MMDV: Interpreting DNNs via Building Evaluation Metrics, Manual Manipulation and Decision Visualization

Keyang Cheng*
School of Computer Science and Communication Engineering, Jiangsu University
Zhenjiang, China
kycheng@ujs.edu.cn

Yu Si†
School of Computer Science and Communication Engineering, Jiangsu University
Zhenjiang, China
siyu@stmail.ujs.edu.cn

Hao Zhou
School of Computer Science and Communication Engineering, Jiangsu University
Zhenjiang, China
zhouhao@stmail.ujs.edu.cn

Rabia Tahir
School of Computer Science and Communication Engineering, Jiangsu University
Zhenjiang, China
rabiatahir074@gmail.com

ABSTRACT

The unexplainability and untrustworthiness of deep neural networks hinder their application in various high-risk fields. The existing methods lack solid evaluation metrics, interpretable models, and controllable manual manipulation. This paper presents Manual Manipulation and Decision Visualization (MMDV) which makes Human-in-the-loop improve the interpretability of deep neural networks. The MMDV offers three unique benefits: 1) The Expert-drawn CAM (Draw CAM) is presented to manipulate the key feature map and update the convolutional layer parameters, which makes the model focus on and learn the important parts by making a mask of the input image from the CAM drawn by the expert; 2) A hierarchical learning structure with sequential decision trees is proposed to provide a decision path and give strong interpretability for the fully connected layer of DNNs; 3) A novel metric, Data-Model-Result interpretable evaluation(DMR metric), is proposed to assess the interpretability of data, model and the results. Comprehensive experiments are conducted on the pre-trained models and public datasets. The results of the DMR metric are 0.4943, 0.5280, 0.5445 and 0.5108. These data quantifications represent the interpretability of the model and results. The attention force ratio is about 6.5% higher than the state-of-the-art methods. The Average Drop rate achieves 26.2% and the Average Increase rate achieves 36.6%. We observed that MMDV is better than other explainable methods by attention force ratio under the positioning evaluation. Furthermore, the manual manipulation disturbance experiments show that MMDV correctly locates the most responsive region in the target item and explains the model's internal decision-making basis. The MMDV not only achieves easily understandable interpretability but also makes it possible for people to be in the loop.

CCS CONCEPTS

• **Networks** → *Network performance modeling*.

KEYWORDS

Deep Neural Networks, Interpretability, Explainability, Human in the loop

ACM Reference Format:
Keyang Cheng, Yu Si, Hao Zhou, and Rabia Tahir. 2022. MMDV: Interpreting DNNs via Building Evaluation Metrics, Manual Manipulation and Decision Visualization. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548260

1 INTRODUCTION

Deep Neural Networks (DNNs) have achieved superior performance in many tasks, such as image classification[13], object detection[7], sentiment analysis[35], and other areas[11, 14, 34]. On the other hand, DNNs are usually uninterpretable and untrustworthy, especially in high-risk domains such as medical diagnosis [10], financial risk forecast[27], and automatic driving[16]. For example, in Figure 1, the misjudgment of the DNN leads to a car accident in automatic driving. Nevertheless, we do not know how to explain the process of decision-making. Users cannot trust self-driving technology. Therefore, it is essential to open the black box and raise trust and transparency among DNNs. In addition, the current interpretable methods are self-consistent, and the corresponding evaluation criteria are also different. Therefore, while exploring the interpretable mechanism within the deep neural network, not only manual intervention is needed, but also objective evaluation indicators are needed to guide the direction.

In this paper, the key features of the deep neural network are manipulated manually, the hyperparameters are adjusted manually, and the image features are visualized. In addition, the decision path is visualized for users to understand. The Manual Manipulation and Decision Visualization (MMDV) of this paper offers three unique contributions :

*Corresponding author.
†Yu Si and Keyang Cheng contributed equally to this work.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548260

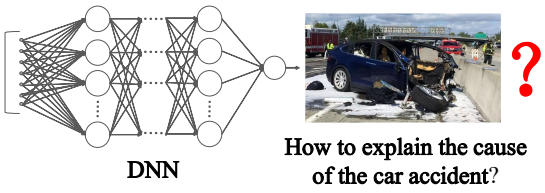

Figure 1: The DNN misjudges decision-making in automatic driving. We cannot explain the internal mechanism of DNN, so we have to distrust it.

- The Expert-drawn CAM (Draw CAM) is presented to manipulate the key feature map and update the convolutional layer parameters, which makes the model focus on and learn the important parts by making a mask of the input image from the CAM drawn by the expert.
- A hierarchical learning structure with sequential decision trees is proposed to provide a decision path and give strong interpretability for the fully connected layer of DNNs.
- A novel metric, Data-Model-Result interpretable evaluation (DMR metric), is proposed to assess the interpretability of data, model and the results. The DMR evaluation criteria is divided into three closely related parts: the data is first-order, the model is second-order and contains data, and the results are third-order including data and model.

The relationship of these contributions is shown in Figure 2. Data, Model, and Result are three parts of the Evaluation Metrics, this section will be described in Section 4. The Manual Manipulation related to the Model part of the Evaluation Metrics will be presented in Section 3.1. The Decision Visualization of result will be introduced in Section 3.2.

