MARGINAL PROBABILITY EXPLANATION: A SALIENCY MAP WITH CLOSED-LOOP VALIDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work, we propose a saliency map with pixel-level resolution, called Marginal Probability Explanation (MPE), for a black-box classifier. MPE visualizes the contribution of each input dimension to the classifier by calculating marginal probabilities when only one dimension is considered. Marginal probabilities are estimated using Monte Carlo by sampling from the training dataset. Based on MPE, we propose typical samples, i.e. samples that maximize their marginal probability in every input dimension. We verify that our proposed MPE is meaningful through closed-loop validation experiments, where replacing a few pixels with the lowest marginal probability with pixels in the typical sample "corrects" the classification. Based on experiments, we found deep neural networks are probably still using pixel-level logics for image classification. Moreover, the critical pixels are not necessary related to the subject.

1 INTRODUCTION

Saliency representation is one of the most intuitive ways to understand how a classifier or deep neural network makes decisions. Saliency representation is usually a visualization of the importance of features in classification. In general, saliency representations can be divided into two categories, local or global, based on the perspective from which the interpreter views the classifer.

Local saliency representations focus on the local behavior of the classifier at samples, for example the local gradients or the local decision boundaries. Simonyan et al. (2013) calculates the gradient of the classifier at the sample as saliency map. Intuitively, the gradient of the classifier locally reflects the sensitivity of each feature. However, the gradients of all features together are too noisy to form semantic-level interpretability. Smilkov et al. (2017) smoothed the gradient by adding noise to the classifier. Sundararajan et al. (2017) defined two axioms, namely sensitivity and implementation invariance, and proposed a integrated gradient that satisfies the axioms. Both works try to add interpretability to the saliency representation while maintaining its mathematical meaning, i.e. the local gradient. On the other hand, the layer-wise relevance propagation (LRP) proposed in Bach et al. (2015) assumes that the output of the classifier is the sum of all feature contributions. Equivalently, LRP calculates the Taylor approximation of the classifier at a nearby sample on the decision boundary. Ribeiro et al. (2016) came up with the local interpretable model-agnostic explanations (LIME) which explains the classifier by approximating local decision boundary with a linear model. Although local saliency representation can provide pixel-level explanations for any given sample, the explanation is subjective. It is difficult verify whether of local saliency representations are accurate or meaningful. Furthermore, since local saliency focuses on the local behavior of classifiers, it is difficult to use local saliency representations to improve the classifier or augment the dataset.

Global saliency representations treat classifiers as black-box or partial black-box functions. Usually, they find the saliency map by perturbing the inputs or observing the output. Adler et al. (2018) estimated the indirect influence of each feature for black-box models by adding or removing features. Ancona et al. (2019) computed the significance of a feature based on shapely value considering all feature combinations with or without the feature. Although shapely valued based explanation is a more comprehensive estimation of the feature significance, the computation cost increases exponentially as the number of feature increases. Guan et al. (2019) extracted multilevel word attributes from the mutual information. Selvaraju et al. (2017) found the localization of the neural network activation is closely related to the object. Comparing with local saliency representations, global

saliency representations have better semantic-level interpretability. Global saliency representations have been used as feature selection, model improvement, etc. However, most of the global saliency representations can not provide pixel-level resolutions. Or, to extract pixel-level feature explanation without further assumption about the feature independency, global saliency representations require huge computation cost.

As discussed above, the meaningfulness or accurateness of saliency maps are usually evaluated by its interpretability. Specifically, saliency representations claim that they are meaningful by the phenomenon that their computed saliency are consistent with the locations of subjects in the image. However, the claim should be under the assumption that the classifier makes decisions based on the subjects instead of the background, which sounds obvious, but does not necessarily hold for neural networks.

To this end, we propose a global saliency representation, namely marginal probability explanation (MPE), which explains the effect of each input dimension to the classifier by calculating the marginal probability when only one dimension is considered. MPE ensures pixel-level resolution while maintaining excellent semantic-level interpretability. Most importantly, MPE can verified to be meaningful by closed-loop validation. In our experiments, we found that the misclassified samples can be corrected by applying an MPE and typical sample inspired modification. The experiment also implies that deep neural networks are still using pixel-level logics for image classification instead of semantic level. Most of the time neural networks make decisions based on only a few pixels in the object background.

