SN Computer Science (2021) 2:47
https://doi.org/10.1007/s42979-021-00449-3

ORIGINAL RESEARCH

Eigen-CAM: Visual Explanations for Deep Convolutional Neural Networks

Mohammed Bany Muhammad[1] · **Mohammed Yeasin**[1]

Received: 19 August 2020 / Accepted: 2 January 2021 / Published online: 20 January 2021

Abstract

The adoption of deep convolutional neural networks (CNN) is growing exponentially in wide varieties of applications due to exceptional performance that equals to or is better than classical machine learning as well as a human. However, such models are difficult to interpret, susceptible to overfit, and hard to decode failure. An increasing body of literature, such as class activation map (CAM), focused on understanding what representations or features a model learned from the data. This paper presents novel Eigen-CAM to enhance explanations of CNN predictions by visualizing principal components of learned representations from convolutional layers. The Eigen-CAM is intuitive, easy to use, computationally efficient, and does not require correct classification by the model. Eigen-CAM can work with all CNN models without the need to modify layers or retrain models. For the task of generating a visual explanation of CNN predictions, compared to state-of-the-art methods, Eigen-CAM is more consistent, class discriminative, and robust against classification errors made by dense layers. Empirical analyses and comparison with the best state-of-the-art methods show up to 12% improvement in weakly-supervised object localization, an average of 13% improvement in weakly-supervised segmentation, and at least 15% improvement in generic object proposal.

Keywords Class activation maps · Explainable AI · Salient features · Visual explanation of CNN · Weakly supervised localization

Introduction

Convolutional neural networks (CNN) are ubiquitous and are designed to learn representations using deep neural network architecture consisting of multiple building blocks, such as convolution layers, pooling layers, fully connected decision layers. They have shown performance equal or surpassing humans in solving visual tasks such as image recognition [1–3], object localization [4–7], image captioning [8–11], semantic segmentation [12–15], 3D action recognition [16, 17], and visual question answering [18, 19].

Arguably, the deeper the network architecture, the better the accuracy and generalization. A deeper model requires an exponentially growing number of parameters (i.e., hundreds of layers and millions of parameters) to learn complex visual tasks. For example, a one-layer CNN model is capable of learning simple features such as edges. A two-layer CNN model is capable of learning texture features. A three-layer CNN model is capable of learning shape features, and so on. AlexNet is the first CNN and consists of eight layers and 62 M parameters [2]. Nowadays, we have models with layers that exceed 150 layers such as Resnet-152 [1], and models with trainable parameters that exceed 144 million as the case in VGG-19 [20]. Besides depth, non-linear elements and techniques such as activation functions, dropout, MaxPooling, and regularization enable CNN models to learn complex representations.

The ability to learn complex representations translates to achieve higher accuracy and better generalization; on the other hand, it makes it harder to decode model failures and hard to make sense of learned representations. The inability to interpret model predictions and diagnose the model failures remains challenging for both designers and end-users.

There is an increasing demand for tools to interpret DL models in general and CNN-based models in particular.

✉ Mohammed Bany Muhammad
mbnymhmm@memphis.edu

Mohammed Yeasin
myeasin@memphis.edu

[1] The University of Memphis, Memphis, USA

Desirable explanations include class discriminative precise visual explanations consistent with ground truth, consistency in the presence of multiple objects and complex background, and class-independent explanations. Figure 1 shows examples illustrating the ability of Eigen-CAM to generate visual explanations for multiple objects in an image.

This paper extends a preliminary edition of this work presented in [21]. We introduced the Eigen-Salience map, combined with Eigen-CAM to obtain a better and consistent explanation, expanded performance evaluation for different tasks, and presented a framework for decoding prediction failure. The main contributions are:

- We present a simple, intuitive Eigen-CAM to obtain detailed CAM and salience features based on lower and higher convolution layers output.
- We demonstrate the consistency and robustness of the Eigen-CAM over state-of-the-art methods using the following tasks:
  - Weakly supervised localization

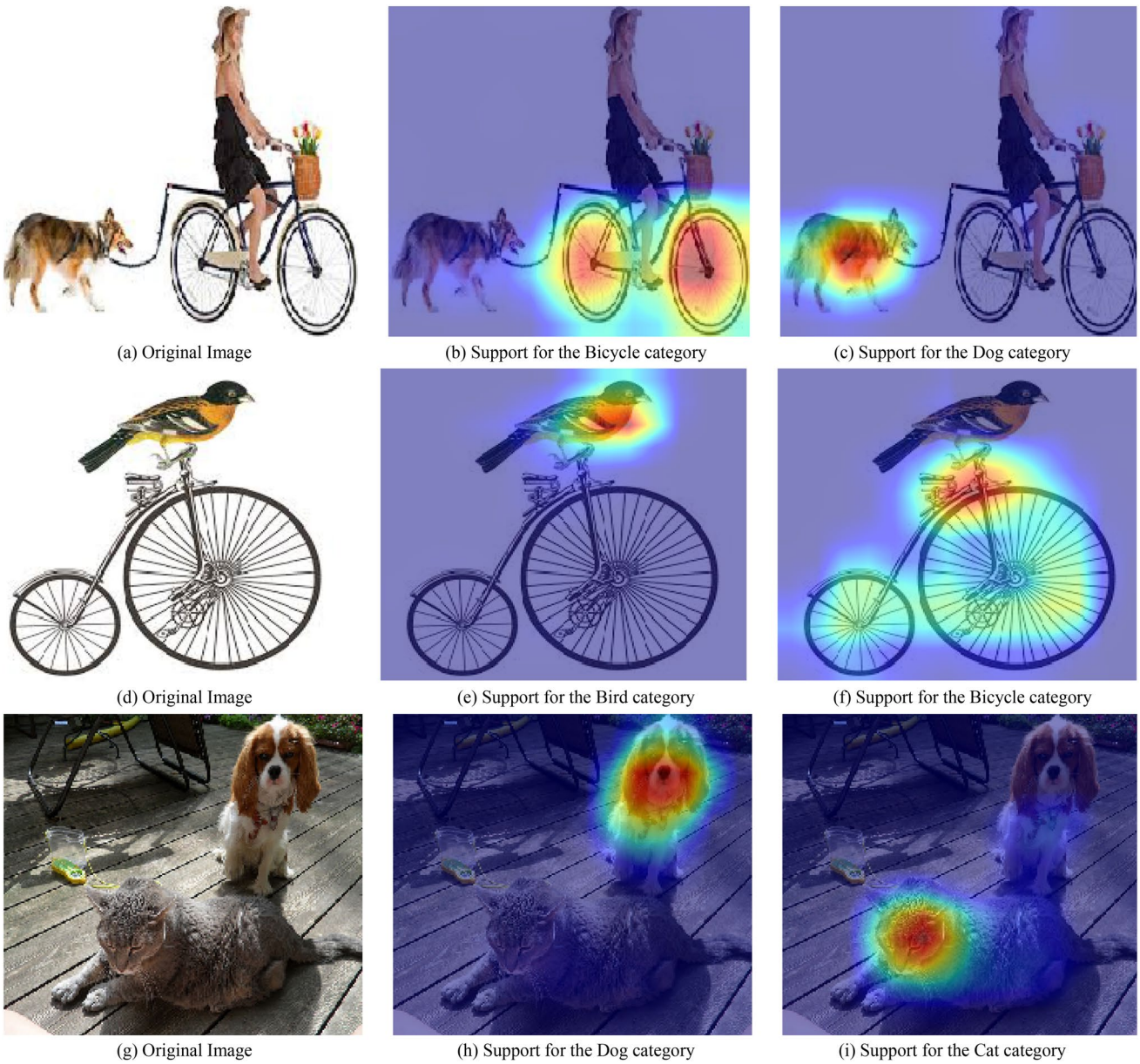

Fig. 1 Eigen-CAM visualizations computed for three sample images. **a**, **d**, **g** [32]. Images. **b**, **c**, **e**, **f**, **h**, **i** Shows class activations computed using Eigen-CAM, middle column images **b**, **e**, **h** show explanation for CNN predictions obtained using the first principal component and right column images **c**, **f**, **i** show an explanation for CNN predictions obtained using the second principal component

- Weakly supervised salient object detection
- Generic object proposal

- We present extensive error analysis that can troubleshoot or trace the error source in CNN architecture and verify the annotation process.

