An Interpretable Semi-Supervised Framework for Patch-Based Classification of Breast Cancer

Radwa El Shawi (radwa.elshawi@ut.ee)
Tartu University
Khatia Kilanava
Tartu University
Sherif Sakr
Tartu University

Research Article

Keywords:

Posted Date: February 18th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1343955/v1

An Interpretable Semi-Supervised Framework for Patch-Based Classification of Breast Cancer

Radwa El Shawi[1,*], **Khatia Kilanava**[1], **and Sherif Sakr**[1]

[1]Institute of Computer Science, Tartu University, Estonia
[*]radwa.elshawi@ut.ee

ABSTRACT

Developing effective methods for Invasive Ductal Carcinoma (IDC) detection remains a challenging problem for breast cancer diagnosis. Recently, there have been notable success of utilizing deep neural networks in various application domains, however, it is well-known that training deep neural networks requires a huge amount of labeled training data in order to achieve high accuracy. Such amounts of manually labeled data are time-consuming and very expensive to obtain especially when domain expertise is required. In order to take advantage of cheap available unlabeled data, we present a novel semi-supervised learning framework for IDC detection using small amounts of labeled training examples. In order to gain trust in the prediction of the proposed framework, we explain the prediction globally. Our proposed framework consists of five main stages including data augmentation, feature selection, dividing co-training data labeling, deep neural network modeling and interpretability of the neural network prediction. The data cohort used in this study contains digitized BCa histopathology slides from 162 women with IDC at the Hospital of the University of Pennsylvania and the Cancer Institute of New Jersey. The experimental evaluation shows that our framework is able to detect IDC with a balanced accuracy of 0.865 and F-measure of 0.773 using only 10% labeled instances from the training dataset while the rest of the training dataset is treated as unlabeled.

1 Introduction

According to the American National Breast Cancer Organization, breast cancer is the second leading cancer type that cause death among women[1]. Breast cancer contributes to around 25% of all types of cancers diagnosed in women. Furthermore, it contributes to 15% of cancer deaths in women[2]. Invasive Ductal Carcinoma (IDC) is the most common type of breast cancer. Among all the patients of breast cancers, around 80% are diagnosed as invasive ductal carcinomas[3]. Breast masses are the most significant findings among different types of breast abnormality. In addition, morphological features of tumor shape play vital roles in the diagnosis of tumor malignancy[4].

Because of its high performance, deep learning has been used extensively in various application domains including medical diagnosis, image recognition and image classification[5–10]. Nowadays, developing high quality deep learning models has become a commodity thanks to the availability of several open-source machine learning frameworks such as `TensorFlow`[11] and `PyTorch`[12]. In general, one of the main challenges for deep learning models is that they require huge amounts of labeled data for fine tuning their architecture and parameters. In practice, these labeled data is expensive and hard to obtain especially in critical domains such as the medical domain. In addition, developing a well performing deep neural network architectures and fine tuning their hyper-parameters is a very challenging and time-consuming task due to the huge search space. In particular, the performance of a deep neural network architecture can significantly vary with different hyper-parameter values. Therefore, Neural Architecture Search (NAS) has become an essential technique for automating the process of finding the best performing neural network architecture along with the best the set of their hyper-parameter values. In practice, NAS has been successfully used to design the model architecture for various image classification and language processing tasks[13–16].

Generative Adversarial Networks (GANs)[17] are a special type of neural networks that consists of two main components: *generator* and *discriminator*. Both generator and discriminator are neural networks in which the generator focuses on generating images while discriminator focuses on discriminating between the synthetic generated images and the original ones. Recently, GANs have received huge attention due to their capability of data generation without explicitly modeling the probability density function. They have shown to be successful in different domains and have achieved the state-of-the-art performance in many image generation tasks, including classification, super-resolution[18] and image-to-image translation[19]. Therefore, GANs have been widely adopted in the medical domain[20] in order to tackle the privacy concerns related to medical image diagnosis in addition to the limited number of positive cases of each pathology. Furthermore, the lack of sufficiently labeled medical images poses another challenge for the adoption of the traditional supervised training techniques and motivates approaches that incorporate unlabeled data that might be available. These approaches include *semi-supervised learning* and *transfer learning*[21]. There have been lots of research that examine the use of generative models in the semi-supervised setting. Salimans, et al.[22]

presented a technique to utilize GANs for solving classification problems with k classes. More specifically, they extended vanilla GAN such that the set of labeled dataset is augmented with the generated samples from the generator (fully connected network). The discriminator is modified in a way to predict $k+1$ classes (the original k classes plus the fake generated class from the generator). Adiwardana et al.[23] utilizes the GANs as in [22] but replaced the fully connected generator network with Deep Convolutional Generative Adversarial Network (DCGAN)[24]. Such change resulted in significant performance boost in supervised image recognition task using small amount of labeled data.

Co-training[25] is one of the most popular techniques for semi-supervised learning in which two classifiers are trained by labelling the unlabeled data for the other classifier and then making the final decision for a particular instance based on the agreement of the two classifiers. One of the main limitations of the co-training is that it requires two complementary models to learn from sufficient independent features. Such independent features may be hard to obtain as there is no clear and effective way to construct independent features from individual images. On other hand, *transfer learning* tries to gain performance from a larger labeled dataset for a related task. It has been empirically observed that features learned from enough training examples by deep learning models can generalize to other related problems. In computer vision domain, it is a common practice to reuse layers from large pre-trained networks such as VGG[26] and Inception[27].

