

Model Doctor: A Simple Gradient Aggregation Strategy for Diagnosing and Treating CNN Classifiers

Zunlei Feng, Jiacong Hu, Sai Wu, Xiaotian Yu, Jie Song, Mingli Song*

Zhejiang University

{zunleifeng, jiaconghu, wusai, yuxiaotian, sjie, brooksong}@zju.edu.cn

Abstract

Recently, Convolutional Neural Network (CNN) has achieved excellent performance in the classification task. It is widely known that CNN is deemed as a ‘black-box’, which is hard for understanding the prediction mechanism and debugging the wrong prediction. Some model debugging and explanation works are developed for solving the above drawbacks. However, those methods focus on explanation and diagnosing possible causes for model prediction, based on which the researchers handle the following optimization of models manually. In this paper, we propose the first completely automatic model diagnosing and treating tool, termed as Model Doctor. Based on two discoveries that 1) each category is only correlated with sparse and specific convolution kernels, and 2) adversarial samples are isolated while normal samples are successive in the feature space, a simple aggregate gradient constraint is devised for effectively diagnosing and optimizing CNN classifiers. The aggregate gradient strategy is a versatile module for mainstream CNN classifiers. Extensive experiments demonstrate that the proposed Model Doctor applies to all existing CNN classifiers, and improves the accuracy of 16 mainstream CNN classifiers by 1% ~ 5%.

Introduction

Image classification is a widely studied topic in the computer vision area. The image classification algorithms are the essential ingredient in many applications, such as, face recognition (Masi et al. 2018), autonomous driving (Bojarski et al. 2016), text recognition (Chen et al. 2020; Ye and Doermann 2015), medical image analysis (Litjens et al. 2017; Feng et al. 2021), product quality inspection (Lei et al. 2018) and so on. In recent years, Convolutional Neural Network (CNN) based classification networks, e.g., AlexNet (Technicolour et al. 2012), VGG-Net (Simonyan and Zisserman 2014), ResNet (He et al. 2016), DenseNet (Huang et al. 2017), SimpleNet (HasanPour et al. 2016), GoogLeNet (Szegedy et al. 2014), MobileNet (Howard et al. 2017), ShuffleNet (Zhang et al. 2017), SqueezeNet (Iandola et al. 2016), MnasNet (Tan et al. 2019), have achieved breakthrough results across a variety of image classification tasks.

Those CNN classifiers achieve high classification performance but often lack straightforward interpretability of the

classifier predictions (Jie et al. 2018; Ras, Haselager, and Gerven 2018; Riccardo et al. 2018). In other words, the classifier acts as a blackbox and does not provide details about why it reaches a specific classification decision.

Some works are developed to explain and debug the machine learning models (Shouling et al. 2019). Most current works (Krause, Perer, and Ng 2016; Krause et al. 2017; Zhang et al. 2019) focus on explaining and diagnosing the deep learning model’s prediction with visual analysis techniques. Based on those visualized diagnoses and analyses, researchers interactively optimize the deep learning models through improving data quality, choosing reliable features, and fine-tuning the model parameters. Another kind of works (Bastani, Kim, and Bastani 2017; Lei et al. 2018) adopted the explainable random forest to approximate the deep learning models and analyze the deep learning model by examining the explainable model. To sum up, all existing deep model explanation and debugging methods require humans interaction. There is a lack of an entirely automatic model diagnosing and treating tool for efficiently and effectively optimize the deep learning models.

In this paper, we put forward a completely automatic model diagnosing and treating tool, termed as Model Doctor, which is based on two discoveries. The first discovery is that the target category is highly correlated with sparse and specific convolutional kernels in the last few layers of the CNN classifier. Meanwhile, there are also some incorrect responses for those corresponding convolutional kernels in the background areas. The second discovery is that adversarial samples are isolated while normal samples are successive in the feature space. So, we devise the gradient aggregation strategy for diagnosing the deficiency of CNN classifiers and treating them automatically.

In the diagnosing stage, the absolute value of the first-order derivatives of the predict category w.r.t. each feature map of each layer are summed into a single correlation index, which denotes the correlation degree between the target category and the corresponding convolutional kernel. To alleviate the disturbance of inaccurate features, we also calculate the relevance index for the disturbed feature maps that are synthesized by adding some small noises to the original feature maps. Then, we aggregate those correlation index values in each layer of the CNN classifier for all training samples of the same category. The aggregated correlation in-

*Corresponding author.

dex values for each convolutional kernel can be seen as the criterion for diagnosing the wrong prediction samples.

In the treating stage, we devise the channel-wise and space-wise constraints for treating CNN classifiers based on the above correlated convolutional kernels distribution for each category. The channel-wise constraint is adopted to restrain the wrong correlation between the convolutional kernels and the target category. The space-wise constraint is adopted to restrain the wrong correlation between the unrelated background features and the target category, which requires extra coarse annotations of the object area. Experiments demonstrate that the proposed Model Doctor can effectively diagnose the possible causes for the failure prediction of the model and improve the accuracy of the CNN classifier effectively.

Our contribution is therefore the first completely automatic model diagnosing and treating tool, termed as Model Doctor. We reveal two discoveries and analyze the relationship between the convolution kernels and categories, which can be used as the criterion for future researches on deep model diagnosing and treating. The gradient aggregation strategy combined with the channel-wise and space-wise constraints is devised for diagnosing and treating CNN classifiers. Extensive experiments demonstrate that the proposed methods effectively improve the accuracy of mainstream CNN classifiers by 1% ~ 5%. It's worth noting that the proposed Model Doctor is built on top of pre-trained CNN classifiers, which is applicable for all existing CNN classifiers.

Related Work

Model Debugging and Explanation. Due to the complex operation mechanism and low transparency of machine learning models, there is usually a lack of reliable causes and reasoning to assist researchers in debugging the models. Some model explanation techniques are developed as tools for model debugging and analysis. Cadamuro, Gilad-Bachrach, and Zhu (2016) proposed a debugging method for machine learning models to identify the training items most responsible for biasing the model towards creating this error. Krause, Perer, and Ng (2016) presented an explanatory debugging method that explains to users how the system made each of its predictions. The user then explains any necessary corrections back to the learning system. Kulesza et al. (2010) presented an explanatory debugging approach for debugging machine-learned programs. Brooks et al. (2015) presented an interactive visual analytics tool FeatureInsight for building new dictionary features (semantically related groups of words) for text classification problems. Paiva et al. (2015) proposed a visual data classification methodology that supports users in tasks related to categorization such as training set selection, model creation, verification, and classifier tuning. The above methods focused on debugging traditional machine learning models. The improvement process needs human interaction.

