n \cdot I \).

Shrikumar et al. proposed DeepLIFT that assigns scores according to the difference between the activation of each unit and its reference [13]. The difference-from-reference can back-propagate even if the gradient is zero. The \( i \)th importance score \( S_i \) of DeepLIFT is given by

\[
S_i = C_{\Delta I_i \Delta O} = m_{\Delta I_i \Delta O} \cdot \Delta I_i
\]

where \( m_{\Delta I_i \Delta O} \) is given by

\[
S_i = \frac{\Delta l_i}{\Delta I_i} \cdot \Delta I_i
\]

where \( \Delta l_i \) is the change in log-odd from removing the \( i \)th input.

The saliency map represents a characteristic of a linear region instead of an input image since the coefficients \( \alpha_{i,c} \) are shared for a whole linear region defined by a polytope in the input space. The saliency maps are not input-specific if the neural network is piece-wise linear. Meanwhile, the gradient\times\text{input} scores select important coefficients of gradients related to the inputs. DeepLIFT takes account of the bias terms along with the gradient of an output respect to the input. Bias terms include the information related to the boundaries of a polytope in the input space. On the other hand, the integrated gradient\(-n \) method considers several different inputs located in the different linear regions. Even though \( n \) evenly-spaced inputs seem similar to each other, they are different inputs to the neural networks.

**IV. EXISTING APPROACHES TO COMPARE THE SCORES**

In this section, we summarize existing approaches for comparing the importance scores. To the best of our effort, there is no standardized method. It is because the interpretation involves human cognitive processes and the importance scores are difficult to be standardized. To assure the objectiveness, researchers developed log-odd method [4], [13] and survey methods [4]. The log-odd methods do not involve the cognitive process whereas the survey method actually surveys human intuition.

**A. LOG-ODD METHOD**

The log-odd method uses two different importance scores of the original \( S_O \) and target \( S_T \) class. The difference of two importance scores \( S_O - S_T \) is used to select the most important elements of inputs to change the prediction from the original to the target class. Normally, \( 20\% \) of inputs are erased and these new inputs are processed with the neural network again. The change in the log-odds scores is evaluated.

The log-odd methods are useful since they can quantify the importance score based on the changes in log-odd values. However, there are two main concerns with the log-odd method to be a generalized method. Firstly, the log-odd method can be used only when the neural network model and its weight values are the same. Generally, two different neural networks have different scales in their logits which are units previous to softmax function. The different scales can result in a biased conclusion. For example, if the values of two logits are 0.1 and 1 for a network A and 100 and 1000 for a network B. Even though their ratio is the same, their log-odd scores are 0.9 and 900. Thus, the network B always has the higher log-odd value. Secondly, log-odd method is based on the assumption that the predicted class is the predicted class...
FIGURE 1. The different types of small images, named as stamps, inserted into MNIST dataset. There are five different types: (a)∼(e). (f) and (g) are examples of the images inserted the stamp (d) and (e) respectively.

because it is not other classes. For instance, if a network classifies an image as a number 1, then the importance scores finding different pixels between other numbers and 1 have higher log-odds values. It means that a number 1 is 1 because it is not 2, 3, ..., 0. As discussed in Introduction, this is not how we perceive the objects.

B. SURVEY METHOD
The survey method is based on surveying people about their intuitions. The survey method has a big advantage; it directly compares the importance scores and human intuitions. The relations between the importance score and human intuition can be shown undoubtedly. However, the survey method also has many disadvantages at the same time. Firstly, only relatively straightforward questions can be qualified since answers can be inherently subjective unless survey questions are carefully designed. In other words, the questions are limited to simple and direct ones. For instance, Lundberg et al. asked a multiple-choice problem to participants and compared their answers to the importance scores [4]. Secondly, the credibility of surveys should be guaranteed. Depending on a representative sample, the results of the survey can vary significantly. Designing the unbiased survey itself is a non-trivial task.

V. PROPOSED METHOD
In this section, we describe our proposed method that handles two issues, namely the standardization and the minimization of a heuristic process. We create new sets of images by adding small and simple images, named as stamps, to the original image data. Trained with both the new and original set, a deep neural network can classify images into two groups; the images with and without a stamp. The stamps have to be large enough to be distinctive, but small enough to be a single piece, which means there are no sub-feature inside stamps. The importance scores of the deep neural network are calculated by different methods such as saliency map, gradient×inputs, and others. Since the stamps are a single piece, pixels of the stamps should have higher scores than the rest of the pixels of the inputs. The importance scores are standardized for the comparison between different deep neural networks or interpretation methods.

