SMOOT: SALIENCY GUIDED MASK OPTIMIZED ONLINE TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep Neural Networks are powerful tools for understanding complex patterns and making decisions. However, their black-box nature impedes a complete understanding of their inner workings. Saliency-Guided Training (SGT) methods try to highlight the prominent features in the model's training based on the output to alleviate this problem. These methods use back-propagation and modified gradients to guide the model toward the most relevant features while keeping the impact on the prediction accuracy negligible. SGT makes the model's final result more interpretable by masking input partially. In this way, considering the model's output, we can infer how each segment of the input affects the output. In the particular case of image as the input, masking is applied to the input pixels. However, the masking strategy and number of pixels which we mask, are considered as a hyperparameter. Appropriate setting of masking strategy can directly affect the model's training. In this paper, we focus on this issue and present our contribution. We propose a novel method to determine the optimal number of masked images based on input, accuracy, and model loss during the training. The strategy prevents information loss which leads to better accuracy values. Also, by integrating the model's performance in the strategy formula, we show that our model represents the salient features more meaningful. Our experimental results demonstrate a substantial improvement in both model accuracy and the prominence of saliency, thereby affirming the effectiveness of our proposed solution.

1 INTRODUCTION

The transformative influence of deep learning on our lives stems from its ability to learn from data and discover complex patterns in complex datasets. Deep Neural Networks (DNN) have enabled us to make more accurate predictions and make better decisions supported by data, serving as a catalyst for societal and technological evolution. Despite their unquestionable utility, the black-box nature of DNNs raises concerns about the trustworthiness and reliability of their outputs and the facets affecting their inference function. Consequently, there exists a significant interest to understand the behavior of these models and identify specific features they use and prioritize when producing an outcome." Generating a reliable explanation is especially important in sensitive domains such as medicine (Caruana et al. (2015)), neuroscience, finance, and autonomous driving (Li et al. (2018)). Not only do these explanations contribute to the critical understanding and trustworthiness of models, but they also aid in the process of model debugging (Zaidan & Eisner (2008); Li et al. (2018)) and model tuning. In light of the aforementioned, a substantial volume of scholarly research has been dedicated to the development of interpretability methods aimed at comprehending the internal mechanisms of DNNs (Bach et al. (2015); Kindermans et al. (2016); Smilkov et al. (2017); Singla et al. (2019); Singh & Lee (2017); LeCun et al. (2015)). A prevailing approach to interpreting model decisions involves identifying significant input features that highly influence the final classification decision (Baehrens et al. (2010); Singla et al. (2019); Smilkov et al. (2017); Zeiler & Fergus (2014); Selvaraju et al. (2017)). Commonly referred to as saliency maps, these methods typically employ gradient calculations to assign an importance score to individual features, thus reflecting their impacts on the model's prediction (Selvaraju et al. (2017); Shrikumar et al. (2017)).

Saliency maps can be unclear due to noise or distracting elements, which makes them harder to comprehend and less accurate. To address this issue, (Singh & Lee (2017)) proposed explanation methods that leverage higher-order backward gradients to give insight into the saliency maps. Also, (Kindermans et al. (2016)) benefits from multiple gradient calculations. An example is the SmoothGrad technique, which mitigates saliency noise by repetitively adding noise to the input and subsequently averaging the resulting saliency maps for each input (Singla et al. (2019)). Other techniques like integrated gradients(Smilkov et al. (2017)), DeepLIFT(Selvaraju et al. (2017)), and Layer-wise Relevance Propagation(Bach et al. (2015)) modify the backpropagation through a different gradient function (Ancona et al. (2017)). However, these methods' effectiveness is intrinsically tied to their reliability and stability (Adebayo et al. (2018); Kindermans et al. (2016)). If saliency maps change drastically for slight perturbations in the input or model, their trustworthiness can be severely compromised (Ghorbani et al. (2019)). Thus, in developing novel interpretability techniques, it is imperative to establish robust and comprehensive sanity checks to ensure their validity and (Adebayo et al. (2018); Thorne et al. (2018)). Furthermore, the quality of explanations generated by these methods can vary significantly depending on the data type (images, text, time series, etc.) and the model architecture (CNN, Recurrent Neural Networks, Transformer-based models, etc.). Hence, it's crucial to adapt and develop new interpretation techniques considering these factors (Ismail et al. (2020); Sundararajan et al. (2017)). Moreover, the quest for better interpretability extends beyond understanding individual predictions. It's about deciphering the learned representations and decision-making logic of the model as a whole (Hooker et al. (2019); Ross et al. (2017)). Neural network distillation into interpretable models like soft decision trees has been studied as a means to improve interoperability(Frosst & Hinton (2017)).

2 Related Works

Interpretability in machine learning refers to the ability to understand a learning model's decisions and actions and explain it in human-comprehensible terms(Chakraborty et al. (2017)). This research paper focuses on the advancement and extension of existing gradient-based methods that serve to define the behavior of models. Such methods have demonstrated their value in facilitating knowledge transfer and enabling the post-hoc interpretation of models. Building upon prior works in this domain, we aim to enhance the effectiveness and applicability of gradient-based approaches in understanding and interpreting model behavior while improving the model's generalization by selecting the most robust features during model training. In this section, we review the relevant prior-art literature on interpretability and saliency-guided training.