The rest of the paper is structured as follows. In "Related Work", we present related reported literature to provide the research context. Following this, we present the specifics of the proposed Eigen-CAM in "Proposed Approach". Subsequently, in "CNN Prediction Explanations", we evaluate CNN prediction explanations generated using Eigen-CAM against Grad-CAM and CNN-fixation. We present the outcomes of the empirical evaluation of Eigen-CAM and compare and contrast the performance against state-of-the-art methods across different localization applications in "Localization Applications". Results of the error analysis to trace the source of error in CNN architecture and verify the annotation process is presented in "Analysis of CNN Prediction Errors". Finally, "Conclusions" concludes the paper with a few remarks on lessons learned and future directions.

Related Work

In general, all methods that provide visual explanations for CNN prediction utilize different methods to weigh each pixel in the input image to reflect the pixel's relative importance to a specific class level.

Class non-discriminative methods calculate the gradient of the SoftMax layer with respect to each pixel in the input image and use these gradients to represent the salience map. On the contrary, class-discriminative methods identify each pixel's relative importance in the input space and weigh each pixel based on a specific class level made at CNN output.

In the actual process of calculating weights, class discriminative or class non-discriminative visualizations implement a two-step procedure. In the first step, all tools utilize CNN forward propagation (forward pass) to propagate input data from the input to the CNN model's output to calculate the output at a particular layer or the output at the SoftMax layer of the CNN model. In the second step, all methods utilize backpropagation to calculate the weights using different mechanisms, starting from the first step's output. The mechanism used in step two makes all the difference between class discriminative and class non-discriminative visualization and also makes the difference between different methods in the literature.

With class non-discriminative tools, several methods rely on backpropagating gradients to locate salience features. Among the first efforts under this class of visualization is the Saliency map [22].

The saliency map weights the image pixels at the input by backpropagated gradients computed using the SoftMax layer's input with respect to the input pixels. Deconvnet [23], on the other hand, does the same function with one difference characterized by the way Deconvnet handles nonlinearity at activation function, where Deconvnet suppresses negative gradients to enhance visualization. A more recent effort. Guided backpropagation [24] adds more constraints on what gradients are allowed in the backpropagation. The additional constraints enabled the Guided backpropagation to outperform Deconvnet and Saliency map.

The second and more beneficial class of visualization, class-discriminative tools, provides a more intuitive visualization that explains CNN predictions and can provide localization and segmentation functionality.

Tools in this class started with class activation maps (CAM) [25]. CAM method computes the dot product of the extracted weights from the SoftMax layer and the feature map to produce the class activation map. To implement the CAM method, the user needs to modify the model by replacing the last MaxPooling layer with a global average pooling (GAP) layer and eliminate dense layers.

CAM is an intuitive yet straightforward idea that inspired methods like Grad-CAM [26] and Grad-CAM++ [27]. The Grad-CAM used the CNN output gradients to weight the extracted feature at the last CNN convolutional Layer. Using the gradients enabled Grad-CAM to work with any CNN model without the need to modify the architecture. The Grad-CAM++ improved the gradient weighting mechanism to account for different feature sizes, and by that modification, they improved visualization for multiple object occurrences.

Besides the gradient approach in the class-discriminative visualizations, other methods adopted the relevance score-based approach to weight features. For example, deep LIFT [28] and layer-wise relevance propagation [29] backpropagate the probabilities at the output of the SoftMax layer to the last convolutional layer or to the input layer to determine the class-discriminative features.

A more recent method named CNN-Fixations was reported in [30] and inspired by the biological vision and loosely analogous to human eye fixations. CNN-Fixations computes binary relevance scores in layer i for each activation in the higher layer j and later combines them to get a real-valued map at layer i. The method creates a model-specific memorization map, designed to keep track of maximum activations using Hadamard product to discard irrelevant information. CNN-Fixations create visual explanations by backtracking activations from the decision layer to the input image pixel space to locate discriminative features during the forward pass rather than resorting to operations such as gradient computation in saliency maps, CAM and

Grad-CAM. In summary, CNN-Fixations computes the positive feature correlations and leaving the negative ones to create a real-valued relevance for each neuron activation to the predicted inference.

In CNN-Fixations, as the probability of the predicted class increases, the better the fixations are, and as the predicted class lowers towards the second prediction probability, the more confusion or overlap between the fixations that explain the top two objects triggering these predictions.

In general, all class-discriminative methods are class-dependent because they rely on the class score to backpropagate their weighting mechanism. For such methods to work effectively, one must assume a correct decision at the CNN output implicitly. A wrong decision will lead to erroneous or distorted CAM, as shown in Figs. 5 and 6. Furthermore, backpropagating gradients requires resources for computation and memory.

We present Eigen-CAM that is intuitive and compatible with all CNN models without any model modifications to address the shortcomings mentioned above.

Proposed Approach

CNN-based deep neural networks outperform all other methods in a range of computer vision tasks [31]. The CNN model's basic structure consists of a convolutional network, represented by layers of filters of varying sizes to learn the representations and classification networks exemplified by dense layers to differentiate learned representations.

The first convolutional layer's role is to learn lower-level spatial representation (features) such as edges and corners. The hierarchical structure allows convolutional layers to learn higher levels of abstraction and possibly features that can produce semantic meaning at a categorical level, such as classification and annotation of objects.

At the top of the hierarchy (last convolutional layer), learned features proceed to the classification networks (dense layers in the CNN model). On the other hand, a classification network's role is to learn the decision boundary to categorize objects or attach semantic meaning to the data. To understand the need and intuition behind Eigen-CAM, let us consider the following observations.

Observation 1 Previously reported visualization methods mentioned in the related work section depend on the backpropagation by tracing of information such as gradients [26, 27], relevance score [29], and maximum activation locations [30] from the output of the CNN to the desired space to generate visual explanations. These methods implicitly assume "correct decisions" at the models' output layer, which is not always true. The failure of this assumption can lead to incorrect or distorted explanations shown in Fig. 5.

Observation 2 The CAM method outlines the notion of linear combinations to generate the visual explanations of CNN predictions. However, the CAM method requires model modification to work [25]. To eliminate the need to modify CNN model architecture Grad-CAM [26] and Grad-CAM++ [27] use extracted features as variables and back-propagated positive gradients of the class score with respect to the last convolutional layer as coefficients. In other words, linear explanations used to generate visual explanations for CNN predictions include only positive terms only using the ReLU function to suppress negative terms.

Observation 3 The Grad-CAM and Grad-CAM++ have a better explanation capability than CAM. Computed gradients are noisy in nature and depend on the order of approximation. The number of dense layers in the CNN model dictates the order of approximation. Tuning the approximation to reduce noise requires changing the model architecture by increasing or decreasing the number of dense layers.

Observation 4 The learning process in any CNN classifier resembles a mapping function. The mapping is done by a transformation matrix (model) that captures salient features from images using Conv layers. The optimizers play a crucial role in this learning process as they are used to adjust the weights of filters used in convolution layers to learn salient features and the weights of fully connected layers to learn the non-linear decision boundary. Based on this observation, the hierarchical representation mapped onto the last convolutional can simply provide visual explanations for CNN prediction.