A. STAMPS
The purpose of adding stamps is to minimize a heuristic process. Searching for the stamps in images is a straightforward task for humans without any room for subjectivity. Its interpretation is also simple; every pixel of the stamps should be present in the images. We are able to know human intuitions already without asking them. The ideal interpretation methods should have higher values for all the pixels of the stamps compared to the other pixels of an input.

The stamps should be unique and composed only of a small number of pixels in order to suppress undesirable effects. If they are too big, then there might be unique patterns inside the stamps. That is, not all pixels of the stamps are necessary to find where they are. On the other hand, if they are too small, they are not unique anymore. The deep neural networks cannot be trained if the stamps are not unique.

Figure 1 shows the 5 different types of stamps. The stamp (a) is composed of 6 pixels. The black pixels are where their values are equal to 255 or 1 depending on the maximum value of the original images. The white pixels represent 0 values. If the number of pixels is reduced below 6, the stamps are not unique anymore and a few stamped images become indistinguishable. The stamp (b) and (c) have 8 pixels and other shapes are as illustrated in Figure 1. The stamp (d) have only 4 pixels, but the distances between pixels are 14 pixels. The stamp (d) is inserted into MNIST dataset as an example shown in Figure 1 (f). The stamp (e) has a cross-shaped 9 pixels.
B. TRAINING AND INTERPRETATION
The deep neural networks are trained to classify the stamped and regular images. Normally, the networks are easily trained since the task is a binary classification and the patterns of stamps are easily distinguishable. Several importance scores are calculated with the methods as we described in Section IV. The detailed experimental setting is described in Section VI-D. It is worthwhile to mention that the method is applicable to any architecture of the deep neural network.

C. STANDARDIZATION
The importance scores evaluated by different methods are on a different scale. For example, the scores of the saliency map are always positive numbers since they are the absolute values of the gradients. The score of the gradient×input can be negative and positive. Standardization is necessary to compare different interpretation methods. The standardized data have a mean of 0 with a standard deviation of 1 to compare measurements that have different units [19]. Especially, the standardization is useful when the data is clustered in order to compare similarities between features based on certain distance measures [20]. If the importance scores reflect human intuitions as we described in Introductions, the scores should be divided into two clusters; the stamped and the rest regions. Ideally, the distribution of the rest region is close to the Gaussian distribution because, with many randomly distributed sample scores, the central limit theorem is close to the Gaussian distribution. Therefore, the distances between two clusters indicate how close the importance scores are to the human intuition.

The basic assumption of the standardization is that the importance scores for pixels of the stamps should be higher than the score of the rest of the pixels. Algorithm 1 shows the pseudocodes for computing standardized scores. We divide the importance scores into two sets: the stamp (T) and the rest (R) set. We transform the rest set to z-scores by

\[
\bar{S} = \frac{S_i - \bar{S}}{\sigma},
\]

where \(\bar{S} = \frac{\sum_{i \in R} S_i}{|R|}\) and \(\sigma^2 = \frac{\sum_{i \in R} (S_i - \bar{S})^2}{|R|}\) are the mean and the variance of the rest set respectively. After the transform, the z-score have a mean of zero and a standard deviation of 1.

In addition, the elements of the stamp are multiplied by either 1 or −1 depending on the color of the pixel unless the interpretation scores produce only positive numbers like saliency map. This process is necessary to handle negative values of the importance score. The negative values mean that the absence of the dot in that pixel is required to have the same shape as the stamps. The processed stamp set is transformed into the z-scores as well to compare. \(z^T_i = \frac{S^T_i - \bar{S}}{\sigma}\) where \(\bar{S}\) and \(\sigma\) are the same average and the same standard deviation as above, and \(S^T_i\) is the processed stamp set. The distance \(d\) between two sets are given by

\[
d = \frac{\sum_{i \in T} z^T_i}{|T|}.
\]

The distance \(d\) is the average of \(z^T_i\). In addition, \(d\) implies how far the rest and stamp sets are apart in standardized data. In other words, \(d\) have higher values if the importance scores of the stamp pixels are relatively higher than the rest pixels. We can directly compare \(d\) of the different importance scores since \(d\) is calculated in standardized scores.