2.1 Interpretability

Previous research has explored different approaches to enhance the performance and interpretability of neural networks. Perkins et al. proposed a grafting method for feature selection, by incrementally expanding the feature set while training a prediction model using gradient descent. This technique accelerates the regularized learning process, making it suitable for large-scale applications (Perkins et al. (2003)). Frosst et al. introduced a method for yielding a soft decision tree through stochastic gradient descent, leveraging neural network predictions (Frosst & Hinton (2017)). Young et al. created a benchmark dataset with annotated "rationales" provided by humans, allowing for the measurement of model justifications against human justifications (DeYoung et al. (2019)). In (Ghaeini et al. (2019)), Ghaeini et al. proposed saliency learning aiming to train models that make predictions with explanations that align with that of the ground truth. Ross et al. developed a method to effectively explain and regularize differentiable models by penalizing input gradients based on expert annotations in an unsupervised manner (Ross et al. (2017)). Wang et al. introduced a solution that emphasizes class discrimination as a fundamental component in training CNNs for image classification, improving model discriminability and reducing visual confusion (Wang et al. (2019)). (DeVries & Taylor (2017)) demonstrated the cutout regularization technique to enhance CNN's robustness and performance. Behrens et al. proposed a method to explain local decisions made by arbitrary classification algorithms, utilizing local gradients as estimated explanations (Baehrens et al. (2010)).

2.2 Saliency Guided Training

In the saliency-guided training, Ismail et al. (Ismail et al. (2021)) introduce a new algorithm incorporating interpretability to enhance models' accuracy and saliency. They established their algorithm based on employing gradients for saliency map detection. Gradients are representative of the degree of "importance" attributed to the input pixels.

Although gradient methods have a lot of positive characteristics when used with visual models, they frequently result in noisy pixel attributions in areas unrelated to the predicted class (Kapishnikov et al. (2021)). Potentially meaningless local fluctuations in partial derivatives may be the cause of the noise observed in a sensitivity map (Smilkov et al. (2017)). The gradient of the model, when trained using a common method based on empirical risk minimization (ERM), may change drastically in response to minor input perturbations. Low-level features contain object position information but are intermingled with noises like backdrops, whereas high-level features display rich semantic information but lack object position information (Liu et al. (2022)). In order to accomplish saliency map extraction using low-level features like color and texture., downstream algorithms need more precise criteria. This is where the saliency guidance training approach comes in. Saliency-guided training aims to decrease the gradient values and maintain the model performance. This approach reduces the impact of irrelevant features on the outcome of the model as the gradients get closer to zero. By incorporating saliency guidance into the training procedure, the model can focus on the most significant aspects of the data, potentially leading to improved performance and generalization.

Algorithm 1 describes the SGT process which uses saliency information in training a neural network model f_θ. In this algorithm $\mathcal{D}_{\mathcal{KL}}(p\|q)$ is the KL divergence between probability distributions p and q. the $\mathcal{D}_{\mathcal{KL}}$ quantifies the difference between the original output distribution $f_\theta(X)$ and the modified output distribution $f_\theta(\widetilde{X})$. The $M_k(I, X)$ is the masking function that removes the bottom k features from the input data X, based on the sorted index I representing the importance of features according to their gradients. The $\widetilde{X}$ is the input data with the least important k features masked out. It is obtained by applying the masking operation $M_k(I, X)$. The L_i is the combined loss function used for training. It includes two terms: the standard loss term $\mathcal{L}(f_\theta(X), y)$ that measures the model's performance on the original input X with corresponding labels y, and a regularization term involving the KL divergence to encourage similarity between the output distributions of X and $\widetilde{X}$.

Algorithm 1 Saliency Guided Training (Original)

```
Training samples $X$, number of features to be masked $k$, learning rate $\tau$, hyperparameter $\lambda$
Initialize $f_\theta$ {Preload or randomize for new training}
for $i = 1$ to epochs do
    for $i = 1$ to epochs do
        {Calculate the sorted index $I$ for the gradient of output w.r.t the input.}
        $I = \text{sort}(\nabla_X f_{\theta_i}(X))$
        {Mask the bottom $k$ features of the original input.}
        $\widetilde{X} = M_k(I, X)$
        {Compute the loss function with regularization term.}
        $L_i = \mathcal{L}(f_{\theta_i}(X), y) + \lambda \mathcal{D}_{\mathcal{KL}}(f_{\theta_i}(X) \| f_{\theta_i}(\widetilde{X}))$
        {Update network parameters using the gradient.}
        $f_{\theta_{i+1}} = f_{\theta_i} - \tau \nabla_{\theta_i} L_i$
    end
end
```

2.3 Motivation and Problem Statement

In Algorithm 1, the number of masked features is determined using parameter k. Authors in Ismail et al. (2021), although noted that parameter k may affect the SGT optimization, have assumed this value constant. The solution in this paper was motivated after we investigated the impact of k and realized that the best choice of k is input image dependent.

