FPN-IAIA-BL: A Multi-Scale Interpretable Deep Learning Model for Classification of Mass Margins in Digital Mammography

Julia Yang
Duke University
Durham, NC, USA
julia.yang@duke.edu

Alina Jade Barnett
Duke University
Durham, NC, USA
alina.barnett@duke.edu

Jon Donnelly
Duke University
Durham, NC, USA
jon.donnelly@duke.edu

Satvik Kishore
Duke University
Durham, NC, USA
satvik.kishore@duke.edu

Jerry Fang
Duke University
Durham, NC, USA
jerry.d.fang@alumni.duke.edu

Fides Regina Schwartz
Brigham and Women's Hospital
Boston, MA, USA
frschwartz@bwh.harvard.edu

Chaofan Chen
University of Maine
Orono, ME, USA
chaofan.chen@maine.edu

Joseph Y. Lo
Duke University
Durham, NC, USA
joseph.lo@duke.edu

Cynthia Rudin
Duke University
Durham, NC, USA
cynthia@cs.duke.edu

Abstract

Digital mammography is essential to breast cancer detection, and deep learning offers promising tools for faster and more accurate mammogram analysis. In radiology and other high-stakes environments, uninterpretable ("black box") deep learning models are unsuitable and there is a call in these fields to make interpretable models. Recent work in interpretable computer vision provides transparency to these formerly black boxes by utilizing prototypes for case-based explanations, achieving high accuracy in applications including mammography. However, these models struggle with precise feature localization, reasoning on large portions of an image when only a small part is relevant. This paper addresses this gap by proposing a novel multi-scale interpretable deep learning model for mammographic mass margin classification. Our contribution not only offers an interpretable model with reasoning aligned with radiologist practices, but also provides a general architecture for computer vision with user-configurable prototypes from coarse- to fine-grained prototypes.

1. Introduction

Digital mammography plays a crucial role in detecting and diagnosing breast cancer, a pervasive health concern worldwide. Advancements in deep learning and computer vision have increased the speed and accuracy of lesion classifications for mammography. However, when used for high-stakes tasks like medical diagnoses, deep learning models should be inherently interpretable so that, among other advantages, models can be "fact checked" [18].

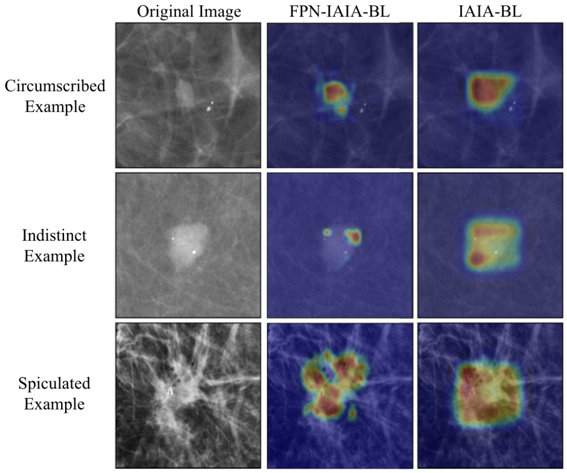


Figure 1. **Activation maps for FPN-IAIA-BL in comparison to IAIA-BL.** FPN-IAIA-BL can learn human interpretable prototypes at any scale, including fine-grained details most salient to mass margin classification.

Recent work has shown that interpretable, case-based machine learning models can provide accurate, human understandable explanations for their predictions while performing on par with other state-of-the-art models [6, 13]. These prototype-based deep learning models have also been applied to digital mammography by Barnett et al. [4], who developed the Interpretable AI Algorithm for Breast Lesions (IAIA-BL) model, an interpretable model for *mass margin* classification. They focused on classification on *margins*, a descriptor of the edges around the mass, because it is a key factor in identifying cancerous lesions under the Breast Imaging Reporting and Data System (BI-RADS). IAIA-BL successfully classified margins using prototypes, as shown in the third column of Figure 1. However, the prototypes often identified more than just the margin or even the entire lesion, leaving any detailed analysis of the margin to the user.

