PLOS ONE

RESEARCH ARTICLE

CAManim: Animating end-to-end network activation maps

Emily Kaczmarek[1]*, Olivier X. Miguel[2], Alexa C. Bowie[2], Robin Ducharme[2], Alysha L. J. Dingwall-Harvey[1,2], Steven Hawken[1,2,3,4], Christine M. Armour[1,5,6], Mark C. Walker[1,2,3,4,7,8,9,10], Kevin Dick[1,6,9]*

1 Children's Hospital of Eastern Ontario Research Institute, Ottawa, Canada, **2** Clinical Epidemiology Program, Ottawa Hospital Research Institute, Ottawa, Canada, **3** School of Epidemiology and Public Health, University of Ottawa, Ottawa, Canada, **4** ICES, Toronto, Canada, **5** Department of Pediatrics, University of Ottawa, Ottawa, Canada, **6** Prenatal Screening Ontario, Better Outcomes Registry & Network, Ottawa, Canada, **7** Department of Obstetrics and Gynecology, University of Ottawa, Ottawa, Canada, **8** International and Global Health Office, University of Ottawa, Ottawa, Canada, **9** BORN Ontario, Children's Hospital of Eastern Ontario, Ottawa, Canada, **10** Department of Obstetrics, Gynecology & Newborn Care, The Ottawa Hospital, Ottawa, Canada

* ekaczmarek@cheo.on.ca (EK); kdick@cheo.on.ca (KD)

Citation: Kaczmarek E, Miguel OX, Bowie AC, Ducharme R, Dingwall-Harvey ALJ, Hawken S, et al. (2024) CAManim: Animating end-to-end network activation maps. PLoS ONE 19(6): e0296985. https://doi.org/10.1371/journal.pone.0296985

Editor: Arka Bhowmik, Memorial Sloan Kettering Cancer Center, UNITED STATES

Received: May 9, 2023

Accepted: December 22, 2023

Published: June 18, 2024

Data Availability Statement: All relevant data are within the manuscript and its Supporting information files.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Deep neural networks have been widely adopted in numerous domains due to their high performance and accessibility to developers and application-specific end-users. Fundamental to image-based applications is the development of Convolutional Neural Networks (CNNs), which possess the ability to automatically extract features from data. However, comprehending these complex models and their learned representations, which typically comprise millions of parameters and numerous layers, remains a challenge for both developers and end-users. This challenge arises due to the absence of interpretable and transparent tools to make sense of black-box models. There exists a growing body of Explainable Artificial Intelligence (XAI) literature, including a collection of methods denoted Class Activation Maps (CAMs), that seek to demystify what representations the model learns from the data, how it informs a given prediction, and why it, at times, performs poorly in certain tasks. We propose a novel XAI visualization method denoted CAManim that seeks to simultaneously broaden and focus end-user understanding of CNN predictions by animating the CAM-based network activation maps through all layers, effectively depicting from end-to-end how a model progressively arrives at the final layer activation. Herein, we demonstrate that CAManim works with any CAM-based method and various CNN architectures. Beyond qualitative model assessments, we additionally propose a novel quantitative assessment that expands upon the Remove and Debias (ROAD) metric, pairing the qualitative end-to-end network visual explanations assessment with our novel quantitative "yellow brick ROAD" assessment (ybROAD). This builds upon prior research to address the increasing demand for interpretable, robust, and transparent model assessment methodology, ultimately improving an end-user's trust in a given model's predictions. Examples and source code can be found at: https://omni-ml.github.io/pytorch-grad-cam-anim/.

Introduction

The popularization of deep learning in numerous domains of research has led to the rapid adoption of these methodologies in disparate fields of scientific research. Convolutional Neural Networks (CNNs) are a class of deep learning models that use convolutions to extract image features, achieving high performance in numerous computer vision applications [1]. However, due to the intrinsic network structure and the complexity of features leveraged for model predictions, CNNs are, consequently, often labeled as uninterpretable or 'black-box' models. Interpretability is crucial for applications in high-criticality fields such as medicine [2], where model decisions have the potential to cause excessive harm if incorrect. In order to be deployed, models must be trustworthy both in their class predictions and in the features used to make those predictions. Therefore, there is a definitive impetus to develop trustworthy explanations of model decisions.

Presently, there exists extensive literature on the use of state-of-the-art deep learning methodologies within healthcare systems and applications. Indeed, there exist entire subfields of computer science and biomedical engineering on computational medicine and medical image analysis. Notable examples from the literature include online medical pre-diagnosis systems [3], 3D deep learning on medical images [4], the development of medical transformers for chest x-ray diagnosis [5], and an emergent trend to adopt generative methods in these high-criticality fields (e.g. GPT-3 as a data generator for medical dialogue summarization [6]). With the emergence of large language models (LLMs) such as the GPT-3 and GPT-4 models developed by OpenAI and made broadly available through the ChatGPT platform, early adopters are actively promoting the transformative opportunities of these AI systems within the healthcare space [7] while others issue active calls for caution in their use [8]. Fundamentally, it is paramount to develop increasingly transparent methods to assist medical practitioners in their use of, critical oversight of, and reliance upon deep learning models.

There have been numerous methods proposed to improve the interpretability of CNNs. Zeiler and Fergus initially investigated network interpretability by using a deconvolutional network to identify pixels activated in CNN feature maps [9]. Thereafter, gradient-based methods were used to develop saliency maps indicating important image regions based on desired output class [10–12]. Class Activation Maps (CAMs) are a group of methods that linearly combine weighted feature activation maps from a given CNN layer [13–22]. Typically, only the final layer(s) are visualized to confer trustworthiness and describe what image features are used for model predictions. However, this provides little detail on the learning process of the model. In addition, selecting the correct final layer to visualize from each CNN model is not straightforward and is often done arbitrarily.

To better interpret how a given model evaluates a given image through each of its layers, we propose expanding these existant Explainable Artificial Intelligence (XAI) methodologies by individually visualizing and analyzing the model's layer-wise activation maps. In a natural extension of this idea, these layer-wise activation maps can be combined as individual frames of a video animating the end-to-end network activation maps; a method we propose in this article and denote CAManim. We develop local and global normalization to understand learned network features on a layer-wise (local perspective) and network-wise scale (global perspective). We experiment and quantify layer-wise performance of CAManim with numerous CNN models and CAM variations to show performance in a variety of experimental conditions, including medical applications wherein model understanding and trustworthiness is critical.

Our contributions are as follows:

- We propose CAManim, a novel visualization method that creates activation maps for each layer in a given CNN. CAManim can be applied to any existing CNN and CAM for any classification task.
- We introduce local and global normalization to understand important learned features at both a layer-wise and network-wise level.
- We perform extensive experimentation to determine the expected time and space requirements to run CAManim.
- We demonstrate the usefulness of CAManim across multiple CAM variations and CNN models, and in high-criticality fields.
- We quantitatively evaluate the performance of each CAM visualization generated per model layer with an analytical process denoted "yellow brick ROAD" (ybROAD) that seeks to improve the understanding of how CNNs learn. This is further extended to selecting the most accurate feature map representation from all possible layers of a CNN.

Related work

The topic of explainable and trustworthy AI has been researched extensively. Lipton *et al.* [23] described the importance for trustworthy and interpretable models, while Ribeiro *et al.* [24] conducted human-based trials to quantify their degree of trust in classifer predictions. Computationally, numerous methods have investigated the improvement of CNN interpretation. In this section, we provide an overview of proposed methods and how CAManim addresses a gap in the current literature.

Earliest explainable AI studies. One of the earliest efforts to interpret CNNs was made by Zeiler and Fergus [9]. In this study, feature maps from convolutional layers are used as input to a deconvolutional network to identify activated pixels in the original image space. Simonyan *et al.* [10] approached network explainability in two ways. First, they proposed class models, which are images generated through gradient ascent that maximize the score for a given class [10]. Next, they produced class-specific saliency maps, calculated using the gradient of the input image with respect to a given class [10].

Guided backpropagation and gradient-based methods. Springenberg *et al.* [11] extended Simonyan's work to Guided Backpropagation, which excludes all negative gradients to produce improved saliency maps. Despite calculating gradients with respect to individual classes, Selvaraju *et al.* showed that the visualizations produced by Guided Backpropagation are not class-discriminative (*i.e.* there is little difference between images generated using different class nodes) [14]. Sundarajan *et al.* [12] proposed integrated gradients, calculated through the integral of the gradient between a given image and baseline, to satisfy axioms of sensitivity and implementation invariance. FullGrad is another gradient-based method that is non-discriminative and uses the gradients of bias layers to produce saliency maps [25].