A. Eigen class activation maps

The unparalleled CNN performance achieved on various computer vision tasks could not happen due to complete memorization. We can assume that that feature extraction network in CNN architecture (Convolutional and MaxPooling layers) will select and preserve relevant features and smooth out irrelevant or redundant features.

The only relevant question is what features go through all local linear transformations and stay relevant or stay in the same direction of maximum variation. In other words, what features will be in the direction of the principal component of the learned representation?

Let I represent the input image of size $(i \times j) I \epsilon R^{i,j}$, and let $W_{L=n}$ represent the combined weight matrix of the first k layers of size (m, n).

The class activated output is the image I projected onto the last convolution layer $L = k$ and is given by

$$O_{L=k} W_{L=k}^{T} I \tag{1}$$

Factorizing $O_{L=k}$ using singular value decomposition to compute the principal components of $O_{L=k}$ gives

$$O_{L=k} = U\Sigma V^T, \tag{2}$$

where U is an$M \times M$ orthogonal encoding matrix, and the columns of U are the left singular vectors, Σ is a diagonal matrix of size $M \times N$ with singular values along the diagonal, V is an $N \times N$ orthogonal matrix, and the column of V are the left singular vectors.

The class activation map, $L_{\text{Eigen-CAM}}$ is given by the projection of $O_{L=k}$ on the first eigenvector

$$L_{\text{Eigen - CAM}} = O_{L=k} V_1, \tag{3}$$

where V_1 is the first eigenvector in the V matrix.

B. Eigen-Salience maps

Salience maps are a general method used to visualize each pixel's unique quality independent from the predicted class. It functions as a transformation where all data in the input space are re-represented using a lower number of dimensions. This transformation removes redundant and irrelevant features and segregates class relevant features from the background. With this logic, principal component analysis (PCA) is the natural choice to achieve the goal. Meanwhile, the dimension reduction comes at the price of losing a higher level of details, which can be adjusted using the explained variance ratio.

In theory, we can calculate Eigen-Salience maps at every structural level in the CNN model (different layers). At higher levels, the per-feature empirical mean produces a coarse level visualization of all essential features. In contrast, at a closer distance from the input image (lower-level convolutional layers), the per-feature empirical mean produces a higher resolution visualization, as shown in Fig. 2b, f.

C. Combined Eigen visualization

Class activation maps produce a coarse visualization with a single advantage represented by locating class-discriminative features. Meanwhile, Salience maps are capable of producing high-resolution visualization [26, 27]. By combining the two methods, we can obtain better visualization.

Fusing Eigen-CAM with Eigen-Salience maps can produce a higher resolution class-discriminative visualization. To fuse Eigen-CAM with Eigen-Salience maps, we used a simple pointwise multiplication, Fig. 2d, h.

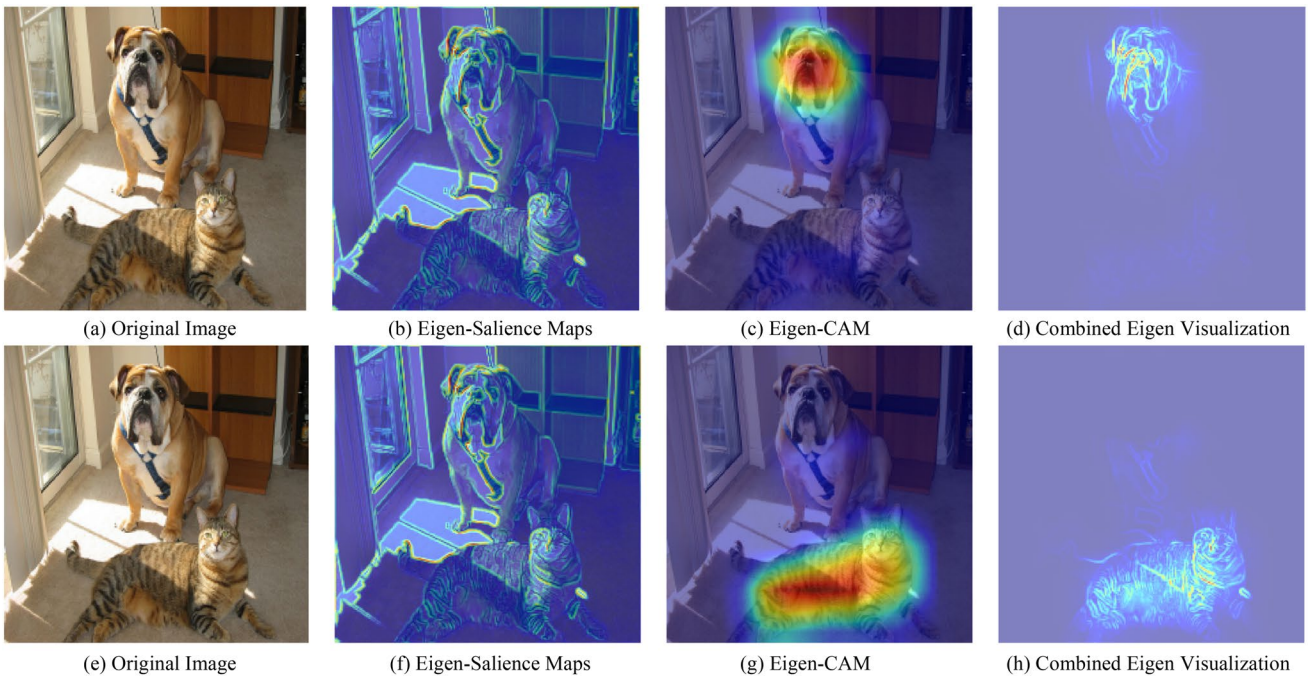

Fig. 2 CNN visualizations computed for misclassified sample images from ILSVRC validation set **a**, **e**, **i** original image. Second column form the left images **b**, **f**, **j** shows class activation maps computed using Grad-CAM. Second column from the right images **c**, **g**, **k** represent class activation maps produced using CNN-Fixations. First column from the right images **d**, **h**, **l** shows class activation maps computed using Eigen-CAM, green box represents ground truth and orange box represents top-1 classification results using VGG-16

CNN Prediction Explanations

We performed empirical analyses using ILSVRC 2014 benchmark dataset (ImageNet) [32] to demonstrate the efficacy of Eigen-CAM in generating visual explanations for correctly classified objects. We show five different examples in Fig. 3, illustrating the precision of discriminative regions in different scenarios and compare results with CNN-fixation and Grad-CAM. In particular, we consider scenarios like single and multiple object detection, detecting objects in the foreground or the background, and detecting objects in images with a crowded or plain background.

It is easy to note from Fig. 3 that Eigen-CAM shows a near-perfect match with the ground-truth shape for "Unicycle" and "Hay" examples compared to the other methods. Similarly, the "Strawberry" example shows the better localization by Eigen-CAM in the presence of background. Similar observations hold for localizing multiple objects within a single image (Power Drill).

The Bighorn example shows three regions detected by Eigen-CAM that are consistent with ground-truth compared to the four regions detected by CNN-Fixations and none detected by Grad-CAM.

Localization Applications

In this section, we evaluate Eigen-CAM against state-of-the-art methods in the task of weakly-supervised object localization, weakly-supervised segmentation, and generic object proposal.

A. Weakly-supervised localization

In weakly-supervised localization, different reported literature uses different techniques for object localization without training on bounding boxes. Instead, they use CNN models trained for classification tasks only to localize objects.

Figures 1 and 2 show explanations for CNN predictions and evidence on accurate localization. In this subsection, we evaluated Eigen-CAM localization capability on the ILSVRC 2014 validation dataset in the context of "image classification".

To localize different objects using Eigen-CAM, we utilize the forward pass of a single image at a time to obtain an explanation for the CNN prediction. Eigen-CAM does not require more than the forward pass. Unlike all other methods that utilize backpropagation starting from the class label.

