Explain Black-box Image Classifications Using Superpixel-based Interpretation

Yi Wei[1], Ming-Ching Chang[1], Yiming Ying[1], Ser Nam Lim[2], and Siwei Lyu[1]

[1]University at Albany, State University of New York, NY, USA
[2]GE Global Research Center, Niskayuna, NY, USA

***Abstract*—How to best understand and interpret the decisions of deep neural networks is a crucial topic, as the impact of intelligent deep network systems is prevalent in many applications. We propose a superpixel based method to interpret and explain the results of black-box deep networks in the widely-applied image classification tasks. We perform probabilistic prediction difference analysis upon one or more superpixels clustered from image pixels. Our method generates a superpixel score map visualization that can provide rich interpretation regarding image components. Such interpretation provides supportive/unsupportive likelihood of image regions upon the decisions performed by the black-box classifier. We compare our method against state-of-art pixelwise interpretation methods over the latest deep neural network classifiers on the ImageNet dataset. Results show that our method produces more consistent interpretations in less computation time. Our method also supports interactive interpretation, where users can acquire explanations on specified regions through a convenient interface for a prompt reaction.**

I. Introduction

Deep learning has achieved great successes in many areas, including computer vision and speech recognition. As research advances, state-of-the-art deep neural network (DNN), *e.g.* the convolutional neural network (CNN) generally comprises many layers [1], and their architecture is becoming more versatile [2], [3] and complex [4].

Although the DNNs have reached human-level performance on several specific tasks, real-world deployments and usage of these systems are still hindered due to a number of technical or non-technical risks and concerns. A major concern is the lack of interpretability of the DNN, which makes them less *trustworthy* for making critical decisions in tasks including disease diagnosis and autonomous driving. How to make end users to better understand these models is becoming critical — What problem is suitable for the DNN to apply? Can one justify and trust the decisions made by the DNN? These issues are in general much less addressed compared to the development of the DNNs themselves.

The DNN *interpretations* are a set of techniques that can explain the decisions or behaviors of a DNN. The interpretation can operate either in the *black-box* or the *white-box* manners. Black-box interpretation is *model-agnostic*, *i.e.*, the interpretation is only obtained based on the sole knowledge of the input/output of the model and does not rely on the knowledge of the model itself. On the contrary, white-box interpretation makes use of the knowledge of model structure, and assumes that the DNN parameters are known. Black-box interpretation is generally difficult to formulate, but favorable in practical applications, as the white-box assumption of knowledge regarding the exact parameters is hard to ensure. The DNN parameters are typically encoded. Due to CPU/GPU implementations and platform constraints, it is hard for end users to obtain exact values in the computation of gradients during the white-box interpretation.

This paper develops a fast black-box interpretation technique that can explain the decisions of a DNN image classifier. Our approach is *example-specific*, as we do not aim to explain or visualize a DNN model overall; instead, we provide visual interpretations over individual test images. The inference is performed on image regions, *i.e.* clusters of pixels specified in one or more *superpixels*. Adopting superpixels has several advantages: (1) Since superpixels better capture semantic contents, the interpretation is more consistent with the image contents and visually intuitive. (2) The inference computation is fast compared to pixelwise methods [5]. (3) It is easy for users to select a Region of Interests (ROI) using superpixels for fast interactive interpretation.

Our interpretation method calculates a significance score for each superpixel, which reflects whether or not it provides supportive information to the classifier regarding the classification. Such superpixel scores can be visualized as a heat map on top of the image, to show intuitive visualization of how individual image components (object parts, foreground/background *etc.*) affect the decisions made by the black-box image classifier. For example in Fig. 1, the bird is recognized as a macaw, and our interpretation suggests that the macaw's characteristic long tail is *supportive* (red) toward its identification from other kinds of birds. On the contrary, the body is *unsupportive* (blue) against such decision. Although this interpretation is counter-intuitive at a first glance, it can be understood as the bird body is less distinctive in separating macaw from other kinds of birds. Our interpretation shows that the highly nonlinear DNN decisions are often made based upon a few key distinctive regions.