Gradient-free methods. While gradient-based methods are quite popular in the field of explainable AI, some studies argue that these methods produce noisy visualizations due to gradient saturation [26, 27]. For this reason, gradient-free methods have been investigated by a number of studies. Zhou *et al.* [28] identified K images with the highest activation at a given neuron in a convolutional layer and occluded patches of each image to determine the object detected by the neuron. Morcos *et al.* [29] used an ablation analysis to remove individual neurons or feature maps from a CNN and quantify the effect on network performance. This study demonstrated that neurons with high class selectivity (*i.e.* highly activated for a single class)

may indicate poor network generalizability. Zhou *et al.* [30] extended this work to show that ablating neurons with high class selectivity may cause large differences in individual class performance.

Class activation maps. A popular class of CNN visualizations are Class Activation Maps (CAMs), which produce explainable visualizations through a linearly weighted sum of feature maps at a given CNN layer [13]. The original CAM was proposed for a specific CNN model, consisting of convolutional, global average pooling, and dense layers at the end of the network [13]. The dense layer weights were used to determine the weighted importance of individual feature maps [13]. However, this required a specific CNN architecture and was not applicable to numerous high-performing models. This led to the development of CNN model-agnostic CAM methods.

Gradient-based methods were the first variation of the original CAM [14–19]. These methods determine importance weights by calculating averaged or element-wise gradients of the output of a class with respect to the feature maps at the desired layer. As discussed previously, gradient methods may produce noisy visualizations due to gradient saturation [20–22, 26, 27]; as a result, perturbation CAM methods have been proposed [20, 21]. In this case, importance weights are calculated by perturbing the original input image by the feature maps and measuring the change in prediction score. In addition, non-discriminative approaches have been investigated to eliminate the reliance of class-discriminative methods upon correct class predictions. For example, EigenCAM produces its CAM visualization using the principal components of the activation maps at the desired layer [22].

While most studies have developed saliency map and/or CAM formulations for a single layer, LayerCAM demonstrated how aggregating feature maps from multiple layers can refine the final CAM visualization to include more fine-detailed information [19, 31]. Gildenblat extended this idea across existing multiple CAM and saliency map methods [17]. While conceptually similar, to the best of our knowledge, our study is the first to consider individual feature maps generated from every CNN layer and combine them into an end-to-end network explanation. Moreover, this end-to-end layer-wise analysis enables a unique view of local and global perspectives and a natural integration of both qualitative and quantitative network-wide explainability. Fig 1 provides a conceptual overview of the CAManim method proposed in this work.

Materials and methods

In this section, we first recall the general formulation for Class Activation Maps and outline notation preliminaries. Next, we explain the generation of CAManim using CAM visualizations from each layer of a CNN, depicted in Fig 1. The concepts of global and local normalization are introduced, and the computational requirements of CAManim are described from large-scale experiments. Lastly, we define the quantitative performance metric for individual CAM visualizations, and propose ybROAD for analyzing end-to-end layer-wise CAManim.

Individual CAM formulation

The general formulation for any CAM method consists of taking a linearly weighted sum of feature maps as follows:

$$L^c_{CAM(A^l)} = \sum_k (\alpha^c_k A^l_k), \textit{where } A^l = f^l(x) \tag{1}$$

For a given input image x and CNN model $f(\cdot)$, a CAM visualization L can be generated through the weighted α summation of k activation feature maps A at layer l. Class

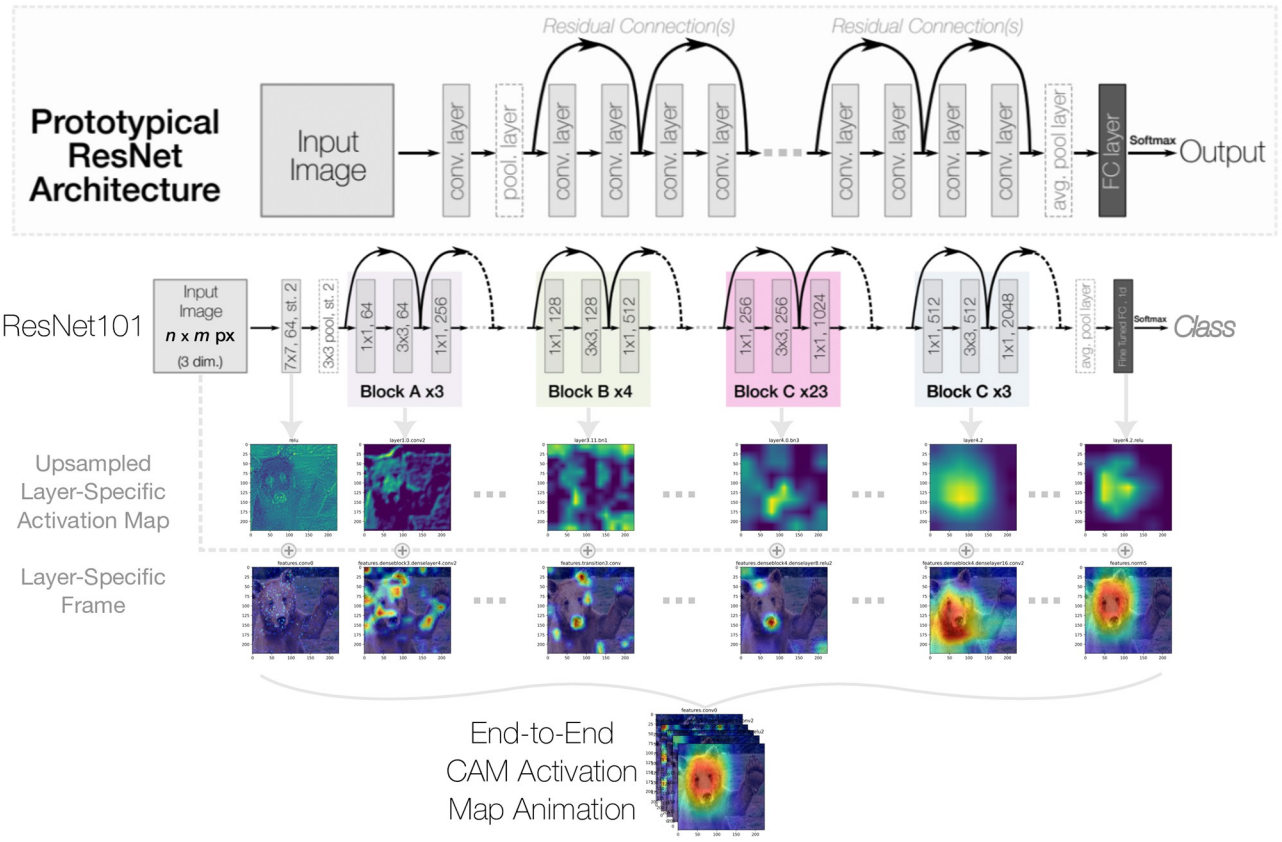

Fig 1. Conceptual overview of generating an animation of a resnet's end-to-end activation maps for a given image and target class.

https://doi.org/10.1371/journal.pone.0296985.g001

discriminative CAM methods further define L per predicted class c. To exclude negative activations, most CAM formulations are followed by a ReLU operation.

End-to-end layerwise activation maps

To formulate CAManim, CAM visualizations are first generated for every differentiable layer l within a given CNN with a total number of layers N:

$$L^c_{CAManim} = L^c_{CAM(A^{l=0})}, \ldots, L^c_{CAM(A^{l=N})} \tag{2}$$

Each CAM visualization is subsequently saved as a PNG image I and concatenated together to create the final CAManim video, as depicted below:

$$CAManim = \Big\|_{l}^{N} I_{L^c_{CAM(A^l)}} \tag{3}$$

For clarity, the concatenation operator, $\|$, is defined in this work in a way analagous to the summation operator, Σ, and product operator, Π, to concisely express the sequential organization of individual frames into the resulting animated video.