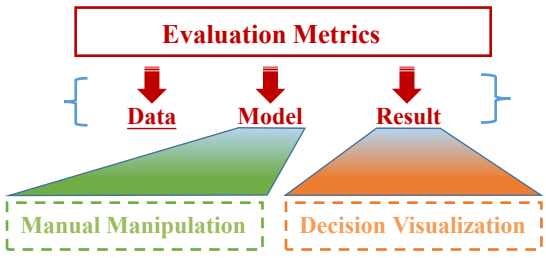

Figure 2: The relationship of the contributions. Model is manipulated and controlled by Manual Manipulation. Result is quantified and displayed through Decision Visualization

2 RELATED WORK

2.1 Human in the loop

Self et al.[21] propose a human-model interactive parameter adjustment mode to facilitate user participation by bridging the gaps between a user's intention and the parameters of a weighted multi-dimensional scaling model. Gentile et al. [9] propose an interactive dictionary expansion tool using two neural language models. Yao et al. [36] point out that iterations between queries may be expensive or time-consuming, making it unrealistic for executing interactions with end-users. They present an interactive object detection architecture to employ individuals to correct a few annotations proposed by a detector for the un-annotated image with the maximum predicted annotation cost. Madono et al. [17] put forward an efficient human-in-the-loop object detection framework composed of bi-directional deep SORT [33] and annotation-free segment identification (AFSID). Humans' role in this architecture is to verify the object candidates that bi-directional deep SORT can not detect automatically. Then train the model over the supplementary objects annotated by individuals. However, the deep network model needs to be involved in the human operation in the process of training.

2.2 Post-hoc explainability analysis

Post-hoc explainability works after the finished black box. Sensitive features of DNNs can be identified through feature analysis. The principle of the model can be explained to a certain extent[39]. D.Bau [2] proposed a technique to evaluate the function of individual network units to extensive understanding of how a network works. Network dissection relies on the emergence of disentangled or human-interpretable units during training[1, 42]. By systematically extracting the internal structure and parameters of the neural networks, it is committed to fault or deviation detection in the neural networks [32, 37]. Numerous efforts have explored the design of saliency maps identifying pixels that most influenced the model's prediction [4, 22, 43]. Nevertheless, the saliency map does not explain the model's decision-making process. Some works analyze the original mechanism of neural networks by the knowledge of mathematics and physics [3, 18, 38]. These methods are related to the theory of mathematical heuristic reasoning. However, it is easy to make unrealistic theoretical assumptions and break away from practical applications. The advantage of the MMDV model is that there is no need to sacrifice interpretability for the sake of predictive performance, but it cannot explain the reasoning process inside the model.

2.3 Ad-hoc interpretable modelling

The ad-hoc interpretable modelling can avoid the bias in post-hoc explainability analysis. The regularization technique of interpretable representation is used to make the constructed neural network more interpretability [15, 25, 40]. Properties such as decomposability, sparsity and monotonicity can enhance interpretability. [5, 12, 31] seek interpretability by designing and deploying more interpretable components. These components include neurons with specially designed activation functions, insertion layers with special functions, modular architecture. Some works replace parts of the DNNs with interpretable machine learning models, such as decision trees[28, 29], rule systems[6]. A model constructs the neural network into a hierarchical structure and runs reasoning by dynamically selecting branches[19]. Other approaches[23, 41] fuse deep learning into each decision tree node. The weakness is that the proxy method needs to pay the extra cost, and it will reduce the prediction accuracy of the original model at the same time.

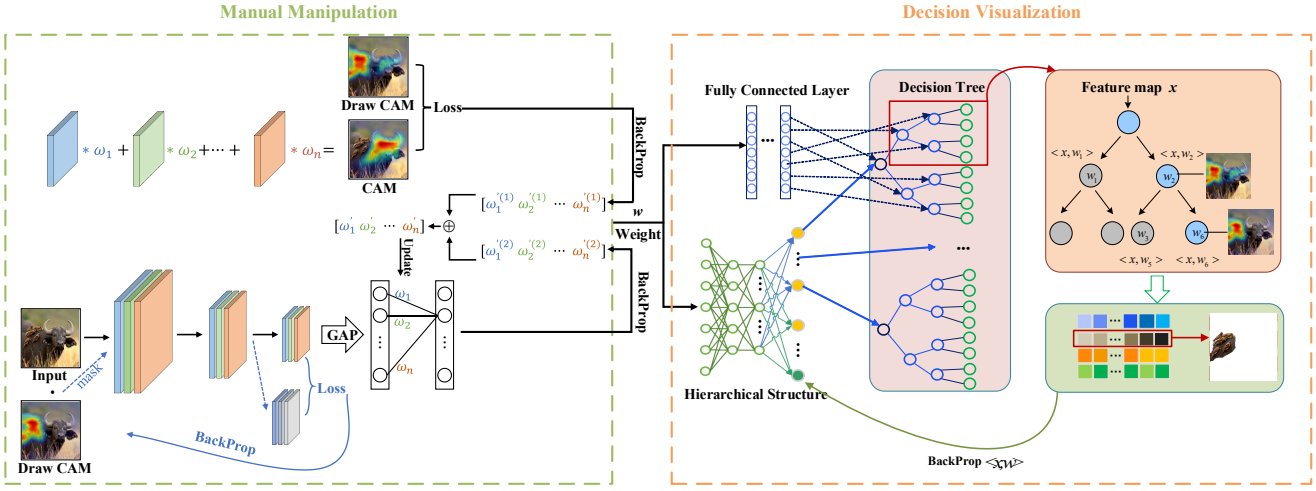

Figure 3: The proposed MMDV overview model. The weight parameters of convolution layer are changed artificially through Manual Manipulation. Decision Visualization provides visual path for decision-making. The model combines the convolution layer human-in-the-loop method and the fully connected layer decision visualization method to realize the rational application of expert knowledge in the model training process, and provides an interpretable visual decision for the classification.