2 MARGINAL PROBABILITY EXPLANATION

2.1 A GENERAL DEFINITION

Assume a classifier f takes input $\boldsymbol{x} \in \mathbb{R}^n$, which follows a probability density function (p.d.f) $p_X(\boldsymbol{x})$, and returns $\hat{\boldsymbol{y}} \in \mathbb{R}^c$. The *marginal probability explanation (MPE)* of f at a specific sample $\boldsymbol{z}$ is defined as $\boldsymbol{M}(f, \boldsymbol{z}) \in \mathbb{R}^{n \times c}$, where

$$M(f, \boldsymbol{z})_{ij} = \int f(\boldsymbol{x}|x_i = z_i)_j p_X(\boldsymbol{x}|x_i = z_i) dx_1 \dots dx_{i-1} dx_{i+1} dx_n \tag{1}$$

The classifier input $(\boldsymbol{x}|x_i = z_i)$ represents an arbitrary sample $\boldsymbol{x} \sim p_X(\boldsymbol{x})$ with $x_i = z_i$.

The classifier output of the sample $\boldsymbol{z}$ is commonly regarded as an estimation of the conditional probability that $\boldsymbol{z}$ belongs to each class given the dataset $\mathcal{D}$, i.e.,

$$\hat{y}_j = f(\boldsymbol{z})_j = Pr(y_j = 1|\boldsymbol{z}, \mathcal{D}), j = 1, 2, \dots, c \tag{2}$$

where $(\boldsymbol{z}, \boldsymbol{y})$ is a sample-label pair, c is the number of classes, and the label $\boldsymbol{y}$ is represented in one-hot. Thus, in equation 1, $M(f, \boldsymbol{z})_{ij}$ is actually the marginal probability of conditional event that $\boldsymbol{z}$ belongs to the class j by only looking at the i-th dimension of the input z_i, i.e.,

$$M(f, \boldsymbol{z})_{ij} = Pr(y_j = 1|z_i, \mathcal{D}) \tag{3}$$

MPE explains the effect of each input dimension to the classifier by calculating the marginal probability when only one dimension is considered. It interprets the classifier from a global aspect of the view while makes no assumption on the independency of the input dimension and no assumption on the classifier.

2.2 MONTE CARLO MARGINAL PROBABILITY EXPLANATION

In most of the applications, a direct calculation for $\boldsymbol{M}(f, \boldsymbol{z})$ is challenging, since the integration dimension can be very high and the probability $p_X(\boldsymbol{x})$ is also unknown. But we can approximate it using the Monte Carlo method. A good sampling dataset is the original training dataset. We assume the samples in the training dataset is independent identical distributed (i.i.d.) and sufficient to reflect the sample distribution. Although the training dataset reflecting the sample distribution can be a strong assumption, at least the classifier is trained under the same assumption. We consider the training dataset a better sampling dataset than synthetic datasets since its sampling can be assumed to follow the distribution of the input variable. Thus, we define the Monte Carlo MPE as following.

The *Monte Carlo marginal probability explanation (MCMPE)* of a classifier f, whose training dataset is $\mathcal{D}$, at the sample $\boldsymbol{z}$ is $\hat{\boldsymbol{M}}(f, \boldsymbol{z}) \in \mathbb{R}^{n \times c}$, where

$$\hat{M}(f, \boldsymbol{z})_{ij} = \text{avg}(f(\boldsymbol{x}|x_i = z_i)_j), \forall(\boldsymbol{x}|x_i = z_i) \in \mathcal{D} \tag{4}$$

MCMPE can be applied to explain any classifier which takes discrete input with computational cost $O(|\mathcal{D}|nc)$, where $|\mathcal{D}|$ is the size of the training dataset, n is the dimension of the input, and c is the number of classes.