Although deep neural networks have been well performing in various application domains[28], however, in the medical domain, physicians still find it hard to trust the prediction of these black-box models and hence prefer white-box models even if they achieve lower performance compared to black-box models. Since May 2018, machine learning interpretability has received lots of attention especially due to the General Data Protection Regulations (GDPR) '*right to explanation*'[29]. In particular, the GDPR requires all decisions made automatically to be explained as a safeguard for the rights and freedom of EU citizens[30]. One way to define machine learning interpretability is the ability to understand and comprehend the decision made by a machine learning model[31]. In general, machine learning interpretability techniques can be broadly categorized into *global* and *local* techniques[32–34]. Global interpretability techniques focus on explaining the model globally and enable users to comprehend an aspect of the whole model at once[34]. On the other hand, local techniques focus on explaining the prediction of a single instance. One common way for local interpretability is saliency methods that have been used intensively in several gradient-based methods[35–38]. Output of saliency methods show the importance of individual outcomes as an overlay on the input image to be explained. Such approaches suffer from being limited and inconsistent to some extent[39,40]. Another line of research shows that linear classifiers can learn meaningful directions that can be mapped to semantically meaningful word embedding[41] or visual concepts[42].

1.1 Motivation and Contribution

In this paper, we hypothesize that combining a small amount of labeled data with a large amount of unlabelled data is one effective way for combatting the scarcity of labeled data and hence enabling the use of deep learning models effectively. Motivated by the current trend for favoring complex machine learning models at the expense of interpretability, we explain the predictions of our proposed framework globally as a way to provide physicians with complementary insights about the model. In particular, the main contribution of this paper is summarized as follows:

- We developed a semi-supervised deep learning framework for IDC detection using a small number of labelled data combined with large number of unlabeled data. The proposed framework outperforms the state-of-the-art supervised models achieving AUC of 0.865 and F-measure of 0.773.
- We used AutoML techniques to design a deep neural network architecture that outperforms the state-of-the-art performance for IDC detection on digitized BCa histopathology slides obtained from 162 women who have been diagnosed with IDC at Cancer Institute of New Jersey and the hospital of the University of Pennsylvania.
- We interpret the predictions of our semi-supervised model globally by learning meaningful high level concepts and using directional derivatives to quantify the degree to which such concepts are important to the IDC prediction.

Ensuring repeatability is one of the main targets of this work. Therefore, we provide access to the source codes and the detailed results for the experiments in our project repository[43]. The reminder of this paper is organised as follows. Section 2 describes the building blocks for our interpretable semi-supervised deep learning framework for IDC detection. The details of our experimental evaluation is described in Section 3. We discus our results in Section 4 before we conclude the paper in Section 5 .

2 Methods

Figure 1 illustrates the architecture of our framework that consists of five main stages including *data augmentation*, *feature selection*, *dividing co-training data labeling*, *deep neural network* and *interpretability* of the neural network prediction. In the following subsections, we explain the different building blocks of our architecture.

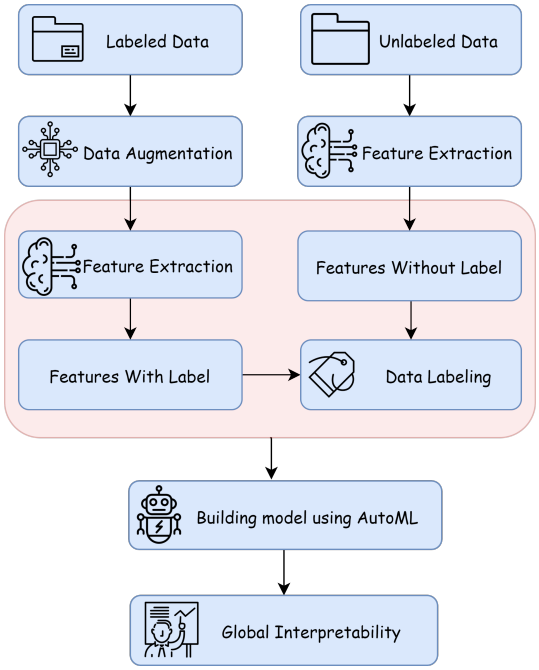

Figure 1. Flowchart of the proposed framework

2.1 Datasets

For IDC detection, we use a data cohort that consists of digitized Bca histopathology slides obtained from 162 women diagnosed with IDC at Cancer Institute of New Jersey and the hospital of the University of Pennsylvania. The cohort was randomly split into three different subsets including 84 training, 29 for validation and 49 for evaluation. The patch-based dataset used in this study obtained from the original cohort consists of 82,883 patches for training, 31,352 patches for validation (i.e. 114,235 patches for full training) and 50,963 instances for testing[44]. From the 114,235 patches of the full training dataset, we randomly select 12k instances (6k instances from each class) as training data and we remove the ground truth of the rest of the full training patches obtaining 102,235 unlabelled patches. For more information about the dataset, we refer the interested readers to[45].

The concepts used by the global interpretability technique are extracted from the nuclear segmentation dataset[46]. The nuclear segmentation dataset consists of 30 whole slide images of digitized tissue samples of several organs obtained from 18 different hospitals. The dataset contains nuclear appearances of seven different organs including breast, liver, kidney, prostate, colon, stomach, and bladder. Since computational requirements for processing WSIs are high, a sub-image of 1000×1000 is cropped from WSIs and more than 21,000 nuclear boundaries are annotated in `Aperio ImageScope`[47]. We extract only 5 images that contains the nuclear appearance of breast cancer which are then segmented into 100 patches in which the concepts are extracted.