In Eigen-CAM, to localize an object in an image using a particular CNN model, we feed the image to the model. In a forward pass, we use the last convolutional layer's output in the model and using the procedure described in the proposed approach. We generate an explanation in the form of a heat map. We scale the heat map values to (0–255) range, reshape to the original image size, and binarized based on different thresholds of (5–15%) of the maximum value of the heat map. Adaptive thresholds are used to account for variances in feature size produced by different models. Binarizing facilitates producing connected segments. To generate a bounding box, we use the largest segment of the arbitrary shapes.

To evaluate the weakly-supervised object localization using Eigen-CAM against state-of-the-art methods, we have implemented the experiment in [26, 30]. We used five of the popular models, namely VGG-16 [20], AlexNet [2], ResNet-101 [1], Inception-V1 aka GoogLeNet [33] and DenseNet-121 [34], all models are pretrained on the ILSVRC dataset. We used Eigen-CAM to localize the Top-1 object for each image in the validation set of the ILSVRC dataset, a total of 50,000 objects for the validation dataset.

We reported results in Table1 in the form of the error rate of the Intersection over Union (IOU) metric (100—accuracy) for the top-1 recognition prediction. The metric requires a minimum of 0.5 IOU between the ground-truth bounding box and the predicted bounding box. Eigen-CAM does not require a correct prediction of the CNN model as the case of other methods due to being class independent.

The results presented in Table 1 show clearly that Eigen-CAM outperforms all other methods in the task of weekly supervised localization except with the ResNet-101 model. One possible explanation is the input preprocessing used to train ResNet-101 on the ILSVRC dataset. The ResNet-101 model requires all images to be center cropped as a resizing step before feeding images to the model for prediction. Meanwhile, annotations for objects (bounding boxes) are created based on full-size images. Eigen-CAM works fine whenever the object sits in the center of the image. When the object sits towards the image's boundary, the principal components of extracted features are affected and hence affect Eigen-CAM localization capability.

B. Weakly-supervised segmentation

Besides generating visual explanations for CNN prediction and localizing objects, we can use Eigen-CAM explanations for salient object segmentation. In this subsection, we demonstrate the effectiveness of Eigen-CAM in performing weakly-supervised segmentation. In particular, we used the VGG-16 model to test our ideas and compare them with other methods.

Following [35] and [30], we have fine-tuned the VGG-16 model to predict the three classes presented in the Graz-2 dataset [36], namely, bike, person, and car. Each

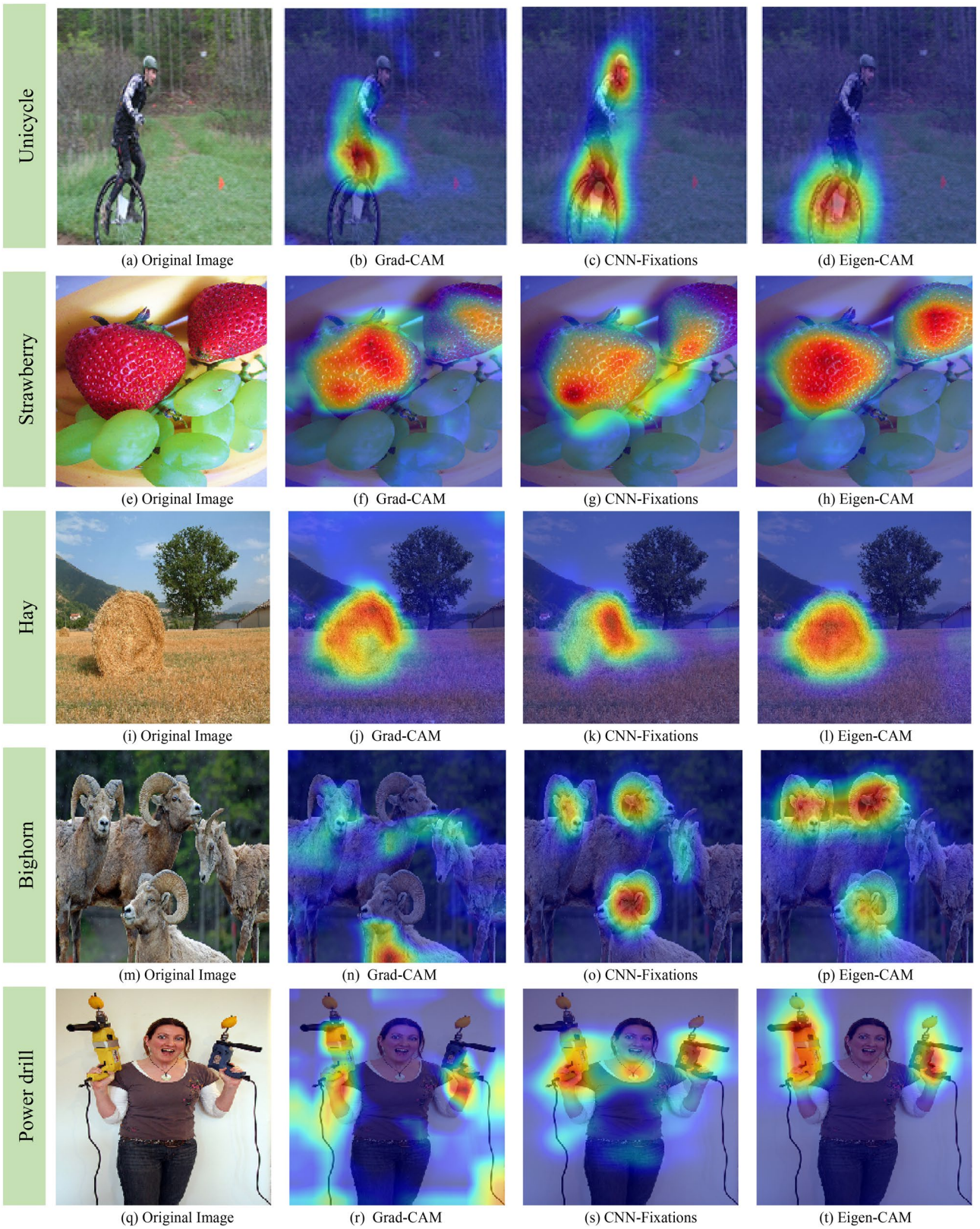

Fig. 3 CNN visualizations computed for a three misclassified example images from ILSVRC validation set **a**, **e**, **i** original image. Second column form the left images **b**, **f**, **j** shows class activation maps computed using Grad-CAM. Second column from the right images **c**, **g**, **k** represent class activation maps produced using CNN-Fixations. First column from the right images **d**, **h**, **l** shows class activation maps computed using Eigen-CAM, green box represents ground truth and orang box represent top-1 classification results using VGG-16

Table 1 Top-1 recognition prediction error rates based on 0.5 IOU for the weakly-supervised localization task of different visualization methods on the ILSVRC validation set

Method	AlexNet	VGG-16	GoogLeNet	ResNet-101	DenseNet-121
cMWP [39]	72.31	64.18	69.25	65.94	64.97
Backprop [22]	65.17	61.12	61.31	57.97	67.49
CAM [25]	67.17	57.2	60.09	**48.34**	55.37
Grad-CAM [26]	71.16	56.51	74.26	64.84	75.29
CNN-Fixations [28]	65.7	55.22	57.53	54.31	56.72
Eigen-CAM	**53.02**	**47.67**	**46.28**	56.31	**55.07**

Boldface numbers represent best results among different models

Table 2 Performance of salience object segmentation for different visualization methods

Method	Bike	Car	Person	Mean
Backprop [22]	39.51	28.50	42.64	36.88
cMWP [39]	61.84	46.82	44.02	50.89
Grad-CAM [26]	65.70	56.58	57.98	60.09
WS-SC [34]	67.50	56.48	57.56	60.52
CNN-Fixations [30]	71.21	62.15	61.27	64.88
Eigen-CAM (without training)	80.97	77.39	55.75	71.37
Eigen-CAM (with training)	**82.40**	**83.54**	**69.48**	**78.47**

Results represent pixel-level precision rates at EER, boldface numbers represent best results among different method

class has 150 example images for training and the same number for testing with a total of 900 images. By fine-tuning the VGG-16 model (trained over the ILSVRC dataset), we freeze the learning process at the convolutional layers and allow learning in the dense layers only and modify the SoftMax layer output to three nodes to match the number of classes in the Graz-2 dataset. We used this experiment to compare the results presented in [30].