In our motivational experiment leading to the solution proposed in this paper, we first used backpropagation to compute the gradient value for all input pixels. We then sorted the pixels based on the gradient value. We then started masking pixels with the highest gradient. The highest gradient pixels are expected to be the most important input features affecting the decision of the learning model. We expected to see a monotonically reducing accuracy curve as we masked additional input features. Interestingly, we observed a different behavior for some input images, in which the accuracy of the model started increasing by masking the high-value gradients initially, and then after reaching a max value, it started reducing. This behavior is captured in both images shown in Figure 1. The orange curve represents the behavior of the majority of images, where by increasing the making percentage, the classification accuracy monotonically reduces, and the blue curve represents the exception minority where the peak accuracy is reached after some initial masking. Figure 1 also illustrates that the peak in accuracy in the exception group can happen before (left) or after (right) the 50% making point.

The next important question we attempted to investigate was the frequency at which such a scenario transpires. To answer this question we trained a two-layer CNN model on the CIFAR-10 dataset featuring a kernel size of 3 and a stride of 1, succeeded by two fully connected layers. For regularization, we introduced two dropout layers with rates set at 0.25 and 0.5. Our experiment indicated that a considerable portion of the images, 16% of test images in our specific experiment, fall into this category. This indicates that the observed behavior is not statistically negligible. This

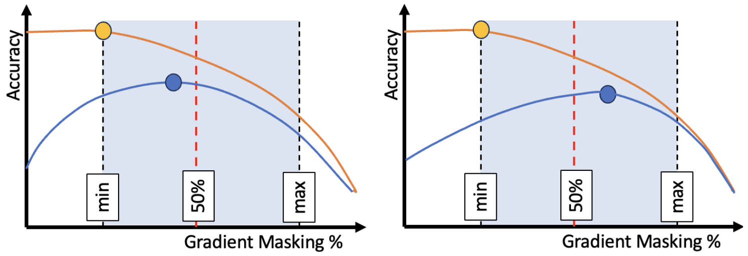

Figure 1: Illustration of how sorted gradient masking could result in a monotonic decrease in accuracy in majority of images (in orange), but an initial increase and then decrease in accuracy in some other images (in blue). The figure on the left captures the case where the peak accuracy in the exception images is reached before the 50% masking point, and the figure on the right captures the case where peak accuracy in the exception images is reached after 50% masking point.

observation also holds logical validity. The dimensions and orientation of an object within an image impact the quantity of input pixels integrated into the input feature, thereby triggering activation of the input image. This indicates that the optimal value for k is contingent upon the specific image. Additionally, when examining the saliency map of an input image, it is frequently noted that there are prominent pixels within the saliency map that lie outside or deviate from the area of interest within the image. This suggests that the model is allocating attention to features that hold no significant relevance for its classification objective. This observation likely elucidates why the removal of certain pixels initially leads to an increase in model accuracy, until the authentic features are discerned, ultimately resulting in a decline in accuracy. Stemming from this exploratory experimentation, we postulated that the parameter k in Algorithm 1 assumes a pivotal role as a hyperparameter, enabling the model to learn better features, while also enhancing the alignment between the saliency map and the location of the object of interest. Given this motivational background this paper addresses the following problem statement:

Problem Statement: Given a learning model f_θ and data $\{(X_i, y_i)\}_{i=1}^{n}$, formulate a saliency-guided training solution that optimizes the number of masked input pixels (k) based on saliency metrics and adjust model parameters θ to enhance the model accuracy and saliency map's fidelity.

3 METHODOLOGY

To enhance the model's generalization and the accuracy of the saliency map by focusing on a broader, more crucial set of features, as well as to minimize noise by teaching the model to ignore irrelevant features for input classification, we introduce an upgraded Saliency Guided Mask Optimized Online Training (SMOOT) approach. In this approach, the hyperparameter k specifies the number of masked pixels.

In the context of the Algorithm 1, the authors fixed parameter K. Their proposal advocates for configuring this parameter to encompass 50% of the pixels. To illustrate, when applied to images of dimensions 28*28, this parameter is set at 392 pixels. As explained in the motivational and problem statement section of this paper, for some images, gradient masking initially increases the classification accuracy and then reduces it. As mentioned, and illustrated in Figure 1, the max accuracy could also be realized by dropping less (left image), or more (right images) than 50% of high gradient pixels. Given this observation, we propose starting with 50% gradient masking and modifying the Algorithm 1 to dynamically adjust the number of masked pixels in the direction where accuracy is maximized.

Our proposed solution, SMOOT, is outlined in Algorithm 2. In this approach, instead of a single hyperparameter k, we use a vector K_i, where $K_i(X)$ is the masking percentage of input image X in epoch i. The objective is to optimize model parameters by finding the best masking percentage for each input image to minimize the loss function. The algorithm starts by initializing model parameters f_θ and by assigning each value in K vector to 50%. We then adjust the masking percentage for each image in the direction that image accuracy increases. As discussed previously, there are two types of images. images where increasing the masking percentage monotonically decreases their classification accuracy. Let us denote this group of images as class I images. We also have the images where masking initially increases the accuracy and then results in a drop in accuracy. Let us denote these images as class II images.