Our black-box interpretation operates based on a superpixel-based probabilistic prediction difference analysis (PDA) [6]. The color invariant characteristic of the superpixels bring another advantage, that we can incorporate a *color histogram* based sampling scheme (§III-B) to improve both the accuracy and stability of the PDA inference. Our method differs from

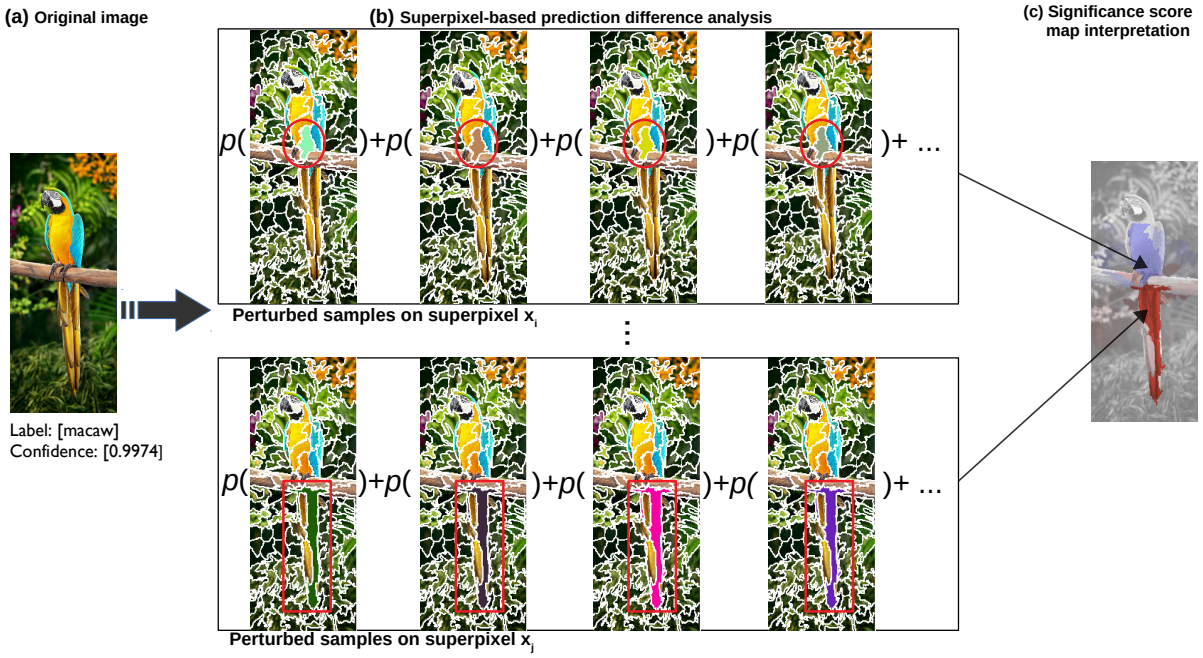


Fig. 1: **Superpixel-based prediction difference analysis (PDA).** (a) The input image with classification results. (b) PDA is performed on each superpixel of the input image. We illustrate the process of inferencing the significance score for superpixel x_i (feet) and x_j (tail) with their perturbed samples using marginalization. (c) The resulting map visualizes the *supportive* (red) *vs. unsupportive* (blue) likelihood (significance score) of the components toward the classification.

existing interpretation works in two main aspects. (1) Existing black-box interpreters [7], [8] that operate on image components can only highlight components that is supportive to the classification, however without a quantitative score (*i.e.* 0 or 1 *vs.* likelihood). (2) The pixelwise method of [5] is slow and the interpretation results tend to be noisy.

This paper has four main contributions. First, our method is model-agnostic in providing a significance score visualization that intuitively illustrates how each image component contributes to the classification decision. Second, our superpixel-based inference improves the consistency and computational efficiency of the black-box interpretation. Third, the superpixel formulation enables users to quickly specify a ROI for interactive interpretation. Fourth, our method is example-specific which brings additional justifications for developers and users to evaluate: (1) how the classifier is *trustworthy* in terms of how it responses in specific test cases, (2) how robustness of the classifier can be improved by adding specific training samples that are reflected from the interpretations.