To address this gap, we develop FPN-IAIA-BL, a multi-scale interpretable deep learning model for mammographic mass margin classification. It can be configured to provide prototypes at various levels of granularity, with multiple scales within the same model. We build the model's architecture using both the Feature Pyramid Network (FPN) and IAIA-BL model. We developed a new training schedule and objective function, as the training methods and loss terms used by these predecessors were insufficient to train the combined architecture. The main contributions of this work are that:

- We develop an inherently interpretable deep learning architecture that learns prototypes at multiple scales.
- We train FPN-IAIA-BL, which provides specific prototype activations for mass margin classification.

2. Related Work

Interpretability of deep learning models is critical for high-stakes applications like breast cancer detection and diagnosis. In recent years, *inherently interpretable* deep neural networks have grown in popularity. As compared to *posthoc explanation* techniques such as saliency visualizations [1, 21–24, 29], activation maximization [9, 17, 25, 27, 28], and image perturbation methods [10, 11] which approximate model reasoning after training, *inherently interpretable* techniques such as [2, 3, 6, 8, 13, 15, 16, 19, 26] provide explanations guaranteed to be faithful to the model's underlying decision-making process.

FPN-IAIA-BL uses inherently interpretable case-based reasoning with prototypes by building upon IAIA-BL [4], a case-based model for mass margin classification. IAIA-BL was limited to learning prototypes at only one scale, with prototypes often identifying more of the image than is relevant for margin classification. In contrast, FPN-IAIA-BL learns prototypes at various scales including highly-localized, fine-grained prototypes that select small details, as shown in Figure 1. This is possible because FPN-IAIA-BL incorporates features at various scales.

Typically, a key challenge in mammogram analysis is capturing information at various scales, since traditional CNN architectures focus on a single image resolution. Multi-scale approaches like [7] and [14] address this challenge by incorporating features extracted at different scales within the network. A foundational architecture for multi-scale predictions is the Feature Pyramid Network (FPN) [14] which introduces a bottom-up and top-down pyramidal architecture that produces multiple feature maps from fine-grained to coarse. As a result, FPN's are able to localize to objects of multiple scales for object detection.

Our FPN-IAIA-BL architecture leverages this bottom-up and top-down pyramidal architecture to learn prototypes at multiple scales by augmenting IAIA-BL's VGG-16 backbone with a similar structure, detailed in Section 3. Furthermore, our model also provides visual, human interpretable, case-based reasoning for each classification.

3. FPN-IAIA-BL Architecture

Inspired by the Feature Pyramid Network (FPN), the FPN-IAIA-BL model adds lateral and top-down connections to the original VGG-16 convolutional layers used in the IAIA-BL architecture as its foundation. Figure 2 illustrates this architecture. The model consists first of an FPN that extracts useful feature maps at multiple scales, allowing for more varied representation than single-scale IAIA-BL. The FPN is followed by the prototype layer g in which the input image's feature maps are compared to learned prototypes to produce similarity scores. Fully connected layer h then uses the similarity scores to produce margin class predictions.

3.1. Multi-Scale Feature Maps from Feature Pyramid Network

IAIA-BL [4] uses a CNN to create a single feature map z which limits the network to prototypes at the scale of that output feature map. In contrast, FPN-IAIA-BL uses the latent feature maps from multiple layers in the CNN, which have different spatial and semantic scales. Thus, the output of the set of convolutional layers f in FPN-IAIA-BL is a set of feature maps of varying spatial scale, which we refer to as the feature pyramid $f(x) = \mathbf{Z} = \{\mathbf{z}^{(2)}, \mathbf{z}^{(3)}, \mathbf{z}^{(4)}, \mathbf{z}^{(5)}\}$. For our implementation, the coarsest feature maps were 14 by 14, and finest were 56 by 56.

For the VGG-16 backbone, we use the output from each block's max-pooling layer to form the intermediate feature map levels in the bottom-up pathway (left column of backbone in Figure 2). We also include the final output of the convolutional layers as a feature map at the top. We denote these bottom-up feature maps as $\mathbf{C} = \{\mathbf{c}^{(2)}, \mathbf{c}^{(3)}, \mathbf{c}^{(4)}, \mathbf{c}^{(5)}\}$ where $\mathbf{c}^{(2)}$ is the base of the bottom-up pyramid, and $\mathbf{c}^{(5)}$ is the top.