For class I images, if we adjust the masking in the direction where accuracy increases, we always move towards the direction of reducing the masking percentage. We define a fixed threshold for minimum (min) and maximum (max) masking. In this case, for the class, I group, the masking percentage always shifts towards the min masking value, as as shown in Figure 1 stays there. For class II, however, the move is towards the direction of max accuracy. As shown in Figure 1, the move could be to the left or the right until the accuracy max (shown with blue dot) in Figure 1 is achieved.

Algorithm 2 SMOOT: Saliency Guided Mask Optimized Online Training

```
Training samples $X$, learning rate $\tau$, hyperparameter $\lambda$, hyperparameter $\alpha$, controls increase or decrease number of masking $\mu$
Initialize $f_\theta$ {Preload or randomize for new training}
Initialize $K$ {50% to be consistent with prior work}
for $i = 1$ to epochs do
    for $i = 1$ to epochs do
        {Get sorted index $I$ for the gradient of output w.r.t the input.}
        1. $I = \text{sort}(\nabla_X f_{\theta_i}(X))$
        {compute $\widetilde{X}$ as the image with bottom $k$ features of the original input masked.}
        2. $\widetilde{X} = M(i, K, I, X)$
        {Compute difference in softmax outputs when $softmax_i$ is ith highest softmax output}
        3. $\delta_1 = softmax_1(\widetilde{X}) - softmax_1(X)$
        4. $\delta_2 = \frac{1}{n-1}\sum_{i=2}^{i=n}(softmax_i(\widetilde{X}) - softmax_i(X))$
        5. $\delta = \alpha\delta_1 + (1-\alpha)\delta_2$
        {Find number of masking for next epoch}
        6. $K_{i+1}(X) = \max(K_{\min}, \min(K_{\max}, K_i + \lfloor \mu\delta \rfloor))$
        {Compute the loss function}
        7. $L_i = \mathcal{L}(f_{\theta_i}(X), y) + \lambda \mathcal{D_{KL}}(f_{\theta_i}(X) \| f_{\theta_i}(\widetilde{X}))$
        {Use the gradient to update network parameters}
        8. $f_{\theta_{i+1}} = f_{\theta_i} - \tau \nabla_{\theta_i} L_i$
    end
end
```

Let us define $softmax_i(X)$ to be the softmax output of image X in epoch i. Let's also define $\widetilde{X}$ to be the masked image X. In this case, the difference between the accuracy of top 1, top 5, or a combination of the 2 could be used to indicate if masking results in improvement or degradation in accuracy. The change in the top 1 accuracy is computed in line 5 of the SMOOT algorithm as:

$$\delta_1 = softmax_1(\widetilde{X}) - softmax_1(X) \tag{1}$$

and the change in the top 2 to top n is computed using

$$\delta_2 = \frac{1}{n-1}\sum_{i=2}^{i=n}(softmax_i(\widetilde{X}) - softmax_i(X)) \tag{2}$$

In this equation, for top 5 accuracy, n should be equal to 5. We then use a weighted representation of change in softmax value using the equation:

$$\delta = \alpha\delta_1 + (1-\alpha)\delta_2 \tag{3}$$

For the generated results, in the result section of this paper, we have used the $n = 5$ and $\alpha = 0.7$, placing more priority on improvement in top 1 accuracy. Given the change in the accuracy, we then change the masking percentage of the input image using the following equation:

$$K_{i+1}(X) = \max(K_{\min}, \min(K_{\max}, K_i + \lfloor \mu\delta \rfloor)) \tag{4}$$

In this equation, K_{min} and K_{max} are the min and max percentages allowed for masking. In our experiment $K_{min} = 20$ and $K_{max} = 80$. Finally, the weight "μ" is a hyperparameter that determines the speed at which the masking percentage changes. The loss of the model is computed similarly to the previous SGT using:

$$L_i = \mathcal{L}(f_{\theta_i}(X), y) + \lambda \mathcal{D_{KL}}(f_{\theta_i}(X) \| f_{\theta_i}(\widetilde{X})) \tag{5}$$

In which $\mathcal{L}$ is the cross entropy loss and $\mathcal{D_{KL}}$ is the KL divergence. The KL divergence or relative entropy is computed using:

$$\mathcal{D_{KL}}(f_{\theta_i}(X) \| f_{\theta_i}(\widetilde{X})) = \sum_{x \in X} f_{\theta_i}(X) log(\frac{f_{\theta_i}(\widetilde{X})}{f_{\theta_i}(X)}) \tag{6}$$

The KL divergence is computed based on the similarity of $f_{\theta_i}(X)$ to $f_{\theta_i}(\widetilde{X})$, first term representing the output for the original input image X, and second representing the output for masked image $\widetilde{X}$ using the $K_i(X)$ masked in the current epoch. Using this updated loss, the gradients are then used to update the network parameters as follows:

$$f_{\theta_{i+1}} = f_{\theta_i} - \tau \nabla_{\theta_i} L_i \tag{7}$$

The algorithm of our proposed solution, including all steps above is shown in Algorithm 2. The algorithm takes as input the training samples (X), the initial number of features to be masked (k), learning rate (τ), and a hyperparameter (λ). It starts by initializing the model parameters (f_θ). Then, for each epoch and mini-batch, the following steps are executed.