2.2 Data Augmentation

Data Augmentation is a widely used technique in various deep learning approaches in the presence of limited amount of labelled training instances to reduce over fitting and improve the accuracy and the robustness of a classifier. In this work, we employ DCGAN to generate synthetic patches. The DCGAN generator consists of a fully connected layer projecting an input of 100-dimensional uniform distribution to four convolution layers with filter sizes of 256, 128, 64 and 32 and kernel size of 5×5. The used activation function is rectified linear unit (Relu) except for the output layer. Batch normalization is performed on all layers except for the last one. We train a DCGAN on 12K patches that are randomly sampled from the available training dataset as a preprocessing step such as the number of instances from each class equals to 6k. The DCGAN is trained separately on each label using multi-channel image patches containing both the acquired image and the ground truth label. The number of synthetic examples generated for each class is 6k patches. Such 12k generated synthetic images are then used to augment the original 12K patches. DCGAN is trained using Stochastic Gradient Descent as an optimizer for 1200 epochs with the *batch size* of 16 and *learning rate* equal to 0.0003. The loss function applied is a binary cross-entropy. Parameters are the same for both discriminator and generator.

2.3 Feature Selection

Complex deep convolution neural network architectures that contain millions of parameters such as Inception, ResNet and VGG have achieved the state-of-the-art performance in different applications[27]. Training these networks requires a huge amount of data, as training with only a small amount of data may lead to overfitting. One approach that is commonly used with such networks with a limited number of training examples is *fine-tuning*, in which only part of the pre-trained neural network is being fitted on the new dataset. In our experiments, we have considered this approach, however, it did not lead to a good performance. Inspired by[48], in this work, we use standard pre-trained VGG-16 network for feature extraction. We follow the same procedure for extracting features from both labeled and unlabeled data. We remove the fully connected layers from VGG network and apply Global Average Pooling operation to the four internal convolutions layers with 128, 256, 512, 512 channels, respectively. Next, we concatenate them to form one vector of length 1408.

2.4 Data Labeling

The original co-training process introduced by Blum and Mitchell[25] starts with two independent attribute subsets, and the unlabelled data is labelled once the two classifiers reach an agreement. It has been shown that independence assumption can be relaxed and the co-training is still powerful under a weaker independence assumption[49]. Since the semi-supervised training is very sensitive on the initial labelled dataset, we developed a co-training procedure that aims to achieve confident labelling. Since the size of the labelled dataset is relatively small, partitioning it into disjoint subsets will result in too small subsets to build a reliable model[50]. First, our procedure starts by shuffling and partitioning the original labelled dataset (12k patches) and the synthetic patches (12k patches) generated by DCGAN into two equal sized subgroups five times, so that the size of each subgroup is half the size of the original labelled dataset. Next, the 10 subgroups of labelled data are used to train 10 gradient-boost classifiers to label the unlabelled data and the common confident data is then appended to the labelled dataset. Then, the common confident instances are chosen by setting a threshold of minimum number of classifiers that should agree on the predicted label of the unlabelled instance. Such threshold is set to 5. The process has been repeated until no more confident data appeared. To control the quality of each of the classifiers and the accuracy of the new labelled instances in each iteration, we calculate the predication accuracy of each classifier using a subset of the originally labelled set and 20% of the newly labelled confident instances. If the accuracy of the classifier decreases compared to the previous iteration, then an expanded labelled dataset (originally labelled plus newly labelled) is re-shuffled and re-divided until the prediction accuracy of the classifier improves. Note that our approach only adds the common confident instances to the labelled dataset, not all unlabelled data are added eventually to the labelled dataset and hence the instances that are not added are considered noisy data and are not considered in building the final model. Note that the synthetic patches which are generated by DCGAN are only used in the labeling process and are not considered in training the final network. So, the final dataset from this stage which is used to train our final model consists of the originally labelled instances (12k instances) in addition to the confident instances labelled by our labeling technique.

2.5 Deep Convolution Neural Network Model

In general, different neural network architectures can have significantly different performance results. In practical, most of the currently employed network architectures are manually developed by human experts which is a very time-consuming and error-prone process. In this work, we use `Neural Network Intelligence(NNI)`[51], an open-source toolkit by Microsoft for automated machine learning, to find the best network architecture along with the best set of hyper-parameters. NNI accelerates and simplifies the whole search space using built-in super-parameter selection algorithm. The architecture of the neural network found by the NNI is illustrated in Figure 2. The network consists of three convolution layers of sizes 32×32, 16×16, and 8×8, respectively. Batch-normalization (BN) is applied before each convolution layer. For each convolution layer, 64 kernel was applied for the previous feature maps. Each of the convolution layers is followed by a max-pooling layer. The first two max-pooling layers are passed thorough the `Relu` non-linear activation function. The last pooling layer is of size 4×4 and followed by global average pooling layer which is followed by a dropout layer. Following the global average pooling layer is two fully connected layers passed by Relu non-linear activation function. The last fully connected layer generates the final membership degree to each class. The parameters used in the network are as follows: *learning rate* = 0.01, *number of epoch* = 80, *batch size* = 32, *momentum* = 0.9 and *decay* = $1e^{-5}$. For finding the best network architecture and hyper-parameters, the NNI has been assigned a budget of 48 hours.

2.6 Global Interpretability

In this work, we use a global interpretability technique which is based on Concept Activation Vector (CAV) and can provide an interpretation of the internal state of neural networks using real human-friendly concepts[52,53]. Such concepts are tied to real-world data that represents interesting and relevant concepts. Testing the concept activation vector uses directional derivatives to quantify the degree of importance of a particular concept to the model prediction. Informally, the key idea of the technique relies on the evidence that feedforward neural network works by gradually disentangling particular concept