II. Related Work

Existing efforts on the understanding, explanation, and interpretation of DNNs can be organized into the following categories: (1) *self-explainable models*, (2) deep feature visualization, example-specific (3) white-box and (4) black-box interpretation methods.

Self-explainable models contain explicit structures or explainable rules that can be easily interpreted. Si and Zhu [9] use hierarchical reconfigurable and-or image templates with simple rules to synthesize images for object detection. The interpretable classifiers of [10] provide a generative framework for case-based reasoning. Zhang *et al.* [11] improve the interpretability of the CNN by modifying the higher convolutional layers to represent specific object parts.

Deep feature visualization methods aim to visualize the learned features inside the CNN layers, *i.e.*, to understand what information the convolutional filters carry inside a CNN. Several works visualize the learned feature maps that maximize the activation of a single unit in the CNN [12], [13]. These methods can generate intriguing images that, to some extent, represent the characteristics of the CNN feature map. However, the meanings of this visualization are still obscure for the explanation of how the CNN classifier works for a specific input image.

Example-specific white-box interpretation methods aim to interpret or visualize how the CNN classifier responds to individual components of the given image *w.r.t.* the classification results. These methods depend on the knowledge of the network architecture and parameters. Simonyan *et al.* [14] propose a gradient based method to compute a saliency map for the input image, which represents the significance of each pixel toward the prediction. Montavon *et al.* [15] apply the Taylor expansion to decompose the classification prediction into the significance of input elements.

Example-specific black-box interpretation methods do not require the knowledge of the model structure or pa-

rameters, while the resulting interpretation can be still as informative as the white-box methods. Ribeiro *et al.* [7] exploit a linear model to estimate the local behavior of the classifier through perturbed samples. Although this method can highlight important image components that affect the decision results, the resulting interpretations is usually unstable, due to the weak assumption of a linear model and the heuristic perturbation method (see §IV for comparisons). Recently, Fong *et al.* [8] interpret a black-box network classification by learning a "meaningful" perturbation mask of the given image. This method infers the mask using the gradient of the classifier, which is hard to obtain in practice. Zintgraf *et al.* [5] apply pixelwise PDA for interpretation. This per-pixel scheme is slow, as it can take $\sim$ 30 minutes on the Titan X GPU to calculate an interpretable evidence map, and the interpretation tends to be noisy and inconsistent. §IV will provide detailed comparisons of these methods to our method.

III. Method

Our superpixel-based interpretation is inspired by Robnik-Šikonja & Kononenko's probabilistic *prediction difference analysis* (PDA) [6]. PDA provides a means to measure how the decision of a black-box classifier changes, while the effect of an individual component of the input is taken out using marginalization. A significance score for each input component can be calculated to represent its relevance (supportive or unsupportive likelihood) to the final classification decision.

A. Probabilistic Prediction Difference Analysis (PDA)

We consider the decomposition of the input image X into a set of superpixels (or pixels) *i.e.*, $X = \{x_i\}$, where i is the index of the superpixels/pixels in the image.[1] We denote the image X excluding a specific superpixel x_i as $X\backslash_i$, *i.e.*, $X\backslash_i = \{X - x_i\}$. For the given black-box image classifier $\mathcal{F}$ operating on X, the classification score of a specific class c is then $p(c|X)$. The *significance* of a superpixel x_i toward the black-box classification can be estimated by:

$$\delta_i(c|X) = p(c|X) - p(c|X\backslash_i). \tag{1}$$

Note that the theoretical approach of calculating $p(c|X\backslash_i)$ by obtaining or re-training a classifier $\mathcal{F}'$ excluding pixel x_i is not practical (nor applicable) in our black-box setup. To resolve this, we adopt a simple efficient method to approximate $p(c|X\backslash_i)$ by marginalizing x_i, using multiple runs of *substituted* RGB colors for superpixel x_i:

$$p(c|X\backslash_i) = \sum_{s=1}^{m_i} p(x_i = v_s|X\backslash_i)\; p(c|X\backslash_i,\; x_i = v_s), \tag{2}$$

where v_s is the substituted RGB color of superpixel x_i in the s-th run, m_i is the number of all possible RGB values for x_i for marginalization. We essentially perturb the input image X by changing the colors of its superpixel x_i (by sampling some distributions) in several runs of the classification $\mathcal{F}$, and marginalize the effect of x_i.[2] By assigning different colors v_s to superpixel x_i, we generate a *perturbed* image X_s that can be fed to the classifier $\mathcal{F}$ for the s-th run, and we perform a total of m_i marginalization runs. Fig. 1 illustrates this process.

