# An Interactive Interpretability System for Breast Cancer Screening with Deep Learning

Yuzhe Lu*
Vanderbilt University

Adam Perer†
Carnegie Mellon University.

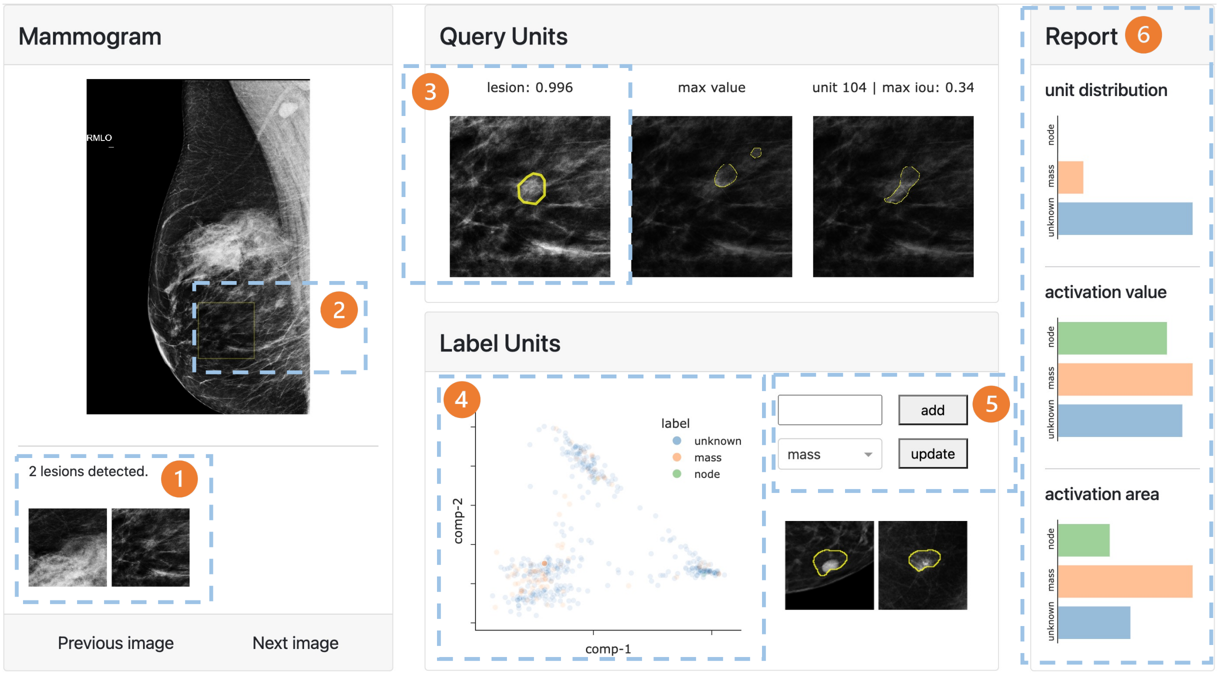

Figure 1: Overview of our interface. (1) potentially malignant patches identified by a patch-based model on the full mammogram; (2) context of a selected patch in the original mammogram; (3) place for users to query model representation with salient regions; (4) visualization of neurons based on their learned representation; (5) place for user to annotate neurons' semantic meaning; (6) explainability reports generated based on neuron annotations.

## Abstract

Deep learning methods, in particular convolutional neural networks, have emerged as a powerful tool in medical image computing tasks. While these complex models provide excellent performance, their black-box nature may hinder real-world adoption in high-stakes decision-making. In this paper, we propose an interactive system to take advantage of state-of-the-art interpretability techniques to assist radiologists with breast cancer screening. Our system integrates a deep learning model into the radiologists' workflow and provides novel interactions to promote understanding of the model's decision-making process. Moreover, we demonstrate that our system can take advantage of user interactions progressively to provide finer-grained explainability reports with little labeling overhead. Due to the generic nature of the adopted interpretability technique, our system is domain-agnostic and can be used for many different medical image computing tasks, presenting a novel perspective on how we can leverage visual analytics to transform originally static interpretability techniques to augment human decision making and promote the adoption of medical AI.