For a fair comparison with CAM and Grad-CAM (relies on dense layers to generate explanations) and CNN-Fixations (require all layers) to segment salient objects, we experiment with two scenarios for weekly supervised segmentation. First, we test using the VGG-16 model trained on the ILSVRC dataset (no training or fine-tuning) to segment the salient object. In the second scenario, we use the hyperparameter optimized trained VGG-16 model on the Graz-2 dataset and utilize the new model to segment the salient objects.

In both scenarios, we use the Eigen-CAM to generate explanations in the form of a heat map. We binarize the heat maps based on the empirically determined threshold on the training dataset. We used the same threshold values from the training set to generate heat maps for the test dataset. We then performed pixel-wise precision at an equal error rate (EER) with the ground truth presented by the segmented objects in the Graz-2 dataset.

Table 2 summarizes the results obtained for the Weakly-supervised segmentation task. We take Rows 1–5 in Table 2 from [30]. To establish the testing protocol and verify the result presented in [30], we reproduced the Grad-CAM results. Then we used the established protocol to evaluate Eigen-CAM performance and compared the results with state-of-the-art methods.

We can easily see in Table 2 that the Eigen-CAM outperforms all previously reported methods. We observed an improvement of 7 mean pixel average precision at EER with the original VGG-16 model trained and tested on the ILSVRC dataset. We can see a much better improvement (14 mean pixel-level precision rates at EER) by training and testing the VGG-16 model from scratch on the Graz-2 dataset.

C. Generic object proposal

In weekly supervised localization, we demonstrate the utility of Eigen-CAM in generating explanations for localization and segmentation of multiple objects. Generic object proposal based deep neural networks reported in [37–39] are capable of localizing hundreds of class agnostic proposals in a single image. However, most images contain at most a few essential objects. In this subsection, we used Eigen-CAM for producing a single proposal representing the dominant object and compare the performance with the state-of-the-art.

To demonstrate the ability of CNNs to serve as a generic object proposal, we use Eigen-CAM to localize the best proposal extracted by the convolutional layers in the CNN network. We adopted the GoogLeNet model trained over the ILSVRC dataset to generate proposals from the PASCAL VOC-2007 dataset. Note that the GoogLeNet model was trained for a classification task only. The target categories in the ILSVRC and PASCAL VOC-2007 dataset [40] are disjoint sets.

We evaluated the performance of Eigen-CAM as a generic object proposal generator in terms of mean average precision and mean average recall, similar to methods

reported in [28, 34, 36]. We used each image in PASCAL VOC-2007 test dataset to compare the results. For each proposal, if the IOU between the ground truth and the proposal bounding box is above 50%, the proposal is considered as true positive, false positive if IOU is less than 50%, false-negative if there is no intersection between the generated proposal and the ground truth. Since each image has at least one object, there are no true negatives. Mean average recall and precision are computed as in the PASCAL VOC-2007 benchmark [40].

Table 3 shows the results obtained using Eigen-CAM and comparison with state-of-the-art methods to generate a single object proposal. Results show that our method outperforms all previously reported methods with a 15% improvement in mean average recall and a 43% improvement in mean average precision. The class-independent nature of Eigen-CAM accounts for substantial improvement achieved in the task of generic object proposal generation.

Analysis of CNN Prediction Errors

This section analyzes different scenarios that lead to recognition failure, whether that failure is caused by a noise like a case with adversarial examples, failure caused by the CNN model, or caused by annotation error (mislabeled images).

A. Adversarial examples

In this subsection, we attempt to show how adversarial examples affect CNN classifiers. In particular, we investigate the part of a CNN model that adversarial examples affect the most. We also investigate the potential effect of adversarial examples in the case of weakly-supervised localization.

Table 3 Generic object proposal performance, models trained on ILSVRC 2015 dataset for the classification task, and evaluated on the PASCAL VOC-2007 validation set, boldface numbers represent best results among different methods, results represent mean average recall and precision based on 0.5 IOU, boldface numbers represent best results among different methods

Method	mRecall	mPrecision
Backprop [22]	0.32	0.36
CAM [25]	0.30	0.33
cMWP [39]	0.23	0.26
Grad-CAM [26]	0.18	0.21
STL-WL [37]	0.23	0.31
Deep Mask [38]	0.29	0.38
CNN-Fixations [30]	0.32	0.36
Eigen-CAM (ours)	**0.47**	**0.79**

Adversarial examples represent the main vulnerability that can endanger CNNs and affect safety–critical applications such as target autonomous vehicles.

In this study, we generate the adversarial examples using small calculated perturbation to fool the model to make a wrong classification. The subtle changes in adversarial examples are indistinguishable by human eyes. To achieve our objectives, we have perturbated two examples from the ILSVRC validation dataset using the DeepFool method [41]. The original images and their perturbated copies were classified using VGG-16. Then Eigen-CAM, Grad-CAM, and CNN-Fixations were used to generate explanations for original images and their perturbated copies, as shown in Fig. 4.

Figure 4 shows that explanations generated using Eigen-CAM are almost identical to human eyes both in the case of original and perturbated images [see Fig. (4d vs. h; Fig. 4i vs. p)]. On the contrary, the Grad-CAM (Fig. 4b vs. f; Fig. 4j vs. n) and CNN-Fixations (Fig. 4c vs. g; Fig. 4k vs. o) produce different activation maps.

We can expect nearly identical explanations produced by the Eigen-CAM because the Eigen-CAM is a robust global method against small local changes. Also, Eigen-CAM does not rely on the class level. Hence any classification error does not get propagated in generating explanations. In contrast, the dependence on CNN correct prediction (Class level) as in the case with Grad-CAM and CNN-Fixations distorts generated explanations.

These results using adversarial examples prove that adversarial noise mainly affects the classification part of CNN (dense layers) since Eigen-CAM visualizations are independent of the dense layer. The unchanged explanations produced using Eigen-CAM demonstrate its robustness against local changes induced by adversarial examples, unlike other methods.

B. Tracing CNN prediction failure

Methods such as Grad-CAM and CNN-Fixations used visual explanations of CNN predictions to trace CNN-based models' prediction failure. However, the results are inconsistent and often pick up background objects. In contrast, Eigen-CAM produces consistent explanations with the ground truth.

Eigen-CAM is class independent in generating explanations. Hence, it can trace error before the dense layers in the CNN model. This added value of traceability can help build robust CNN-based models using a two-step process consisting of design and visualization.

To gain insight into what causes prediction failures in CNN, we need to distinguish different error sources (i.e., error resulting from misclassification and miss-annotation).