In Eq.(2), the calculation of $p(c|X\backslash_i)$ requires the prior probability $p(x_i|X\backslash_i)$, which can be approximated as $p(x_i|X)$, since the slight perturbation is negligible, *i.e.*, $X\backslash_i \simeq X$. This way, instead of estimating $p(x_i|X\backslash_i)$ every time for each x_i, the prior term $p(x_i|X)$ only needs to be calculated once for a test image:

$$p(c|X\backslash_i) \approx \sum_{s=1}^{m_i} p(x_i = v_s|X)p(c|X\backslash_i,\; x_i = v_s). \tag{3}$$

Weight of Evidence. The naive computation of prediction difference of δ_i in Eq.(1) can be improved by using a refined metric. We adopt the *weight of evidence* [6] ω_i for superpixel x_i as the logarithm of *odds ratio*, which is commonly used in information theory and logistic regression. Specifically,

$$\omega_i(c|X) = log_2\,[odds(c|X)] - log_2\,[odds(c|X\backslash_i)], \tag{4}$$

where the odds function is the odds ratio of a probability:

$$odds(z) = \frac{p(z)}{1 - p(z)}.$$

Both prediction difference terms $\delta_i(c|X)$ and $\omega_i(c|X)$ calculate the significance of a superpixel for the prediction of given a class c. In general, the use of ω_i strengthens the contrast of the resulting interpretation map in comparison to the original δ_i. For a superpixel x_i with larger prediction differences, *i.e.* $p(c|X) > p(c|X\backslash_i)$, the significance of x_i is larger in supporting the classification result. In contrast, negative prediction difference suggests that superpixel x_i is unsupportive toward the classification result. An interpretable heatmap visualization can be produced from the superpixels with resulting prediction difference scores.

Superpixels represent salient perceptual clusters, thus it helps to produces perceptually consistent visualization for interpretation. We emphasize the advantages of using superpixels in the PDA interpretation. (1) The uniform colors of superpixels directly benefits the color substitutions in PDA, as assigning a single color to a superpixel is intuitive and effective. (2) The computation speed up is significant — from a slow pixelwise iterations to the process of typically a few hundred superpixels. One can freely control the interpretation speed *vs.* desired details by setting superpixel sizes and merging parameters.

There are two parameters that control the inference time in our method — the number of superpixels s and the marginalization sampling rounds m_i for each superpixel x_i in Eq.(2). In implementation, we replace the traversal of m_i samples with a fixed number m. Denote the running time of the black-box classifier $\mathcal{F}$ as t. The complexity of our interpretation inference is then $O(mst)$.

[1] A pixel can be regarded as the degenerate case of a size $n = 1$ superpixel.

[2] To run the black-box classifier $\mathcal{F}$ multiple times for excluding superpixel x_i, we must discard the original pixels of x_i and fill new pixel values in x_i. There obviously exists many ways for doing so. Here we simply assign a single RGB value to every pixel in x_i.

Algorithm 1 Color histogram sampling to estimate $p(c|X\setminus_i)$, for all superpixels x_i, $i \in [1, ..., s]$.

Input: image X; classifier $\mathcal{F}(X)$ which calculates $p(c|X)$; class of interest c; histogram bins b, superpixels s, and samples m.
Initialization: $B = b^3$, $H = zeros(B)$, $D = zeros(B)$.
// Calculate color histogram H.
Evenly split R,G,B values from $[0\ 255]^3$ into B bins.
for k = 1 to B **do**
  H[k] $\leftarrow$ number of pixels in color bin k.
  $D[k] = H[k] + D[k-1]$.
end for
$D = D/|X|$.
Segment image X **into** s **superpixels.**
// Sample histogram and estimate posterior probability.
for i = 1 to s **do**
  $sum_i = 0$.
  for j = 1 to m **do**
    Take a sample u from uniform distribution $U(0, 1)$.
    if $D[\lambda] > u > D[\lambda - 1]$ **then**
      $X'_j \leftarrow$ assign superpixel x_i the RGB value of bin λ.
      Run classifier $\mathcal{F}$ on X'_j to calculate $p(c|X'_j)$.
      $sum_i \mathrel{+}= p(c|X'_j)$.
    end if
  end for
  $p(c|X\setminus_i) = sum_i/m$.
end for
Output: $p(c|X\setminus_i)$ for all superpixels x_i, $i \in [1, ..., s]$.