**Index Terms:** Human-centered computing—Visualization—Visualization systems and tools—Visualization toolkits; Computing methodologies—Machine learning—Machine learning approaches—Neural networks;

## 1 Introduction

Breast cancer is the most common cancer among women worldwide and their second leading cause of death. Although early detection and treatment can improve the prognosis of a patient, screening tests have high error rates. Recently, the use of deep learning and big data has made it possible to develop high-performing models for breast cancer screening. In particular, convolutional neural networks (CNNs) have achieved remarkable performance in screening mammography [1, 14]. Moreover, recent reader studies have shown that deep neural networks could enhance radiologists' performance in breast cancer screening [18].

However, the deployment of CNNs in medical domain has its unique challenges. One of the key obstacles is how to effectively allow models and physicians to collaborate effectively on their complementary set of strengths. While CNNs have achieved high performance in various medical imaging domains, it is hard for physicians to understand the model's decision-making process due to its black-box nature. To tackle this challenge, both machine learning (ML) and visualization researchers have made great efforts. Many ML researchers have proposed novel methods to visualize and inter-

[*] e-mail: yuzhe.lu@vanderbilt.edu
[†] e-mail: adamperer@cmu.edu

pret convolutional neural networks [3, 6, 11, 20]. Mostly, the use of saliency maps (i.e, highlighting image features important to the model's decision) plays a central role in these methods. With this observation, a recent work [4] proposes a system that compares saliency maps from deep neural networks to ground truth segmentations of image components to measure Human-AI alignment.

While many ML researchers continued to polish interpretability techniques for deep learning models, few have considered these techniques being applied to medical domains in actual clinical workflows. For example, Wu et al. [17] proposed DeepMiner, where they applied Network Dissection [2] on breast cancer screening models to uncover implicitly learned finer-grained medical concepts to improve interpretability. Although this framework provides an efficient human-in-the-loop paradigm to understand medical CNNs, it is not practical in real clinical settings as the framework requires radiologists to finish labeling neurons of a model before even using it, which adds a huge time overhead. Meanwhile, numerous visual analytics interfaces [5, 9] have been proposed for machine learning models in healthcare applications such as electronic medical records to improve user understanding and support clinical workflow. Given the huge potential of interactive data visualizations in promoting human understanding of complex deep learning models, it's appealing to find a visual analytics solution to integrate state-of-the-art interpretability techniques into clinical workflows with deep learning components to promote Human-AI collaboration and maximize the utility of powerful medical AI models.

To this end, we proposed a novel visual analytics system to transform a generic yet powerful interpretability methodology, Network Dissection [2, 3], and integrate it into radiologists' workflow assisted by deep learning models. Such a system may promote appropriate reliance and adoption of breast cancer screening models. The merits of our system are listed as follows:

- The system empowers radiologists to interactively probe medical AI models by asking whether the model pays attention to certain features when making decisions.
- The system leverages interactions from the radiologist in their workflow to progressively accumulate understanding of the model to provide additional insights.

## 2 Methods

In this section, we will describe the dataset and the deep learning model used for our task, and the interpretability technique utilized in our interactive system.

### 2.1 Dataset

Our mammography dataset is from the University of Pittsburgh Medical Center (UPMC) and has been properly deidentified. Each patient in this dataset has multiple imaging views stored in DICOM format. When suspicious regions are present, a low-resolution copy with radiologist's annotations (ellipses with white boundaries) are saved. After filtering the dataset based on the DICOM header and label information, we identified 2237 patients assigned BIRADS score 0 (whose imaging contains possibly malignant findings) and 2237 patients assigned BIRADS score 1 (whose imaging contains no findings) and BIRADS score 2 (whose imaging contains benign findings) as relevant to our study.

Given these data, we aimed to build a binary classifier that differentiates mammograms of BIRADS 0 patients (possibly malignant) from those of BIRADS 1 and 2 combined (no finding or benign). As it is challenging to fit full high-resolution mammograms into a standard GPU's memory to train deep neural networks, we followed the procedures in [1] to build a patch-based model. To extract patches from mammograms with BIRADS score 0, we first detected radiologists' annotation (typically a white ellipse) in the low-resolution copy and mapped the region to the corresponding high-resolution mammograms; then, we cropped square patches within the ellipse region of the high-resolution mammograms in a sliding-window fashion. We used a patch size of 512 and a step size of 256, extracting 4710 patches from 2237 BIRADS 0 patients' mammograms. To extract normal patches, we cropped $\lceil 4710/2237 \rceil$ patches of size 512 x 512 from all-tissue areas from mammograms of 2237 BIRADS 1 and 2 patients and uniformly sampled 4710 normal patches to create a balanced dataset.