3 METHOD

We design the Manual Manipulation and Decision Visualization (MMDV) model to achieve Human-in-the-loop and interpretable decision-making. The overview model is shown in Figure 3. First, the Manual Manipulation model is applied to each convolution layer of DNNs. The feature map of the convolution layer is extracted for manual drawing intervention, so as to update the parameter weights of different parts of the convolution layer. The DrawCAM method is proposed to control the feature map in the Manual Manipulation model. DrawCAM updates the weights manually through people's expert knowledge, which can not only increase the weight of positive samples but also reduce the weights of negative samples. Second, the Decision Visualization model is applied to the last fully connected layer. The hierarchical structure is proposed to build decision trees to make decision reasonable. The Optimization CAM is introduced in decision trees to improve the understandability of decision-making. The MMDV model combines the convolution layer human-in-the-loop method and the fully connected layer decision visualization method to realize the rational application of expert knowledge in the model training process, and provides an interpretable visual decision for the final model results.

3.1 Manual Manipulation

Class Activation Mapping is a conventional visualization approach that creates an attention map by combining the feature maps created by the final convolutional layer in a weighted average. We create k feature maps $A^k \in \mathbb{R}^{u \times v}$ from the penultimate layer, width u and height v for every class c, with i and j indexing each element. As a result, $A^k_{i,j}$ refers to the activation of feature maps at a certain location (i, j). Then, as illustrated in Formula 1, global average pooling is used to spatially pool these feature maps, and linear transformation is applied to obtain the score Y^c of each class c.

$$Y^c = \sum_k W^c_k \frac{1}{Z} \sum_i \sum_j A^k_{i,j} \tag{1}$$

where W^c_k is the weight connecting the k^{th} feature map with the c^{th} class. Z is the number of pixels in the feature map ($Z = \sum_i \sum_j \mathbf{1}$)
.

However, the feature map would lose spatial information and label more background noise after dimensioning the global average pooling layer. It inflicts severe harm to the model's attention force's spatial information. As shown in Formula 2, we propose Optimization Class Activation Mapping algorithm to convert the weight after global average pooling to matrix value-based weight $W^{(c)}_k$.

$$W^{(c)}_k = \sum_{i,j} \frac{\partial Y^{(c)}}{\partial A^k_{i,j}} \tag{2}$$

After passing through the model's last convolution layer, we remove the global average pooling layer and visualize the output of the convolution layer to save the spatial location information of the last convolution layer. To emphasize the high-weight area, we use the Hadamard product to multiply the feature map and the weight $W^{(c)}_k$, which is indicated as $W^{(c)}_k \circ A^k$. The Hadamard product is used to multiply the weight $W^{(c)}_k$ with the feature map to output the weight feature map, as shown in Formula 3.

$$L^{(c)} = ReLU\left(\sum_k W^{(c)}_k \circ A^k\right) \tag{3}$$

The proposed Optimization Class Activation Mapping retains the *ReLU* activation function. Moreover, it is applied to the linear combination of the feature map because the *ReLU* activation

function can effectively distinguish between relevant and unrelated pixels and extract the features that contribute to the classification result class. Following that, activation mapping is generated in the feature map group, and the saliency mapping sum of the feature map is obtained using the activation function, as given in Formula 4.

$$S : L_{\text{sum}}^{(c)} = \sum ReLU\left(\sum_k W_k^{(c)} \circ A^k\right) \tag{4}$$

In the visualization stage of the decision path, we use the above algorithm to output the attribution graph to select the weight of the key parts. The Optimization Class Activation Mapping algorithm is shown as Algorithm 1.

Figure 4: Drawn area in DrawCAM. Increase the weight of important areas, and the image is displayed in red. Reduce the weight of the negative sample area and the image is displayed in blue.

Based on this algorithm, the CAM drawn by the expert (as shown on the right of Figure 4) is used to multiply with the input image to obtain the mask image (as shown in the middle of Figure 4). Input this mask image into the model to calculate the last feature map of the convolution layer. Using this feature map to compare the loss with the feature map of the original input image, the weights of the convolution layer are updated, as shown in Formula 5.

$$L_c = Loss(f_c(X \cdot M), f_c(X)) \tag{5}$$

where $f_c(\cdot)$ is the convolution layer of the model, X is the input image, M is the mask (i.e. expert-drawn CAM).