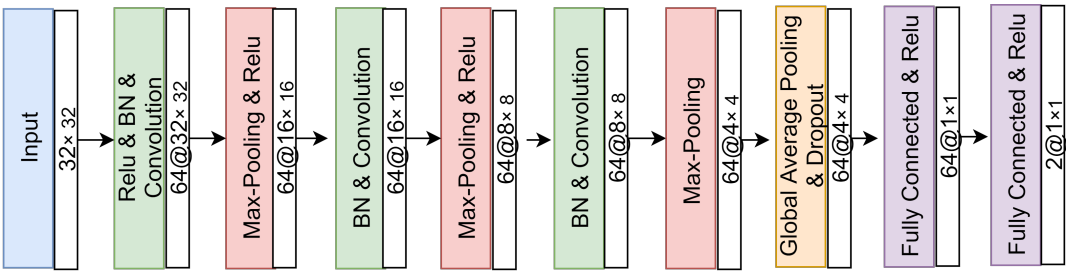

Figure 2. The structure of the CNN used in this study

across layers[54]. Such learnt concept is not necessary acquired by a single neuron but more generally in linear combinations of neurons[55,56]. Hence, the space of neurons activations in neural network layers can have meaningful global linear structure. Such structure can be uncovered by training a model that can map the representation in a single network layer to meaningful user-defined concepts. We adopted the technique of Graziani et al.[53] which is summarized as follows. In this approach, the first step is to define the set of interesting concepts which are relevant to IDC prediction. Nuclear morphometric and appearance features such as average size, and pleomorphism can help in assessing cancer grades and predicting treatment[57,58]. In this study, we use concepts by referring to the Nottingham Histologic Grading system (NHG)[59]. Such concepts are extracted from the nuclear segmentation dataset[46] that quantify the impact of variations in nuclei size, area and texture. The second step is to compute the Pearson product-moment correlation coefficient ρ between each of the concepts and the network prediction for each input patch. If the correlation coefficient for a particular concept is low, then this concept is not relevant for the model prediction. A high correlation value for a particular concept refers to whether such concept is positively or negatively affecting the prediction. We repeat the following steps for each concept of interest. Let $\phi_l(x)$ be the activations for input patch x at layer l. The third step is to find a unit vector v_c^l in the space of activations of layer l in the network that represents the increasing direction for a particular concept of interest. Such vector is computed as the least squares linear regression fit $\{\phi_l(x_i), c_i\}$ on the nuclear segmentation dataset, where c_i is the concept measure for a particular concept C for input image x_i. The fourth step is to calculate the sensitivity to changes in each input x_i along the direction of the increasing values of the concept measures, at neural network activation layer l. The sensitivity score $S_{C,l,i}$ is calculated as the directional derivative along the direction of v_c^l.

$$S_{C,l,i} = \frac{\partial f(x_i)}{\phi_l(x_i)} \cdot v_c^l \tag{1}$$

where $f(x_i)$ is the network prediction for instance x_i. The sensitivity sign is interpreted as the direction of the change whereas the magnitude of the sensitivity reflect the rate of change. In this work, we use the bidirectional relevance score Br[53] that is defined as the ratio between the coefficient of determination of the least squares regression, R^2, and the coefficient of variation $\frac{\sigma}{\mu}$ of the sensitivity scores calculated in the previous steps over all the test instances in the testing dataset.

$$Br = R^2 \times \frac{\sigma}{\mu} \tag{2}$$

Where σ and μ are the standard deviation and the mean of the sensitivity scores respectively. The Br scores are calculated for all of the concepts of interest and scaled over the range of $[-1, 1]$.

3 Experimental Evaluation

3.1 Experimental setup

We conducted our experiments on two hardware environments: a CPU environment and a GPU environment. The CPU environment runs on CentOS release 7.5.1804 with 64 core Intel Xeon Processor (Skylake, IBRS) @ 2.00GHz;240 GB DIMM memory; and 240 GB SSD data storage. The GPU experiments are performed on a single machine running on Debian GNU/Linux 9 (stretch) with an 8 core Intel(R) Xeon(R) CPU @ 2.00GHz; NVIDIA Tesla P4;36 GB DIMM memory; and 300 GB SSD data storage.

3.2 Results

3.2.1 Data augmentation

Examples of real and synthetic patches generated by DCGAN is illustrated in Figure 3. The data labeling performance of using only labelled instances (12K instances) and varying amounts of additional synthetic data is shown in Figure 4. The