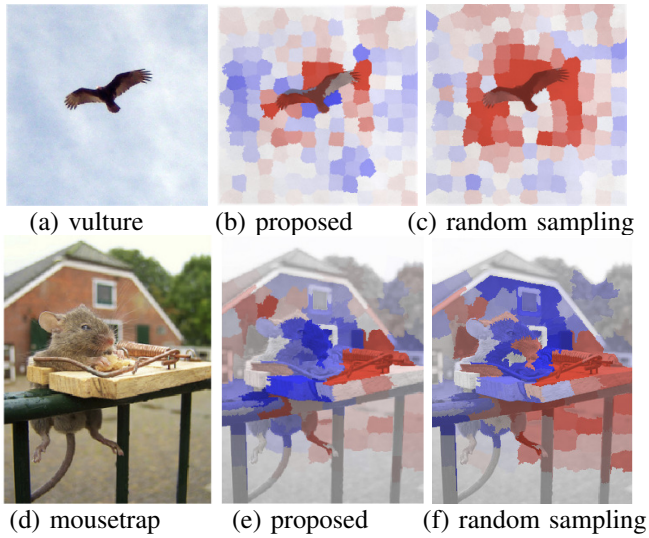
B. Color histogram sampling for superpixel substitution

We calculate $p(c|X\setminus_i)$ marginalization in Eq.(2) using a novel color histogram sampling scheme. Given input image X with $|X|$ pixels, and a black-box classifier $\mathcal{F}$ that predicts $p(c|X)$ for a specific class c that we desire to interpret, the algorithm consists of the following steps. (1) We first discretize each image RGB channel into b bins (so there are totally $B = b^3$ bins). Calculate the color histogram H of X as a vector of length B, where $H[k]$ is the number of pixels in color bin k. (2) For each color bin k, we use the median RGB value as the representative color. (3) Convert color histogram H to cumulative probability D, where $D[B] = 1$. (4) Generate a sampling color (for superpixel color substitution) by taking a sample from an uniform distribution $U(0, 1)$, and refer to the corresponding color bin in D. Replace all belonging pixels in superpixel x_i to the representative color of the sampling color bin. (5) Repeat Step.4 for m times to sample colors for each superpixel x_i, where the sampling is guided by the image color distribution specified within the histogram. Finally, marginalize x_i out to estimate $p(c|X\setminus_i)$. Algorithm 1 lists the pseudo code for these steps.

Fig. 2 compares the interpretations of the ResNet101 classifier using the proposed color histogram sampling approach *vs.* naive random color sampling. Our color histogram sampling produces interpretations that are visually more consistent and informative. The results validate that our interpretation method based on Eq.(3) can generate more accurate prior probability to interpret how the decisions are made in ResNet101.

C. Interactive multi-superpixel interpretation

We extend our method for users to interpret only a key small set of regions for a fast turnaround time. This ability to query the significance of a specified sets of superpixels leads to an interactive interpretation tool that is useful in practice.



(a) vulture (b) proposed (c) random sampling
(d) mousetrap (e) proposed (f) random sampling

Fig. 2: **Comparison of color histogram sampling *vs.* random sampling on ResNet101.** The interpretation visualizations from our color histogram sampling (b,e) are more consistent and informative than results using random sampling (c,f). Random sampling produces counter-intuitive visualizations that in (c) most of the sky surrounding the vulture is highlighted, and in (f) a large portion of the ground is highlighted.