### 2.2 Model

We used a classic convolutional neural network (CNN), VGG16 [15], which consists of 5 blocks of $3 \times 3$ convolutional filters and max pooling layers, 3 fully connected layers, and a softmax activation. As the extracted patches are gray-scale images (512 x 512 x 1), we modified the VGG16 model to accept single-channel input.

To train the model, we split our dataset into train, validation, and test set with a 8:1:1 ratio. During training, we applied data augmentations (random horizontal flip, Gaussian blur) to increase sample diversity. All data are normalized to have $mean = 0.5$ and $std = 0.5$ before sent to the model. We used cross entropy loss as the objective function and trained the model using stochastic gradient descent with the Adam optimizer [8]. After sweeping hyperparameters using Ray Tune [10], we set the batch size to 32 and the learning rate to $1\mathrm{e}{-4}$ and trained the model for 50 epochs. The model with the lowest loss on the validation set was selected for testing. Our model achieved an area under the ROC curve (AUC) score of 0.943 on our test set. Note that our goal is not to build the best-performing model, but rather to develop a well-performing model to prototype and experiment with our interface.

### 2.3 Decode Individual Neurons

The interpretability technique used in this work is mainly based on the Network Dissection methodology [2], which aims to quantify interpretability of individual neurons in a deep CNN. One way of determining the semantic meaning of neurons in a neural network is to look at the characteristics of images that the neuron consistently activates on. A more straightforward formulation is to simply look at the top activated images for each neuron. The process of identifying these images can be described as the following:

For a set of images $X = \{x_i\}_{i=1}^{n}$ and a set of neurons $N = \{n_i\}_{i=1}^{m}$. For each neuron $n_i$, we gather the maximum activation value $a_i$ on each $x_i$. Then, images with the topk activations will be used as the top k activated images for neuron $n_i$.

With top activated images, people could decide which concept a neuron captures. While deciding on concepts captured by a neuron can be done in a scalable way with segmentation models [2, 3] on natural dataset, such a task can be challenging on a medical data set, as training a segmentation model would require a more densely labeled data set, which is generally not available. Thus, the nature of medical image dataset necessitates novel strategies to decode the semantic meaning of neurons in medical AI models to assist user's understanding.

## 3 System Design

The key challenge in deciding the semantic meaning of neurons in a CNN is their shear number. Classic CNNs often have tens or even hundreds of layers, each containing between 512 and 2048 neurons. While previous work [3] has found that neurons in the last CNN layer tend to have the highest level of interpretability, it is still a label-intensive and time-consuming process for domain experts to inspect and label concepts for hundreds if not thousands of neurons in the last layer. To deal with these challenges, we propose an interactive system that takes advantage of the Network Dissection methodology for users to understand their model while avoiding the costly overhead of inspecting and labeling individual neurons. An