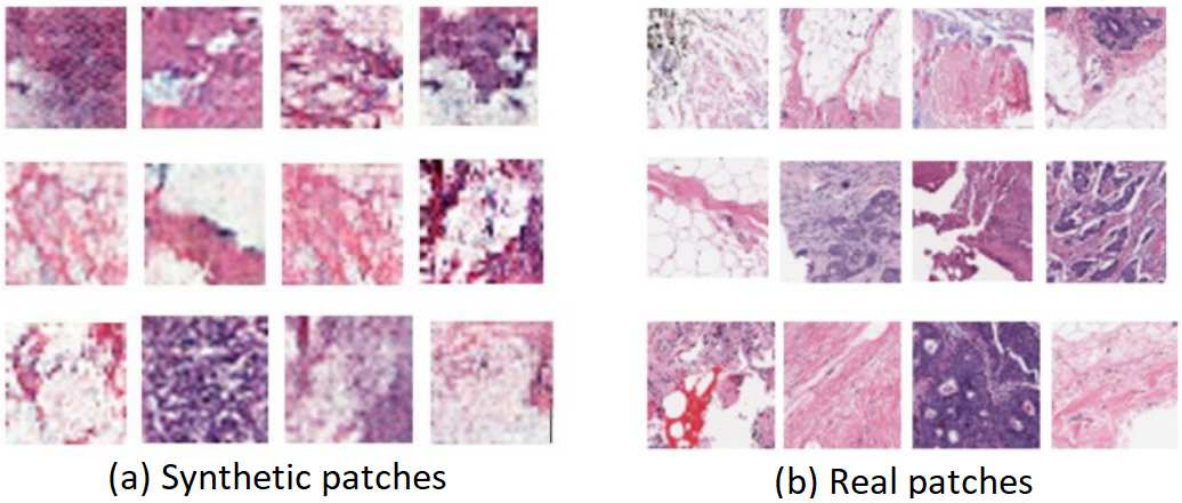

Figure 3. Examples of real and GAN generated synthetic patches

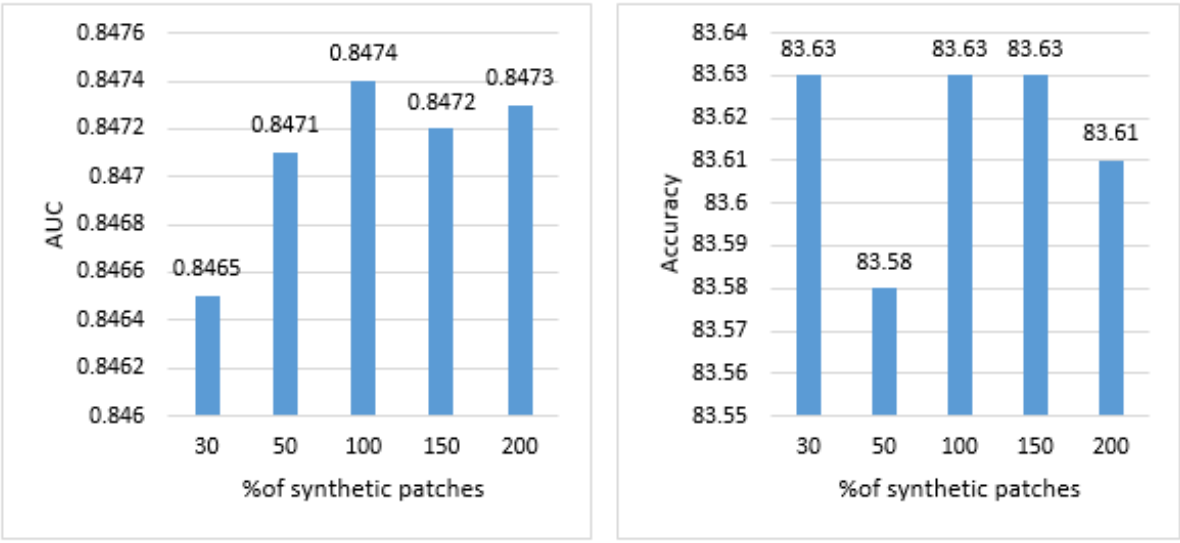

Figure 4. The data labeling performance with varying amounts of additional synthetic data

		Dimension			
		50	100	200	250
PCA	AUC	0.8069	0.8072	0.8047	0.8041
	Accuracy	80.87	80.96	79.78	80.58
LDA	AUC	0.6412	0.6412	0.6412	0.6412
	Accuracy	46.51	46.51	46.51	46.51

Table 1. The performance of newly labelled data using LDA and PCA feature extraction techniques

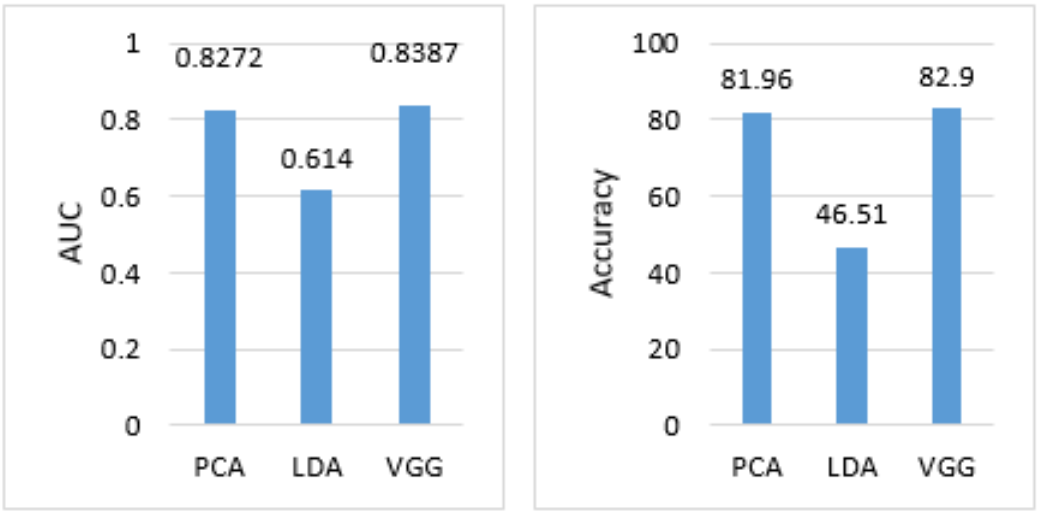

Figure 5. Comparison of the data labeling performance of different feature extraction techniques

performance is measured by calculating the accuracy and the AUC of the newly labelled data. Note that the labelling model remains unchanged when examining the effect of different amounts of synthetic patches generated by the DCGAN. This provides a fair platform upon which to observe the effects of GAN augmentation by ensuring that any changes in performance are due to the additional synthetic patches, and not due to changes in the labelling model itself. It is notable that increasing the amount of synthetic generated patches to 100% increases the labeling performance to accuracy = 83.63% and AUC = 0.8474 compared to accuracy = 81.51% and AUC = 0.815 without using synthetic patches. The labeling performance slightly drops when increasing the percentage of synthetics generated patches to 150% and 200% as shown in Figure 4.