For the deep learning models, Bastani, Kim, and Bastani (2017); Lei et al. (2018) adopted the explainable random forests to approximate the blackbox model and debug the blackbox model by examining the explainable model. Krause, Perer, and Ng (2016) proposed an interactive partial

dependence diagnostics to understand how features affect the prediction overall. Krause et al. (2017) proposed the visual analytics workflow that leverages instance-level explanations, measures of local feature relevance that explain single instances. Zhang et al. (2019) proposed a framework that utilizes visual analysis techniques to support the interpretation, debugging and comparison of machine learning models in an interactive manner. Those methods focused on diagnosing the possible causes for models' prediction, based on which researchers handle the following model debugging and optimization manually. Different from the above methods, we focus on automatically treating the deep models based on the diagnosed results.

Attribution Method. Existing attribution methods contain perturbation-based and backpropagation-based methods. Perturbation-based methods (Zeiler and Fergus 2014; Zhou and Troyanskaya 2015; Zintgraf et al. 2017; Lengerich et al. 2017) directly compute the attribution of an input feature by removing, masking, or altering them and running a forward pass on the new input, measuring the difference with the actual output.

For the backpropagation-based method, Baehrens et al. (2010) first applied the first-order derivative of the predicted category w.r.t. the input to explain classification decisions of the Bayesian classification task. Furthermore, Simonyan, Vedaldi, and Zisserman (2013) extended the same technique into the CNN classification network to extract class aware saliency maps. Sundararajan, Taly, and Yan (2017) took the (signed) partial derivatives of the output w.r.t the input and multiplying them with the input itself (i.e., Gradient \times Input), which improves the sharpness of the attribution maps. Shrikumar et al. (2016) computed the average gradient while the input varies along a linear path from a baseline, which is defined by the user and often chosen to be zero, to the initial input (i.e., Integrated Gradients). Shrikumar, Greenside, and Kundaje (2017) recently proposed DeepLift for decomposing the output prediction of a neural network on a specific input by backpropagating the contributions of all neurons in the network to every feature of the input. Bach et al. (2015); Feng et al. (2018) proposed an approach for propagating importance scores called Layerwise Relevance Propagation (LRP). Kindermans et al. (2016) showed that the original LRP rules were equivalent within a scaling factor to the gradient \times input.

With the above techniques, Song et al. (2019) adopted the Saliency Map, Gradient \times Input, and ϵ -LRP to calculate the attribution map for estimating the transferability of deep networks. Ancona et al. (2018) analyzed above Gradient \times Input, ϵ -LRP, Integrated Gradients and DeepLIFT (rescale) from theoretical and practical perspectives, which shows that these four methods, despite their different formulation, are firmly related, proving conditions of equivalence or approximation between them.

CAM and Grad-CAM. Another related area is visual feature localization, which includes gradient-free technique and gradient-based technique. For the gradient-free technique, Zhou et al. (2016) proposed the first visual feature localization technique, called Class Activation Mapping (CAM),

to visualize the predicted class scores on any given image, highlighting the discriminative object parts detected by the CNN. CAM computes a weighted sum of the feature maps of the last convolutional layer to obtain the class activation maps. Wang et al. (2020) proposed the score-CAM, a novel CAM variant, which uses the increase in confidence for the weight of each activation map. Desai and Ramaswamy (2020) introduced an enhanced visual explanation for visual sharpness called SS-CAM, which produces centralized localization of object features within an image through a smooth operation. Desai and Ramaswamy (2020) proposed the Ablation-CAM that uses ablation analysis to determine the importance (weights) of individual feature map units w.r.t. class, which is time-consuming.

For the gradient-based technique, Selvaraju et al. (2020) proposed the Grad-CAM, which utilizes a local gradient to represent the linear weight and can be applied to any average pooling-based CNN architectures without re-training. Chattopadhyay et al. (2018) proposed the Grad-CAM++ that used a weighted combination of the positive high order derivatives of the last convolutional layer feature maps w.r.t a specific class score as weights to generate better object localization as well as explaining occurrences of multiple object instances in a single image. Omeiza et al. (2019) proposed the Smooth Grad-CAM++ that calculates these maps by averaging gradients from many small perturbations of a given image and applying the resulting gradients in the generalized Grad-CAM algorithm.

Deconvolutional Visualization. The deconvolutional network (Zeiler et al. 2010; Zeiler, Taylor, and Fergus 2011) was originally proposed to learning representation in an unsupervised manner and later applied to visualization (Zeiler and Fergus 2014). Zeiler and Fergus (2014) proposed the first deconvolution visualization approach DeConvNet to better understand what the higher layers in a given network have learned. DeConvNet makes data flow from a neuron activation in the higher layers down to the image. Springenberg et al. (2014) extended this work to guided backpropagation which helped understand the impact of each neuron in a deep network w.r.t. the input image. Mahendran and Vedaldi (2016) extended DeConvNet to a general method for architecture reversal and visualization.

Method

Discovery and Analysis

Discovery 1: *The target category is only correlated with sparse and specific convolutional kernels in the last few layers of the CNN classifier. Meanwhile, unrelated background features will disturb the prediction of the CNN classifier.*

Analysis: In this section, we first give the analysis between the convolutional kernels with the target category. For the input image I , the feature maps of the $(r-1)$ -th layer are denoted as $\{m_1^{r-1}, m_2^{r-1}, m_3^{r-1}, \dots, m_T^{r-1}\}$, as shown in Fig. 1. With the convolutional kernels $\{c_1^r, c_2^r, c_3^r, \dots, c_K^r\}$, feature maps $\{m_1^r, m_2^r, m_3^r, \dots, m_K^r\}$ of the r -th layer are calculated as follows:

$$m_k^r = f(c_k^r \otimes \{m_1^{r-1}, m_2^{r-1}, m_3^{r-1}, \dots, m_T^{r-1}\}), \quad (1)$$

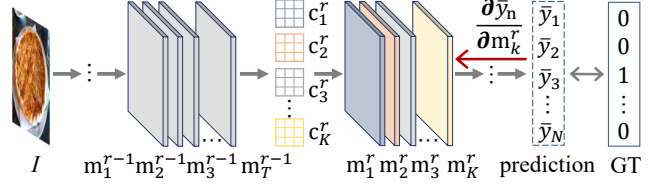


Figure 1: The relationship between the predicted category and feature maps (convolutional kernels) in each layer.

where \otimes denotes the convolutional operation, f denotes the following operation, such as pooling operation, activation function. With feature maps $\{m_1^r, m_2^r, m_3^r, \dots, m_K^r\}$ of the r -th layer, the predicted category \bar{y}_n is denoted as $\bar{y}_n = F(\{m_1^r, m_2^r, m_3^r, \dots, m_K^r\})$, where F denotes the following operations: convolution, pooling, activation function, and fully connected layer, decided by the specific network architecture.