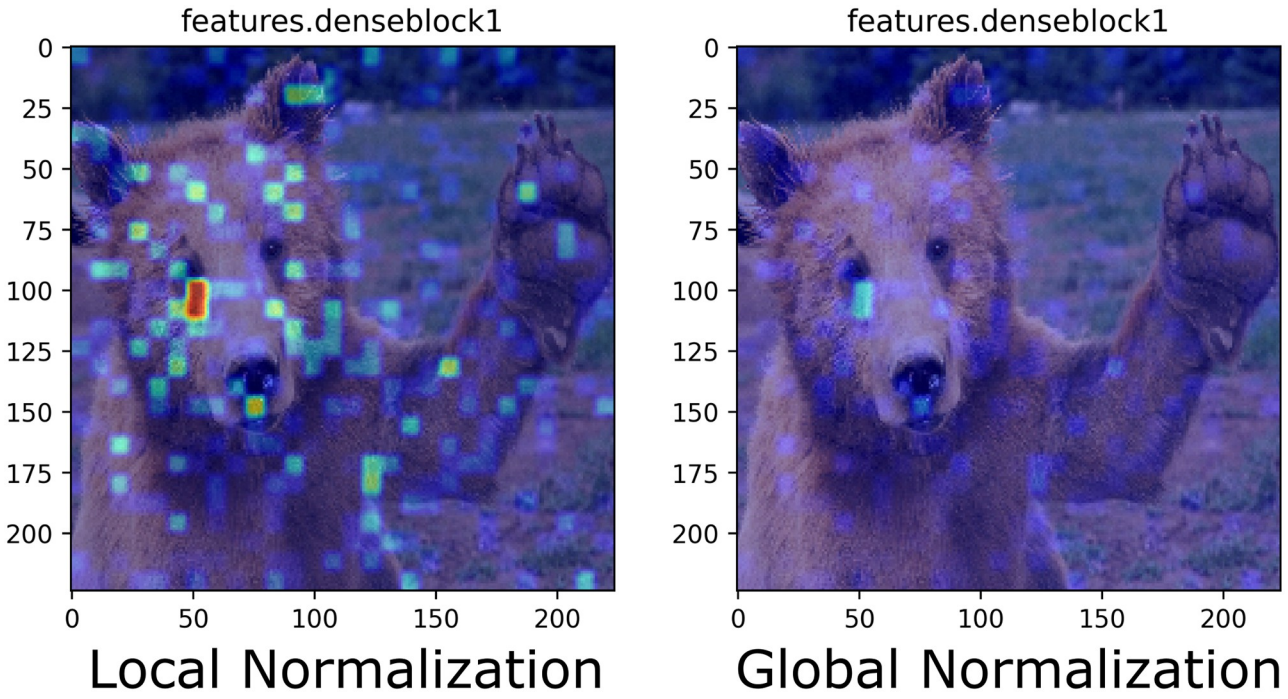

Fig 2. Difference between local and global normalization for the feature map generated from layer *features.denseblock1* in DenseNet161.

https://doi.org/10.1371/journal.pone.0296985.g002

Global- vs. Local-Level Normalization

For a model end-user to correctly interpret what importance the model attributes to particular pixels at a given layer in the network, they must be provided the appropriate context. To this end, the model interpreter may wish to know "*what importance does the model place on particular pixels at a given layer?*" or "*what importance does the model place upon particular pixels overall?*". Consequently, two normalization approaches can be leveraged, each with the intent of correctly relaying information to answer one of these two questions, and both complementary to the other. Thus CAManim visualizes the CAM activations of each layer using two different types of normalization: Local-Level Normalization and Global-Level Normalization. Global normalization is performed using the minimum and maximum activation value across all activations generated, which is practical for determining and visualizing which layer contains the strongest network-wide activation for a given class. Local normalization uses the minimum and maximum values of the activations of each specific layer. Local normalization, contrary to global-level normalization, depicts the strongest layer-level activation and therefore provides layer-wise information.

Fig 2 shows the difference between global and local normalization for the first denseblock of DenseNet161 [32]. The global normalization (right) displays an attenuated version of the local normalization (left). This example demonstrates that the layer-wise information focuses upon learning small pattern-like features, whereas the network-wise information indicates that the activations of this layer are generally attenuated with respect to all other layers within the DenseNet161 model.

Model- & CAM-specific interpretation and computational complexity

To appreciate how varying CNN architectures and CAM methods produce differing visual explanations for a given image x, target class c, and CNN model $f(\cdot)$, we ran large-scale

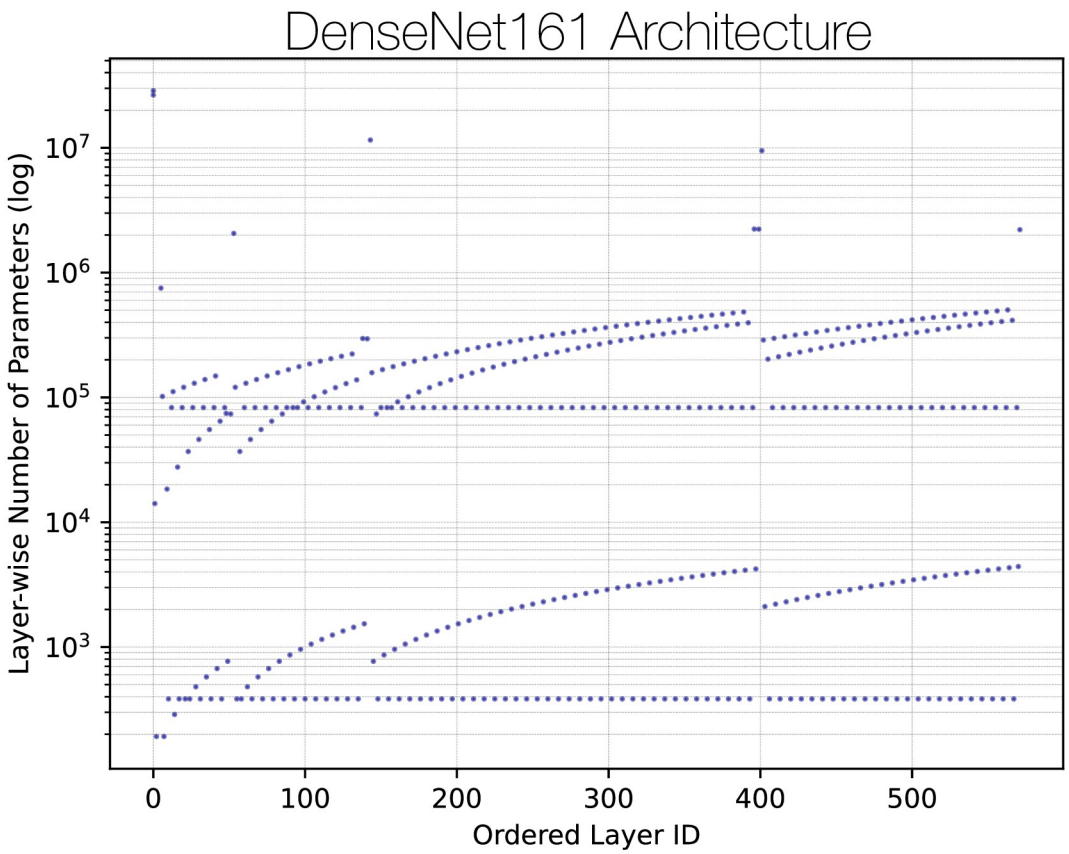

Fig 3. Layer-wise depiction of DenseNet161 parameters.

https://doi.org/10.1371/journal.pone.0296985.g003

experiments producing numerous layer-wise and globally normalized CAManim videos/image sequences. Consequently, this additionally enabled the benchmarking of key model-specific metrics such as layer-level number of parameters and CAManim run-time.

Fig 3 illustrates the layer-wise parameter number along a log-scale where we can explicitly visualize the four general `DenseBlocks` comprised of a varying number of `DenseLayers`. Since CAManim computation will vary by layer number n, layer-wise parameters p, and CAM-specific compute runtime r, we generally estimate that CAManim will have a simplified computational time complexity of $\mathcal{O}(n\bar{p}\bar{r})$. For clarity, the overbar notion expresses averages for the number of parameters and CAM-specific compute time, respectively. Image-specific dimension will also impact runtime, however, given that the majority of models reshape their input to a consistent size, this constant factor may be subsumed within term n. To provide general estimates on the overall runtime for a given CNN and CAM, we tabulate in Table 1 our experimental benchmarks using an Intel Xeon CPU 2.20 GHz, 13GB RAM, Tesla K80 GPU accelerator, and 12GB GDDR5 VRAM.

Quantitative evaluation

To quantitatively evaluate the performance of each CAM visualization and demonstrate the information gained through deeper layers in a CNN, we calculate the Remove and Debias (ROAD) score [33]. This metric has superior computational efficiency and prevents data leakage found with other CAM performance metrics [33]. ROAD perturbs images through noisy

Table 1. Total number of parameters, CAManim runtime, and average parameters and runtime across all layers included in CAManim calculated for six CNN models using HiResCAM.