Let R denote the specified region consisting of superpixel set, *i.e.*, $R = \{x_r\} \subseteq X$. Following the predictive difference analysis, we use $p(c|X\setminus_{\{x_r\}})$ to measure the prediction of X, and here all specified superpixels $\{x_r\}$ need to be marginalized. We make an assumption that the color of the superpixel is independent to each other. This way, $p(\{x_r\} = \{v_s\}|X)$ can be approximated by the multiplication of the probability of individual superpixels.

$$p(c|X\setminus_{\{x_r\}}) \approx \sum p(\{x_r\}{=}\{v_s\}|X)\ p(c|X\setminus_{\{x_r\}}, \{x_r\}{=}\{v_s\})$$
$$\approx \sum \prod_{x_r} p(x_r{=}v_s|X)\ p(c|X\setminus_{\{x_r\}}, \{x_r\}{=}\{v_s\}) \quad (5)$$

The $\prod_{x_r}$ computation in Eq.(5) is exponential to the number of superpixels $\{x_r\}$ within the ROI. We observe that the ROI of 2 to 4 superpixels leads to an acceptable interpretation time (30 sec to 3 min).

IV. Experiment

We apply our interpretation method to three state-of-the-art CNN models: VGG [2], GoogleNet [3], and ResNet [1]. We compare our method with two recent example-specific, black-box interpretation methods — pixelwise PDA [5] and LIME [7]. We use the public pre-trained CNN models implemented in Tensorflow as black-box classifiers. We use the weight of evidence ω_i as the metric for all experiments, except §IV-C (see below). Test images are taken from the validation set of the ILSVRC challenge [16]. All experiments are conducted on

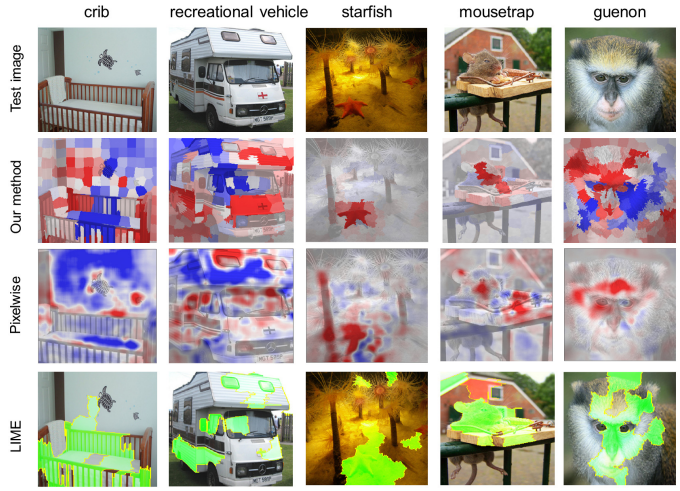


Fig. 3: **Visual comparison of interpretations generated from our method, pixelwise PDA [5], and LIME [7].** Both our method and pixelwise PDA generate a supportive/unsupportive likelihood map visualization. In comparison, LIME generates a discrete (0 or 1) map, where green shows supportive regions, and red shows unsupportive regions (which only appears in the mousetrap image).

a workstation with Intel Xeon X5570 2.93 GHz CPU and a NVIDIA Titan X GPU.

A. *Visual validation*

We compare our method with pixelwise PDA [5] and LIME [7] on GoogleNet [3] using default settings. SLIC segmentation [17] is used to generate $s = 200$ superpixels, with $b = 8$ histogram bins and $m = 10$ samples for marginalization in estimating $p(c|X\backslash_i)$. Test images are randomly chosen from the ILSVRC 2012 validation set.

Observe in Fig. 3 that our visual interpretations are more reasonable regarding the consistency within and across image regions. Our method highlights more of the visually consistent parts that coincide with human intuition. Since superpixels provide a natural way to segment out primitive semantic regions, the interpretations based on them as basic units are visually more coherent and plausible. In the "starfish" image in Fig. 3, the starfish body is highlighted most accurately with our method. In comparison, the pixelwise PDA method generates noisy visualization, since the difference within a single pixel is too small, and thus the provided information is not reliable for robust inference. The LIME method is also operated using superpixels, however its interpretation results are very unstable. It produces visually different visualizations across repeated runs of an input image. This instability is mainly due to its weak assumption on the linear model and the heuristic sampling procedure.