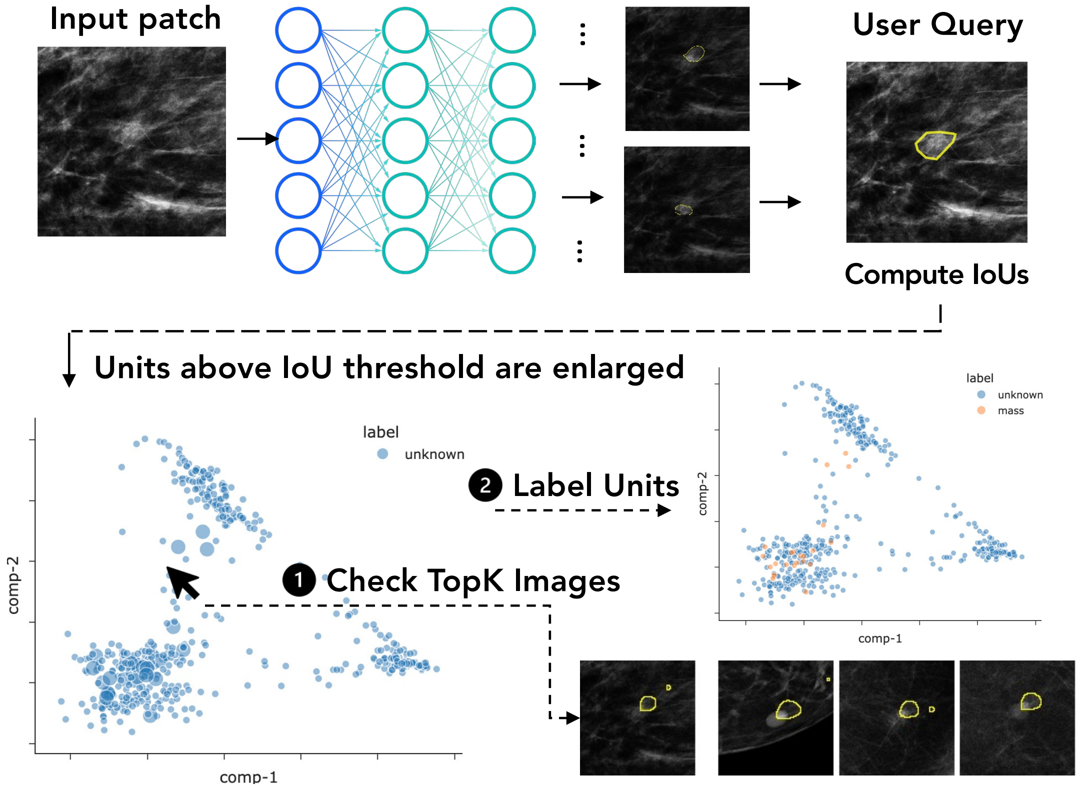

Figure 2: We provide an illustration of our system's main workflow. Each time a patch is feed to the model, its feature maps and corresponding activation maps in the last layer will be retained. When the user initiated a query, the user-defined activation map will be compared with all the retained activation maps based on IoU. Neurons whose activation maps have a high IoU (above the threshold) with one drew by the user will be highlighted in the scatter plot. The user can then click on these neurons to check their top activated images. When the user is convinced, they can annotate these neurons with the concept they detect. These annotations will be used in generating explainability reports in the interface.

overview of our interface is shown in Fig 1, and the design goals behind our system are listed in the following sections.

## 3.1 Goal 1: Query Neuron's Learned Representations

The primary goal of our proposed system is to allow radiologists to answer the question: are there neurons that activate on features that users deem important? A key hypothesis to our query mechanism is that neurons with similar semantic meanings tend to have similar activation maps on a given image. Thus, the problem of finding neurons that align with human reasoning boils down to identifying neurons whose activation maps have a high degree of overlap with regions deemed important by a domain expert for a given image.

### 3.1.1 Activation Map

After identifying top activated images for each neuron, we need to show an activation map for each image to signal what features in the image cause a neuron to activate. We followed the same neuron visualization technique as [3]. Essentially, using the same notation from the previous section, we can compute a global activation quantile $q_{n_i}$ for each $n_i$ on all images in $X$. Then, for each $x_i$, we can generate the activation map by highlighting pixels with an activation value higher than a predetermined quantile value. We used the 99% quantile value to generate activation map of images for each neuron.

### 3.1.2 Query Metric

Given our proposed query mechanism, the metric we used to compare the salient regions defined by the user and the activation maps of each neuron is Intersection over Union (IoU), which is widely used in evaluating segmentation tasks. We use $A$ to denote the user-defined region and $B$ to denote the activation map of a neuron. Then, IoU can be defined as the following:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

## 3.2 Goal 2: Label Meaningful Neurons for Explanation

### 3.2.1 Label Groups of Neurons

Another function that our interface provides is that it allows the user to label neurons. The process is as follows: after the user highlights a region in the image, we compute the IoU of the region and the activation maps of all neurons. If a neuron's activation map has an IoU score above our specified threshold (0.2), we increase the size of the dot of the corresponding neuron in the scatter plot. Then, the user will be able to click on these neurons to see their top activated images. If these neurons indeed detect consistent concepts, the user can label these neurons. If the concept is not yet provided in the dropdown menu, the user has the option to add a new one.

### 3.2.2 Generate Explainability Reports

One of the key benefits of labeling semantically meaningful neurons is the promise of detailed explainability reports. We implemented two ways of generating explainability reports using neurons' labels to provide users with additional insights. To streamline the workflow, our strategy is purely post-hoc and thus is different from concept extraction strategies such as [19] that leverage active learning.