Model	Num. Params.	Time (s)	Avg. Params. per Layer	Avg. Layer Time (s)
AlexNet	61,100,840	16.67	329,292.8	0.2622
ConvNeXT	88,591,464	392.78	1,140,535.1	0.2665
DenseNet161	28,681,000	514.99	180,814.6	0.0997
EfficientNet-b7	66,347,960	972.42	345,325.1	0.1375
MaxViT-t	30,919,624	596.84	292,907.1	0.1080
SqueezeNet	1,248,424	52.54	48,030.8	0.0144

https://doi.org/10.1371/journal.pone.0296985.t001

linear imputations, blurring regions of the image based on neighbouring pixel values [33]. The confidence increase or decrease C in classification score using the perturbed image with the *least relevant pixels* (LRP) or *most relevant pixels* (MRP) is then used to evaluate the accuracy of a CAM visualization. Since the percentage of pixels perturbed affects the ROAD performance, we evaluate ROAD at $p = 20\%, 40\%, 60\%$ and 80% pixel perturbation thresholds. As proposed by Gildenblat [17], we combine the LRP and MRP scores for our final metric:

$$ROAD(L^c_{CAM(A^l)}) = \sum_p \frac{(C^p_{L_{LRP}} - C^p_{L_{MRP}})}{2} \tag{4}$$

A ROAD score is calculated for each CAM generated. Therefore, for N differentiable layers in a CNN, there will be N ROAD scores calculated within CAManim. Given that this network-wide sequence of ROAD values represents a journey-like traversal of the network, we denote this series of ROAD values as the 'yellow brick ROAD', or ybROAD for brevity:

$$ybROAD = \Big\|_l^N ROAD(L^c_{CAM(A^l)}) \tag{5}$$

The ybROAD scores can be used to analyze performance of an experiment with given image x, target class c, and CNN model $f(\cdot)$ over all layers of the network. Consequently, this analysis enables the quantitative identification of the CNN layer that maximally visualizes features with the largest impact on model performance through `max`(ybROAD). The `mean` (ybROAD) score is also calculated to summarize the overall model end-to-end ROAD performance. Interestingly, variant metrics derived from ybROAD values may yield new insights into the quantification of a model's ability to predict particular classes.

Results & discussion

In this section, we first define the pre-trained models and datasets used to evaluate CAManim. Next, we demonstrate CAManim in high-criticality fields using a ResNet50 model fine-tuned to perform breast cancer classification. We then show example visualizations from CAManim for 10 different CAM variations and discuss abnormal visualizations. Lastly, we discuss the ybROAD performance of CAManim and future directions building upon this work.

Pre-trained models and datasets

To evaluate CAManim, we use models from Pytorch pre-trained on the 2012 ImageNet-1K dataset [34]. Specifically, results are shown for AlexNet [35], ConvNeXT [36], DenseNet161 [32], EfficientNet-b7 [37], MaxViT-t [38], and SqueezeNet [39]. The CAManim videos for an additional 14 models and publicly available code can be found here: https://omni-ml.github.

io/pytorch-grad-cam-anim/. All results in this study (apart from the high-criticality case study) leverage a popular brown bear-containing image typically used in the CAM research community; following an emergent standard, the image is preprocessed by resizing to 224 × 224 and normalized. Next, we demonstrate the utility of CAManim in a high-criticality field.

Case study: End-to-end BC-ResNet50 visualization for malignant tumour prediction

We leverage a ResNet50 model [40] initially trained on the 2012 ImageNet-1K dataset [34] and fine-tune the model using the Kaggle breast ultrasound data to classify malignant vs. normal images [41]. For simplicity, we refer to this network as BC-ResNet50 ("Breast Cancer-ResNet50"). This dataset comprises 133 normal images and 210 malignant images that are split into a 80–10-10% train-validation-test split. Images are preprocessed to a size of 224 × 224 and various augmentations are applied to the training set.

Pre-processing and training steps are selected based on MONAI recommendations. Following fine-tuning, CAManim is run with an example test image of the malignant class to visualize and interpret how the resultant CNN arrives at producing the correct prediction of malignant cancer. Fig 4 illustrates the layer-wise activations that BC-ResNet50 considers when determining the 'malignant' tumour.

It is important to emphasize that for high-criticality applications such as medical imagery, the initial resizing of input imagery can dramatically alter the information available to the model and impact model out and its interpretability. While this work builds upon previous XAI literature and adopts their methodological approach, we recommend that for high-criticality applications, the initial image size be kept closely aligned with original input image sizes (no/limited downsizing) so as not to alter image resolution and to provide a clinical decision support system as a visual explanation method.

Visualizing end-to-end network activation maps

We further demonstrate the performance of CAManim on 10 different CAM methods, including seven gradient-based methods (EigenGradCAM, GradCAM, GradCAMElementWise, GradCAM++, HiResCAm, LayerCAM, and XGradCAM), two perturbation methods (AblationCAM and ScoreCAM), a principal components method (EigenCAM), and RandomCAM. RandomCAM generates random feature activation maps from a uniform distribution between [−1, 1].

As expected, Figs 5 & 6 depicts early model layers as activating general patterns and edges while middle and final layers progressively focus the activation maps to regions highly characteristic of the brown bear contained within. Such a layer-wise approach enables the pair-wise or multi-wise comparison of visual-explanation methods and how these individual activation maps compare globally across all activation maps.

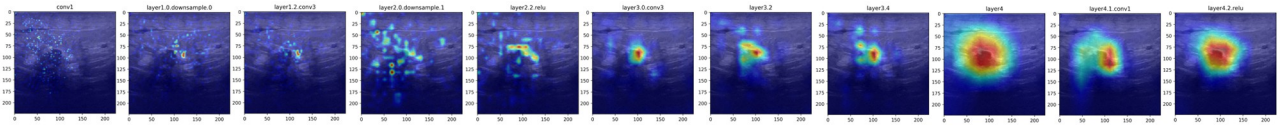

Fig 4. Visualization of the activation maps from BC-ResNet50 to visually depict how the model predicts the 'malignant' tumour class. Only the 10th percentile layers are illustrated for concision.

https://doi.org/10.1371/journal.pone.0296985.g004

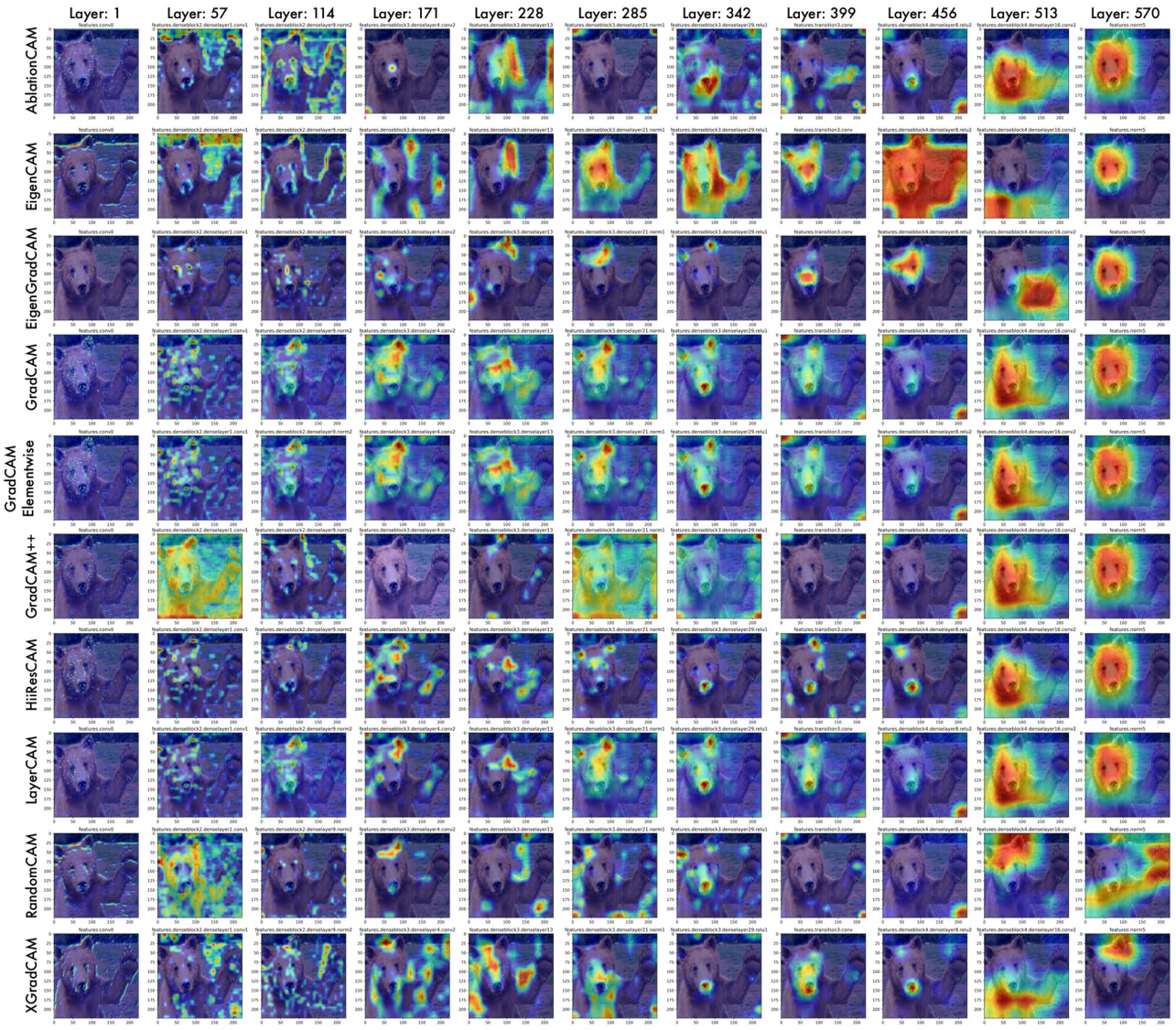

Fig 5. End-to-end activation map visualization for 10 CAMs using DenseNet161. Every 10th percentile map is depicted, from left to right.