2.3 TYPICAL SAMPLE

Denote the maximum entropy probability distribution of the classifier f given its marginal probabilities $M(f, \boldsymbol{v})_{ij}$ as

$$q(\boldsymbol{v})_j := \prod_{i=1}^{n} M(f, \boldsymbol{v})_{ij} \tag{5}$$

where $\boldsymbol{v} \in \mathbb{R}^n$. The maximum entropy probability distribution has entropy that at least as great as that of all other distributions with the same marginal distributions. It considers the contributions of all inputs independently and works as the least-information default. We define *typical sample* as the sample who maximize $q(\boldsymbol{v})_j$, i.e., $\boldsymbol{V}^*_{\cdot j} = \arg\max_{\boldsymbol{v}} q(\boldsymbol{v})_j$, $\boldsymbol{V}^* \in \mathbb{R}^{n \times c}$. As the input value for each dimension can be chosen arbitrarily, we can choose the value to maximize each corresponding marginal probability, i.e., $V^*_{ij} = \arg\max_{v_i} M(f, \boldsymbol{v})_{ij}$. Thus, the typical sample demonstrates the best value selection of each input dimension for each class. However the classifier doesn't necessarily categorize typical samples into the corresponding classes, even though most of time it does.

Similarly, the typical sample can be estimated through Monte Carlo by

$$\hat{V}^*_{ij} = \arg\max_{v_i} \text{avg}(f(\boldsymbol{x}|x_i = v_i)_j), \forall(\boldsymbol{x}|x_i = v_i) \in \mathcal{D} \tag{6}$$

3 CLOSED-LOOP VALIDATION OF MCMPE

3.1 MCMPE ON MNIST AND A CLOSED-LOOP VALIDATION EXAMPLE

We first demonstrate MCMPE with a simple 2-layers convolution neural network (CNN) on MINST dataset. The CNN is trained until it has training accuracy 98.74%, validation accuracy 99.20%, and test accuracy 99.22%. Figure 1(a) shows the first misclassified sample in the test dataset. The sample has true label "2" but misclassified as "7".

In Figure 1(c), MCMPE is applied to explain the misclassification. Each subplot in Figure 1(c) represents the marginal probability of the sample belongs to each class with the lighter color the higher probability. For instance, the first subplot represents the marginal probability of the sample belongs to the class "0" given each pixel of the sample. The dark green circle with its shape similar as a "0" is telling that these pixels do not support classifying the sample as "0". In the original sample, the corresponding area has dark pixels. It means that in order to be classified as "0", these pixels should preferably not be dark. The center area and the bottom area of the first subplot in Figure 1(c) have the darkest pixels. These pixels oppose the "0" classification. While in the original sample, these pixels are the bright part of the digit, which are less possible to happen for digit "0".

In the 3rd subplot of Figure 1(c), MCMPE suggests that the sample is not being classified as "2" because of the bottom pixels. But in the 8th subplot, these pixels highly supports the class "7". Both the facts can result the misclassification. To verity the result of MCMPE, we remove some of the bottom pixels which do not support the correct classification. The modified digit is shown in Figure 1(b) and its MCMPE of the class "2" and "7" are shown in (d) and (e). The sample is correctly classified back to the class "2" after those non-supportive pixels were removed.

MCMPE is also tested on the 2nd, 3rd, and 4th misclassified samples of the MNIST test dataset. In the first row of Figure 2, the sample is misclassified as "0" but has true label "6". Intuitively, we may think it was caused by the big circle in the digit. However, MCMPE suggests that the misclassification is resulted from the bottom-left corner of the digit. After several pixels removed, the digit can be correctly classified. Similar phenomena happens to other samples. The sample in the second row has true label "6" but classified as "5". MCMPE finds the reason in the upper-right corner. The third misclassified sample is caused by not wide enough on the bottom part.