With neurons and associated labels, we can compute the mean max activation values of neurons belonging to the same label. Given an image containing a medical phenomenon $P$, we should expect that the mean activation of neurons labeled as $P$ have higher values than those of neurons with other labels. We show this information with a bar chart, as in section (6) of Fig 1, under "activation value".

The other explainability report depends on the user-defined region $S$ of the incoming image. With this additional input, we can compute the mean IoU between $S$ and activation maps of neurons with the same label. Similarly, given an image containing medical phenomena $P$, we should expect the mean IoU between $S$ and activation maps of neurons labeled $P$ will have higher values than those of neurons with other labels. Similarly, we use a bar chart to encode this information, as in section (6) of Fig 1, under "activation area".

### 3.3 Goal 3: Decode Semantic Connections of Neurons

Finally, to help users understand the relation between neurons and view each neuron's highly activated images, we introduced an embedding for neurons and utilized a dimension reduction technique.

#### 3.3.1 Neuron Embedding

Our method is based on the intuition that, on a diverse set of images, neurons detecting different concepts will have different activation maps on these images. Therefore, given a test set $X$ with $n$ samples, for each neuron, we can compute the maximum activation value on all $n$ samples. Each neuron will be associated with a discriminative vector $\in R^n$. If we have $m$ units, we end up with an embedding of shape $m \times n$. While previous work [12] also explored the idea of neuron embedding, our method presents a more computationally efficient solution without the need for an optimization process.

#### 3.3.2 Dimension Reduction for Visualization

For visualization purposes, we apply PCA with 2 components, generating a matrix of shape $m \times 2$. The visualization is shown in the Label Units section of the interface. Each dot in the scatter plot represents a neuron in the last layer of our model.

### 3.4 Implementation Details

We implemented the system purely in Python. We used PyTorch [13] to develop models and Plotly Dash [7] to build the front-end.

## 4 Interface Workflow

In this section, we will provide an overview of how our interface can potentially fit into a radiologist's workflow (Fig. 1).

In the Mammogram component on the left, the radiologist can browse all the mammograms in the data set. Each time the radiologist switches to a new mammogram, our system's backend will split it into non-overlapping $512 \times 512$ patches and feed them to the trained model for inference. The lesion patches are shown in area (1). If the user wants to have a closer look at a specific patch, they can click on it in (1). Then, the corresponding region will be highlighted in the full mammogram to provide context as in (2).

Furthermore, an enlarged version of the selected patch will be presented in area (3), together with the softmax score. In addition, the activation map of the most activated neuron on the input image will appear in the middle of the Query Units component. With this information, the radiologist will get a sense of what features of the patch led to the model's final decision.

At this stage, the radiologist may be satisfied with the justification provided by the activation map and can simply move on to subsequent mammograms. However, the activation map may not be perfect. As we discussed in Section 4, the user might be interested in knowing whether there are neurons in the model that focus on a region that is not well covered by the activation map of the most activated neuron. In this case, our system provides a solution by allowing the radiologist to define their region of interest in (3). Once the region is defined, our system will follow the methods laid out in section 3.1 to identify neurons whose activation map has a high overlap with what the user defined. The most aligned activation map is then shown on the right of the Query Units component.

When the IoU scores between the user-defined region and all neurons' activation maps are computed, the scatter plot in (4) will be updated. As discussed in 3.3, this scatter plot is a 2D projection of our proposed neuron embedding using PCA. This plot, with each point denoting a specific neuron, provides several important interactions to assist users' visual exploration of the model's learned representations (Fig 2). When the user selects the patch for further investigation, the patch will be fed to the model, and an activation map will generated for each of the neuron in the last convolutional layer. If the user selects a salient region and queries neurons with similar semantic meanings, the system backend will compute the IoU between the region and activation maps of all neurons. The points representing neurons whose activation maps have high IoU values (over the specified threshold) will be enlarged. Then, the user may click on the relevant point to inspect both the neuron's activation map on the input patch and its top activated images. In this way, the user can confirm whether a neuron consistently captures a concept. When the user is convinced, they may endow those neurons with a label for the concept they detect, after which the points will return to their original size but put on a different color. The user can perform the annotation in (5), where they can either select preexisting labels in the drop-down menu or add new labels if needed. As the user can label these units in groups, little annotation effort is required.