The fully connected layer gradients $W_1' = \left[\omega_1'^{(1)}, \omega_2'^{(1)}, \cdots, \omega_n'^{(1)}\right]$ and gradients $W_2' = \left[\omega_1'^{(2)}, \omega_2'^{(2)}, \cdots, \omega_n'^{(2)}\right]$ are derived by calculating the output of the model and the loss and backpropagation of the labeled, CAM and expert-drawn CAM, respectively. Then the gradient W_1' and the gradient W_2' are weighted to obtain W', which is used to update the parameters of the fully connected layer. The weight α is used to coordinate whether the model is biased towards learning CAM or towards learning the contribution of CAM to prediction, as shown in Formula 6.

$$\begin{aligned} \text{Loss}(f(X), Y) &\rightarrow W_1' = \left[\omega_1'^{(1)}, \omega_2'^{(1)}, \cdots, \omega_n'^{(1)}\right] \\ \text{Loss}\left(M, Y_{CAM}\right) &\rightarrow W_2' = \left[\omega_1'^{(2)}, \omega_2'^{(2)}, \cdots, \omega_n'^{(2)}\right] \\ W' &= \alpha W_1' + (1-\alpha) W_2' \end{aligned} \tag{6}$$

where $f(\cdot)$ is the whole model, Y_{CAM} is the class activation mapping of input images, and α is the weighted parameter (decision preference for results or visualization).

3.2 Decision Visualization

We replace the last fully connected layer of the backbone network with the Hierarchical Structure. Hierarchical Structure represents a certain number of decision trees. Hierarchical Structure is a full binary tree, which has $N = 2^{d-1} - 1$ split nodes with depth d. Each of Decision Tree consists of a set of split nodes $N = n_1, n_2, ..., n_N$ and a set of leaf nodes $L = l_1, l_2, ..., l_N$. We associate a weight vector n_i with each node, $n_i = w_i$. The weight vector from the fully connected layer's weight W. The softmax inner products give the child probabilities in the leaf node. The sigmoid function gives the child probabilities in the inner node. For example, x_q is the qth image sample. The loss of leaf node $\mathbf{a}_q$ in Decision Tree is shown in Formula 7.

$$L_{\text{leaf}}\left(\mathbf{a}_q\right) = -\log\left(\frac{exp(\mathbf{a}_q \mathbf{W}_{\cdot y_q})}{\sum_{j=N-|C_l|+1}^{N} exp(\mathbf{a}_q \mathbf{W}_{\cdot j})}\right) \tag{7}$$

where $y_q \in \{N - |C_l| + 1, \ldots, N\}$ denotes the leaf category index of x_q. Besides, the inner node loss for $\mathbf{a}_q$ in Decision Tree is defined as shown in Formula 8.

$$\begin{aligned} L_{inner}\left(\mathbf{a}_q\right) = -\sum_{j=1}^{N-|C_l|} \Big[& \mathbf{H}_{j,y_q} \log\left(Sigmod\left(\mathbf{a}_q \mathbf{W}_{\cdot j}\right)\right) \\ & + \left(1 - \mathbf{H}_{j,y_q}\right) \log\left(1 - Sigmod\left(\mathbf{a}_q \mathbf{W}_{\cdot j}\right)\right) \Big] \end{aligned} \tag{8}$$

where the hierarchical representations of all categories $\mathbf{H} \in \{0,1\}^{N \times N}$.

In the part of decision tree visualization, Algorithm 1 is used to obtain the feature visualization map of the decision process. Visual semantics are given to the decision-making process by Optimization Class Activation Mapping, as shown in Figure 5. On the basis of constructing sequential decision tree, key nodes are extracted for visualization. The "aircraft" category is classified according to the key information obtained, and the trusted visual images are provided to the user.

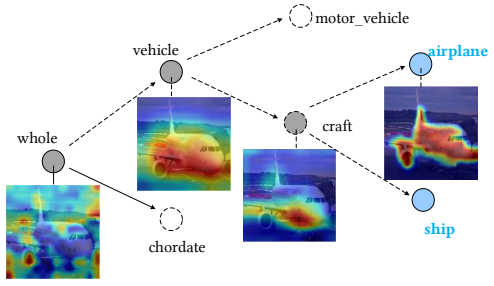

Figure 5: Deep decision tree Optimization Class Activation Mapping inference diagram.

4 DMR EVALUATION METRICS

The existing interpretable work is aimed at the credibility of the data distribution, the transparency of the model and the interpretability of the results. Our idea is that data, models, and results cannot be

Algorithm 1 Optimization Class Activation Mapping algorithm

Input: Image X_0, Class c, layer l
Output: $L^c_{Opti-CAM}$

1: Initialization:Let the penultimate layer to create k feature maps $A^k \in \mathbb{R}^{u \times v}$, $A^k_{i,j}$ refers to the activation of feature maps at a certain location (i, j);
2: Get target layer feature maps A, importance weights W^c;
3: **for** k in $[0, ..., C-1]$ **do**
4: $Linear(GlobalAvgPool(A^k_{i,j}))$;
5: $Y^c \leftarrow \sum_k W^c_k \frac{1}{Z} \sum_i \sum_j A^k_{i,j}$;
6: Use Optimization Class Activation Mapping algorithm to convert the weight after global average pooling to matrix value-based weight $W^{(c)}_k$,
$W^{(c)}_k \leftarrow \sum_{i,j} \frac{\partial Y^{(c)}}{\partial A^k_{i,j}}$;
7: Use the Hadamard product to multiply the feature map and the weight $W^{(c)}_k$,
$L^{(c)} \leftarrow ReLU\left(\sum_k W^{(c)}_k \circ A^k\right)$;
8: $k \leftarrow k + 1$;
9: **end for**
10: **return** $ReLU(L^c_{Opti-CAM})$

discussed separately. The quality of the dataset affects the performance of the model, and the transparency of the model affects the interpretability of the results. As shown in Formula 9, the evaluation criteria are divided into three parts, the first part is the data, the second part is the model, and the third part is the result evaluation.

$$M_{DMR} = \alpha \cdot f_{Data} + \beta \cdot f_{Model}(f_{Data}) + \gamma \cdot f_{Result}(f_{Data}, f_{Model}) \tag{9}$$

where f_{Data} represents the evaluation function of Data and α is its corresponding regular parameter. f_{Model} represents the evaluation function of Model, and β is its corresponding regular parameter. Because the interpretability of Data affects the performance of Model, f_{Data} is the variable parameter of f_{Model}. f_{Result} is the confidence evaluation function of Result, and γ is its corresponding regular parameter. Because the result of the model is determined by the data and the performance of the model, the internal variables of the Result function contain f_{Data} and f_{Model}.