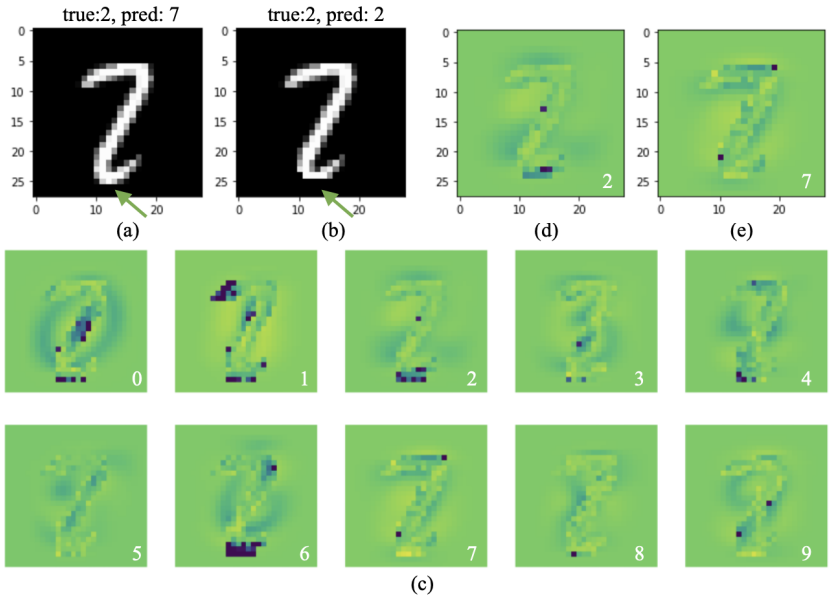

Figure 1: MCMPE for a misclassified MNIST test sample. (a) the sample has true label "2" but misclassified as "7"; (b) the classification is corrected after the sample being modification based on MCMPE; (c) MCMPE for each class of the misclassified sample; (d) MCMPE for class "2" of the modified sample; (e) MCMPE for class "7" of the modified sample. For better visualization, we use logarithm for MCMPE and clip from -10 to 0.

3.2 CLOSED-LOOP VALIDATION

In the above example, we show that MPE is a pixel-level saliency map with excellent interpretability for black-box classifiers. Most importantly, as demonstrated above, when the sample is modify based on its MPE, we can actually correct its prediction. Thus, MPE is a saliency representation which can be closed-loop validated. We specify the closed-loop validation as following.

Assume $\boldsymbol{M}(\boldsymbol{z})$ is a saliency representation of the classifier f, $\boldsymbol{M}$ can be *closed-loop validated* if there exists a saliency representation inspired binary mask $\boldsymbol{m_M}$ with the same size as $\boldsymbol{z}$, an operator ϕ, and a target class $j \neq f(\boldsymbol{z})$, such that

$$\arg\max f(\boldsymbol{m_M} \odot \phi(\boldsymbol{z}) + (1 - \boldsymbol{m_M}) \odot \boldsymbol{z}) = j, \ \|\boldsymbol{m_M}\|_1 \leq \varepsilon \tag{7}$$

Equation 7 is similar as the representation of adversarial attacks, but there are a few differences. We disturb the original sample through a mask since saliency maps are representations of the contributions for each input dimension. As the perturbation is only induced at the saliency map inspired mask, it is quantified by L1 norm of the mask. Some examples of the operator ϕ can be

$$\phi(\boldsymbol{z})_i = \begin{cases} 0, & \text{constant,} \\ \mathcal{U}(0,1), & \text{uniformly distributed,} \\ V^*_{ij}, & \text{typical sample} \end{cases} \tag{8}$$

Many saliency maps claim that they are meaningful by the phenomenon that their computed saliency are consistent with the locations of subjects in the image. However, the claim should be under the assumption that the classifier makes decisions based on the subjects, which does not necessarily hold for neural networks. Therefore, closed-loop validation is important to verify that the saliency map is functioning objectively.

3.3 CLOSED-LOOP VALIDATION: TYPICAL SAMPLE INSPIRED CLASSIFICATION CORRECTION

As shown in Figure 1 and Figure 2, MPE can be used to find masks to modify samples and correct their classifications. Intuitively, if we adjust the value of the input dimension which has the smallest