We use the VGG-16 model and compare explanations generated using Grad-CAM, CNN-Fixations, and

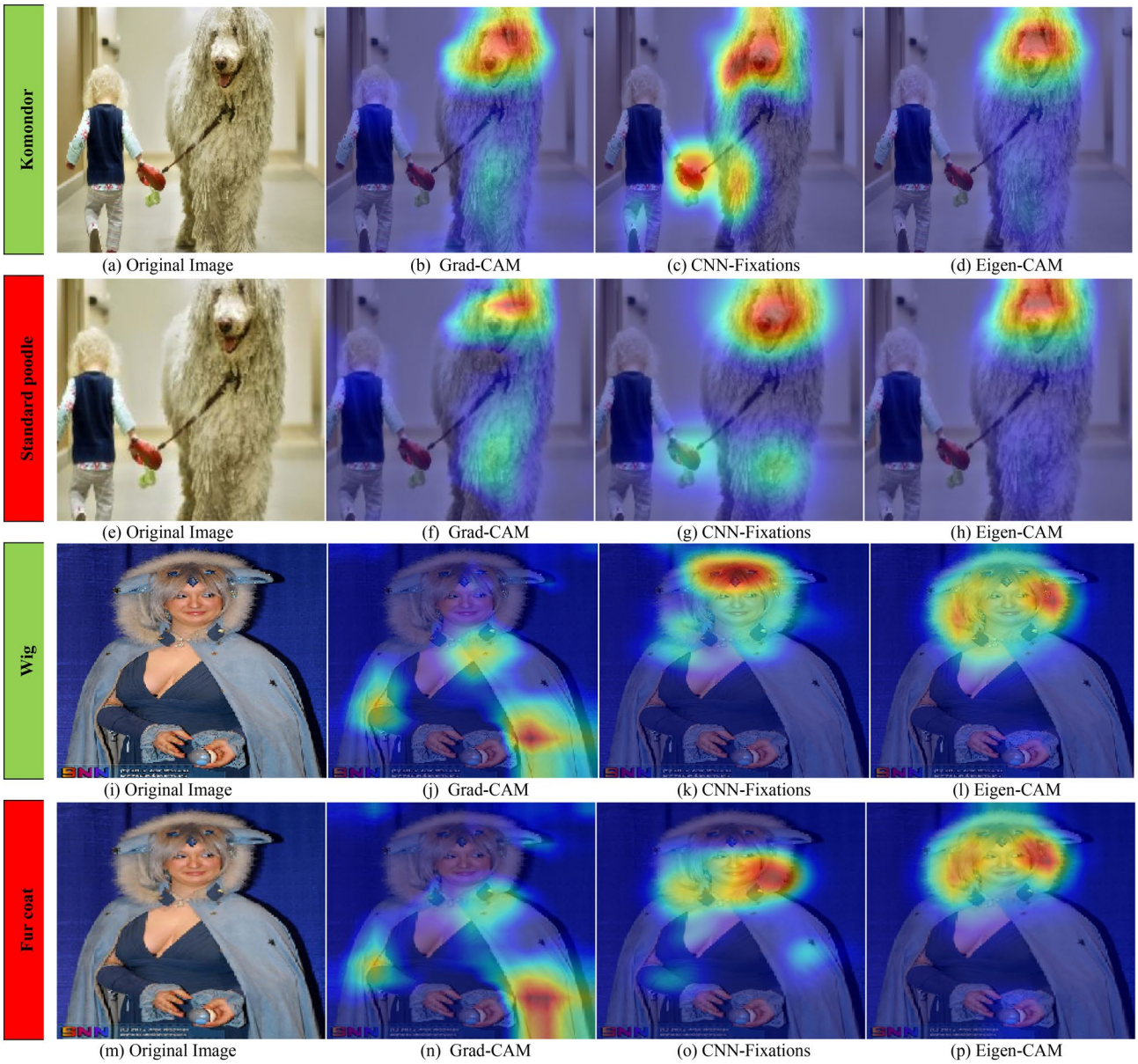

Fig. 4 CNN visualizations computed for five sample images from ILSVRC validation set **a**, **e**, **i**, **m**, **q** Original image. Second column form the left images **b**, **f**, **j**, **n**, **r** shows class activations computed using Grad-CAM. Second column from the right images **c**, **g**, **k**, **o**, **s** represent activations produced using CNN Fixations. First column from the right images **d**, **h**, **l**, **p**, **t** shows class activations computed using Eigen-CAM, green box represents ground truth and top 1 classification results using VGG-16

Eigen-CAM on the ILSVRC validation dataset to explore the traceability.

In this study, we analyze errors seen in classification with and without annotation error. We assume perfect annotation and use the convolution layers' weights as input to all visual explanation methods in studying classification errors. Eigen-CAM and other methods reported in the literature review section can help identify classification error by simple visual verification of CNN predictions and the generated explanations. Figure 5 illustrates the verification process with examples from the ILSVRC validation dataset.

Figure 5 shows that Eigen-CAM produces visual explanations matching the ground truth, and the heat map projects on the right object despite the presence of other objects.

The ground truth for the image shown in Fig. 5a is Cradle, but the VGG-16 predicted it as Grand piano. Despite the error in the classification, all methods provided reasonable visual explanation matching the ground truth. However, from Fig. 5b–d, it can be observed that the heat map produced by Eigen-CAM is right on the target, but that is not true for Grad-CAM and CNN-Fixations (Heat map on the image background).

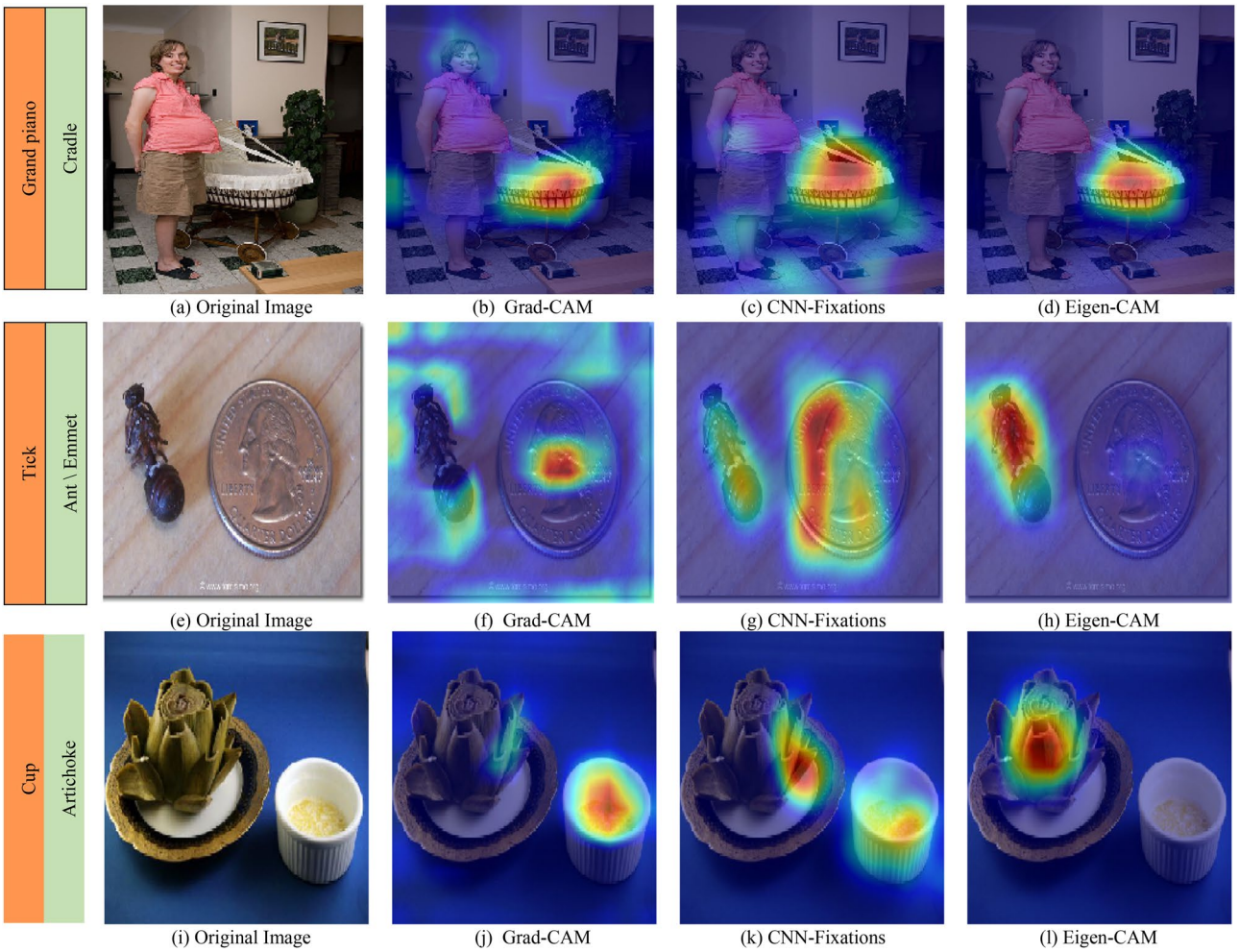

Fig. 5 Eigen visualizations computed for a dog and cat image **a**, **e** [26]. Second column from the left images **b**, **f** shows non-discriminative visualization computed using Eigen-Salience maps. Second column from the right images **c**, **g** represent CNN prediction explanations produced using Eigen-CAM. First column from the right images **d**, **h** shows class-discriminative activations computed by fusing Eigen-Salience maps and Eigen-CAM. Images in the first row corresponds to the first Eigen component and second row correspond to the second Eigen component