https://doi.org/10.1371/journal.pone.0296985.g005

Layer-type visualization issues

Certain differentiable layers may produce unanticipated CAM visualizations, as depicted in Fig 7. In these layers, images are compressed to 1-dimensional (1D) representations; consequently, 2D feature visualization of a non-convolutional layer is effectively meaningless. Instead, individual neurons that are highly activated show up as vertical or horizontal lines across the image. While these images are uninformative, they simply depict visualizations of 1D vectors and should be filtered out.

ybROAD quantitative evaluation

Fig 8 displays the ybROAD for 11 trials of generating CAManim for the bear image using ResNet152 (mean ybROAD: 0.204; max ybROAD: 0.473 at layer 402). Initially, the layer-wise ROAD performance is very high ($\sim$0.4). At this point, the CNN layer is activating many small regions throughout the image; when each of these areas is perturbed, it is difficult to correctly

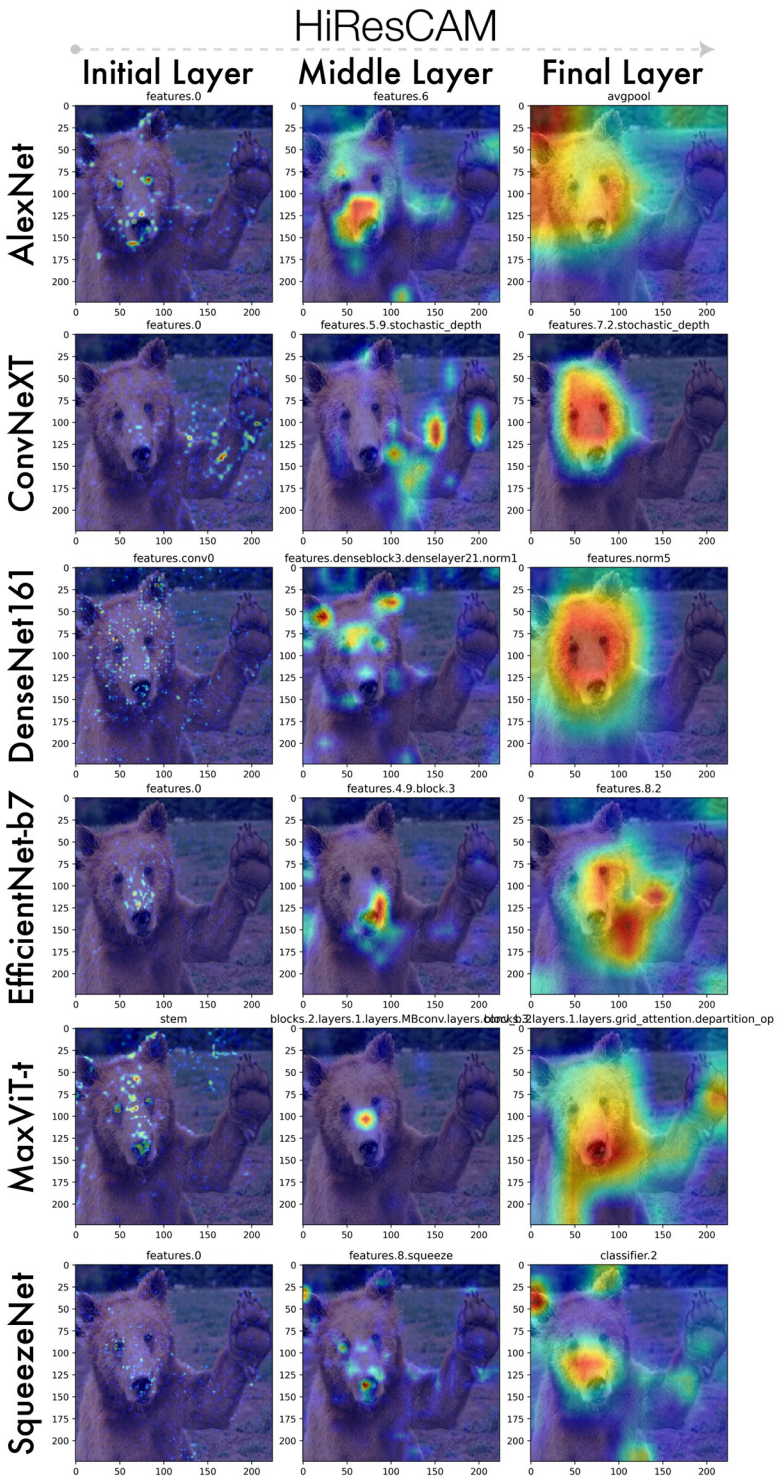

Fig 6. Initial, middle, and final activation maps applying a single CAM, HiResCAM, to various model architectures.

https://doi.org/10.1371/journal.pone.0296985.g006

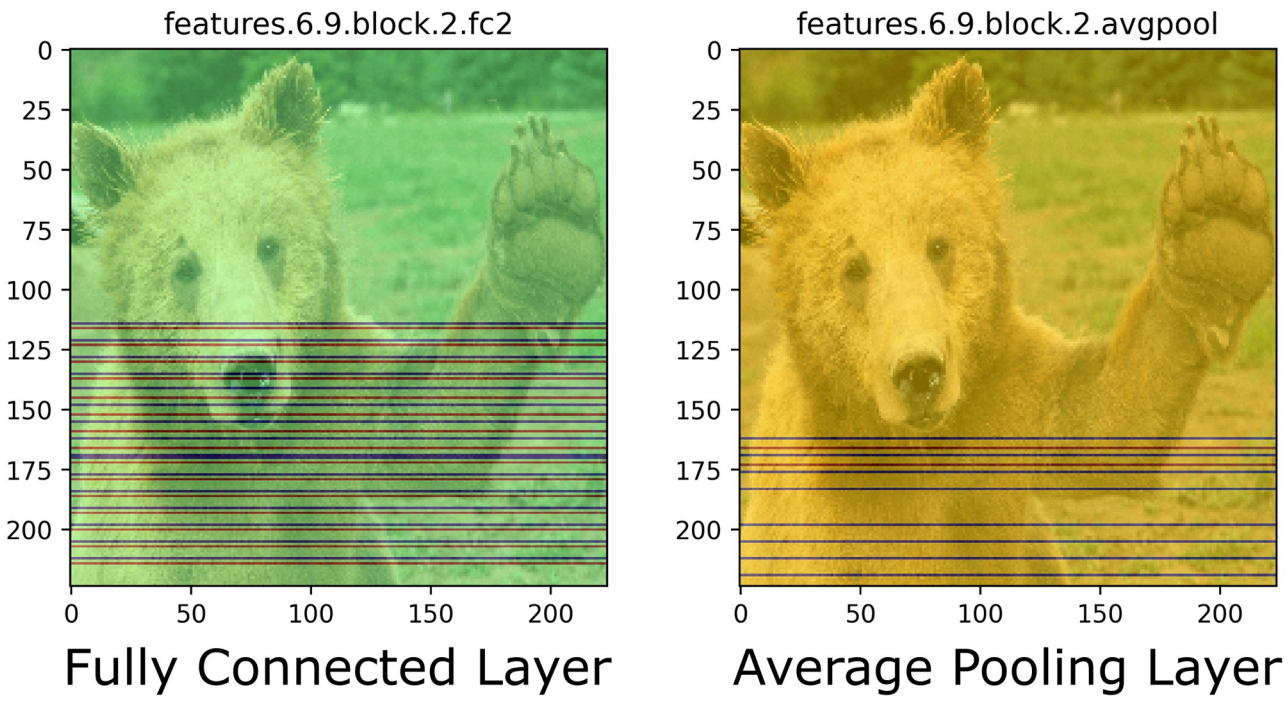

Fig 7. Visualization of CAManim for fully connected and average pooling layers.

https://doi.org/10.1371/journal.pone.0296985.g007

classify the image, and the ROAD score increases. As the network starts learning larger features, less of the bear image is perturbed, and the ROAD score decreases. Towards the end of the network, the ROAD score increases again and reaches its maximum value as the small activations are combined together to encapsulate the entire bear. This demonstrates how the ybROAD score can provide more information on how the network progressively learns.