For feature maps $\{m_1^r, m_2^r, m_3^r, \dots, m_K^r\}$ of the r -th layer, the first-order derivative $(\bar{y}_n)'_{m_k^r}$ of the predicted category \bar{y}_n w.r.t the k -th feature map m_k^r is calculated as follows:

$$(\bar{y}_n)'_{m_k^r} = \frac{\partial \bar{y}_n}{\partial m_k^r}. \quad (2)$$

The magnitude of the derivative $(\bar{y}_n)'_{m_k^r}$ indicates which feature value in m_k^r needs to be changed the least to affect the class score \bar{y}_n the most.

Then, the summed value $a_k^{n,r}$ of k -th first-order derivative $(\bar{y}_n)'_{m_k^r}$ is calculated as follows:

$$a_k^{n,r} = \bigoplus \left| \frac{\partial \bar{y}_n}{\partial m_k^r} \right|, \quad (3)$$

where \bigoplus denotes summing values of the matrix $\left| \frac{\partial \bar{y}_n}{\partial m_k^r} \right|$. The magnitude of $a_k^{n,r}$ indicates the importance of the feature map m_k^r to affect the class score \bar{y}_n . From Eqn. (1), we can see that, with the same input feature maps $\{m_1^{r-1}, m_2^{r-1}, m_3^{r-1}, \dots, m_T^{r-1}\}$ and operation f , the convolutional kernel c_k^r determines the different feature maps $\{m_1^r, m_2^r, m_3^r, \dots, m_K^r\}$. So, the magnitude of $a_k^{n,r}$ also indicates the degree of correlation between the k -th convolutional kernel and the predicted category \bar{y}_n .

Fig. 2(a) shows the statistical correlation degree between all categories and the convolutional kernels in the last layer of GoogLeNet (Szegedy et al. 2015) on the CIFA10 dataset. For each category, we calculate the sum value $\sum_1^{100} a_k^{n,r}$ for each convolutional kernel c_k^r of 100 images, prediction confidence of which are larger than 0.90. From Fig. 2(a), we can observe that each category is only correlated with sparse and specific convolutional kernels (bright color), while most convolutional kernels are not correlated (dark color). For each image, the convolutional kernels with high correlation are almost identical with the statistical high correlation kernels in each layer of the network. Furthermore, the number of convolutional kernels with high correlation reduces along with the network layer goes deeper. More visual correlation diagrams about a single image and each layer are given in the *supplements*.

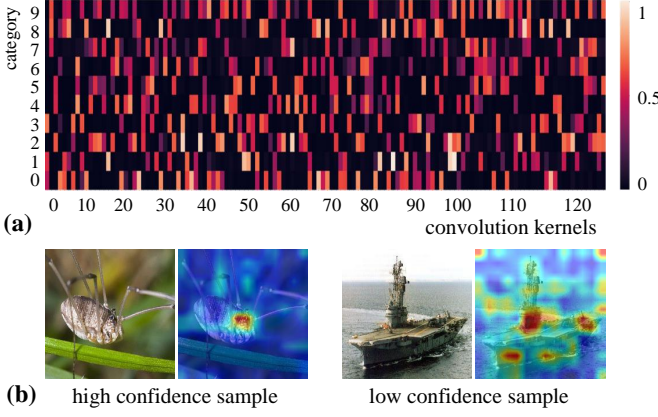


Figure 2: (a) the statistical correlation diagram between all categories and the convolution kernels in last layer of GoogLeNet. (b) the high and low-confidence images covered with summed derivative maps.

For the image with inaccurate prediction, the summed derivative map $\sum_{k=1}^K \left| \frac{\partial \bar{y}_n}{\partial m_k^r} \right|$ is mapped into the original image. Fig. 2(b) shows the high-confidence and low-confidence prediction images and the corresponding image covered with the summed derivative maps. We can find that some background areas have a relatively high correlation. For images with accurate predictions, there are hardly any background areas with high correlations. The above phenomena indicate that those unrelated background features will disturb the prediction of the CNN classifier.

Discovery 2: *Adversarial samples are isolated while normal samples are successive in the feature space.*

Analysis: In the experiment, we find that the performance of the normal sample with disturbances is more robust than the adversarial sample. So, we assume that adversarial samples are isolated while normal samples are successive in the feature space. For the normal and corresponding adversarial samples obtained with fast gradient sign method (Goodfellow, Shlens, and Szegedy 2015), noises of different ranges are added into those samples to verify their robustness with different disturbances. For each image, the noises are randomly added to the arbitrary layer of the network 10 times separately.

Fig. 3 shows the accuracy curves of normal and adversarial samples with different disturbances. The accuracy curves are the average results of 100 samples. From Fig. 3, we can see that a normal sample achieves accuracy close to 100% when the noise range in value from 0 to 0.3. However, with some noises, the prediction of the adversarial sample will turn back to its' original true label and has accuracy close to 100%. Only less than 10% samples are still be predicted to the same false label as the adversarial sample. So, we conclude the secondary discovery that adversarial samples are isolated while normal samples are successive in the feature space.

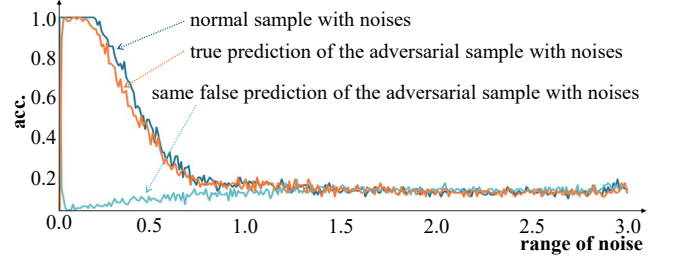


Figure 3: The accuracy curves of normal and adversarial samples with different disturbances.