Data. The evaluation function in Data comes from the trusted distribution of different datasets. The number of datasets, dimensioning accuracy, and coverage will affect the interpretability of the data. In addition, the preprocessing of data can also enhance its interpretability. There may be many different processing methods for different datasets, so there is no unified strategy for dealing with the data.

Model. There are many evaluation indicators of the DNNs model, but the interpretable evaluation can not be evaluated separately from the dataset. The Model part is represented based on Data evaluation metric, such as it contains the content of an interpretable evaluation of the dataset. The model can be interpreted and evaluated uniformly, and the impact of data needs to be considered, so we use Average Drop (AD) and Average Increase (AI) as quantitative disturbance evaluation metrics. They are indicators to measure the influence degree of adding disturbance to the model input on the classification results. Those metrics aim to conduct quantitative analysis on the influence degree of the disturbance model.

Average Drop represents the maximum positive difference between the prediction using the input image and the prediction through saliency map. AD= $\frac{1}{N}\sum_{i=1}^{N}(\frac{1}{Y_i^c}\max\left(0, Y_i^c - O_i^c\right)) \times 100$. Average Drop measures how the saliency map causes changes in scores. The lower the degree of decline in Average Drop scores, the higher the model's credibility.

Average Increase describes the condition in which the saliceny map results in a higher score. A higher score indicates greater confidence in the model's ability to generate an interpretation. AI= $\sum_{i=1}^{N}(\frac{1}{N}\text{Sign}\left(Y_i^c < O_i^c\right)) \times 100$, where Y_i^c is the predicted score for class c on image i, and O_i^c is the predicted score for class c with the explanation map region as input. N is the total number of images in dataset.

Result. The interpretable evaluation of the results comes from the interpretability of the dataset and the interpretability of the model. Therefore, the result is composed of datasets and models.

Inspired by **Energy-Based Pointing Game (EBPG)**, we binarize the sample image with the bounding box of the predicted category. For example the middle picture in Figure 4, the inside area is set to 1 and the outside area is set to 0. We assign the ground-truth mask as G and the predicted mask as P. Then, the metric represents how much attention area is in the predicted bounding box, shown as Proportion= $(\|G \odot P\|_1 / \|P\|_1) \times 100$.

The recognized metric in image segmentation, **mean Intersection over Union(mIoU)**, is used to analyze the localization ability and meaningfulness of the attributions captured in an explanation map. mIoU = $\frac{1}{k}\sum_{i=1}^{k}(P \bigcap G)/(P \bigcup G) \times 100$, where k is the number of categories, which we can get global evaluation.

We use **Bounding box (Bbox)** to quantify how well attribution methods identify and localize the object of interest. Assuming that the bounding box contains n pixels, we assess how many of the top k pixels in predicted mask P, where $k \leq n$. We can get the Bbox score by $Bbox = (k/n) \times 100$.

Visual attribution assessment metric(VAAM) is a metric proposed for assessing the visual quality of the explanation maps in this paper. This metric evaluates the suitability and denoising ability of Decision Visualization in MMDV. Then, VAAM combines the advantages of the area proportion in EBPG, the segmentation bounding box in mIoU and the pixel attribution in Bbox. VAAM is formed as $\frac{1}{3}(Proportion + mIoU + Bbox) \times 100$.

5 EXPERIMENTS

In this section, we introduce experimental settings first. Second, manual manipulation and visualization of convolution layer features in deep neural network are drawn by DrawCAM. Then, the Optimization Class Activation Mapping of a deep decision tree in hierarchical structure is visualized and compared. Furthermore, we compare similar visualization methods to demonstrate the superiority of MMDV in localization evaluations. Finally, a series of

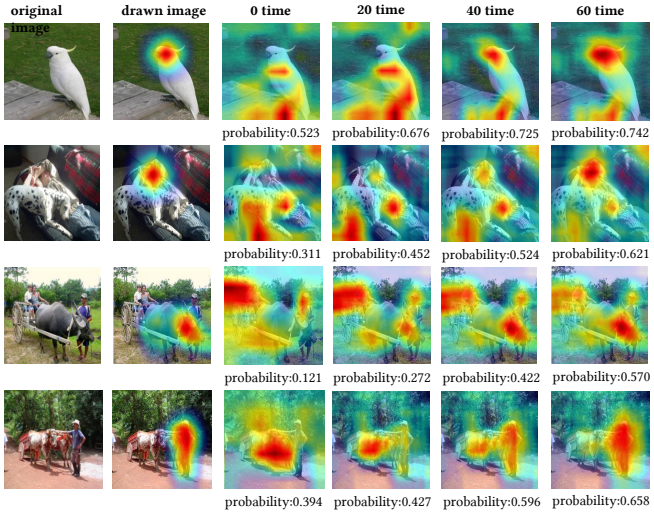

Figure 6: With the influence of draw and manipulation, the probability of classification accuracy is improved gradually.

convolution kernel disturbances are conducted to show the faithfulness of MMDV.