3.2.2 Feature extraction

We first evaluate the influence of three different feature extraction techniques including principle component analysis (PCA), Linear discriminant analysis (LDA) and VGG. The performance of these techniques is measured by calculating the labeling performance of the newly labelled data without including GAN synthetic patches. Table 1 shows the data labeling performance (accuracy and AUC) of PCA and LDA tested at reduced dimensions of 50, 100, 200 and 250. As shown in Table 1, LDA achieves the same performance across all tested dimensions and that is mainly because the number of feature projections outputs by LDA is at most equals to the number of classes - 1. PCA has shown to achieve the best performance at 100 components with AUC of 0.8072 and accuracy of 80.96%. Figure 5 shows the labelling performance of the newly labelled data using PCA, LDA, and VGG-16. The labeling performance using VGG-16 features outperforms other techniques achieving AUC of 0.815 and Accuracy = 81.51%.

3.2.3 Deep Convolution Neural Network Model

To examine the performance of the neural network architecture and the hyper-parameters found by NNI, we run three different experiments. The first one is just feeding the whole training and validation datasets (114,235 labelled patches) and test the performance of the network on the testing dataset (50,964 patches) averaged over 100 runs. Quantitatively, we present the F-score and the balanced accuracy for our method compared to Curz et al.[45] and Janowczuk et al.[60] in Table 2. To provide a fair comparison, we use the same training, validation and testing splits used by the baselines[45,60]. For each metric (column), we highlighted the highest value in bold font. The performance of the network found by NNI achieves performance of AUC=0.8696 and F-meaure=0.7923 and outperforms the customized deep neural network obtained by Cruz-Roa et al.[45], and the customized

	F-measure	AUC
Curz et al.[45]	0.7180	0.8423
Janowczuk et al.[60]	0.7648	0.8468
Our approach	**0.7923**	**0.8696**

Table 2. Performance comparison between our network and other approaches

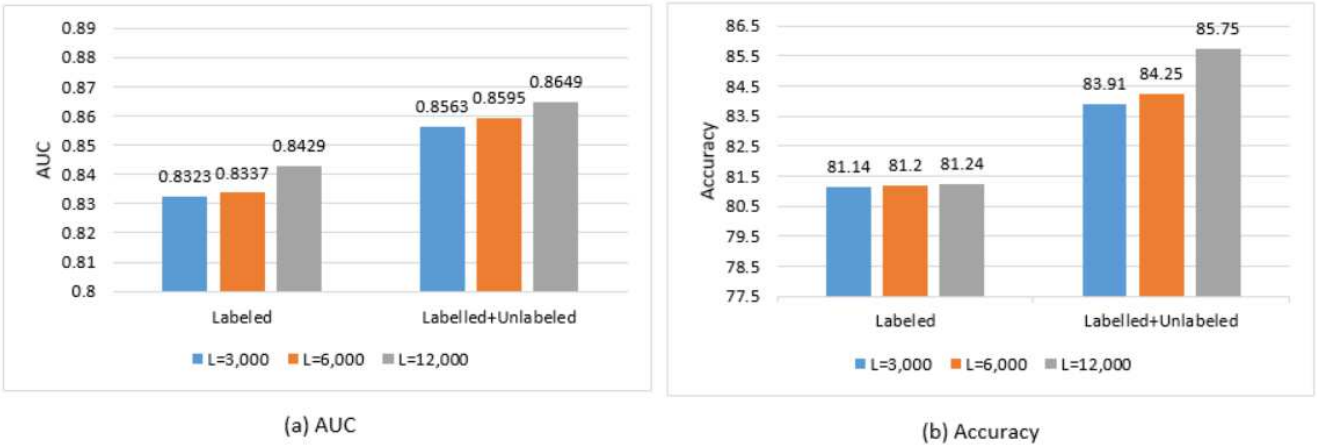

Figure 6. Neural network performance using labelled data only and mixed data(labelled and unlabelled) using different number of labelled data.

AlexNet architecture[10] used by Janowczyk et al. [60].

In the second experiment, we examine the effect of adding unlabelled data to the training dataset, we compare the performance of the neural network model found by NNI using a different amount of originally labelled data (originally labelled 12k patches) and using labelled data combined with unlabelled data (labelled by our data labeling model) in Figure 6. The performance of the labeling gradually increases by increasing the number of labelled instances. It is notable that the performance of the network using unlabelled data combined with labelled data outperforms the performance of the network using the same amount of labelled data only. The performance of the networks using 12,000 labelled instances only achieves AUC of 0.8429, accuracy = 81.24% and F-score = 72.79% while the network achieves AUC of 0.8649, accuracy = 85.75 and F-score = 77.29 using the same amount of labelled data combined with unlabelled data.

In the third experiment, we test the proposed semi-supervised approach against three baseline classification networks for fully supervised learning using only the labelled data (12k patches) and one baseline generative network for semi-supervised learning. The three chosen fully-supervised networks are the network architecture found by NNI (used in the proposed approach), customized AlexNet[10], a shallow VGG network[61] modified to be fully convolutional and to also include batch-normalization[62]. The semi-supervised DCGAN baseline involves a generative model trained along with a discriminator using both labeled (12k patches) and unlabeled data. The discriminator is used to compute the loss of the classification, in addition to the adversarial loss[23]. The three fully-supervised baselines trained on 12k labeled patches only. The semi-supervised base-line and our proposed approach are trained on the same 12k labeled patches (the same used for the supervised baselines) in addition to the set of unlabeled data (102,235 patches). In order to provide a fair comparison, the testing dataset used for testing the models performance is the same across all baselines networks and our proposed approach (50,963 patches).

Table 3 shows the performance of different supervised and semi-supervised networks. For each row metric, we highlighted the highest performance in bold font and underlined the lowest performance. The results show that our proposed approach outperforms all the supervised and semi-supervised approaches in both metrics achieving AUC of 0.8649 and F-measure of 0.7729 while the supervised customised AlexNet approach achieves the lowest performance (AUC of 0.8121 and F-measure of 0.6932). It is notable from the results the effect of using the unlabeled data in semi-supervised approaches to improve the model performance.