Model Doctor

Based on the above two discoveries, we put forward a simple gradient aggregation strategy for diagnosing and treating the mainstream CNN classifiers. In the diagnosing phase, we accumulate the correlation between the predicted category and the convolution kernels in each layer. In most of the case, the accurate prediction is only correlated with sparse and specific convolution kernels in the last few layers of the CNN classifier (*discovery 1*). Then, we can adopt the accumulated correlation distribution of each layer to diagnose the cause of why the sample is misclassified. In the treating stage, two constraints are devised for treating CNN classifiers based on diagnosed results and the *discovery 2*. The channel-wise constraint is proposed to restrain the incorrect correlative convolution kernels different from the accumulated correlation distribution. The space-wise constraint is proposed for restraining the incorrect correlation with the background features. The details about diagnosing and treating are elaborated as follows.

Diagnosing stage. Inspired by the *discovery 1*, we accumulate the correlation between the target category and convolution kernels in each layer. To alleviate the disturbance of misclassified features, average statistics are adopted to accumulate the correlations between the target category and the convolution kernels. For the disturbed feature map $m_k^r + \sigma$, the noise matrix σ sampled from the interval $[-\delta, \delta]$ will be added to the feature map m_k^r . The average correlation value $\bar{a}_k^{n,r}$ between the convolution kernel c_k^r and the category \bar{y}_n is calculated as follows:

$$\bar{a}_k^{n,r} = \frac{1}{J} \sum_{j=1}^J \left| \frac{\partial \bar{y}_n}{\partial (m_k^r + \sigma^j)} \right|, \quad (4)$$

where σ^j denotes noise matrix sampled at the j -th time.

For convolution kernels in each layer, Eqn.(4) is adopted to calculate the correlation with the target category. For each category, T samples are used for calculating the average correlation distribution, which can illustrate the relation between the target category and the convolution kernels in each layer. T is usually set to 100 according to the experiment results. The illustrated relationship can be used to diagnose the possible reasons for the misclassified samples.

Constraint strategy. Before the treating stage, two constraints are devised for treating CNN classifiers based on the

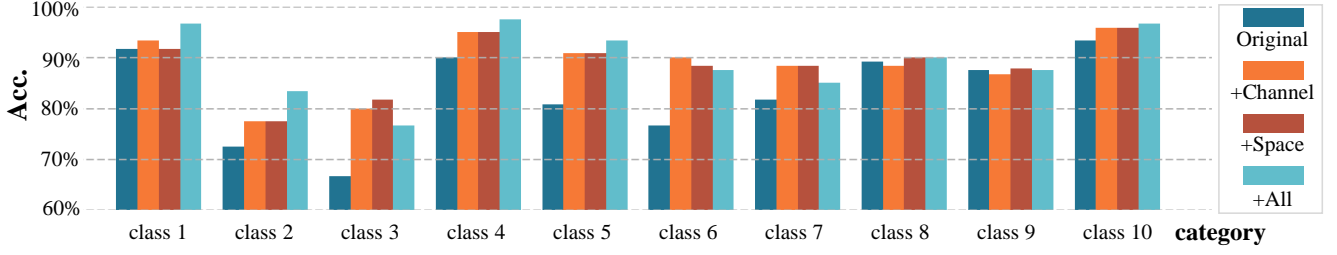


Figure 4: The accuracy and increased accuracy of each category with different constraints on the mini-ImageNet dataset.

accumulated correlation distribution between the target category and the convolution kernels in each layer. It's worth noting that the treating is applied to the trained classifier. For the trained CNN classifier and training samples with high-confidence predictions, the average correlation value $\bar{a}_k^{n,r}$ between each convolution kernel c_k^r and each category \bar{y}_n is firstly calculated. Then, for the training image I with GT label y_n , the r -th feature maps are denoted as $\{m_1^r, m_2^r, m_3^r, \dots, m_K^r\}$ and the predicted label is denoted \bar{y}_n . The channel-wise constraint \mathcal{L}_{ch}^r on the r -th layer for the image I is denoted as follows:

$$\mathcal{L}_{ch}^r = \sum_{k=1}^K \mathbf{1}[\bar{a}_k^{n,r} < v] * \sum_{j=1}^J \biguplus \left| \frac{\partial \bar{y}_n}{\partial (m_k^r + \sigma^j)} \right| + \sum_{k=1}^K \sum_{j=1}^J \biguplus \left| \frac{\partial \bar{y}_s}{\partial (m_k^r + \sigma^j)} \right|, \quad (5)$$

where, the first term is used for restraining the wrong correlation between the predicted category \bar{y}_n and convolution kernels of r -th layer that are different from average correlation, the second term is used to restrain all the correlations between the second high-confidence prediction \bar{y}_s and the convolution kernels of the r -th layer, K denotes the number of the convolution kernel in each layer, σ^j denotes noise matrix sampled at the j -th time, $\mathbf{1}[\bar{a}_k^{n,r} < v]$ equals 1 if $\bar{a}_k^{n,r}$ is less than the threshold value v ; otherwise equals 0.

Furthermore, the space-wise constraint \mathcal{L}_{sp}^r on the r -th layer is proposed for restraining the incorrect correlation with the background features, which is formulated as follows:

$$\mathcal{L}_{sp}^r = \biguplus \sum_{k=1}^K \sum_{j=1}^J I_{bg}^r \otimes \left| \frac{\partial \bar{y}_n}{\partial (m_k^r + \sigma^j)} \right|, \quad (6)$$

where, I_{bg}^r denotes the rescaled background mask that has the same size as m_k^r , \otimes denotes the Hadamard product. In the mask I_{bg}^r , the background area has value one, and the mask I_{bg}^r is eroded to preserve the object boundary features. It's worth noting that the space-wise constraint requires additional annotations, which can be rough boundaries.

Treating Stage. In the treating stage, all constraints can be applied to any layer of the CNN classifier. Based on the fact that deep layers of the CNN classifier usually contain high-level semantic features, the space-wise and channel-wise constraints are adopted to constrain the deep layers.