5.1 Experimental settings

Two hierarchically organized datasets are considered in our experiments: (1) ILSVRC2012 is a subset of the large hand-labeled ImageNet dataset organized according to the WordNet hierarchy, in which each node of the hierarchy is depicted by hundreds and thousands of images. The dataset with 1,000 leaf node categories, and the whole category hierarchy consists of 1,860 nodes. (2) PASCAL VOC 2007 is a popular object detection dataset containing 4952 test images belonging to 20 object classes. For datasets, all images are resized to 224 × 224 × 3, and then transformed to tensors and normalized to the range [0,1]. The pre-training model VGG16 and ResNet-50 are used in the following experiments.

5.2 Draw CAM to manipulate weights

The picture features drawn manually, as shown in Figure 4. The middle picture is a hand-drawn area. Increase the weight of important areas, and the image is displayed in red. Reduce the weight of the negative sample area and the image is displayed in blue. In the model training phase, manually modify the weight of the input picture for calibration. This effect can be shown in the test phase. For example, in Figure 6, DrawCAM shows that manual manipulation is characterized by the position of some multi-objective samples in ILSVRC2012 dataset. Manually increase the weight of the target category parameters, successfully classify the target category, when there are multiple classification targets in the picture. With the influence of continuous manipulation, the feature learning of the extracted test images in the convolution layer focuses on the drawing area. It has brought beneficial effects to the improvement of classification accuracy. Experiments show that manual drawing can increase the control power of people in the loop and make the model more trustworthy and understandable.

5.3 Hierarchical structure visualization

Figure 7 depicts the differences between attention maps formed by traditional Class Activation Mapping and Optimization Class Activation Mapping. The traditional Class Activation Mapping after global average pooling dimension reduction lost much spatial information. For example, traditional Class Activation Mapping in Figure 7, the saliency map with many sky parts. In fact, the context information is not clear, and the target area coverage produces a large offset. However, in the Optimization Class Activation Mapping, many high-weight parts are marked with "machine compartment", which fully reflects the focus area of the model. Meanwhile, the annotated areas of the model have obvious changes at different node levels, and good explanations have been obtained at any node. The decision of aircraft class is derived from the "landing gear" part at "Hierarchy 3" in the process of hierarchical reasoning.

5.4 Localization evaluations

In addition, this part of the experiments measure the location evaluation accuracy of the significance by the objective quantitative. About 70% of the contribution points of the saliency map are generated by Decision Visualization. These areas are within the ground-truth surrounding frame of the target object. The result is about 6.5% higher than the most advanced similar algorithm. The algorithm completes spatial information and avoids background noise interference, resulting in a more accurate note to mark and good interpretability.

Besides, Table 1 shows the results of interpretable evaluation metrics for state-of-the-art XAI methods along with MMDV on VGG16 model trained on the PASCAL VOC 2007 dataset. Table 2 shows the results on ResNet-50. For each metric, the best is shown

Table 1: Results of the state-of-the-art methods compared with MMDV on VGG16 model.

Metric	Grad CAM [22]	Grad CAM++ [4]	Extremal Perturbation [8]	RISE [20]	Score CAM [30]	Integrated Gradient [26]	FullGrad [24]	MMDV (Ours)
EBPG	55.44	46.29	**61.19**	33.44	46.42	36.87	38.72	60.60
mIoU	26.52	**28.1**	25.44	27.11	27.71	14.11	26.61	27.85
Bbox	51.70	55.59	51.20	54.59	54.98	33.97	54.17	**55.75**
VAAM	44.55	43.33	45.94	38.38	43.04	28.32	39.83	**48.07**

Table 2: Results of the state-of-the-art methods compared with MMDV on ResNet-50 model.

Metric	Grad CAM [22]	Grad CAM++ [4]	Extremal Perturbation [8]	RISE [20]	Score CAM [30]	Integrated Gradient [26]	FullGrad [24]	MMDV (Ours)
EBPG	60.08	47.78	63.24	32.86	35.56	40.62	39.55	**66.02**
mIoU	**32.16**	30.16	26.29	27.40	31.0	15.41	20.20	31.35
Bbox	60.25	58.66	52.34	55.55	60.02	34.79	44.94	**61.65**
VAAM	50.83	45.53	45.94	47.29	42.19	30.27	34.90	**53.01**

Table 3: Results of the state-of-the-art methods compared with MMDV on VGG16 model.

Metric	Grad CAM [22]	Grad CAM++ [4]	Extremal Perturbation [8]	RISE [20]	Score CAM [30]	Integrated Gradient [26]	FullGrad [24]	MMDV (Ours)
AD(%)	49.47	60.63	43.90	39.62	39.79	64.74	60.78	**38.55**
AI(%)	31.08	23.89	32.65	37.76	36.42	26.17	22.73	**37.80**

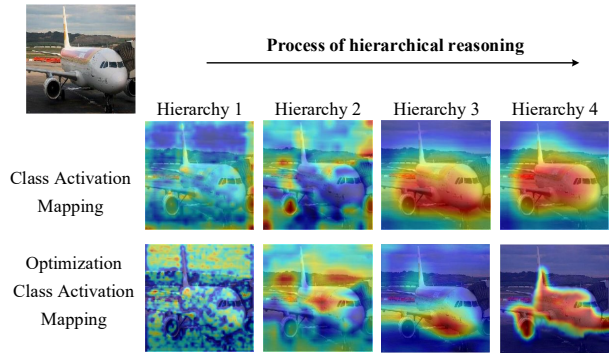

Figure 7: The process of hierarchical reasoning between Class Activation Mapping and Optimization Class Activation Mapping.

in bold, and the second-best is underlined. All values are reported in percentage.