Interestingly, the layer-wise depiction of ROAD values may be used to investigate how various model layers contribute to the overall discrimination of a given target class within an image for a pre-trained model of a given architecture and selected CAM. To quantify the improvement of our ybROAD method against standard practise (*i.e.*, considering the activation map of the final layer of a model), we sumarize 12 diverse experimental conditions in Table 2. The difference in the ybROAD vs. final layer-ROAD values is indicative of the performance improvement from our proposed layer-wise approach. Fig 9 additionally depicts the general improvement and convergence of ROAD values across all model layers. Interestingly, this layer-wise series of values affords greater insight into the general functioning and utility of various model layer contributions across experiments. While Fig 9A, 9B, 9D and 9E all seem to generally improve in ROAD performance from model layer beginning to end, Fig 9C and 9F both appear relatively consistent in their value distribution, perhaps indicative that within these instances, the model/CAM combination had greater difficulty in discriminating the target class within the given image. Certainly, across all experiments we observe a noisy time-series signal suggesting that future work investigate moving average smoothing as a technique to make these curves more interpretable, albeit, as a trade-off for the layer-specific resolution of ROAD values.

The combined consideration of quantitative ROAD and qualitative CAM at every layer enables end-users to identify the best representation for their particular image, target class, and model in a manner less arbitrary than selecting one of several terminal layers. For

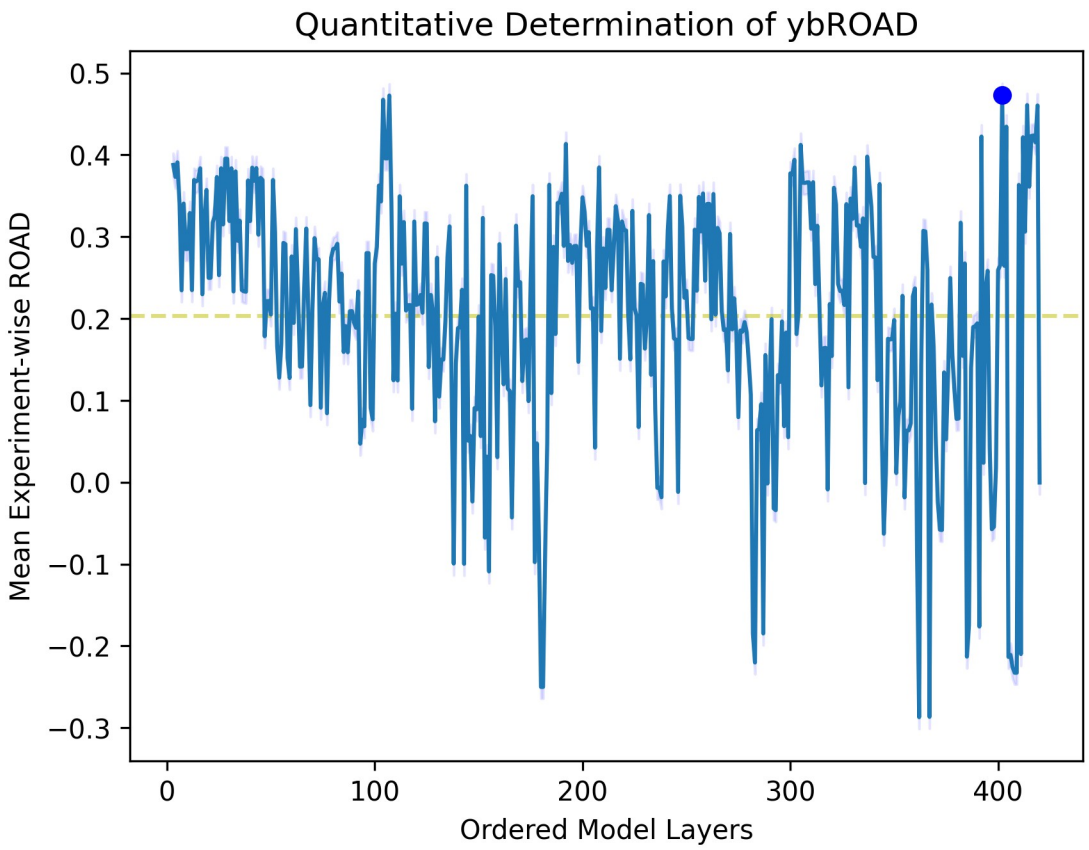

Fig 8. Depiction of end-to-end ROAD values, denoted ybROAD.

https://doi.org/10.1371/journal.pone.0296985.g008

example, a healthcare professional might identify a better representative feature map for a predicted tumour than they might otherwise from a potentially poorer last-layer visualization. This approach effectively allows an end-user to peer across the network and determine those layers that best capture the story as opposed to relying on the final output alone. We caution

Table 2. Quantitative comparison of ybROAD values and SOTA CAM methods for various model architectures, images, and target classes.

Model Architecture	Image Name	Target Class	Selected CAM	Mean Layer-wise ROAD	Final Layer ROAD	ybROAD	Difference (ybROAD—Final)
ResNet152	catdog	dog	ScoreCAM	0.133	-3.50E-07	0.499	+0.499
ResNet152	bear	bear	ScoreCAM	0.107	-6.26E-07	0.491	+0.491
ResNet152	bear	bear	EigenCAM	0.153	-5.96E-08	0.486	+0.486
ResNet152	catdog	cat	ScoreCAM	0.029	-2.92E-07	0.301	+0.301
ResNet152	catdog	dog	GradCAMElementWise	0.120	4.26E-05	0.228	+0.228
ResNet152	catdog	cat	HiResCAM	0.030	1.74E-06	0.222	+0.222
DenseNet169	catdog	dog	EigenCAM	0.043	0.124	0.334	+0.210
DenseNet169	catdog	dog	LayerCAM	0.110	0.111	0.309	+0.197
ResNet152	catdog	cat	LayerCAM	0.049	1.20E-04	0.184	+0.184
DenseNet169	catdog	dog	GradCAMElementWise	0.106	0.107	0.267	+0.160
DenseNet169	bear	bear	EigenCAM	0.167	0.429	0.470	+0.042
DenseNet169	bear	bear	HiResCAM	0.128	0.452	0.463	+0.011