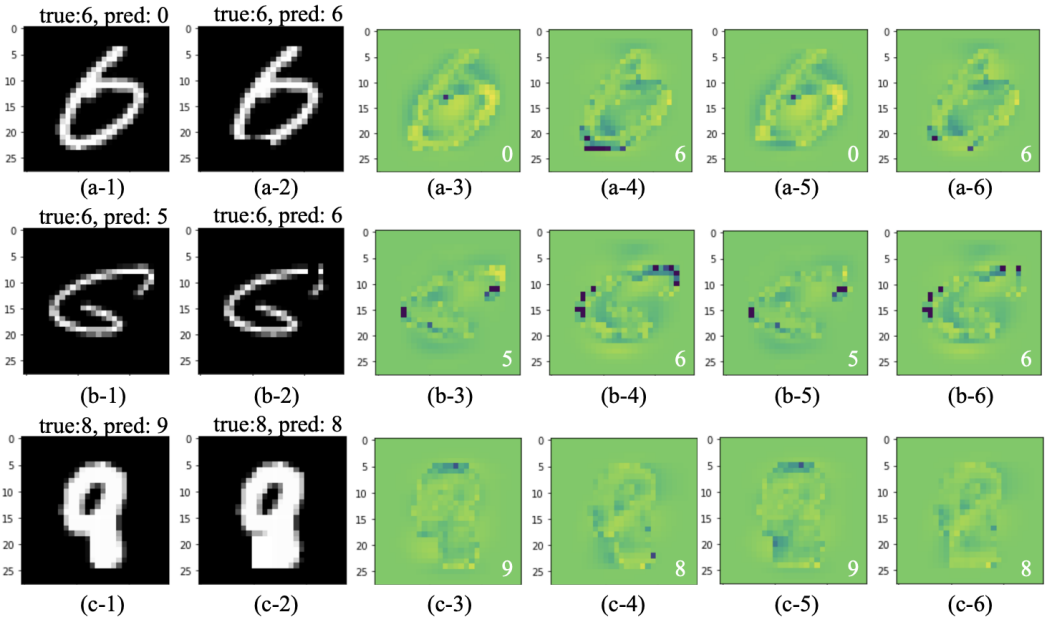

Figure 2: (a-1) A sample with label "6" but misclassified as "0"; (a-2) the classification is corrected after the sample being modification based on MCMPE; (a-3) MCMPE for the misclassified class of the original sample; (a-4) MCMPE for the true class of the original sample; (a-5) MCMPE for the misclassified class of the modified sample; (a-5) MCMPE for the true class of the modified sample; (b-1) A sample with label "6" but misclassified as "5"; (c-1) A sample with label "8" but misclassified as "9".

marginal probability, there will be a higher chance that the output of the model becomes higher. As the typical sample has the highest marginal probabilities in every input dimension, replacing some of the input values with the values in the typical sample may correct the classification. Thus, the typical sample inspired classification correction can be formulated as

$$\arg\max f(\boldsymbol{m}_M \odot \boldsymbol{V}^*_{\cdot j} + (1 - \boldsymbol{m}_M) \odot \boldsymbol{z}) = j, \ \|\boldsymbol{m}_M\|_1 \leq \varepsilon \tag{9}$$

where $j = \arg\max \boldsymbol{y}$ and $\boldsymbol{m}_M$ masks the input dimensions except the ones with the smallest MPEs.

4 EXPERIMENTS

4.1 TYPICAL SAMPLES

The typical samples for MNIST dataset and the 2-layers CNN which the dataset is trained on are shown in Figure 3. To avoid biased Monte Carlo estimation caused by insufficient samples, the best value for each input dimension is selected only if there are at least 200 samples for the value. Although, typical samples have the maximum marginal probabilities, they are not necessarily being predicted into the corresponding classes. However for the MNIST and 2-layers CNN experiment, the typical samples have been predicted into the correct classes.

As the locations and patterns of the subjects in CIFAR dataset can be very different, typical samples of CIFAR do not look like the subjects, while typical samples of MNIST dataset look a lot like the subjects. As shown in Figure 4, we trained a 4-layers CNN and a ResNet50 on CIFAR, whose test accuracies are 80.19% and 86.02% respectively. The RestNet50 were trained with pixel-means subtracted. Although the neural network structure are different, the misclassified samples of 4-layers CNN and ResNet50 are more than 60% in common. From the perspective of typical samples, the 4-layers CNN and ResNet50 have almost the same best value selections for every input dimensions. Similar typical samples can indicate similarity structures of the two neural networks as typical samples are maximizing marginal probabilities of the neural network.