Therefore, the standardized distance \(d\) is a metric to measure how close the importance scores are to the human intuition. We will show that the distance \(d\) experimentally reflects human intuitions as well. Furthermore, the execution time for the log-odd method is as fast as the subtraction between two matrices. Thus, the complexity for the log-odd method is \(O(n)\), where \(n\) is the number of pixels of the image. Whereas, survey methods require people to mark the image, which is impossible to calculate execution time. The execution time for the proposed methods is also \(O(n)\) since the necessary computation involves sorting, calculating averages, and subtraction between two matrices.

VI. EXPERIMENTS
In this section, we demonstrate that the standardized distance \(d\) is practically useful to compare the different interpretation methods. Firstly, there are two pieces of evidences that the distance \(d\) can quantify how good the importance screen out the backgrounds. Secondly, we check how each stamp is applicable to be used as a standard. Finally, we indicate the most matching methods to human intuitions out of 5 different interpretation methods of neural networks: the gradient×input, gradients, guided gradient, DeepLIFT, and integrated gradient-5.
A. CAN WE TRUST DISTANCES?

The distance $d$ is theoretically designed to compare the importance scores of the stamped images regardless of their scales. We compare the distances between different images qualitatively in Figure 3. Each row contains the stamped original image and five different interpretation methods as indicated above. The stamp (a), (c), (d), and (e) are used from the first row to the last row. Corresponding distances of each method are indicated below the images. As shown, the distances get higher value if only stamped regions have high values. Gradient and guided gradient methods show lower distances since the importance scores of unwanted regions have comparable values to those of stamped regions. On the other hand, gradient×input and integrated gradient-5 methods show higher distances since only stamped regions are high values for these methods. When the pixels of the inputs are equal to zeros, the importance scores of these pixels are also zeros in case of the gradient×input method. Whereas, Deep LIFT scores are in-between. The most of stamped regions are high-valued, but some of the other regions are also high-valued in the Deep LIFT scores. Therefore, the distance $d$ reflects how good the interpretation methods spot the stamps in the images.

Furthermore, Figure 4 also demonstrates that the distance $d$ is a quantitative measure for recognizing the stamps in the image. In Figure 4, there are five graphs of the average distances of the entire test data vs. the accuracy of the neural network. Each graph is based on the 5 different importance scores of the same neural network. The average distance $d$ is calculated based on those scores. Since it is a binary classification, the minimum accuracy is approximately about 50%. The neural network is trained to spot the stamps in the images. In other words, training improves the accuracy of the network. Similarly, the distance $d$ is also an indicator of how the networks successfully search the stamps. Therefore, if the accuracy increases, the distance should increase at the same time as shown in Figure 4. All five methods show the same tendency since the neural networks actually are improved on recognizing the stamps.

The multiplication of a constant number to the importance scores does not change the distance $d$ by the definition. In addition, Figure 3 and 4 suggest that the distance $d$ is...
FIGURE 4. The graphs of the distances vs. the accuracies for the five different interpretation methods finding the stamp (a). As the accuracy improves, the distance is increased. In other words, if the neural networks are trained well to spot the stamps, then the distances increase.

a measure of how much the stamps are important in the classification. Therefore, the distance \( d \) is useful to compare the different importance scores. However, we have to examine the effect of the stamps’ shape to compare the methods without bias.

**B. WHICH STAMPS?**

As discussed in Section V-A, the stamps are required two properties: uniqueness and minimum size. The uniqueness of the stamp means the absence of stamp-like patterns in the image. The minimum size requires that all pixels of the stamps are necessary to be recognized. These properties are required since the distance \( d \) is to quantify only the stamped regions in the image. If the stamp is not unique, then the distance \( d \) can disfavor to the images which include similar patterns since their importance scores are also high-valued. For example, Figure 5 (a) and (b) are the images that included the stamp-like parts highlighted with red boxes. Especially, Figure 5 (a) shows an example that the original image contains the same shape as the stamp. The presence of these patterns will deteriorate the accuracy of the model and credibility of the importance scores. Therefore, Stamp (b) and (d) are not suitable for distance measurements.