As in FPN [14], the top-down pathway produces a second feature pyramid. For each level, an upsampled feature map with spatially coarser information is combined with

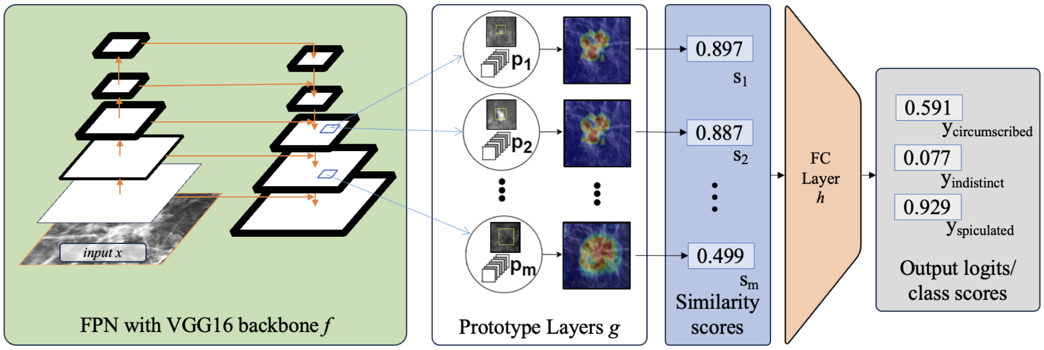


Figure 2. **FPN-IAIA-BL Architectre.** The input image $\mathbf{x}$ passes through convolutional layers f consisting of an FPN with a VGG-16 backbone, which creates an pyramid of feature maps $f(\mathbf{x})$. Each patch of each level of the feature pyramid (referred to as FPN level) is then compared to each prototype of the same FPN level using a cosine distance to produce an activation map. The activation map is then used to calculate an overall similarity score s_j between the input image and the prototype for each prototype. Finally, a set of fully connected last layer produces logits y_{margin} for each margin class.

a corresponding laterally connected feature maps from the bottom-up pyramid. Then, each combined feature map is passed through a 3×3 convolution to reduce the aliasing effect of upsampling and output the feature map $\mathbf{z}^{(l)}$.

$$\mathbf{z}^{(5)} = \text{Conv1x1}(\mathbf{c}^{(5)})$$
$$\mathbf{z}^{(l)} = \text{Conv3x3}\Big(\text{Up}(\mathbf{z}^{(l+1)}) + \text{Conv1x1}(\mathbf{c}^{(l)})\Big); l \in \{2,3,4\} \tag{1}$$

3.2. Prototype Layer

In the prototype layer g, we have m prototypes where each prototype can be configured to represent a specific class c and FPN level l. For m prototypes, let $S = \{(c_j, l_j, j)\}_{j=1}^{m}$ represent our prototype configuration, and denote our prototypes as $\mathbf{P} = \{\mathbf{p}^{(c,l,j)}\}_S$ where the j-th prototype is from class c with FPN level l. Each prototype is $1 \times 1 \times d$ so that each prototype has the same feature dimension d as the convolutional feature pyramid. As in IAIA-BL [4], the prototypes can be interpreted as a characteristic pattern representing a specific class. It can be visually understood by examining a segment of the training image where this pattern was derived.

Once we have computed each feature map in the convolutional feature pyramid $f(x)$, we compute the similarity between each prototype in prototype layer g and the corresponding feature map. The FPN-IAIA-BL similarity score s_j differs from that of IAIA-BL in three ways.