First, in line 1, the sorted index I corresponding to the gradient of the output concerning the input, denoted as $\nabla_X f_{\theta_i}(X)$, is found. In this algorithm, $sort(\cdot)$ is a sorting function. $sort(\nabla_X f_\theta(X))$ denotes the sorted gradient. In line 2, we generate a new masked image from the input image. $M(\cdot)$ represents the input masking function. $M(i, K, I, X)$ generate $X(i)$ with a mask distribution, removing the $K(i)$ lowest features as indicated by sorting vector I to generate $\widetilde{X}$. In lines 3-6, based on the contrast between the accuracy of the input and the masked input, the algorithm determines and applies the magnitude and direction of change in the masking percentage. In line 7 the loss function is computed by adding a weighted KL divergence to the standard loss term. The KL divergence is computed by comparing the output of the model when using original input X versus masked input Xm, using the mask value $K_i(X)$ generated in epoch i. The weighted KL divergence terms account for The saliency-guided regularization in the overall loss function in line 7. Finally, the network parameters are updated using gradient descent in line 8.

4 Experiments and Results

Our primary objective is to assess the performance of SMOOT relative to SGT and traditional models across different datasets. To this end, we evaluate the efficacy of our proposed solution by retraining models on the MNIST (LeCun et al. (2010)), Fashion MNIST (Xiao et al. (2017)), CIFAR-10, and CIFAR-100 (Krizhevsky et al. (2009)) datasets.

4.1 Model Architecture

For the MNIST and Fashion-MNIST Datasets, we employed a two-layer CNN with a kernel size of 3 and a stride of 1. The CNN layers were succeeded by two fully connected layers. Additionally, two dropout layers were incorporated with rates of 0.25 and 0.5. For the CIFAR datasets, we turned to the Tiny Transformer, adopting the original configurations: a minimalist 'tiny' dimension of $(L = 12, d = 192, h = 3)$. The pre-trained model for this setup was sourced from the Facebook research repository[1] and is based on the *deit* architecture, which was pre-trained on ImageNet (He et al. (2016)). As the concluding layer, we incorporated a 10-neuron classifier. Both architectures were trained on a single NVIDIA A100 GPU for a span of 100 epochs with batches of 256. Adadelta (Zeiler (2012)) served as our optimizer, operating at a learning rate of 1 for MNIST and Fashion-MNIST datasets and 0.001 for the CIFAR datasets. Throughout the training process, for each epoch, low gradient value pixels were substituted by random values within the spectrum of the remaining pixels. These gradients were computed using the Captum library (Kokhlikyan et al. (2020)).

In the context of our paper, with regard to the MNIST and Fashion MNIST datasets, which inherently exhibit lower complexity, it is appropriate to configure the hyperparameter (α) to a high value, such as 0.95. This strategic decision enables us to place a greater emphasis on the label. For the CIFAR-10 and CIFAR-100 datasets, we recommend setting the hyperparameter (α) to a substantial value, like 0.8. This choice effectively guides the model to allocate more attention to classes beyond the label. Because the SGT article (Ismail et al. (2021)) considered $K = 50\%$ of the total pixel count, in our paper, we also adopted an initial value of n equal to 50% of the total pixels. Specifically, the initial value of n was set to 392 for the MNIST dataset and 512 for CIFAR. We set the μ to 10 and λ to 1, following the approach described in SGT's paper (Ismail et al. (2021)). Table 1 displays the model architecture and hyperparameters utilized in our paper, which are based on the SGT.

4.2 Saliency Guided Training for Images

In the context of image classification using saliency, it is common to encounter redundant features that are not crucial for the model's prediction. Take the example of an object's background in an image, which occupies a significant portion but typically holds little relevance to the classification task.

[1] `https://github.com/facebookresearch/deit`

Dataset	Model	Initialize K	τ	α	λ
MNIST	CNN	392	1	95%	1
Fashion-MNIST	CNN	392	1	95%	1
CIFAR10	Transformer	512	0.001	80%	1
CIFAR100	Transformer	512	0.001	80%	1

Table 1: Model Architecture

When the model's attention is directed toward the object itself, it is desirable for the background gradient (representing most of the features) to be close to zero, indicating its diminished importance. Figure 2 illustrates a comparison between the saliency map generated using our approach, the SGT in (Ismail et al. (2021)), and Traditional training (no saliency-guided training). Figure 2 provides this comparison for images selected from the MNIST dataset and Fashion MNIST dataset.