https://doi.org/10.1371/journal.pone.0296985.t002

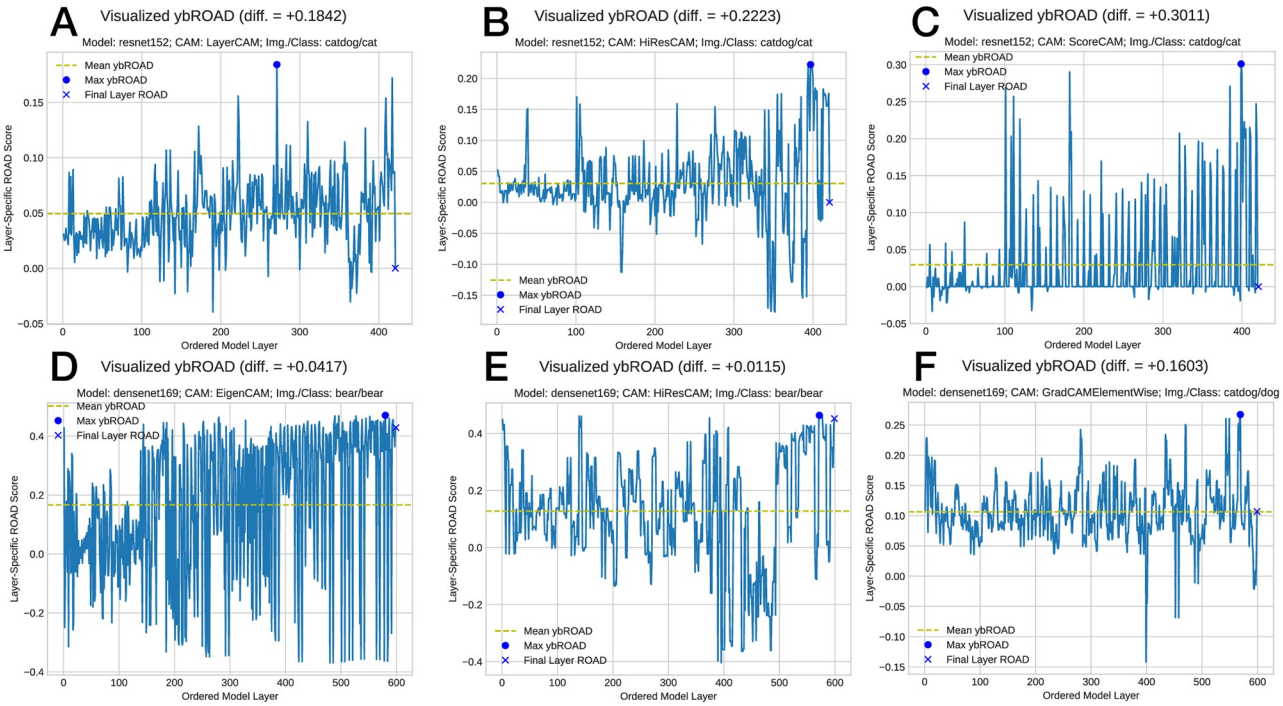

Fig 9. Quantitative determination of ybROAD & visualization of model convergence to the target class.

https://doi.org/10.1371/journal.pone.0296985.g009

that this may introduce additional risk for confirmation bias, however, this has broadly been a challenge within the XAI community.

CAM failure cases

Interestingly, EigenCAM incorrectly highlights the dog in the image, instead of the desired cat class. This explains the negative ROAD value for EigenCAM in Fig 10. EigenCAM is a non-

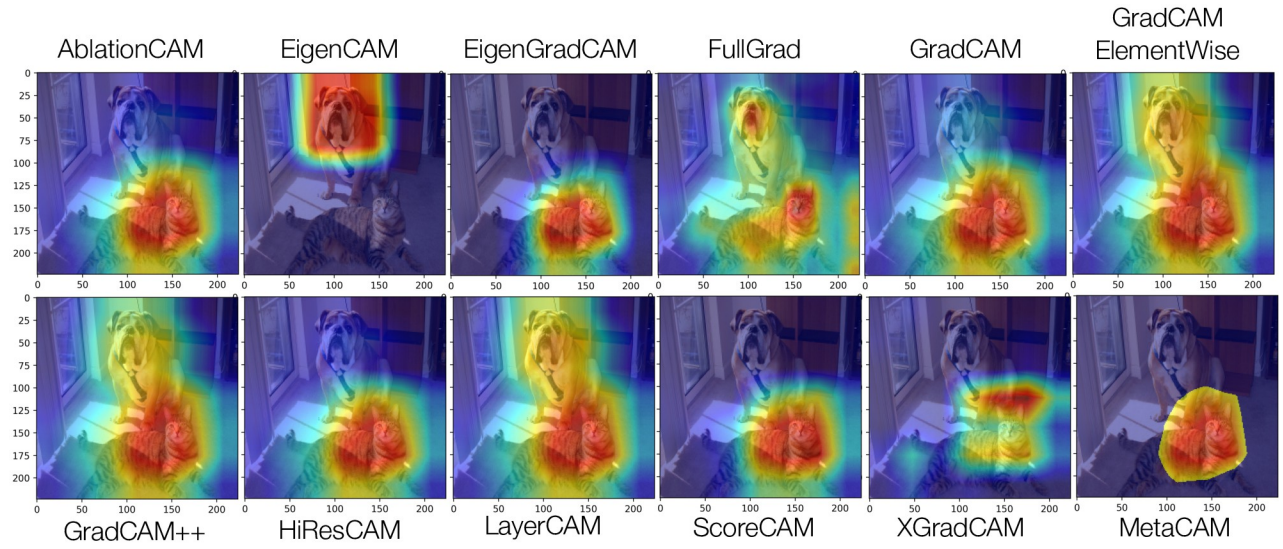

Fig 10. Multiple CAM demonstration with varied target classes ('catdog' image with target classes) and EigenCAM failure case.

https://doi.org/10.1371/journal.pone.0296985.g010

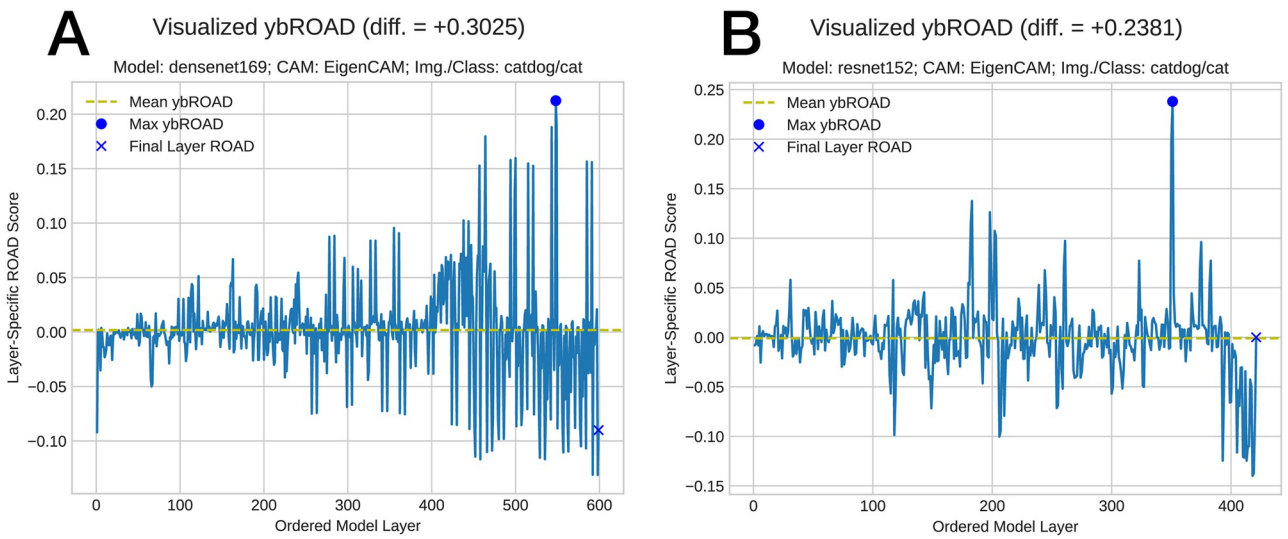

Fig 11. Failure case where the ybROAD plot indicates that the model was unable to correctly distinguish the correct object within this image, class, model, and CAM combination.

https://doi.org/10.1371/journal.pone.0296985.g011

discriminative CAM method that uses principal components to create activation maps. However, when there are multiple classes within the same image, the order of principal components must be specified (*e.g.*, first principal components vs. second principal components). EigenCAM performs well on images with a "single-subject", but otherwise requires a user to determine the number and rank of various components within an image to perform successfully. This requires a level of hand-engineering and data leakage to correctly align the appropriate principal component with the intended class.

The ybROAD plots proposed within this work can be leveraged to better understand whether a model adequately distinguishes a given class or whether it fails across all layers of the model. As visualized in the ybROAD plots of Fig 11 the mean layerwise ROAD value around 0 effectively demonstrate that the model was unable to identify the correct class within the image. Consequently, the ybROAD quantitative metrics derived from the ybROAD plots may be useful in elucidating the impact of model architecture (and their learned parameters) on a class-specific basis. As part of future work, this concept could be extended to consider epoch-wise ybROAD plots to better determine how specific layers through model training contribute to the discrimination of the target class.

Future directions

The proposed future directions for research represent individual contributions that can significantly advance the use of CAMs for CNNs. Supporting materials in continuation of this work can be found in S1 File. Foremost, conducting more in-depth studies on the activation maps statistics at different layers and for different images can provide a better understanding of how CNNs attend to images in varying applications and contexts. Secondly, designing an algorithm to efficiently compute CAM-based videos would greatly improve the applicability of this technique in various fields, particularly those that require inference or interpretability in near-real-time. Thirdly, using activation maps sequences to identify useless layers/filters represents a novel approach towards network compression purposes. Fourthly, exploring the behavior of activation maps sequences for wrong classes and finding ways to exploit this information for

classification purposes is a unique contribution. Lastly, coupling CAM-based videos with expert feedback in specific applications can result in a more interpretable and accurate model. Overall, these individual research contributions have the potential to improve the performance, efficiency, and interpretability of CNN models, leading to advancements in various image classification tasks and promoting large-scale and transparent adoption of these models.