First, because the prototypes are assigned to specific FPN levels l, similarities for a set of prototypes $\mathbf{p}^{(\cdot,l,\cdot)}$ are computed only using the feature map from the same FPN level $\mathbf{z}^{(l)}$. Second, instead of using inverted L_2 distance based similarity, we use a cosine similarity as described in [8, 26]. The cosine similarity is calculated between a prototype and each $1 \times 1 \times d$ patch within the corresponding feature map. We denote the patch in a feature map of size $\eta_l \times \eta_l \times d$ as $n \in \{(1,1), \dots, (1,\eta_l), (2,1), \dots, (\eta_l, \eta_l)\}$. Thus, the cosine similarity for a single patch is:

$$s_{j,n}^{(l)} = \frac{\mathbf{z}_n^{(l)}}{||\mathbf{z}_n^{(l)}||} \cdot \frac{\mathbf{p}^{(c,l,j)}}{||\mathbf{p}^{(c,l,j)}||} \tag{2}$$

Third, in order to focus activation on the most salient features in each image, we use focal similarity as introduced in ProtoPool [19]. Retaining the top-k average pooling from Kalchbrenner et al. [12] and IAIA-BL [4], focal cosine similarity is computed as:

$$g(l,j) = \frac{1}{k}\sum \text{top}_k(\{s_{j,n}^{(l)}\}_{n=(1,1)}^{(\eta_l,\eta_l)}) - \frac{1}{\eta_l^2}\sum_{n=(1,1)}^{(\eta_l,\eta_l)}(\{s_{j,n}^{(l)}\}_{n=(1,1)}^{(\eta_l,\eta_l)}) \tag{3}$$

The last stage of FPN-IAIA-BL is a fully connected layer h which weights the similaritie scores and applies a softmax to predict probabilities for each mass-margin class.

4. Data and Training

The dataset, previously studied in [4], includes 2D digital breast x-rays from patients at the Duke University Health System taken between 2008 and 2018. Data collection was approved by Duke Health IRB and labeled by a fellowship-trained breast imaging radiologist. While IAIA-BL used only the subset of the images that contained a lesion, we also introduce a negative class which consists of images of tissue without lesions. Supplement Section C details how the data for this class were generated.

The training of FPN-IAIA-BL consists of three stages: (A) a warmup stage, (B) a projection of prototypes, and (C)

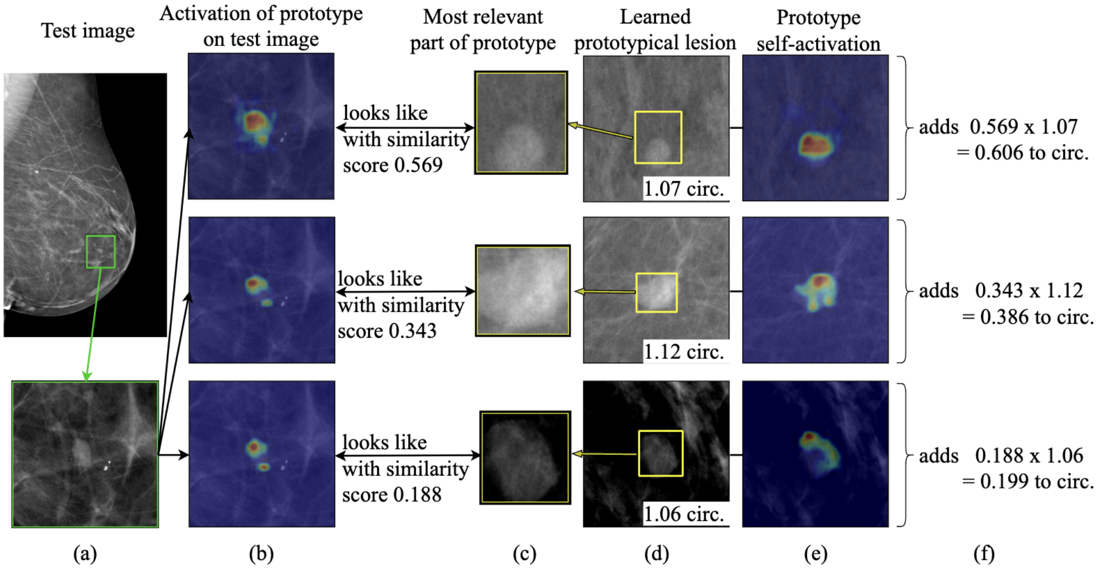

Test image
Activation of prototype on test image
Most relevant part of prototype
Learned prototypical lesion
Prototype self-activation
looks like with similarity score 0.569
1.07 circ.
adds 0.569 x 1.07 = 0.606 to circ.
looks like with similarity score 0.343
1.12 circ.
adds 0.343 x 1.12 = 0.386 to circ.
looks like with similarity score 0.188
1.06 circ.
adds 0.188 x 1.06 = 0.199 to circ.
(a)
(b)
(c)
(d)
(e)
(f)