5.5 Faithfulness evaluations

The attention map's reliability in the decision area is verified. The Hadamard dot product of the input sample and the attention force is mainly used to block the input sample and evaluate the accuracy variations on the target class when measuring the confidence of the model attention force in the classification task. The partial correlation convolution kernel of the model is adjusted to zero to reduce the impact of the convolution kernel of the most significant correlation area. In the contrast technique, the attention force's 50 per cent effective pixels are occluded and dot multiplied with the original input, and the ILSVRC2012 dataset is used for experiments. Comparison methods like LIME, RISE, and ScoreCAM are used to perturb the model input. As indicated in experiments, the algorithm proposed in this paper achieves an AD rate of 26.2% and an AI rate of 36.6%. AD rate is 5.3% better than others. As well as AI rate is 6% higher than other works. Because the comparison algorithm generates the noisy feature map due to sampling approximation.

According to the value of the saliency map, we replace 3.6% pixels in the input image with a highly blurred version at a time until there are no pixels. Some of the results are in Figure8. Each picture contains an input picture with some pixels deleted and a prediction curve for the corresponding time. It can be found from the curve that the prediction accuracy will drop suddenly after the necessary attribution pixels are deleted. In contrast to the deletion experiment, the insertion test replaces 3.6% pixels of the blurred image with the original image until the image is restored. Some of the results are in Figure9. From the insertion curve, we can see that the insertion of necessary attribution pixels leads to improving prediction accuracy.

Combined with the experimental data of the above datasets, models and results, the M_{DMR} can be calculated according to Formula 9.

Table 4: Results of the state-of-the-art methods compared with MMDV on ResNet-50 model.

Metric	Grad CAM [22]	Grad CAM++ [4]	Extremal Perturbation [8]	RISE [20]	Score CAM [30]	Integrated Gradient [26]	FullGrad [24]	MMDV (Ours)
AD%	35.80	41.77	39.38	39.77	35.36	66.12	65.99	**30.90**
AI%	36.58	32.15	34.27	37.08	37.08	24.24	25.36	**40.25**

Figure 8: Delete critical area pixels. The classification confidence of the picture after the pixel is deleted.

Figure 9: Inserted critical area pixels. The classification confidence of the picture after the pixel is inserted.

Its confidence scale is normalized to [0-1]. The three hyperparameter α, β, γ are taken as 1/3. Since the two datasets PASCAL VOC 2007 and ILSVRC2012 are public datasets and we have not performed data processing on them, we assume that the confidence level of f_{Data} is 1. Confidence of f_{Model} comes from Tables 3 and Table 4. To ensure consistent representation of AD and AI, let $f_{Model} = (100-AD)+AI$. Confidence of $f_{Result} = VAAM = \frac{1}{3}(Proportion + mIoU + Bbox)$ comes from Tables 1 and Table 2. The results are shown in Table 5.

Table 5: Results of interpretable evaluation M_{DMR} metric

Data	Model	Result	M_{DMR}
PASCAL VOC 2007	VGG16	VAAM	**0.4943**
PASCAL VOC 2007	ResNet-50	VAAM	**0.5280**
ILSVRC2012	VGG16	VAAM	**0.5445**
ILSVRC2012	ResNet-50	VAAM	**0.5108**

6 CONCLUSION

The Manual Manipulation and Decision Visualization (MMDV) method is proposed in this paper, which makes Human-in-the-loop improve the interpretability of deep neural networks. The Expert-drawn CAM (Draw CAM) is presented to manipulate the key feature map and update the convolutional layer parameters, which makes the model focus on and learn the important parts by making a mask of the input image from the CAM drawn by the expert. A hierarchical learning structure with sequential decision trees is proposed to provide a decision path and give strong interpretability for the fully connected layer of DNNs. The DMR metric is proposed to assess the interpretability of data, model and the results. The experimental results show that the MMDV not only achieves easily understandable interpretability but also makes it possible for people to be in the loop.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (Grant No.61972183).

REFERENCES

[1] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6541–6549.

[2] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. 2020. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences* 117, 48 (2020), 30071–30078.

[3] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences* 116, 32 (2019), 15849–15854.

[4] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. 2018. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 839–847.

[5] Fenglei Fan, Mengzhou Li, Yueyang Teng, and Ge Wang. 2020. Soft Autoencoder and Its Wavelet Adaptation Interpretation. *IEEE Transactions on Computational Imaging* 6 (2020), 1245–1257.

[6] Fenglei Fan and Ge Wang. 2020. Fuzzy logic interpretation of quadratic networks. *Neurocomputing* 374 (2020), 10–21.

[7] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. 2020. Few-shot object detection with attention-RPN and multi-relation detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4013–4022.

[8] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. 2019. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF international conference on computer vision*. 2950–2958.