In the second example, the ground truth for the image shown in Fig. 5e is Ant. The VGG-16 predicted it as a Tick with a large coin in the background that is more prominent than the object. In this example, Eigen-CAM produced a perfect visual explanation (Fig. 5h) despite the classification error. While mismatch explanations produced by CNN-Fixations and Grad-CAM detect two objects (Fig. 5f and g). In the third example, the ground truth for the image shown in Fig. 5i is Artichoke. The VGG-16 predicted it as Cup. Similar results (shown in Fig. 5f–h) was observed (Fig. 5j–l).

We rely on visual verification between generated explanations, classification results, and ground truth for images with single or multiple similar or different objects in studying annotation errors. Figure 6 illustrates the verification process with examples from the ILSVRC validation dataset classified using VGG-16.

The ground truth for the image shown in Fig. 6a is “Chihuahua”, the VGG-16 predicted it as "Norwich terrier". Visual inspection shows that the classification result (“Norwich terrier”) is correct, and the annotation is wrong. Since the classification result is correct, all methods provided reasonable visual explanation matching the classification result. However, from Fig. 6b–d, it can be observed that the heat map produced by Eigen-CAM detected the presence of three “Norwich terriers”, and the heatmap is right on the targets, unlike Grad-CAM and CNN-fixations.

In the second example, the ground truth for the image shown in Fig. 6e is the “Toy terrier”. The VGG-16 top-1 prediction is an “Ibizan hound”. The image contains two objects (“Ibizan hound” and a stuffed “toy without

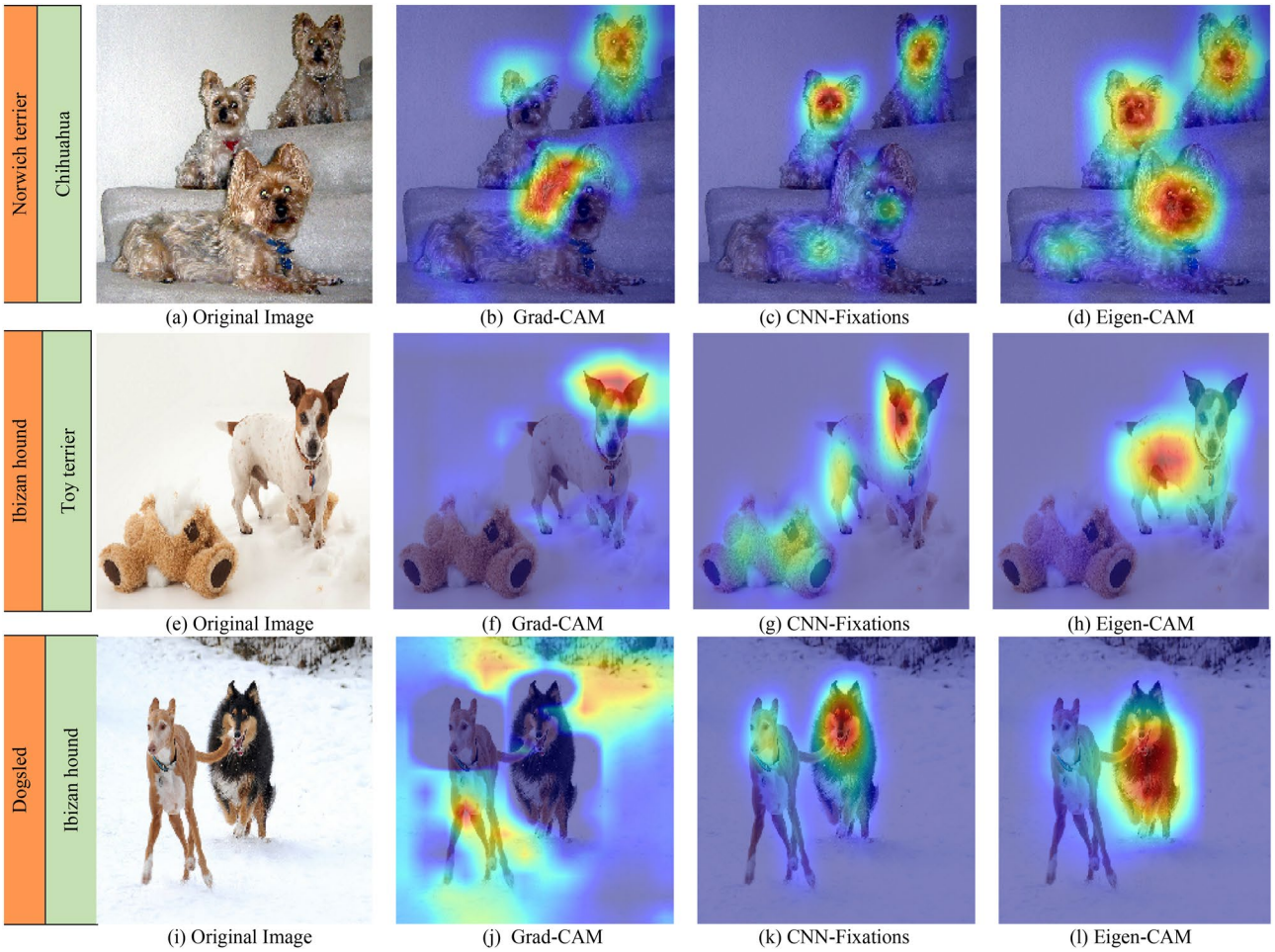

Fig. 6 CNN visualizations computed for two sample images from ILSVRC validation set with adversarial noise, first column from the left represent original images and their perpetuated copies. Original images are (**a**, **i**), perpetuated copies are (**e**, **m**). Second column form the left images **b**, **f**, **j**, **n** shows class activation maps computed using Grad-CAM. Second column from the right images **c**, **g**, **k**, **o** represent activation maps produced using CNN-Fixations. First column from the right images **d**, **h**, **l**, **p** shows class activation maps produced using Eigen-CAM. Green box represents original example top-1 classification results using VGG-16 and red box represents top-1 classification results using VGG-16 for perpetuated example

a head"). Human visual inspection shows that the classification result (Ibizan hound) is correct. Similar results (shown in Fig. 6b–d) was observed (Fig. 6f–h).

In the third example, the ground truth for the image shown in Fig. 6i is an "Ibizan hound". The VGG-16 top-1 prediction is "Dogsled". The image in Fig. 6i shows neither an "Ibizan hound" nor a "Dogsled", so both the classification result and the ground truth are wrong. Visual explanations generated using Eigen-CAM highlight the collie category, CNN-Fixations highlight both dogs, and Grad-CAM highlight the background.