If the stamp is bigger than the minimum size, then sub-patterns can exist inside of the stamp. Figure 5 (c) and (d) show the shape of Stamp (c) and (e) and their importance scores. The importance scores suggest that there exists the sub-pattern inside of the stamp. The neural network search for the smaller patterns instead of the stamps. For example, the network tends to find Stamp (a) instead of Stamp (c) even though they are trained with Stamp (c). Two pixels in Stamp (c) are not essential parts. Similarly, the presence of sub-pattern in the stamps systematically will produce biased results.

Therefore, the Stamp (a) is suitable for the comparison between different interpretation methods since it is a minimal unique pattern.

**C. BETTER METHODS?**

We measure the distance \( d \) of Stamp (a) for various models to figure out which interpretation methods are close to human intuition. Figure 6 shows the box plots of the distance \( d \) for model 1 and 2. Table 1 shows the average distance \( d \) and their standard deviations for model 3-8. The architectures of models are discussed in the following section. The same tendency is observed regardless of the models. As expected, Gradient and Guided Gradient methods show the small distances since they are input-specific. The DeepLIFT method is better than these two methods but its interpretation performance varies depending on how the model is trained. The distance \( d \) for the Integrated Gradient-5 method is the highest value.
FIGURE 6. The distance $d$ measurement of Stamp (a) for model1 (a) and model 2 (b).

TABLE 1. Distance $d$ depending on the interpretation methods. The same tendency is observed regardless of the models.

| Model | Gradient | Deep LIFT | Guided grad. | Grad x input | Int. Grad-5 |
|-------|----------|-----------|---------------|--------------|-------------|
| 3     | 8.44±1.1 | 8.76±1.4  | 12.21±0.6     | 36.28±1.9    | 37.07±1.3   |
| 4     | 13.15±0.6 | 31.09±3.3 | 14.49±0.7     | 31.12±1.6    | 31.38±1.4   |
| 5     | 11.82±1.2 | 36.71±1.9 | 12.76±1.0     | 36.69±0.9    | 37.21±0.9   |
| 6     | 7.52±1.1  | 7.26±1.4  | 9.51±0.7      | 30.83±2.6    | 31.72±2.5   |
| 7     | 14.71±0.7 | 39.78±1.5 | 14.65±0.6     | 41.43±0.5    | 41.59±0.5   |
| 8     | 15.83±1.5 | 27.20±9.1 | 17.75±1.9     | 33.08±1.7    | 33.09±1.8   |

regardless of models. Therefore, the interpretation method close to human intuition is the Integrated Gradient-5 method.

D. EXPERIMENTAL DETAILS

The Modified National Institute of Standards and Technology database (MNIST) is a database of 28 × 28 size images of handwritten digits [21]. We use 120,000 images for training and 20,000 images for testing. There are 60,000 and 10,000 stamped images in the training and testing set.

The architectures of the models in this article are listed in Table 3. For model 1, three convolutional and max-pooling layers are followed by an input layer. $3 \times 3 \times 1 \times 32$, $3 \times 3 \times 32 \times 64$ and $3 \times 3 \times 64 \times 128$ filters are used for each convolutional layer respectively with an appropriate padding. For model 2, only the first two convolutional layers are used. We use $2 \times 2$ max-pooling after the convolutional layers. One dense layer is followed after the convolutional layers. The

learning rate was 0.001 and 1 or 2 epochs were reached. All five stamped images are well-trained. The achieved training accuracies are listed in Table 2.

Figure 4 is based on model 1 with Stamp (a) and the accuracy of the model is achieved by training the model repeatedly and changing the number of epochs and the learning rate. The optimal learning rate is 0.001 and the deviation from the optimal value degrades the accuracy. Thus, it is possible to control accuracy by shifting the learning rate from 0.0001 to 0.01. In general, a large learning rate can reduce the cost fast, but the minimization process is more likely to end up with local minima. On the other hand, a small learning rate may result in a long training process that could also get stuck local minima. The proper learning rate will put more weight on the previous search direction and can have the momentum to overcome the barrier of the local minima.

Figure 6 is based on model 1-8 in Table 3. All models are based on the convolutional neural networks since they are sufficient to spot the stamps. The fully-connected models failed to find Stamp (a) since they require more parameters to be trained [22]. For model 3, only the first convolutional layers are used. For model 4-6 and 8, we change the kernel size to $5 \times 5$. We use $2 \times 2$ max-pooling after the convolutional layers. One dense layer is followed after the convolutional layers. We train the models to find the images with Stamp (a) and calculate the distance $d$ depending on the interpretation methods.