Figure 3. **Case-based explanation generated by FPN-IAIA-BL**. This circumscribed (circ.) lesion is correctly classified as circumscribed. a, Test images. b, Activation of prototype on test images. c, Most relevant part of prototype. d, Learned prototypical lesion. e, Prototype self-activation. f, Contribution to class score. This visualization format for this figure matches that of [4].

full network fine-tuning. Because we use a trained VGG-16 backbone from IAIA-BL to construct our FPN, we first freeze the VGG-16 backbone in Stage A to warm up all the other layers. Stage B projects the learned prototype vectors onto a patch from any input image's corresponding feature map in the same fashion as in [4, 6]. Stage C continues these two stages and unfreezes the VGG-16 backbone to allow for fine-tuning of the full network.

For stages A and C, we minimize the loss function:

$$\ell = \text{CE} + \lambda_1 \ell_{clust} + \lambda_2 \ell_{sep} + \lambda_3 \ell_{ortho} + \lambda_4 \ell_{fine} \quad (4)$$

where cross entropy (CE) penalizes misclassification and $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are coefficients chosen empirically to balance the cluster (ℓ_{clust}), separation (ℓ_{sep}), and orthogonality (ℓ_{ortho}) losses as defined in [8] and fine-annotation loss (ℓ_{fine}) modified from [4]. The modifications to the fine-annotation loss introduce user-configurable coefficients which encourage and penalize the model for activating inside and outside the fine annotations differently for each class pair. Supplement Section B details the fine-annotation coefficients.

These loss terms have not previously been combined.

5. Experiments and Results

In our experiments, we find that FPN-IAIA-BL is able to learn localized prototypes that achieve acceptable performance. An interpretable visual result of the FPN-IAIA-BL is shown in Figure 3 and is compared to baselines in Figure 4. The best performing FPN-IAIA-BL model was able to achieve an average AUROC of 0.88 with one-vs-rest AUROC's of 0.865 for circumscribed, for indistinct, and 0.908 for spiculated margin classes. A further comparison of the performance with IAIA-BL and an uninterpretable baseline (VGG16) is presented in Table 1. The confusion matrix of this model is shown in Supplement Section A.

	Avg. AUROC	Circ.	Ind.	Spic
FPN-IAIA-BL	0.88	0.87	0.86	0.91
IAIA-BL	0.95	0.97	0.93	0.96
VGG16	0.95	0.95	0.94	0.95

Table 1. AUROC metrics for FPN-IAIA-BL as compared to IAIA-BL and the uninterpretable baseline (VGG16).

As shown in Figure 5, prototypes from each FPN level represent relevant features from multiple scales. FPN-level 2 localizes to the most fine-grained features, and FPN-level 5 activations cover large swaths of the image. **The model successfully learned prototypes at each FPN-level that captured information of different scales. In our application for mass margin classification, FPN-level 3 provided prototypes that activated on the most specific and salient parts of the margin**. In other applications, the FPN-level of each prototype can be configured such that the prototypes capture the most relevant scale of information for the application. Figure 4 compares the activation maps provided by FPN-IAIA-BL, IAIA-BL, ProtoPNet, GradCAM and GradCAM++. The explanations from FPN-IAIA-BL highlight the most important parts of the lesion margin.