Our method is also advantageous over the comparison methods in the running time. It takes about 3 minutes on Titan X GPU to analyze a test image on GoogleNet. In comparison, the pixelwise PDA takes about 30 minutes to finish. The LIME method runs as fast as ours however with less interpretation information.

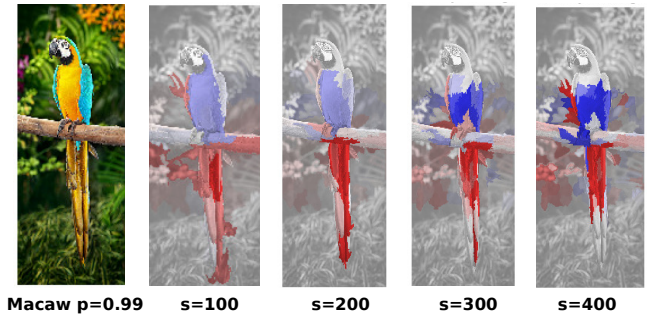


Fig. 4: **Comparison of the superpixel numbers** s on the macaw using $m = 10$, $b = 8$.

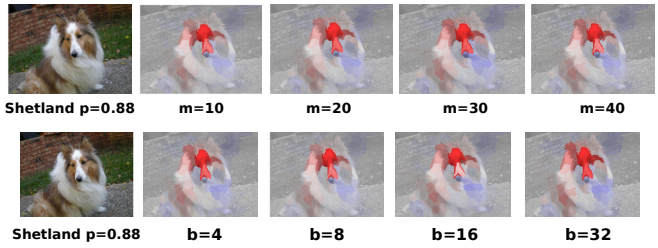


Fig. 5: **(top) Comparisons of the marginalization samples** m using ResNet101 with $s = 200$ and $b = 8$. **(bottom) Comparisons of the histogram bins** b using ResNet101 with $s = 200$ and $m = 10$.

B. *Effects of model parameters*

We evaluate how the parameters of our method influence the interpretation results. Specifically, we evaluate the number of superpixels s, the number of marginalization samples m, and the number of color histogram bins b. All comparison experiments are conducted based on pre-trained ResNet101.

The **number of superpixels** s determines the level of details of the interpretation visualization. We use SLIC segmentation [17] to produce superpixels. Larger s leads to smaller image regions, and the adjacent superpixel colors are more similar to each other. In contrast, smaller s leads to coarser superpixels with increased color variation within a superpixel. Fig. 4 shows visual comparison of the macaw image interpretations generated with $s = 100, 200, 300, 400$. Although larger s leads to better visualization details, the inconsistency and noise also grow. The extreme case of $s = |X|$ reduces our method to the pixelwise PDA of [5]. We choose $s = 200$ as a default value to achieve a good trade-off between interpretation quality and inference time.

The **number of marginalization samples** m controls the stability of marginalizing estimation during the inference. Fig. 5 (top) shows that our method generates visually similar interpretation results for $m = 10, 20, 30, 40$. This shows robustness of our method with regards to the number of samples. Also we can speed up the overall interpretation without much quality degradation by reducing m to 10.

The **number of histogram bins** b controls the accuracy of color splitting in the marginalization sampling step. Fig.