The following ablation study on layer depth also verifies the practicability and effectiveness of the above constraining way. Furthermore, the above two constraints are usually appended to the original training loss \mathcal{L}_{orig} as follows:

$$\mathcal{L}_{all} = \mathcal{L}_{orig} + \sum_{r \in \mathbb{S}_{sp}} \mathcal{L}_{sp}^r + \sum_{r \in \mathbb{S}_{ch}} \mathcal{L}_{ch}^r, \quad (7)$$

where, \mathbb{S}_{sp} and \mathbb{S}_{ch} denote the constrained layer sets of space-wise constraint and channel-wise constraint, respectively. The two constraints also could be appended to the original training loss separately. In the treating stage, the parameter setting and the optimizer are the same as the setting of the original CNN classifier.

Experiments

In the experiment, the adopted classifiers, datasets, and experiment settings are listed as follows.

Classifier. The selected 16 classifiers cover mainstream classification network architectures, which are listed as follows: AlexNet (Technicolor et al. 2012), VGG-16 (Simonyan and Zisserman 2014), ResNet-34 (He et al. 2016), ResNet-50 (He et al. 2016), WideResNet-28 (Zagoruyko and Komodakis 2016), ResNeXt-50 (Xie et al. 2017), DenseNet-121 (Huang et al. 2017), SimpleNet-v1 (Hasan-Pour et al. 2016), EfficientNetV2-S (Tan and Le 2021), GoogLeNet (Szegedy et al. 2014), Xception (Chollet 2017), MobileNetV2 (Howard et al. 2017), Inception-v3 (Szegedy et al. 2016), ShuffleNetV2 (Zhang et al. 2017), SqueezeNet (Iandola et al. 2016), and MnasNet (Tan et al. 2019).

Dataset. The datasets we adopted contain MNIST (Lecun et al. 2001), Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), CIFAR-10, CIFAR-100 (Krizhevsky 2009), SVHN (Netzer et al. 2011), STL-10 (Coates, Ng, and Lee 2011) and mini-ImageNet (Vinyals et al. 2016), which are commonly used datasets for the classification task.

Experiment setting. Unless stated otherwise, the default experiment settings are given as follows: $J = 10$, $T = 100$, the expanded pixel number is 30. More experiment details are given in the *supplements*.

The Effect of Model Doctor for SOTA Classifiers

In this section, we conduct massive experiments of 16 mainstream CNN classifiers on 7 datasets. All results are averages of three runs. For the space-wise constraint, we only annotated 20 low-confidence images for each category.

Dataset	MNIST	Fashion-MNIST	CIFAR-10	CIFAR-100	SVHN	STL-10	mini-ImageNet
AlexNet / +All	99.60 / -	93.32 / -	86.32 / -	55.04 / -	93.44 / -	67.59 / +0.66	76.92 / +3.84
+Space / +Channel	-1.63 / +0.01	-1.44 / +0.09	-1.91 / +0.31	-6.81 / +1.87	-1.65 / +0.08	+0.59 / +0.62	+3.74 / +3.21
VGG-16 / +All	99.71 / -	95.21 / -	93.46 / -	70.39 / -	94.72 / -	77.65 / +0.79	83.00 / +5.92
+Space / +Channel	-0.98 / +0.03	-1.91 / +0.10	-2.52 / +0.74	-7.92 / +1.08	-1.67 / +0.07	+0.69 / +0.73	+5.58 / +5.50
ResNet-50 / +All	99.73 / -	95.33 / -	94.85 / -	77.08 / -	94.81 / -	82.14 / +0.63	90.25 / +3.71
+Space / +Channel	-1.20 / +0.00	-0.61 / +0.15	-3.44 / +0.54	-4.70 / +0.95	-0.92 / +0.04	+0.61 / +0.58	+3.61 / +3.59
SENet-34 / +All	99.75 / -	95.35 / -	94.76 / -	74.95 / -	94.67 / -	81.67 / +0.92	89.23 / +2.73
+Space / +Channel	-0.88 / +0.01	-3.63 / +0.15	-1.89 / +0.41	-6.21 / +1.22	-1.30 / +0.08	+0.87 / +0.73	+2.12 / +2.63
WideResNet-28 / +All	99.47 / -	93.81 / -	94.26 / -	77.48 / -	94.11 / -	79.34 / +0.85	88.47 / +3.26
+Space / +Channel	-1.72 / +0.19	-3.70 / +1.45	-2.94 / +0.35	-5.01 / +0.39	-2.30 / +0.02	+0.82 / +0.81	+3.24 / +3.15
ResNeXt-50 / +All	99.69 / -	95.37 / -	94.34 / -	74.76 / -	94.25 / -	83.21 / +0.49	89.72 / +3.42
+Space / +Channel	-1.34 / +0.01	-2.01 / +0.19	-1.77 / +1.21	-6.05 / +2.26	-1.81 / +0.03	+0.34 / +0.47	+3.19 / +3.21
DenseNet-121 / +All	99.72 / -	95.43 / -	95.22 / -	76.92 / -	95.18 / -	84.03 / +0.91	89.83 / +3.29
+Space / +Channel	-0.70 / +0.01	-3.38 / +0.00	-3.81 / +0.52	-5.79 / +0.75	-0.94 / +0.01	+0.84 / +0.69	+2.98 / +2.97
SimpleNet-v1 / +All	99.72 / -	95.39 / -	94.61 / -	75.29 / -	94.51 / -	81.92 / +0.67	87.92 / +2.19
+Space / +Channel	-1.28 / +0.01	-2.75 / +0.11	-3.67 / +0.80	-4.93 / +1.51	-1.36 / +0.07	+0.66 / +0.53	+1.97 / +1.82
EfficientNetV2-S / +All	99.66 / -	93.82 / -	91.07 / -	61.01 / -	92.14 / -	79.32 / +0.78	86.33 / +4.84
+Space / +Channel	-1.11 / +0.05	-1.21 / +0.08	-1.09 / +0.33	-3.91 / +2.65	-1.55 / +0.05	+0.55 / +0.71	+4.73 / +4.67
GoogLeNet / +All	99.75 / -	95.20 / -	94.36 / -	75.28 / -	94.20 / -	83.33 / +0.93	91.33 / +2.32
+Space / +Channel	-0.90 / +0.01	-2.98 / +0.17	-4.67 / +0.58	-6.34 / +0.71	-0.82 / +0.13	+0.63 / +0.92	+1.95 / +2.21
Xception / +All	99.75 / -	95.23 / -	93.35 / -	74.15 / -	92.38 / -	79.62 / +0.96	84.25 / +3.81
+Space / +Channel	-1.04 / +0.02	-1.40 / +0.05	-3.81 / +0.57	-5.45 / +0.42	-1.20 / +0.04	+0.84 / +0.93	+4.52 / +4.72
MobileNetV2 / +All	99.68 / -	94.58 / -	91.95 / -	68.26 / -	90.43 / -	78.92 / +0.67	86.21 / +4.79
+Space / +Channel	-2.83 / +0.00	-2.91 / +0.39	-2.08 / +1.07	-7.91 / +2.75	-0.73 / +0.08	+0.65 / +0.52	+4.61 / +4.51
Inception-v3 / +All	99.70 / -	95.25 / -	94.77 / -	76.93 / -	93.54 / -	89.46 / +0.78	89.41 / +4.10
+Space / +Channel	-1.20 / +0.06	+1.29 / +0.08	-2.56 / +0.52	-7.24 / +1.20	-1.72 / +0.01	+0.72 / +0.69	+3.83 / +3.70
ShuffleNetV2 / +All	99.69 / -	94.63 / -	92.28 / -	67.39 / -	92.47 / -	75.21 / +0.64	87.08 / +2.69
+Space / +Channel	-1.01 / +0.00	-3.62 / +0.19	-2.81 / +0.64	-3.23 / +0.43	-1.26 / +0.06	+0.56 / +0.43	+2.60 / +2.57
SqueezeNet / +All	99.71 / -	94.74 / -	91.93 / -	67.87 / -	92.65 / -	78.75 / +0.59	88.58 / +3.52
+Space / +Channel	-0.92 / +0.01	-1.45 / +0.18	-2.76 / +0.92	-6.81 / +0.60	-0.98 / +0.06	+0.57 / +0.59	+3.19 / +3.01
MnasNet / +All	99.66 / -	93.53 / -	85.55 / -	53.60 / -	91.08 / -	74.32 / +0.31	87.33 / +3.49
+Space / +Channel	-1.14 / +0.01	-2.53 / +0.03	-2.58 / +1.37	-3.90 / +0.50	-1.04 / +0.03	+0.29 / +0.19	+2.08 / +3.27