While the above workflow allows the user to gain understanding of the model's decision-making process, the neuron annotations can be used to generate explainability reports in (6) as discussed in section 3.2. Notably, the user does not need to label all neurons to generate such explainability reports; instead, they can label neurons gradually as they perform diagnosis to help the system generate better, finer-grained reports over time.

## 5 Conclusion

In this paper, we proposed a novel visual interface to leverage state-of-the-art interpretability techniques to help radiologists screen for breast cancer in an AI-assisted workflow. Instead of letting experts passively interpret saliency maps from a deep learning model, our system allows them to actively query relevant learned representations to understand the model's decisions. With the proposed neuron embedding and visualizations, users can easily inspect the model's learned representations and provide annotations to groups of neurons with minimal effort. We demonstrate that our system can leverage user annotations to provide explainability reports naturally as they perform diagnosis over time. We believe that our system sheds light on how we can leverage the fruitful research in machine learning to maximize its potential in transforming healthcare by human-centric software that considers its applications in actual clinical settings. Meanwhile, we point out that our system currently lacks extensive evaluations. On the user side, it is critical to evaluate whether they indeed feel more transparency about the model's decision with various informative interactions provided by the interface. In terms of system design, an interesting alternative to labeling neurons is to train a finer-grained classification model with the same patches in an active learning setting, which might also produce fine-grained reports with little annotation effort. We intend to pursue these ideas in our future work.

## Acknowledgments

This work used the Extreme Science and Engineering Discovery Environment (XSEDE) [16], which is supported by National Science Foundation grant number ACI-1548562. Specifically, it used the Bridges-2 system, which is supported by NSF award number ACI-1445606, at the Pittsburgh Supercomputing Center (PSC).

## References

[1] R. Agarwal, O. Diaz, X. Lladó, M. H. Yap, and R. Martí. Automatic mass detection in mammograms using deep convolutional neural networks. *Journal of Medical Imaging*, 6(3):031409, 2019.

[2] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

[3] D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou, and A. Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020.

[4] A. Boggust, B. Hoover, A. Satyanarayan, and H. Strobelt. Shared interest: Measuring human-ai alignment to identify recurring patterns in model behavior. In *CHI Conference on Human Factors in Computing Systems*, pp. 1–17, 2022.

[5] F. Cheng, D. Liu, F. Du, Y. Lin, A. Zytek, H. Li, H. Qu, and K. Veeramachaneni. Vbridge: Connecting the dots between features and data to explain healthcare models. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):378–388, 2021.

[6] F. Hohman, H. Park, C. Robinson, and D. H. P. Chau. S ummit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. *IEEE transactions on visualization and computer graphics*, 26(1):1096–1106, 2019.

[7] P. T. Inc. Collaborative data science, 2015.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] B. C. Kwon, M.-J. Choi, J. T. Kim, E. Choi, Y. B. Kim, S. Kwon, J. Sun, and J. Choo. Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):299–309, 2019. doi: 10.1109/TVCG.2018.2865027

[10] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

[11] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. *IEEE transactions on visualization and computer graphics*, 23(1):91–100, 2016.

[12] H. Park, N. Das, R. Duggal, A. P. Wright, O. Shaikh, F. Hohman, and D. H. P. Chau. Neurocartography: Scalable automatic visual summarization of concepts in deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):813–823, 2021.

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[14] D. Ribli, A. Horváth, Z. Unger, P. Pollner, and I. Csabai. Detecting and classifying lesions in mammograms with deep learning. *Scientific reports*, 8(1):1–7, 2018.

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[16] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, et al. Xsede: accelerating scientific discovery. *Computing in science & engineering*, 16(5):62–74, 2014.

[17] J. Wu, B. Zhou, D. Peck, S. Hsieh, V. Dialani, L. Mackey, and G. Patterson. Deepminer: Discovering interpretable representations for mammogram classification and explanation. *arXiv preprint arXiv:1805.12323*, 2018.

[18] N. Wu, J. Phang, J. Park, Y. Shen, Z. Huang, M. Zorin, S. Jastrzebski, T. Févry, J. Katsnelson, E. Kim, et al. Deep neural networks improve radiologists' performance in breast cancer screening. *IEEE transactions on medical imaging*, 39(4):1184–1194, 2019.

[19] Z. Zhao, P. Xu, C. Scheidegger, and L. Ren. Human-in-the-loop extraction of interpretable concepts in deep learning models. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):780–790, 2021.

[20] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.