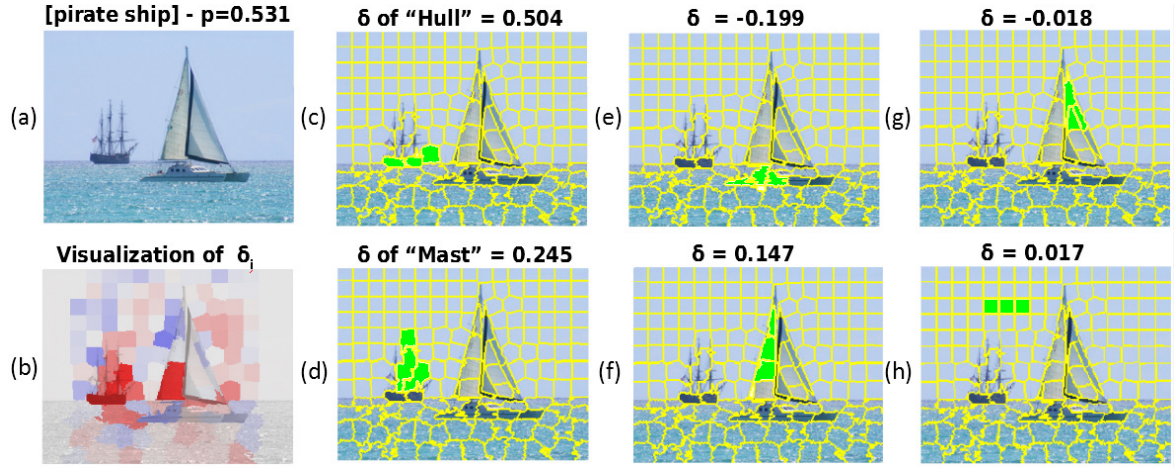


Fig. 6: **Multi-superpixel interactive interpretation** regarding the classification of a "pirate ship". (a) the input image (b) a full significance visualization. (c-h) individual interactive results on various components of the ships, where the user selected superpixels are marked with green. Note that the ship *hull* in (c) is more significant than the *mast* in (d) for the classification as a "pirate ship". Other components in (e-h) have minor or even negative effects toward such decision.

5 (bottom) shows that the change of b has little effects on interpretation results. We choose $b = 8$ as a default parameter.

C. Interactive interpretation

Fig. 6 shows an example of our multi-superpixel interactive interpretation. Given an image containing two ships that is classified as a "pirate ship", we want to find out which ship and what component of the ship (the hull or the mast) has stronger effect toward such classification. The user can select the hull or the mast for a quick interrogation regarding their significance scores. To make explicit the raw differences of significance scores, we use δ_i instead of the weight of evidence ω_i in this experiment. As shown in Fig. 6 (c,d), the hull has a larger significance score than the mast. It is reasonable that the classifier pays more attention on the hull toward the decision of a "pirate ship". Other components in Fig. 6 (e-h) show minor or even negative significance toward the decision. This way, users can quickly investigate various image components to interpret the behaviors of the black-box classifier. This interactive tool is useful for understanding and explaining the behaviors of the black-box classifier, as well as generating insights for improving the training and design of the classifier.

V. Conclusion

We present a superpixel-based interpretation method for understanding the decisions of black-box image classifiers. Our method is effective in generating informative visual interpretations. Our interactive interpretation tool is versatile for quick probing of key components in the image that aids better understanding of the deep classifier. Our black-box method is extensible to the interpretation of new deep learning models. This method can assist model designers to inspect a complex model and gain insights on whether the model learns reasonable decision rules.

Future work includes the extension of this model-agnostic method for the interpretation of other machine learning and AI tasks beyond image classification.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016, pp. 770–778.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE CVPR*, 2015, pp. 1–9.

[4] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *IEEE CVPR*, 2017.

[5] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," in *ICLR*, 2017.

[6] M. Robnik-Šikonja and I. Kononenko, "Explaining classifications for individual instances," *IEEE Trans. on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 589–600, 2008.

[7] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *ACM SIGKDD*, 2016, pp. 1135–1144.

[8] R. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *IEEE ICCV*, 2017.

[9] Z. Si and S.-C. Zhu, "Learning and-or templates for object recognition and detection," *IEEE PAMI*, vol. 35, no. 9, pp. 2189–2205, 2013.

[10] B. Letham, C. Rudin, T. H. McCormick, D. Madigan *et al.*, "Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model," *The Annals of Applied Statistics*, vol. 9, no. 3, pp. 1350–1371, 2015.

[11] Q. Zhang, Y. N. Wu, and S.-C. Zhu, "Interpretable convolutional neural networks," *arXiv preprint arXiv:1710.00935*, 2017.

[12] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*. Springer, 2014, pp. 818–833.

[13] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *arXiv preprint arXiv:1506.06579*, 2015.

[14] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *ICLR workshop*, 2014.

[15] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep Taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[17] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE PAMI*, vol. 34, no. 11, pp. 2274 – 2282, 2012.