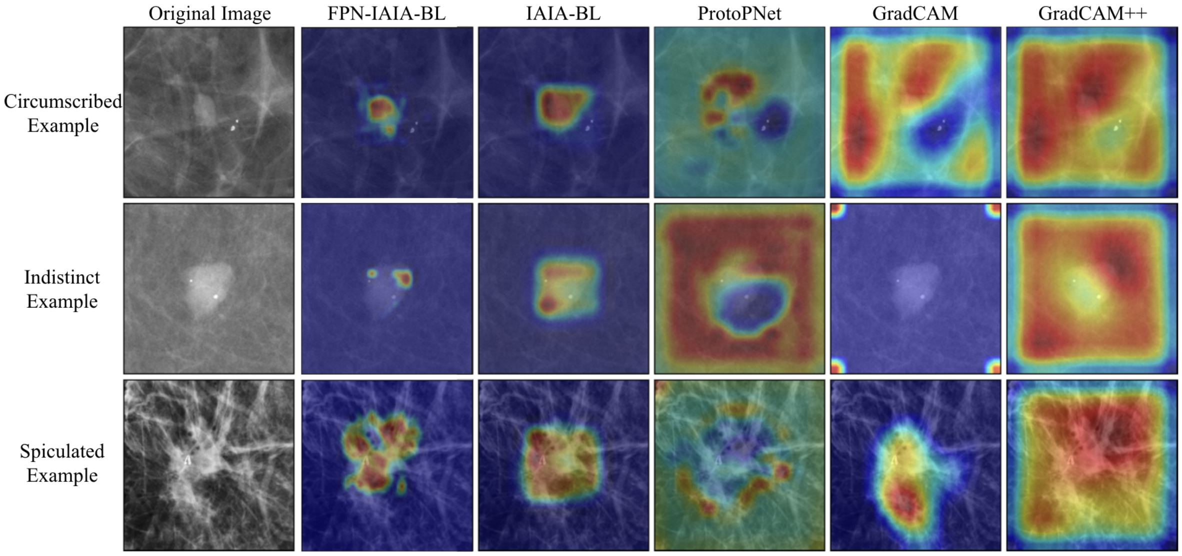
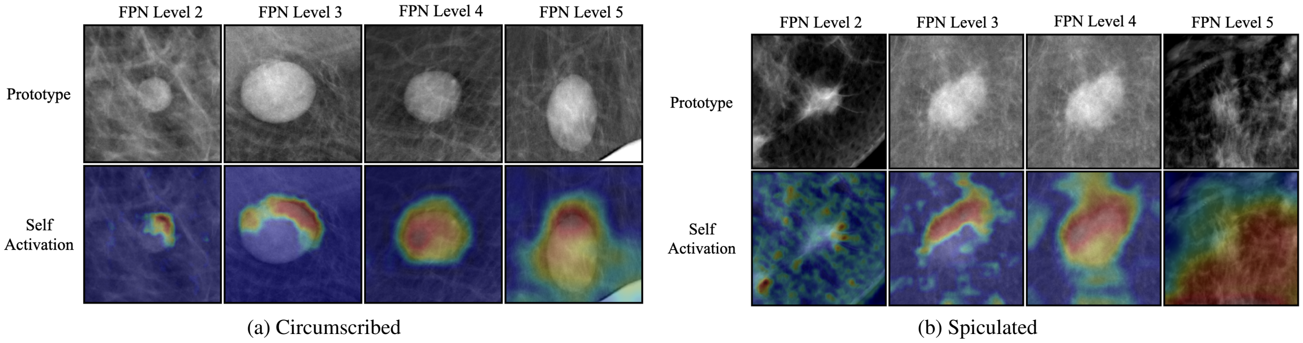


Figure 4. **FPN-IAIA-BL in comparison to other saliency methods (adapted from [4]).** We compare explanations from FPN-IAIA-BL with GradCAM [20], GradCAM++ [5], ProtoPNet [6], and IAIA-BL [4]. GradCam and GradCAM++ are two popular saliency explanation methods, and ProtoPNet and IAIA-BL are case-based explanation methods. The explanations from FPN-IAIA-BL highlight the most important parts of the lesion margin.



(a) Circumscribed

(b) Spiculated

Figure 5. **Learned prototypes at different FPN-levels.** FPN-level 2 prototypes are more localized because they are learned from the base of the feature pyramid which is a finer-grained feature map while FPN-level 5 prototypes are learned from the top of the feature pyramid, a coarser-grained feature map.