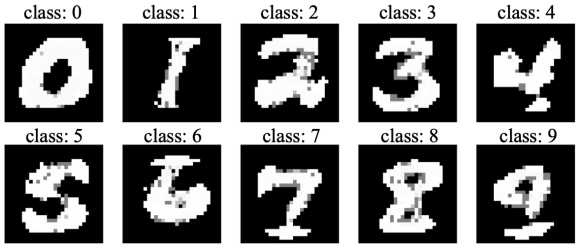

Figure 3: Typical samples for MNIST dataset and the 2-layers CNN.

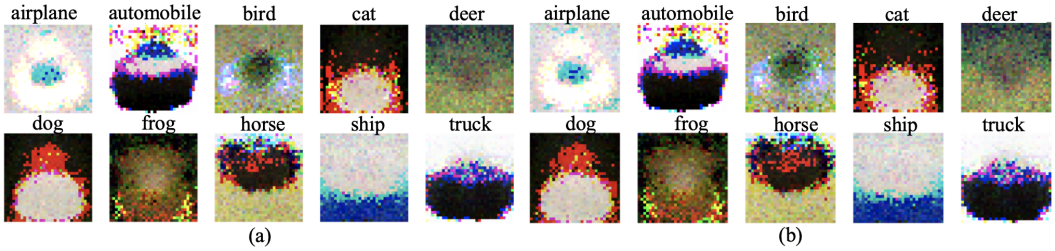

Figure 4: (a) Typical samples for CIFAR dataset and the 4-layers CNN; (b) typical samples for CIFAR dataset and ResNet50.

4.2 TYPICAL SAMPLE INSPIRED CLASSIFICATION CORRECTION

Typical sample inspired classification correction is tested on the 2-layers CNN and MNIST dataset. We set the maximum number of the pixel replacement as 10, i.e. $\epsilon = 10$ (in equation 9). Further, if a pixel replacement does not improve the prediction, it will not be replaced. In the test dataset, 84 out of 10,000 samples are misclassified by the trained CNN. Among them, more than half of the samples can be corrected through typical sample inspired classification correction. For comparison, if we randomly replace pixels with the pixels in typical samples, or if we select the pixels based on MPE but replace them with uniformly distributed random values, or if the selection and the replacement value are both uniformly random, only a few samples can be corrected.

As shown in Figure 5, the typical sample inspired classification correction, i.e. selecting pixels based on MPE and replacing values based on typical samples, corrects significantly more samples than the comparison methods. Moreover, the typical sample inspired classification correction takes fewer pixel replacements in average to achieve the correction. Figure 6 demonstrates the pixel replacement process of the typical sample inspired classification for a sample with true label "9" but predicted as "4". With one pixel adjusted from bright to dark and the another three pixels adjusted from dark to bright, the prediction probability of the sample for the class "9" gradually increased from .042 to .683.

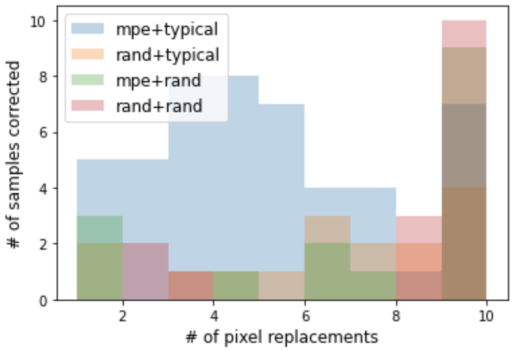

Figure 5: Histogram of the number of pixels used to correct the classification.

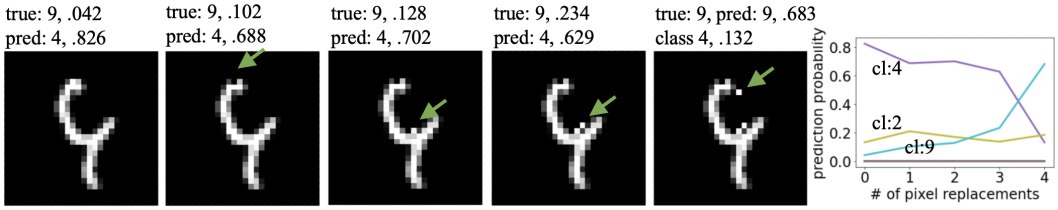

Figure 6: Pixel replacement process of the typical sample inspired classification for a sample with true label "9" but predicted as "4". The left most subplot is the original sample which predicted as "4" with probability .826. As four pixels were replaced by pixels in the typical sample, the prediction for class "4" decreases from .826 to .132. The prediction for class "9" increases from .042 to .682 while the predictions for all other classes remain in the similar levels.