	Supervised			Semi-supervised	
	VGG[61]	**Customised AlexNet**[10]	**NNI**	**Semi-supervised DCGAN**[23]	**Our approach**
F-measure	0.701	0.6932	0.7297	0.7421	**0.7729**
AUC	0.8235	0.8121	0.8429	0.8435	**0.8649**

Table 3. Performance comparison between different supervised and semi-supervised approaches

	Correlation	ASM	Contrast
Correlation Coefficient	**-0.38**	0.32	**0.44**
P-Value	≤ 0.001	≤ 0.001	≤ 0.001

Table 4. Pearson correlation between concepts and network prediction

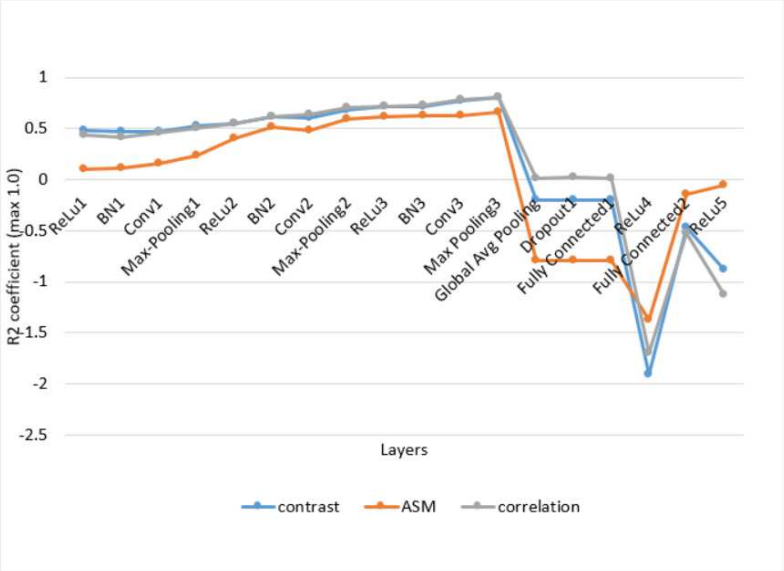

Figure 7. Determination coefficient of linear regression at all layers found by NNI.

3.2.4 Interpretability

We express the NHG criteria for nuclei pleomorphism as the average statistics of nuclei morphology and texture features. We compute some concepts from the nuclear segmentation dataset which are average area, perimeter, Euler coefficient, axis length and eccentricity. In addition, we calculate three of Haralick's texture features[63] including Angular Second Moment (ASM), contrast and correlation[63]. Nuclei pleomorphism features do not correlate with network prediction. Table 4, shows the correlation between the texture concepts measures and the prediction of the network found by NNI. Concept contrast has the largest correlation coefficient of 0.44.

In order to identify the network layer in which the concepts are learnt, we measure the performance of each the linear regression model at each layer. The determination coefficient of the regression expresses the percentage of variation of the regression capture. The determination coefficient is calculated with 10-fold validation averaged across 50 runs as shown in Figure 7. All of the three concepts are learnt at the last Max-pooling layer.

We compute the sensitivity for each test patch in the testing dataset. The global relevance is tested with *Br* score as shown in Figure 8. Concepts contrast and correlation have *Br* scores of 1 and -1 respectively. These values are inline with the values of Pearson correlation in Table 4. The *Br* score of the concept shows that constant is relevant to classification which comes inline with NHG grading system that identifies hyperchromatism as a main cause of nuclear atypia. The *Br* score signs show that concept correlation negatively contributed to the prediction of IDC while concept contrast positively contributed to the prediction of IDC. In other words, if the correlation concept increases significantly, a patch may change from high risk of IDC to low risk of IDC.

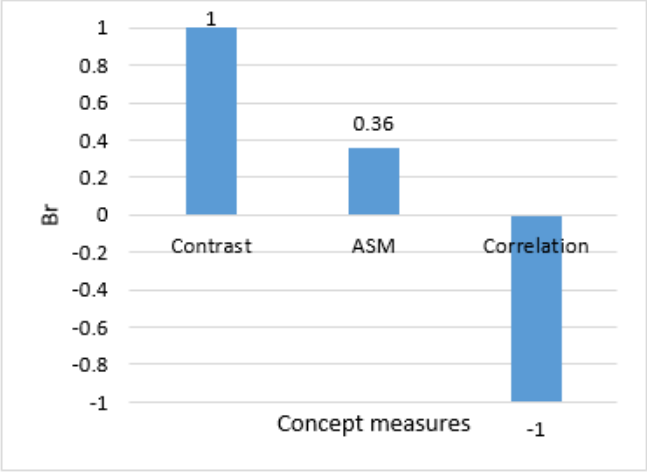

Figure 8. Br Score for texture concepts

4 Discussion

In this study, we presented an interpretable semi-supervised deep convolution neural network model for IDC detection that uses a large amount of unlabelled data to improve the model performance. Availability of abundant labelled data for supervised learning is a challenging problem, especially in medical domain. Most of current techniques for abnormality detection requires manually annotated images for training. In order to best utilize the unlabelled instances, we developed a labeling technique to label the unlabelled data and then append it to the small set of labelled instances to train a neural network model for IDC detection. Such labeling technique uses only small amount of labelled data in addition to the synthetic data generated by DCGAN. Using such labeling technique, our neural network model achieves performance of AUC = 0.865 and accuracy = 85.75% compared to AUC = 0.843 and accuracy = 81.28% when using only the same amount of labelled (12k labelled instances). In order to achieve a high labeling performance, we tested different feature extraction techniques including PCA, LDA and VGG-16. VGG-16 features achieve the best labeling performance of AUC = 0.815. Besides, we examine the labeling performance by testing the effect of increasing the size of the training dataset by generating synthetic patches using DCGAN. Results show that increasing the percentage of synthetic patches to 100% increases the labeling performance to AUC = 0.847 and accuracy = 83.63% compared to AUC = 0.815 and accuracy = 81.51% without using synthetic patches. It is concluded that the amount of unlabelled instances has significant impact on the model performance as shown in Figure 6. Splitting the initial labelled dataset is an essential part of our labeling technique and hence if the size of the initially labelled dataset is extremely small (3k patches), then the ability to incorporate significant information from the unlabelled data would be very limited as shown in Figure 6. That is because extremely small initially labelled instances results in smaller splits that are more sensitive to noise.