Table 1: The base and improved accuracy of 16 classifiers on 7 datasets. For the small-size image, due to the expanded receptive field of convolution operation, the feature map of the last few layers will contain mixture features of object and backgrounds, which leads to negative effective of the space-wise constraint. So, “+All” are not applied to the small-size image datasets.

Of course, the classifier can achieve higher accuracy with more annotations. Before diagnose and treating, all classifiers have achieved the optimal classification performance in the setting of original training. From Table 1, we can see that Model Doctor helps all CNN classifiers improve accuracy by 1% ~ 5%, which verifies the versatility and effectiveness of the proposed gradient aggregation strategy. For datasets with small-size images, only the channel-wise constraint can improve the accuracy. On the contrary, the space-wise constraint has negative effects on those small-size image datasets. The root reason is that the expanded receptive field of convolution operation brings the mixture features of objects and backgrounds in the feature map of the last few layers, which leads to the wrong scope constraint of the small-size image. What’s more, Model Doctor has little effect on high-performance networks such as, 99.7% accuracy on MNIST dataset. For the regular size image datasets (STL-10 and mini-ImageNet), the space-wise constraint achieves a significant accuracy increase, which indicates that background area features indeed have some distractions for clas-

sifiers. Model Doctor only achieves accuracy close to 1% on STL-10. The possible reason is that the object almost takes up the whole image.

To verify the effectiveness of Model Doctor on various categories, we conduct experiments on randomly selected 10 categories 10 times for the mini-ImageNet dataset, which contains 100 categories and 480 samples for each category. Table 1 shows the average results of 10 times. More experiments on mini-ImageNet are given in the *supplements*. Fig. 4 shows the accuracy and increased accuracy of VGG16 on a 10 category classification task. We can see that Model Doctor can improve higher accuracy increase on low accuracy categories, which indicates that Model Doctor can accurately diagnose the classifier failures and effectively treat them.

Visual Results

Furthermore, Fig. 5 shows the visual results of original images with different constraints and Grad-CAM for visualizing the differences intuitively. We can see that the space-wise constraint successfully restrains the irrelevant back-

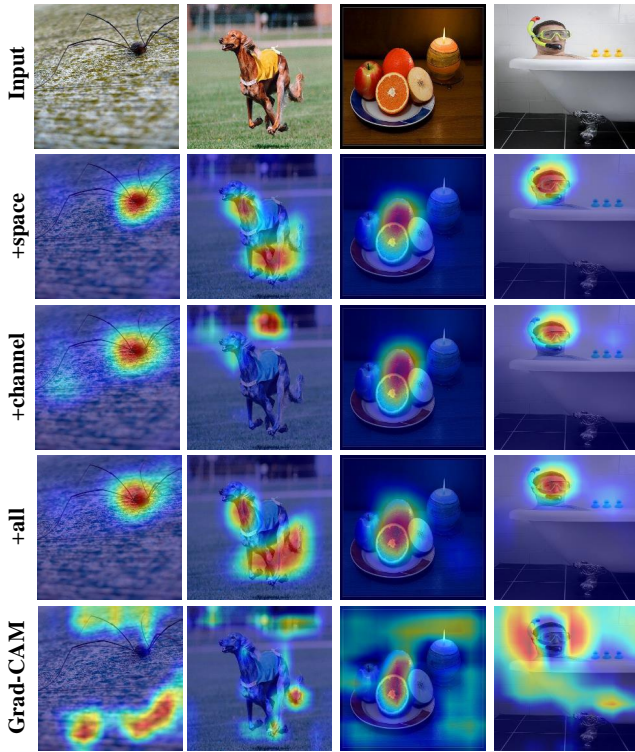


Figure 5: Visual results of original images with different constraints and Grad-CAM on mini-ImageNet datasets.

ground areas. For the visual results of channel-wise constraint, most irrelevant areas are also restrained. The association maps calculated with Grad-CAM contain some background areas. The proposed Model Doctor not only can improve the classification accuracy but also regularize the association between the category and the input features.

The Ablation Study

This section conducts the ablation study of channel-wise constraint and space-wise constraint on mini-ImageNet with VGG16. The *layer depth impact* for different constraints is shown in Fig. 6, where we can see that the increased accuracy of all constraints will increase when the layer goes deep. It reveals that the proposed Model Doctor is more suitable for deep layers of classifiers. The possible reason is that the deep and shallow layers contain more high-level semantic features and low-level features, respectively. Constraining shallow layer will disturb feature extraction ability of classifier on essential features.