We also tested typical sample inspired classification correction on CIFAR dataset with both 4-layers CNN and ResNet50. As shown in Table 1, finding pixels based on MPE and replacing pixels based on typical samples can significantly correct more samples. Although 4-layers CNN and ResNet50 has similar typical samples, ResNet50 allows 52.8% misclassification correction using typical sample inspired classification correction while 4-layers CNN allows 42.0% misclassification correction.

Pixel replacement process for CIFAR dataset is shown in Figure 7. With two pixels replaced, the classification is corrected from "ship" back to "airplane". These two critical pixels are not within the subject, instead one of them is within the sky while the other one is within the ground. The phenomenon that critical pixels are not associate with the subjects is common in the corrected samples. It implies that, neural networks, even deep neural networks, still make decisions based pixel-level logics. The critical pixels are not necessarily within the subject implies that the saliency maps are not necessary consistent with the subject, since classifiers may not making decisions based on the subject.

Table 1: Typical sample inspired classification correction on CIFAR dataset.

	mpe + typical	rand + typical	mpe + rand	rand + rand
4-layers cnn - cifar	**42.0**%	24.2%	33.6%	30.8%
resnet 50 - cifar	**52.8**%	31.0%	43.6%	43.2%

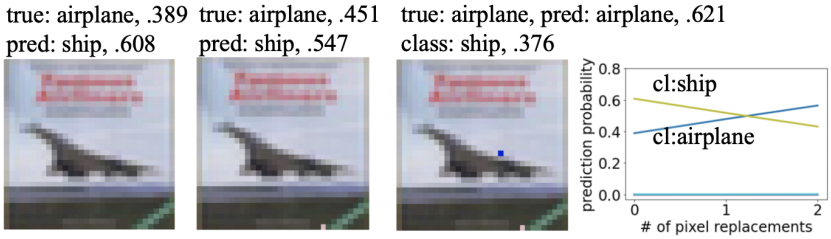

Figure 7: Pixel replacement process of the typical sample inspired classification for a sample with true label "airplane" but predicted as "ship". The prediction for class "airplane" increases from .389 to .621. The prediction for class "ship" decrease from .608 to .376, while the predictions for all other classes remain in the similar levels.

4.3 MPE AND ADVERSARIAL ATTACK

Although our experiments demonstrate classification "correction", MPE can also be used as adversarial attack. However, from the perspective of constructing adversarial examples, MPE is quite different from current adversarial attacks. Current adversarial attacks try to find a minimal perturbation that directs the sample to the other side of the classification manifold. What it cares the most is the local classification boundary surrounding the sample. MPE comes into play from a broader

perspective, i.e. marginal probabilities per pixel, which is more interpretable but less reliable for adversarial attack. Moreover, if there is a model which actually learns from semantic features, MPE shouldn't work at all, but the current CNN structures are not there yet.

One may relate MPE to the one-pixel adversarial attack method (Su et al. (2019)). Su et al. (2019) is a semi-blackbox attack, which requires the neural network output as feedback. MPE however requires the training dataset and the neural network output for the entire training dataset. We design a simple adversarial attack method based on MPE. The MPE of the RGB image is first calculated. We then replace the pixel with the highest MPE on any channel with salt or pepper noise until the classification changes or the threshold for the maximum iteration is reached. We tested the attack method on CIFAR-10 and the 4-layer CNN with a successful attack rate of about 60%. As shown in Figure 8, most of the time, the noise in adversarial samples is in the background not the object. Figure 9 shows the sample with true label "ship" is predicted as "airplane" after 10 pixels with the highest MCMPE were replaced with pepper noise. The classifier output for the class "ship" decreased as the output for the class "airplane" increase.