VII. CONCLUSION

There are many interpretation methods to analyze the deep neural network. The interpretation of the deep neural network is difficult since the hidden layers and activation function in the deep neural networks make nonlinear and complex functions. The interpretation involves heuristic processes that make it difficult to quantify. By adding stamps to the images,
we can minimize the vagueness of human intuitions. There-fore, we are able to compare 5 different interpretation meth-
ods and find out better methods.

REFERENCES

[1] M. D. Zeiler, G. W. Taylor, and R. Fergus, “Adaptive deconvolutional net-
works for mid and high level feature learning,” in Proc. Int. Conf. Comput.
Vis., Nov. 2011, pp. 2018–2025, doi: 10.1109/ICCV.2011.6126474.

[2] J. Tobias Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmüller,
“Striving for simplicity: The all convolutional net,” 2014, arXiv:1412.6806.[Online]. Available: http://arxiv.org/abs/1412.6806

[3] M. Sundararajan, A. Taly, and Q. Yan, “Grad-cam: Visual explanations from deep networks: Visualising image classification models and saliency maps,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2017, pp. 1803–1831, 2010.

[4] D. Batra, “Grad-CAM: Visual explanations from deep networks through propagating activation differences,” 2016, arXiv:1611.07270.[Online]. Available: http://arxiv.org/abs/1611.07270

[5] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. Müller, and W. Samek,
“How to explain individual classification decisions,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., 2016, pp. 1135–1144.[Online]. Available: https://www.aclweb.org/anthology/N16-3020/

[6] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan, “Learning to explain: An information-theoretic perspective on model interpretation,” in Proc. Int. Conf. Mach. Learn., (ICML), 2018, pp. 1–10.

[7] J. H. Jung and Y. Kwon, “Color, edge, and pixel-wise explanation of predictions based on interpretable neural network model,” in Proc. Int. Conf. Pattern Recognit. (ICPR), 2020.

[8] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. Müller, and W. Samek,
“On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,” PLOS ONE, vol. 10, no. 7, pp. 1–46, 2015.

[9] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2014, pp. 1–8.

[10] P. J. Kindermans, K. Schütt, K.-R. Müller, and S. Dahne, “Investi-
gating the influence of noise and distractors on the interpretation of neural networks,” 2016, arXiv:1611.07270.[Online]. Available: http://arxiv.org/abs/1611.07270

[11] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” Int. J. Comput. Vis., vol. 128, no. 2, pp. 336–359, Feb. 2020, doi: 10.1007/s11263-019-01228-7.

[12] D. Bahrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, “How to explain individual classification decisions,” J. Mach. Learn. Res., vol. 11, no. 6, pp. 1803–1831, 2010.

[13] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, “Not just a black box: Learning important features through propagating activation differences,” 2016, arXiv:1605.01713.[Online]. Available: http://arxiv.org/abs/1605.01713

[14] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” in Proc. Int. Conf. Mach. Learn. (ICML), 2017, pp. 1–9.[Online]. Available: http://proceedings.mlr.press/v70/shrikumar17a.html

[15] G. Montavon, W. Samek, and K.-R. Müller, “Methods for interpreting and understanding deep neural networks,” Digit. Signal Process., vol. 73, pp. 1–15, Feb. 2018, doi: 10.1016/j.dsp.2017.10.011.

[16] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio, “On the number of linear regions of deep neural networks,” in Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS), 2014, pp. 2924–2932.[Online]. Available: http://papers.nips.cc/paper/5422-on-the-number-of-linear-regions-of-deep-neural-networks

[17] R. Pascanu, G. Montúfar, and Y. Bengio, “On the number of response regions of deep feed forward networks with piece-wise linear activations,” 2014, arXiv:1312.6098.[Online]. Available: https://arxiv.org/abs/1312.6098

[18] J. H. Jung, Y. Shin, and Y. Kwon, “A metric to measure contribution of nodes in neural networks,” in Proc. IEEE Symp. Ser. Comput. Intell. (SSCI), Dec. 2019, pp. 1508–1515.

[19] D. M. J. de Smith, Statistical Analysis Handbook, Edinburgh, Scotland: The Winchelsea Press, Drumlin Security Ltd, 2018.

[20] C. Heumann and M. S. Shalahb, Introduction to Statistics and Data Analysis. Cham, Switzerland: Springer, 2016.

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Dec. 1998.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2019, pp. 84–90.