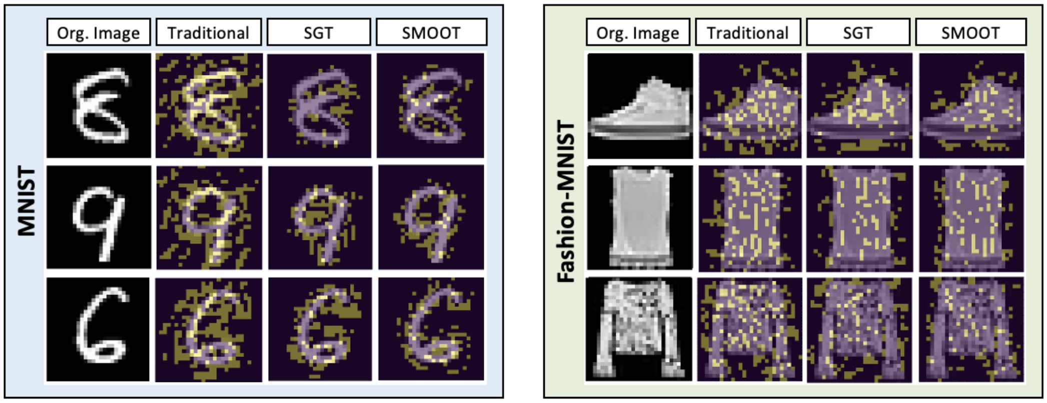

Figure 2: MNIST and Fashion-MNIST: The displayed images compare our model (SMOOT) with an SGT and Traditional model. For saliency, the best model employs a high gradient to focus on the most prominent pixels. In the case of MNIST and Fashion- MNIST, it prioritizes edge detection.

4.3 Model Accuracy Drop

In our study, we investigate the efficacy of our approach, SMOOT, against the prior art SGT (Ismail et al. (2021)) and conventional baseline training methods. We utilize various saliency techniques for this purpose, including modification-based evaluation (Petsiuk et al. (2018); Kindermans et al. (2016)). We evaluated the impact on model accuracy by ranking and eliminating features based on their saliency values. The experiment is conducted on datasets such as MNIST and Fashion MNIST, both of which have known uninformative feature distributions (e.g., black background). The results show that SMOOT leads to a steeper drop in accuracy compared to traditional training and SGT, regardless of the saliency method employed. This steeper drop indicates that our method is more effective in identifying and eliminating less informative features, resulting in a more refined model. However, it's important to emphasize that this experiment is most relevant for datasets with known uninformative features and might not be applicable to datasets with varying or unknown backgrounds. Because in datasets with varying or unknown backgrounds, the uninformative features may not be as consistent or easily distinguishable. This means that saliency methods might struggle to effectively rank and eliminate them, potentially leading to different results.

MNIST	# Test	Min(K)	Median(K)	Max(K)	Accuracy	AUC
Traditional	10K	0	0	0	99.40%	36.35
SGT	10K	392	392	392	99.35%	34.67
SMOOT	10K	234	388	544	99.40%	33.16

Table 2: MNIST: Traditional and Saliency Model Training Accuracy and saliency (A smaller AUC value indicates better performance in saliency)

Table 2 provides an overview of the Area Under accuracy drop Curve (AUC) and accuracy results on the MNIST dataset for different gradient-based training approaches. The table compares traditional training, training with a saliency-guided procedure, and our own method. It's noteworthy that a lower AUC value is indicative of superior performance, largely due to its representation of a more pronounced drop in accuracy. Upon careful analysis of the table, it is clear that SMOOT demonstrates superior accuracy and saliency when compared to the traditional and SGT approach.

Table 3 displays the outcomes of AUC and the achieved accuracy on the Fashion-MNIST dataset. Analyzing the table, it becomes clear that our model exhibits superior accuracy and saliency in comparison to both the SGT and traditional models, as indicated by the smaller AUC values.

Fashion-MNIST	**# Test**	**Min(K)**	**Median(K)**	**Max(K)**	**Accuracy**	**AUC**
Traditional	10K	0	0	0	93.60%	40.79
SGT	10K	392	392	392	93.35%	39.91
SMOOT	10K	223	372	576	93.65%	36.18

Table 3: Fashion-MNIST: Comparing Traditional and Saliency and SMOOT Model Training Accuracy and Saliency (Smaller AUC = Better Saliency)

Based on the changes in accuracy due to the number of masking, we present our findings in Figure 3 for MNIST and Fashion-MNIST datasets. These figures illustrate that our model exhibits a more significant drop in accuracy compared to both the traditional and SGT models, demonstrating its higher Salient in comparison to them.