5.1. Limitations

While FPN-IAIA-BL consistently produces prototypes for circumscribed and spiculated lesion that our radiology team finds compelling, the prototypes for indistinct margins often activate outside of the lesion. This could be because an indistinct margin is defined as a faded, soft boundary between the lesion and normal tissue, and soft boundaries can occur in healthy breast tissue. Additionally, the AUROC for FPN-IAIA-BL is lower than that of IAIA-BL (0.951 overall) and the uninterpretable baseline (0.947 overall). This is because FPN-IAIA-BL architecture is larger and harder to train than IAIA-BL and the baseline.

6. Conclusion

We presented FPN-IAIA-BL, a novel neural network architecture for multi-scale case-based reasoning. We showed its effectiveness for the task of breast lesion margin classification, creating a model that can articulate more detailed reasoning behind its predictions, improving interpretability.

Acknowledgements

This study was supported by National Science Foundation (grant HRD-2222336), Duke TRIPODS CCF-1934964, Duke MEDx: High-Risk High-Impact Challenge, and the Duke Incubation Fund.

References

[1] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015. 2

[2] Alina Jade Barnett, Zhicheng Guo, Jin Jing, Wendong Ge, Cynthia Rudin, and M Brandon Westover. Interpretable machine learning system to eeg patterns on the ictal-interictal-injury continuum. *arXiv preprint arXiv:2211.05207*, 2022. 2

[3] Alina Jade Barnett, Zhicheng Guo, Jin Jing, Wendong Ge, Cynthia Rudin, and M Brandon Westover. Mapping the ictal-interictal-injury continuum using interpretable machine learning. *arXiv preprint arXiv:2211.05207*, 2022. 2

[4] Alina Jade Barnett, Fides Regina Schwartz, Chaofan Tao, Chaofan Chen, Yinhao Ren, Joseph Y Lo, and Cynthia Rudin. A case-based interpretable deep learning model for classification of mass lesions in digital mammography. *Nature Machine Intelligence*, 3(12):1061–1070, 2021. 2, 3, 4, 5

[5] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018. 5

[6] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This Looks Like That: Deep Learning for Interpretable Image Recognition. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 8930–8941, 2019. 2, 4, 5

[7] Zhicheng Cui, Wenlin Chen, and Yixin Chen. Multi-scale convolutional neural networks for time series classification. *CoRR*, abs/1603.06995, 2016. 2

[8] Jon Donnelly, Alina Jade Barnett, and Chaofan Chen. Deformable protopnet: An interpretable image classifier using deformable prototypes, 2022. 2, 3, 4

[9] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing Higher-Layer Features of a Deep Network. Technical Report 1341, University of Montreal, June 2009. Also presented at the ICML 2009 Workshop on Learning Feature Hierarchies, Montreal, Canada. 2

[10] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks, 2019. 2

[11] Maksims Ivanovs, Roberts Kadikis, and Kaspars Ozols. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recogn. Lett.*, 150(C):228–234, oct 2021. 2

[12] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, 2014. 3

[13] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep Learning for Case-Based Reasoning through Prototypes: A Neural Network that Explains Its Predictions. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 2

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017. 2

[15] Chiyu Ma, Brandon Zhao, Chaofan Chen, and Cynthia Rudin. This looks like those: Illuminating prototypical concepts using multiple visualizations. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[16] Meike Nauta, Ron Van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14933–14943, 2021. 2

[17] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 3387–3395, 2016. 2

[18] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. 1

[19] Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment, 2022. 2, 3

[20] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 5

[21] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising Image Classification Models and Saliency Maps. In *International Conference on Learning Representations (ICLR) Workshop*, 2014. 2

[22] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. SmoothGrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[23] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for Simplicity: The All Convolutional Net. *arXiv preprint arXiv:1412.6806*, 2014.

[24] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. 2

[25] Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 875–884, 2021. 2

[26] Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 895–904, 2021. 2, 3

[27] Naoya Yoshimura, Takuya Maekawa, and Takahiro Hara. Toward understanding acceleration-based activity recognition neural networks with activation maximization. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. 2

[28] Jason Yosinski, Jeff Clune, Thomas Fuchs, and Hod Lipson. Understanding Neural Networks through Deep Visualization. In *In ICML Workshop on Deep Learning*, 2015. 2

[29] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 818–833, 2014. 2