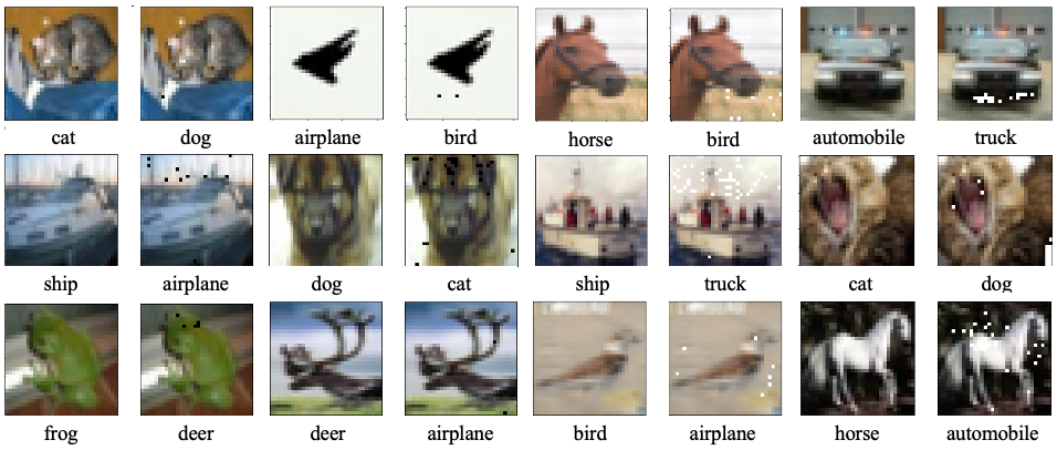

Figure 8: Adversarial attack method tested on CIFAR-10 and the 4-layer CNN. The maximum number of pixel change is set as 60. The attack method is tested on 78 sample with 47 of them successfully attacked.

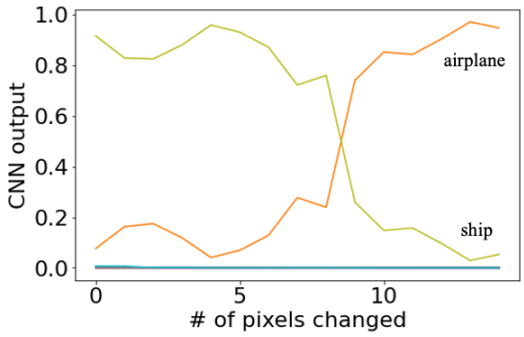

Figure 9: The classifier output varies with the number of top MCMPEs replaced by pepper noise.

5 DISCUSSION AND FUTURE WORK

We proposed a saliency map, marginal probability explanation (MPE), which visualize the saliency of each input dimension as the marginal probabilities when only one dimension is considered. We verified the meaningfulness of MPE with closed-loop validations. In the experiments, we found that those critical pixels which can be used to correct the classification are not necessarily within the

subject in the images. It implies that the interpretability of saliency maps should be verified with closed-loop validation instead of the consistency with the location of the subject.

As MPE and its closed-loop validations pointed out, the current CNN structures are probably still learning pixel-level features despite of the deep structure. In ideal, a classifier shouldn't make decisions simply based on one or two pixels. For an ideal classifier, its MPE shouldn't contain any marginal probability that is close to 0 or 1 and its closed-loop validation shouldn't work at all. Thus, MPE can be a good indicator of whether a classifier learns "deep" or not.

From the CIFAR-10 experiments, the object background in images seems to have a strong influence on decision-making. This is reasonable for neural networks since the background definitely contains category information. But in the human recognition process, decisions are usually made based on the object itself. The background context is only used when the object is ambiguous. Thus, MPE can be a guide for the models and techniques that try to focus the decision on the object itself, which including data augmentations, regularizations, etc.

We demonstrated MPE for computer vision tasks, but MPE can be easily generalized in other domains, such as natural language processing. With appropriate assumptions about the classifier, MPE may be able to be used for continuous input as well. In addition, as the neural network interpretation is quite subjective, closed-loop validation is a necessary process to validate the interpretation is at least correct.

REFERENCES

Philip Adler, Casey Falk, Sorelle A Friedler, Tionney Nix, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. Auditing black-box models for indirect influence. *Knowledge and Information Systems*, 54(1):95–122, 2018.

Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pp. 272–281. PMLR, 2019.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. Towards a deep and unified understanding of deep neural models in nlp. In *International conference on machine learning*, pp. 2454–2463. PMLR, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos 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.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.