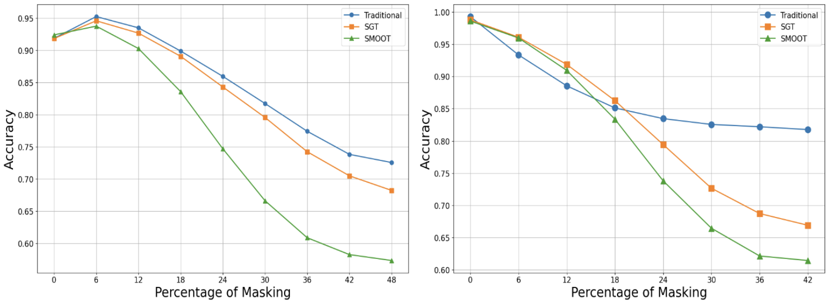

Figure 3: Comparing accuracy drop For MNIST and Fashion-MNIST using three training approaches: Standard cross-entropy based training, SGT (Ismail et al. (2021)), and Our proposed (SMOOT). A more pronounced decrease is indicative of superior performance.

4.4 Saliency Guided Mask Optimized Online Training for Transformers

The Transformer model represents a revolutionary computational paradigm in the realm of deep learning, demonstrating robust performance across an array of computer vision tasks. With its roots in natural language processing (NLP), the Transformer model(Vaswani et al. (2017)) has shown an unparalleled ability to capture long-range dependencies through the self-attention mechanism. Its related large-scale counterparts, such as BERT (Devlin et al. (2018)), GPT-3 (Tomsett et al. (2020)) and GPT-4 (Author1 et al. (2023)), have set new standards in harnessing powerful language representations from unlabeled textual data. This remarkable success of Transformer models in the NLP landscape has piqued the curiosity of the vision community, leading to its effective application in various computer vision tasks. These encompass areas like image recognition (Dosovitskiy et al. (2020); Touvron et al. (2021)), object detection (Carion et al. (2020)), and even image generation (Chen et al. (2021)). Within this transformative framework, the Tiny Transformer (Touvron et al. (2022))., stands out as a streamlined variant. This model is distinguished by its significantly reduced size, all while maintaining performance levels comparable to its standard Transformer counterparts. In Table 4 and 5, the implementation of the SGT method resulted in CIFAR10 and Cifar100 in a notable improvement in accuracy. Notably, through the application of the SMOOTH method and optimization of masked values, we were able to achieve even higher accuracy values, Furthermore, Figure 4 presents a comparison among traditional, SGT, and SMOOT models through the application of saliency maps to images.

CIFAR10	**# Test**	**Min(K)**	**Median(K)**	**Max(K)**	**Accuracy**
Traditional	10K	0	0	0	95.65%
SGT	10K	512	512	512	96.05%
SMOOT	10K	204	488	753	96.35%

Table 4: CIFAR10: The accuracy compares our model (SMOOT) with a Saliency-Guided Training (SGT) and Traditional model by using a transformer.

CIFAR100	# Test	Min(K)	Median(K)	Max(K)	Accuracy
Traditional	10K	0	0	0	75.75%
SGT	10K	512	512	512	78.10%
SMOOT	10K	362	432	682	79.65%

Table 5: CIFAR100: The accuracy compares our model(SMOOT) with a Saliency-Guided Training(SGT) and Traditional model by using transformer.