Conclusions

This paper presents an intuitive and user-friendly tool capable of providing insight into the CNN inner workings through visual explanations of the learned representations and accurate visual explanation of model predictions. In addition, the Eigen-CAM is effective in weakly-supervised localization, weakly-supervised segmentation, generic object proposal, and analysis of CNN prediction errors.

Empirical analyses on different tasks show that the Eigen-CAM provides an enhanced and accurate visual explanation for CNN predictions and is also robust against classification errors made by the CNN models, annotation errors, and the presence of adversarial noise. It can generate prediction explanations from any CNN-based models without the need for model modification.

Comparison with the state-of-the-art methods shows significant improvements up to 12% improvement in weakly-supervised object localization over several models trained for image classification over ILSVRC dataset, an average of 13% improvement on weakly-supervised segmentation for VGG-16 model trained to segment three objects in the Graz-2 dataset, and at least 15% improvement for the task of generic object proposal using GoogLeNet model trained over ILSVRC dataset to localize objects from PASCAL VOC-2007.

Finally, we believe that Eigen-Cam and similar methods could be used to identify models that overfit (memorize the data) and also models that learn patterns when used on different models trained on the same data by simple visualization of model predictions. This capability is thought-provoking that merits further investigation and implementation.

Acknowledgements The authors thank Felix Havugimana for having helpful discussions while conducting the performance evaluation for this research.

Funding The authors acknowledge the funding and research support provided by the Dept. of EECE at the Herff College of Engineering, University of Memphis.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition. 2016. p. 770–8.
2. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM. 2017. https://doi.org/10.1145/3065386.
3. Wang Q, Li Q, Li X. Hyperspectral image super-resolution using spectrum and feature context. IEEE Trans Industr Electron. 2020. https://doi.org/10.1109/TIE.2020.3038096.
4. Girshick R. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. 2015. p. 1440–8.
5. Ren S, He K, Girshick R, Sun J. Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. 2015. p. 91–9.
6. Liu W, Anguelov D, Erhan D, et al. SSD: single shot multibox detector. arXiv. 2016. https://doi.org/10.1007/978-3-319-46448-0_2.
7. Wang Q, Gao J, Lin W, Li X. NWPU-crowd: a large-scale benchmark for crowd counting and localization. IEEE Trans Pattern Anal Mach Intell. 2020. https://doi.org/10.1109/TPAMI.2020.3013269.
8. Aneja J, Deshpande A, Schwing AG. Convolutional image captioning. In: IEEE/CVF conference on computer vision and pattern recognition. 2018, pp 5561–5570
9. Fang H, Gupta S, Iandola F, et al. From captions to visual concepts and back. IEEE conference on computer vision and pattern recognition (CVPR). 2015, pp 1473–1482
10. Johnson J, Karpathy A, Fei-Fei L. DenseCap: fully convolutional localization networks for dense captioning. In: IEEE conference on computer vision and pattern recognition. 2016, pp 4565–4574
11. Vinyals O, Toshev A, Bengio S, Erhan D. Show and tell: lessons learned from the 2015 MSCOCO image captioning challenge. IEEE Trans Pattern Anal Mach Intell. 2017. https://doi.org/10.1109/TPAMI.2016.2587640.
12. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, editors. Medical image computing and computer-assisted intervention – MICCAI 2015. Springer: Cham; 2015. p. 234–41.
13. Han C, Duan Y, Tao X, Lu J. Dense convolutional networks for semantic segmentation. IEEE Access. 2019. https://doi.org/10.1109/ACCESS.2019.2908685.
14. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, pp 3431–3440
15. Abdulla W. Title of subordinate document. In: Mask_RCNN: mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. Matterport. 2017. https://github.com/matterport/Mask_RCNN. Accessed 18 Dec 2020
16. Wang J, Liu Z, Chorowski J, et al. Robust 3D action recognition with random occupancy patterns. In: Fitzgibbon A, Lazebnik S, Perona P, et al., editors. Computer vision—ECCV 2012. Berlin: Springer; 2012. p. 872–85.
17. Xia L, Chen C-C, Aggarwal JK. View invariant human action recognition using histograms of 3D joints. In: IEEE computer society conference on computer vision and pattern recognition workshops. 2012, pp 20–27
18. Antol S, Agrawal A, Lu J, et al. VQA: visual question answering. In: IEEE international conference on computer vision (ICCV). 2015, pp 2425–2433
19. Anderson P, He X, Buehler C, et al. Bottom-up and top-down attention for image captioning and visual question answering. In: IEEE/CVF conference on computer vision and pattern recognition. 2018, pp 6077–6086
20. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv:2015.14091556 [cs].
21. Muhammad MB, Yeasin M. Eigen-CAM: class activation map using principal components. In: International joint conference on neural networks (IJCNN). 2020, pp 1–7
22. Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: visualising image classification models and saliency maps. In: International conference on learning representations. 2014. p. 1–8.
23. Zeiler MD, Taylor GW, Fergus R. Adaptive deconvolutional networks for mid and high level feature learning. In: IEEE international conference on computer vision. 2011, pp 2018–2025
24. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M. Striving for simplicity: the all convolutional net. 2014. arXiv preprint arXiv:1412.6806.
25. Zhou B, Khosla A, Lapedriza A, et al. Learning deep features for discriminative localization. In: IEEE conference on computer vision and pattern recognition (CVPR). 2016, pp 2921–2929
26. Selvaraju RR, Cogswell M, Das A, et al. Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. 2017. p. 618–626.
27. Chattopadhyay A, Sarkar A, Howlader P, Balasubramanian VN. Grad-CAM++: improved visual explanations for deep convolutional networks. In: IEEE winter conference on applications of computer vision (WACV). 2018. https://doi.org/10.1109/WACV.2018.00097
28. Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences. In: Proceedings

of the 34th international conference on machine learning. 2017. p. 3145–53.
29. Bach S, Binder A, Montavon G, et al. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE. 2015;10:e0130140. https://doi.org/10.1371/journal.pone.0130140.
30. Mopuri KR, Garg U, Venkatesh BR. CNN fixations: an unraveling approach to visualize the discriminative image regions. IEEE Trans Image Process. 2019. https://doi.org/10.1109/TIP.2018.2881920.
31. Voulodimos A, Doulamis N, Doulamis A, Protopapadakis E. Deep learning for computer vision: a brief review. Comput Intell Neurosci. 2018. https://doi.org/10.1155/2018/7068349.
32. Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. Int J Comput Vis. 2015. https://doi.org/10.1007/s11263-015-0816-y.
33. Szegedy C, Wei Liu, Yangqing Jia, et al. Going deeper with convolutions. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR). 2015. p. 1–9.
34. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). 2017. p. 2261–9.
35. Cholakkal H, Johnson J, Rajan D. Backtracking ScSPM image classifier for weakly supervised top-down saliency. In: IEEE conference on computer vision and pattern recognition (CVPR). 2016, pp 5278–5287
36. Marszałek M, Schmid C. Accurate object recognition with shape masks. Int J Comput Vis. 2012. https://doi.org/10.1007/s11263-011-0479-2.
37. Bazzani L, Bergamo A, Anguelov D, Torresani L. Self-taught object localization with deep networks. In: 2016 IEEE winter conference on applications of computer vision (WACV). 2016. p. 1–9.
38. Pinheiro PO, Collobert R, Dollár P. Learning to segment object candidates. In: Advances in neural information processing systems. 2015. p. 1990–8.
39. Zhang J, Bargal SA, Lin Z, et al. Top-down neural attention by excitation backprop. Int J Comput Vis. 2018;126:1084–102.
40. Everingham M, Gool L, Williams CK, et al. The Pascal visual object classes (VOC) challenge. Int J Comput Vis. 2010. https://doi.org/10.1007/s11263-009-0275-4.
41. Moosavi-Dezfooli S-M, Fawzi A, Frossard P. Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 2574–82.