Furthermore, we conduct the ablation study on the *annotated sample number* and the *expanded pixel number* of the background mask. From Table 2, we can see that 12% annotated samples for space-wise constraint have achieved near-optimal accuracy increase, which indicates that only small part annotations are enough for the space-wise constraint. Table 3 shows that 30-pixel expansion achieves the best performance and the expanded mask with less than 5 pixels achieves a negative effect, the reason of which is that boundary features are crucial for the classification task.

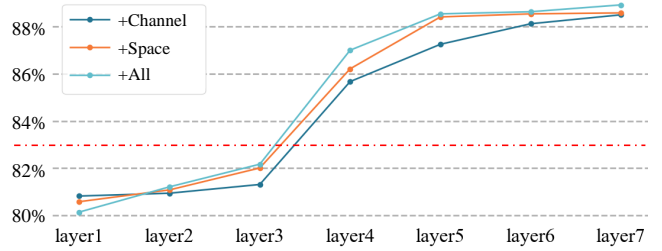


Figure 6: The ablation study of constraints on layer depth. Red dash line denotes the original accuracy (83%).

VGG16	Number Ratio	100	200	400	600	800	1000
	+Space	+3.22%	+5.58%	+6.28%	+7.53%	+7.96%	+8.03%

Table 2: The ablation study of space-wise constraint on annotation samples.

VGG16	Pixel	0	5	10	20	30	40
	+Space	-7.48%	+0.43%	+3.45%	+4.23%	+5.58%	+4.07%

Table 3: The ablation study of space-wise constraint on mask expansion.

Discussion and Future Work

From Table 1, we can see that the treated classifier still can't achieve 100% accuracy. The possible reasons contain multiple aspects: 1) hard samples have different correlations with convolution kernels from the correlations of normal samples; 2) the training samples are not sufficient for the classifier; 3) the classifier may be over-/under-fitting. We will explore actual causes in future work.

In this paper, we verify Model Doctor on mainstream CNN classifiers. Actually, Model Doctor can be extended to a unified framework for diagnosing and treating deep models for different tasks. We will focus on exploring the extension of Model Doctor on segmentation, detection, and key points identification tasks in the future.

Conclusion

In this paper, we put forward a universal Model Doctor for diagnosing and treating CNN classifiers in a completely automatic manner. Firstly, we explore and validate two discoveries that 1) each category is only correlated with sparse and specific convolution kernels, and 2) adversarial samples are isolated while normal samples are successive in the feature space, which can serve as grounds for future researches. Based on the above two discoveries, a simple gradient aggregation strategy is devised for effectively diagnosing and optimizing CNN classifiers. Massive experiments reveal that the devised two constraints are suitable for deep layers of the classifier. For trained classifiers, the proposed Model Doctor can effectively help them improve accuracy by 1% ~ 5%. Model Doctor can be used as a convenient tool for researchers to optimize their CNN classifiers.

Acknowledgments

This work is funded by the National Key R&D Program of China (Grant No: 2018AAA0101503) and the Science and technology project of SGCC (State Grid Corporation of China): fundamental theory of human-in-the-loop hybrid-augmented intelligence for power grid dispatch and control.

References

- Ancona, M.; Ceolini, E.; Zitelli, C.; and Gross, M. 2018. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *ICLR*.
- Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K. R.; and Samek, W. 2015. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, 10(7): 130140.
- Baehrens, D.; Schroeter, T.; Harmeling, S.; Kawanabe, M.; Hansen, K.; and Müller, K.-R. 2010. How to Explain Individual Classification Decisions. *Journal of Machine Learning Research*, 11(61): 1803–1831.
- Bastani, O.; Kim, C.; and Bastani, H. 2017. Interpretability via Model Extraction. *arXiv preprint arXiv:1706.09773*.
- Bojarski, M.; Testa, D. D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L. D.; Monfort, M.; Müller, U.; Zhang, J.; Zhang, X.; Zhao, J.; and Zieba, K. 2016. End to End Learning for Self-Driving Cars. *CoRR*, abs/1604.07316.
- Brooks, M.; Amershi, S.; Lee, B.; Drucker, S. M.; Kapoor, A.; and Simard, P. 2015. FeatureInsight: Visual support for error-driven feature ideation in text classification. In *VAST*, 105–112.
- Cadamuro, G.; Gilad-Bachrach, R.; and Zhu, X. 2016. Debugging machine learning models. In *ICML Workshop*.
- Chattopadhyay, A.; Sarkar, A.; Howlader, P.; and Balasubramanian, V. N. 2018. Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. In *WACV*, 839–847.
- Chen, X.; Jin, L.; Zhu, Y.; Luo, C.; and Wang, T. 2020. Text Recognition in the Wild: A Survey. *CoRR*, abs/2005.03492.
- Chollet, F. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. In *CVPR*, 1800–1807.
- Coates, A.; Ng, A. Y.; and Lee, H. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, 215–223.
- Desai, S.; and Ramaswamy, H. G. 2020. Ablation-CAM: Visual Explanations for Deep Convolutional Network via Gradient-free Localization. In *WACV*, 983–991.
- Feng, Y.; Wu, F.; Shao, X.; Wang, Y.; and Zhou, X. 2018. Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network. In *ECCV*, 557–574.
- Feng, Z.; Wang, Z.; Wang, X.; Zhang, X.; Cheng, L.; Lei, J.; Wang, Y.; and Song, M. 2021. Edge-competing Pathological Liver Vessel Segmentation with Limited Labels. In *AAAI*.
- Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR*.
- HasanPour, S. H.; Rouhani, M.; Fayyaz, M.; and Sabokrou, M. 2016. Lets keep it simple, Using simple architectures to outperform deeper and more complex architectures. *arXiv*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. *CVPR*.
- Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CVPR*.
- Huang, G.; Liu, Z.; Laurens, V.; and Weinberger, K. Q. 2017. Densely Connected Convolutional Networks. In *CVPR*.
- Iandola, F. N.; Han, S.; Moskewicz, M. W.; Ashraf, K.; Dally, W. J.; and Keutzer, K. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and 0.5MB model size. *ICLR*.
- Jie, L.; Zhe, W.; Zunlei, F.; Mingli, S.; and Jiajun, B. 2018. Understanding the prediction process of Deep Networks by Forests. *ICMBD*.
- Kindermans, P.-J.; Schütt, K.; Müller, K.-R.; and Döhne, S. 2016. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270*.
- Krause, J.; Dasgupta, A.; Swartz, J.; Aphinyanaphongs, Y.; and Bertini, E. 2017. A Workflow for Visual Diagnostics of Binary Classifiers using Instance-Level Explanations. In *VAST*, 162–172.
- Krause, J.; Perer, A.; and Ng, K. 2016. Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models. In *HFCSS*, 5686–5697.
- Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images.
- Kulesza, T.; Stumpf, S.; Burnett, M.; Wong, W.-K.; Riche, Y.; Moore, T.; Oberst, I.; Shinsell, A.; and McIntosh, K. 2010. Explanatory Debugging: Supporting End-User Debugging of Machine-Learned Programs. In *SVLHCC*, 41–48.
- Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 2001. Gradient-based learning applied to document recognition. *Intelligent Signal Processing*, 306–351.
- Lei, J.; Gao, X.; Feng, Z.; Qiu, H.; and Song, M. 2018. Scale Insensitive and Focus Driven Mobile Screen Defect Detection in Industry. *Neurocomputing*, 294: 72–81.
- Lei, J.; Wang, Z.; Feng, Z.; Song, M.; and Bu, J. 2018. Understanding the Prediction Process of Deep Networks by Forests. In *BigMM*, 1–7.
- Lengerich, B. J.; Konam, S.; Xing, E. P.; Rosenthal, S.; and Veloso, M. M. 2017. Visual Explanations for Convolutional Neural Networks via Input Resampling. *arXiv preprint arXiv:1707.09641*.
- Litjens, G.; Kooi, T.; Bejnordi, B. E.; Setio, A.; and Sanchez, C. 2017. A Survey on Deep Learning in Medical Image Analysis. *MIA*, 42(9): 60–88.
- Mahendran, A.; and Vedaldi, A. 2016. Salient Deconvolutional Networks. In *ECCV*.
- Masi, I.; Yue, W.; Hassner, T.; and Natarajan, P. 2018. Deep Face Recognition: A Survey. In *CGPI*.

- Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading Digits in Natural Images with Un-supervised Feature Learning. *NeurIPS Workshop*.
- Omeiza, D.; Speakman, S.; Cintas, C.; and Weldemariam, K. 2019. Smooth Grad-CAM++: An Enhanced Inference Level Visualization Technique for Deep Convolutional Neural Network Models. *arXiv preprint arXiv:1908.01224*.
- Paiva, J. G. S.; Schwartz, W. R.; Pedrini, H.; and Minghim, R. 2015. An Approach to Supporting Incremental Visual Data Classification. *TVCG*, 21(1): 4–17.
- Ras, G.; Haselager, P.; and Gerven, M. V. 2018. Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges. *The Springer Series on Challenges in Machine Learning*.
- Riccardo, G.; Anna, M.; Salvatore, R.; Franco, T.; Fosca, G.; and Dino, P. 2018. A Survey Of Methods For Explaining Black Box Models. *ACM Computing Surveys*, 51(5): 1–42.
- Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2020. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *IJCV*, 128(2): 336–359.
- Shouling, J.; Jinfeng, L.; Tianyu, D.; and Bo, L. 2019. Survey on Techniques, Applications and Security of Machine Learning Interpretability. *JCRD*, 56(10): 2071.
- Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. In *ICML*.
- Shrikumar, A.; Greenside, P.; Shcherbina, A.; and Kundaje, A. 2016. Not Just a Black Box: Learning Important Features Through Propagating Activation Differences. *arXiv preprint arXiv:1605.01713*.
- Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *ICLR Workshop*.
- Simonyan, K.; and Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ICLR*.
- Song, J.; Chen, Y.; Wang, X.; Shen, C.; and Song, M. 2019. Deep Model Transferability from Attribution Maps. In *NeurIPS*, volume 32, 6179–6189.
- Springenberg, J.; A, D.; Brox, T.; and Riedmiller, M. 2014. STRIVING FOR SIMPLICITY: THE ALL CONVOLUTIONAL NET. *eprint arxiv*.
- Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *ICML*.
- Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; and Rabinovich, A. 2014. Going Deeper with Convolutions. *CVPR*.
- Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*, 1–9.
- Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *CVPR*, 2818–2826.
- Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; and Le, Q. V. 2019. MnasNet: Platform-Aware Neural Architecture Search for Mobile. In *CVPR*, 2820–2828.
- Tan, M.; and Le, Q. 2021. EfficientNetV2: Smaller Models and Faster Training. In *ICML*, 10096–10106.
- Technicolor, T.; Related, S.; Technicolor, T.; and Related, S. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *NeurIPS*.
- Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching networks for one shot learning. In *NeurIPS*, volume 29, 3637–3645.
- Wang, H.; Wang, Z.; Du, M.; Yang, F.; Zhang, Z.; Ding, S.; Mardziel, P.; and Hu, X. 2020. Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks. In *CVPR Workshops*, 111–119.
- Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747*.
- Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; and He, K. 2017. Aggregated Residual Transformations for Deep Neural Networks. In *CVPR*, 5987–5995.
- Ye, Q.; and Doermann, D. 2015. Text Detection and Recognition in Imagery: A Survey. *TPAMI*, 37(7): 1480–1500.
- Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. In *BMVC*.
- Zeiler, M. D.; and Fergus, R. 2014. Visualizing and Understanding Convolutional Networks. In *ECCV*.
- Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. *ECCV*.
- Zeiler, M. D.; Krishnan, D.; Taylor, G. W.; and Fergus, R. 2010. Deconvolutional networks. In *CVPR*.
- Zeiler, M. D.; Taylor, G. W.; and Fergus, R. 2011. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, 2018–2025.
- Zhang, J.; Wang, Y.; Molino, P.; Li, L.; and Ebert, D. S. 2019. Manifold: A Model-Agnostic Framework for Interpretation and Diagnosis of Machine Learning Models. *TVCG*, 25(1): 364–373.
- Zhang, X.; Zhou, X.; Lin, M.; and Sun, J. 2017. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *CVPR*.
- Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning Deep Features for Discriminative Localization. *IEEE Computer Society*.
- Zhou, J.; and Troyanskaya, O. G. 2015. Predicting effects of noncoding variants with deep learning based sequence model. *Nature Methods*, 12(10): 931–934.
- Zintgraf, L. M.; Cohen, T. S.; Adel, T.; and Welling, M. 2017. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. In *ICLR*.