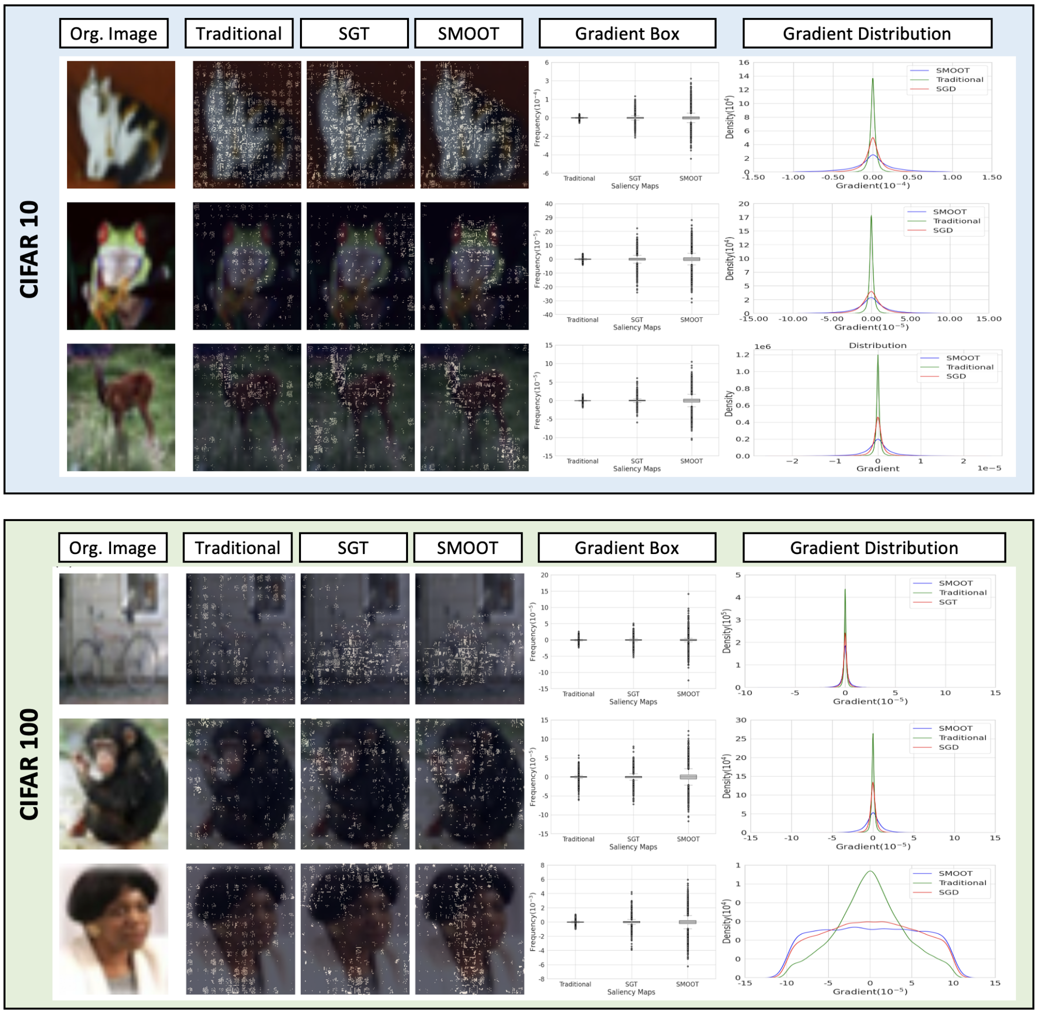

Figure 4: CIFAR10 and CIFAR100: The displayed image compares our model (SMOOT) with a Saliency-Guided Training (SGT) and Traditional model by using a transformer. For saliency, the best model employs a high gradient to focus on the most prominent pixels. Gradients approaching zero often signify uninformative features, whereas exceptionally large or exceedingly small gradient values tend to highlight the informativeness of these features.

5 Conclusion

In the framework of SGT, the hyperparameter k is of paramount importance as it represents the ”number of masking.” This hyperparameter is instrumental in the learning process by pinpointing the most relevant pixels in each image. Also, such identification proves vital for improving accuracy, especially in datasets with larger image dimensions. Therefore, careful optimization of k is crucial to ascertaining its optimal value. In this paper, we present a novel solution for input-based selection of k, illustrating that such optimization improves both the accuracy of the model and the fidelity of the saliency map across all tested benchmarks.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017.

Author1, Author2, and Author3. Gpt-4: Technical report. Technical report, OpenAI, San Francisco, CA, USA, July 2023.

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.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1721–1730, 2015.

Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuveer M Rao, et al. Interpretability of deep learning models: A survey of results. In *2017 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computed, scalable computing & communications, cloud & big data computing, Internet of people and smart city innovation (smartworld/SCALCOM/UIC/ATC/CBDcom/IOP/SCI)*, pp. 1–6. IEEE, 2017.

Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12299–12310, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.

Reza Ghaeini, Xiaoli Z Fern, Hamed Shahbazi, and Prasad Tadepalli. Saliency learning: Teaching the model where to pay attention. *arXiv preprint arXiv:1902.08649*, 2019.

Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3681–3688, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, and Soheil Feizi. Benchmarking deep learning interpretability in time series predictions. *Advances in neural information processing systems*, 33:6441–6452, 2020.

Aya Abdelsalam Ismail, Hector Corrada Bravo, and Soheil Feizi. Improving deep learning interpretability by saliency guided training. *Advances in Neural Information Processing Systems*, 34: 26726–26739, 2021.

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5050–5058, 2021.

Pieter-Jan Kindermans, Kristof Schütt, Klaus-Robert Müller, and Sven Dähne. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270*, 2016.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. att labs, 2010.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9215–9223, 2018.

Deyin Liu, Lin Wu, Farid Boussaid, and Mohammed Bennamoun. Improved and interpretable defense to transferred adversarial examples by jacobian norm with selective input gradient regularization. *arXiv preprint arXiv:2207.13036*, 2022.

Simon Perkins, Kevin Lacker, and James Theiler. Grafting: Fast, incremental feature selection by gradient descent in function space. *The Journal of Machine Learning Research*, 3:1333–1356, 2003.

Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.

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.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMLR, 2017.

Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE international conference on computer vision (ICCV)*, pp. 3544–3553. IEEE, 2017.

Sahil Singla, Eric Wallace, Shi Feng, and Soheil Feizi. Understanding impacts of high-order loss approximations and features in deep learning interpretation. In *International Conference on Machine Learning*, pp. 5848–5856. PMLR, 2019.

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

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

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

Richard Tomsett, Dan Harborne, Supriyo Chakraborty, Prudhvi Gurram, and Alun Preece. Sanity checks for saliency metrics. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6021–6029, 2020.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 32–42, 2021.

Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *European Conference on Computer Vision*, pp. 516–533. Springer, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Lezi Wang, Ziyan Wu, Srikrishna Karanam, Kuan-Chuan Peng, Rajat Vikram Singh, Bo Liu, and Dimitris N Metaxas. Sharpen focus: Learning with attention separability and consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 512–521, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Omar Zaidan and Jason Eisner. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 conference on Empirical methods in natural language processing*, pp. 31–40, 2008.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.