Conclusion

This work proposes CAManim as a novel XAI visualization method enabling end-users to better interpret CNN predictions by animating the CAM-based network activation maps through all layers. We demonstrate that CAManim works with any CAM-based method and various CNN architectures. We additionally introduce a quantitative end-to-end assessment inspired from the ROAD metric, denoted "yellow brick ROAD" (ybROAD). Our experiments demonstrate the utility of these methods for improved interpretation and understanding of CNN predictions, not only in their final layers but across their layer-specific and global-wise perspectives. Visualizations and source code can be found at: https://omni-ml.github.io/pytorch-grad-cam-anim/.

Supporting information

S1 File. Code and and data availability.
(PDF)

Acknowledgments

The authors would like to acknowledge Dr. Katherine Muldoon for her support of this work. The authors also acknowledge that this study took place on unceded Algonquin Anishinabe territory.

Author Contributions

Conceptualization: Emily Kaczmarek, Olivier X. Miguel, Mark C. Walker, Kevin Dick.

Data curation: Emily Kaczmarek, Olivier X. Miguel, Kevin Dick.

Formal analysis: Emily Kaczmarek, Kevin Dick.

Investigation: Emily Kaczmarek, Olivier X. Miguel, Kevin Dick.

Methodology: Emily Kaczmarek, Steven Hawken, Kevin Dick.

Project administration: Kevin Dick.

Resources: Emily Kaczmarek, Alexa C. Bowie, Robin Ducharme, Alysha L. J. Dingwall-Harvey, Mark C. Walker, Kevin Dick.

Software: Emily Kaczmarek, Olivier X. Miguel, Kevin Dick.

Supervision: Christine M. Armour, Mark C. Walker, Kevin Dick.

Validation: Emily Kaczmarek, Kevin Dick.

Visualization: Emily Kaczmarek, Olivier X. Miguel, Kevin Dick.

Writing – original draft: Emily Kaczmarek, Kevin Dick.

Writing – review & editing: Emily Kaczmarek, Olivier X. Miguel, Alexa C. Bowie, Robin Ducharme, Alysha L. J. Dingwall-Harvey, Steven Hawken, Christine M. Armour, Mark C. Walker, Kevin Dick.

References

1. Khan A, Sohail A, Zahoora U, Qureshi AS. A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review. 2019; 53:5455–5516. https://doi.org/10.1007/s10462-020-09825-6
2. Walker MC, Willner I, Miguel OX, Murphy MS, El-Chaâr D, Moretti F, et al. Using deep-learning in fetal ultrasound analysis for diagnosis of cystic hygroma in the first trimester. Plos one. 2022; 17(6): e0269323. https://doi.org/10.1371/journal.pone.0269323 PMID: 35731736
3. Zhou X, Li Y, Liang W. CNN-RNN based intelligent recommendation for online medical pre-diagnosis support. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2020; 18(3):912–921. https://doi.org/10.1109/TCBB.2020.2994780
4. Singh SP, Wang L, Gupta S, Goli H, Padmanabhan P, Gulyás B. 3D deep learning on medical images: a review. Sensors. 2020; 20(18):5097. https://doi.org/10.3390/s20185097 PMID: 32906819
5. Hou B, Kaissis G, Summers RM, Kainz B. Ratchet: Medical transformer for chest x-ray diagnosis and reporting. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VII 24. Springer; 2021. p. 293–303.
6. Chintagunta B, Katariya N, Amatriain X, Kannan A. Medically aware GPT-3 as a data generator for medical dialogue summarization. In: Machine Learning for Healthcare Conference. PMLR; 2021. p. 354–372.
7. Cheng K, Guo Q, He Y, Lu Y, Gu S, Wu H. Exploring the Potential of GPT-4 in Biomedical Engineering: The Dawn of a New Era. Annals of Biomedical Engineering. 2023; p. 1–9. PMID: 37115365
8. Haupt CE, Marks M. AI-Generated Medical Advice—GPT and Beyond. JAMA. 2023; 329(16):1349–1350. https://doi.org/10.1001/jama.2023.5321 PMID: 36972070
9. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13. Springer; 2014. p. 818–833.
10. Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:13126034. 2013;.
11. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M. Striving for simplicity: The all convolutional net. arXiv preprint arXiv:14126806. 2014;.
12. Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. In: International conference on machine learning. PMLR; 2017. p. 3319–3328.
13. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning Deep Features for Discriminative Localization. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA, USA: IEEE Computer Society; 2016. p. 2921–2929. Available from: https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.319.
14. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. 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.
15. Chattopadhay A, Sarkar A, Howlader P, Balasubramanian VN. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE winter conference on applications of computer vision (WACV). IEEE; 2018. p. 839–847.
16. Fu R, Hu Q, Dong X, Guo Y, Gao Y, Li B. Axiom-based Grad-CAM: Towards Accurate Visualization and Explanation of CNNs; 2020.
17. Gildenblat J. PyTorch library for CAM methods; 2021. https://github.com/jacobgil/pytorch-grad-cam.
18. Draelos RL, Carin L. Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks; 2020. Available from: https://arxiv.org/abs/2011.08891.
19. Jiang PT, Zhang CB, Hou Q, Cheng MM, Wei Y. LayerCAM: Exploring Hierarchical Class Activation Maps For Localization. IEEE Transactions on Image Processing. 2021;. https://doi.org/10.1109/TIP.2021.3089943 PMID: 34156941
20. Wang H, Wang Z, Du M, Yang F, Zhang Z, Ding S, et al. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops; 2020. p. 24–25.

21. Desai S, Ramaswamy HG. Ablation-CAM: Visual Explanations for Deep Convolutional Network via Gradient-free Localization. In: 2020 IEEE Winter Conference on Applications of Computer Vision (WACV); 2020. p. 972–980.

22. Muhammad MB, Yeasin M. Eigen-cam: Class activation map using principal components. In: 2020 International Joint Conference on Neural Networks (IJCNN). IEEE; 2020. p. 1–7.

23. Lipton ZC. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue. 2018; 16(3):31–57. https://doi.org/10.1145/3236386.3241340

24. Ribeiro MT, Singh S, Guestrin C. "Why should i trust you?" Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. p. 1135–1144.

25. Srinivas S, Fleuret F. Full-gradient representation for neural network visualization. Advances in neural information processing systems. 2019;32.

26. Adebayo J, Gilmer J, Muelly M, Goodfellow I, Hardt M, Kim B. Sanity checks for saliency maps. Advances in neural information processing systems. 2018;31.

27. Kindermans PJ, Hooker S, Adebayo J, Alber M, Schütt KT, Dähne S, et al. The (un) reliability of saliency methods. Explainable AI: Interpreting, explaining and visualizing deep learning. 2019; p. 267–280.

28. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Object detectors emerge in deep scene CNNs. arXiv preprint arXiv:14126856. 2014;.

29. Morcos AS, Barrett DG, Rabinowitz NC, Botvinick M. On the importance of single directions for generalization. arXiv preprint arXiv:180306959. 2018;.

30. Zhou B, Sun Y, Bau D, Torralba A. Revisiting the importance of individual units in CNNs via ablation. arXiv preprint arXiv:180602891. 2018;.

31. Kaczmarek E, Miguel OX, Bowie AC, Ducharme R, Dingwall-Harvey AL, Hawken S, et al. MetaCAM: Ensemble-Based Class Activation Map. arXiv preprint arXiv:230716863. 2023;.

32. Huang G, Liu Z, Weinberger KQ. Densely Connected Convolutional Networks. CoRR. 2016;abs/1608.06993.

33. Rong Y, Leemann T, Borisov V, Kasneci G, Kasneci E. A Consistent and Efficient Evaluation Strategy for Attribution Methods. In: Proceedings of the 39th International Conference on Machine Learning. PMLR; 2022. p. 18770–18795.

34. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. Ieee; 2009. p. 248–255.

35. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012;25.

36. Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S. A ConvNet for the 2020s. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022;.

37. Tan M, Le QV. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv. 2019.

38. Tu Z, Talebi H, Zhang H, Yang F, Milanfar P, Bovik A, et al. MaxViT: Multi-Axis Vision Transformer. ECCV. 2022;.

39. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv:160207360. 2016;. https://doi.org/10.48550/arXiv.1905.11946

40. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. CoRR. 2015;abs/1512.03385.

41. Al-Dhabyani W, Gomaa M, Khaled H, Fahmy A. Dataset of breast ultrasound images. Data in Brief. 2020; 28:104863. https://doi.org/10.1016/j.dib.2019.104863 PMID: 31867417