[9] Anna Lisa Gentile, Daniel Gruhl, Petar Ristoski, and Steve Welch. 2019. Explore and exploit. Dictionary expansion with human-in-the-loop. In *European Semantic Web Conference*. Springer, 131–145.

[10] Branka Hadji Misheva, Ali Hirsa, Joerg Osterrieder, Onkar Kulkarni, and Stephen Fung Lin. 2021. Explainable AI in Credit Risk Management. *Credit Risk Management (March 1, 2021)* (2021).

[11] Peter Hase and Mohit Bansal. 2020. Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior?. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[12] Chengyue Jiang, Yinggong Zhao, Shanbo Chu, Libin Shen, and Kewei Tu. 2020. Cold-start and interpretability: Turning regular expressions into trainable recurrent neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 3193–3207.

[13] Young Jae Kim, Jang Pyo Bae, Jun-Won Chung, Dong Kyun Park, Kwang Gi Kim, and Yoon Jae Kim. 2021. New polyp image classification technique using transfer learning of network-in-network structure in endoscopic images. *Scientific Reports* 11, 1 (2021), 1–8.

[14] Kacper Jacek Kubara, Blazej Manczak, Blazej Dolicki, and Kacper Sawicz. 2021. Towards Transparent and Explainable Attention Models, ML Reproducibility Challenge 2020. (2021).

[15] Isaac Lage, Andrew Slavin Ross, Been Kim, Samuel J Gershman, and Finale Doshi-Velez. 2018. Human-in-the-loop interpretability prior. *Advances in neural information processing systems* 31 (2018).

[16] Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. 2020. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters* 5, 2 (2020), 3153–3160.

[17] Koki Madono, Teppei Nakano, Tetsunori Kobayashi, and Tetsuji Ogawa. 2020. Efficient Human-In-The-Loop Object Detection using Bi-Directional Deep SORT and Annotation-Free Segment Identification. In *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 1226–1233.

[18] Song Mei and Andrea Montanari. 2019. The Generalization Error of Random Features Regression: Precise Asymptotics and the Double Descent Curve. *Communications on Pure and Applied Mathematics* (2019).

[19] Meike Nauta, Ron van Bree, and Christin Seifert. 2021. Neural Prototype Trees for Interpretable Fine-grained Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14933–14943.

[20] Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *Proceedings of the British Machine Vision Conference (BMVC)*.

[21] Jessica Zeitz Self, Radha Krishnan Vinayagam, James Thomas Fry, and Chris North. 2016. Bridging the gap between user intention and model parameters for human-in-the-loop data analytics. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. 1–6.

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

[23] Wei Shen, Yilu Guo, Yan Wang, Kai Zhao, Bo Wang, and Alan Loddon Yuille. 2019. Deep differentiable random forests for age estimation. *IEEE transactions on pattern analysis and machine intelligence* (2019).

[24] Suraj Srinivas and François Fleuret. 2019. Full-gradient representation for neural network visualization. *Advances in neural information processing systems* 32 (2019).

[25] Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. 2018. Spine: Sparse interpretable neural embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

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

[27] Neus Llop Torrent, Giorgio Visani, and Enrico Bagli. 2020. PSD2 Explainable AI Model for Credit Scoring. *arXiv preprint arXiv:2011.10367* (2020).

[28] Alvin Wan, Lisa Dunlap, Daniel Ho, Jihan Yin, Scott Lee, Henry Jin, Suzanne Petryk, Sarah Adel Bargal, and Joseph E Gonzalez. 2021. NBDT: Neural-backed decision trees. (2021).

[29] Alvin Wan, Daniel Ho, Younjin Song, Henk Tillman, Sarah Adel Bargal, and Joseph E Gonzalez. 2020. SegNBDT: Visual Decision Rules for Segmentation. *arXiv preprint arXiv:2006.06868* (2020).

[30] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. 2020. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 24–25.

[31] Tong Wang. 2019. Gaining free or low-cost interpretability with interpretable partial substitute. In *International Conference on Machine Learning*. PMLR, 6505–6514.

[32] Yulong Wang, Hang Su, Bo Zhang, and Xiaolin Hu. 2018. Interpret neural networks by identifying critical data routing paths. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8906–8914.

[33] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*. IEEE, 3645–3649.

[34] Mike Wu, Sonali Parbhoo, Michael Hughes, Ryan Kindle, Leo Celi, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. 2020. Regional tree regularization for interpretability in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6413–6421.

[35] Ashima Yadav and Dinesh Kumar Vishwakarma. 2020. Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review* 53, 6 (2020), 4335–4385.

[36] Angela Yao, Juergen Gall, Christian Leistner, and Luc Van Gool. 2012. Interactive object detection. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 3242–3249.

[37] Jiaxuan You, Jure Leskovec, Kaiming He, and Saining Xie. 2020. Graph structure of neural networks. In *International Conference on Machine Learning*. PMLR, 10881–10891.

[38] Shujian Yu and Jose C Principe. 2019. Understanding autoencoders with information theoretic concepts. *Neural Networks* 117 (2019), 104–123.

[39] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. 2018. Interpreting cnn knowledge via an explanatory graph. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[40] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. 2018. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8827–8836.

[41] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. 2019. Interpreting cnns via decision trees. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6261–6270.

[42] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. 2018. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence* 41, 9 (2018), 2131–2145.

[43] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2921–2929.