Since we use an AutoML framework to find the best network architecture and hyper-parameters on our dataset, we examine this architecture on the whole originally labelled training and validation datset and compared to the state-of-the-art performance by Cruz-Roa et al.[45] and Janowczyk et al. [10]. The results show that the network architecture found by NNI (AUC = 0.87 and F-score = 79%) outperforms the best baseline (AUC = 0.85 and F-score = 76%). It is clear that the unlabelled data can not replace the labelled data and using unlabelled data is just supplement. Labelled data contains precise and accurate information obtained from radiologist compared to automatically labelled data, which has been proven to improve the performance.

Complex machine learning models such as neural network models are hard to understand their behaviour and hence may pose a problem on their adoption in critical domains due to trust reasons. Our approach provides a global interpretability technique for explaining the behaviour of the neural network model based on meaningful concepts extracted from nuclear segmentation dataset. The main strength of this approach is the flexibility of extracting relevant features from a small set of images.

5 Conclusion

We presented a novel interpretable semi-supervised learning algorithm for IDC detection that uses a small amount of labelled data and relatively large amount of unlabelled data to train a neural network model. The results of our experimental evaluation shows that the performance of the neural network model improves when combining the unlabelled data to the originally labelled

instances. Our approach significantly tackles the challenge of deep learning models on requiring a large amount of data for their training, something which is not always easily to obtain especially in the medical domain. Our labeling model utilizes synthetic images generated by DCGAN to improve the labeling performance. Our proposed framework allows users to utilize unlabelled data in deep learning training dataset and increase the overall performance. In order to build trust in the developed framework, we explain the neural network model globally using texture concepts extracted from nuclear segmentation dataset.

6 Data availability

For IDC detection, the batch-based dataset used in the current study is available at http://www.andrewjanowczyk.com/use-case-6-invasive-ductal-carcinoma-idc-segmentation/. The concepts used by the global interpretability technique are extracted from the nuclear segmentation dataset available at https://nucleisegmentationbenchmark.weebly.com/dataset.html. The implementation of the method, acknowledgement files for the IDC detection used is provided at https://github.com/DataSystemsGroupUT/DC-classification.

References

1. https://www.nationalbreastcancer.org/breast-cancer-facts. [Online; accessed 21-December-2021].
2. Ferlay, J. *et al.* Cancer incidence and mortality worldwide: sources, methods and major patterns in globocan 2012. *Int. journal cancer* **136**, E359–E386 (2015).
3. https://www.breastcancer.org/symptoms/types/idc. [Online; accessed 21-December-2021].
4. Tang, J., Rangayyan, R. M., Xu, J., El Naqa, I. & Yang, Y. Computer-aided detection and diagnosis of breast cancer with mammography: recent advances. *IEEE Transactions on Inf. Technol. Biomed.* **13**, 236–251 (2009).
5. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062* (2014).
6. Girshick, R., Donahue, J., Darrell, T. & Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587 (2014).
7. Long, J., Shelhamer, E. & Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440 (2015).
8. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).
9. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708 (2017).
10. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105 (2012).
11. Abadi, M. *et al.* Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 265–283 (2016).
12. Paszke, A., Gross, S., Chintala, S. & Chanan, G. Pytorch (2017).
13. Zoph, B. & Le, Q. V. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578* (2016).
14. Zoph, B., Vasudevan, V., Shlens, J. & Le, Q. V. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8697–8710 (2018).
15. Cai, H., Chen, T., Zhang, W., Yu, Y. & Wang, J. Efficient architecture search by network transformation. In *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).
16. Liu, C. *et al.* Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 19–34 (2018).
17. Goodfellow, I. *et al.* Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680 (2014).
18. Ledig, C. *et al.* Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4681–4690 (2017).
19. Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232 (2017).

20. Yi, X., Walia, E. & Babyn, P. Generative adversarial network in medical imaging: A review. *arXiv preprint arXiv:1809.07294* (2018).

21. Quellec, G., Cazuguel, G., Cochener, B. & Lamard, M. Multiple-instance learning for medical image and video analysis. *IEEE reviews biomedical engineering* **10**, 213–234 (2017).

22. Salimans, T. *et al.* Improved techniques for training gans. In *Advances in neural information processing systems*, 2234–2242 (2016).

23. Adiwardana, D., Matsukawa, A. & Whang, J. Using generative models for semi-supervised learning. In *Medical image computing and computer-assisted intervention–MICCAI*, vol. 2016, 106–14 (2016).

24. Radford, A., Metz, L. & Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

25. Blum, A. & Mitchell, T. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 92–100 (ACM, 1998).

26. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

27. Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence* (2017).

28. Domhan, T., Springenberg, J. T. & Hutter, F. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Twenty-Fourth International Joint Conference on Artificial Intelligence* (2015).

29. https://ec.europa.eu/commission. [Online; accessed 21-December-2021].

30. Goodman, B. & Flaxman, S. European union regulations on algorithmic decision-making and a "right to explanation". *AI Mag.* **38**, 50–57 (2017).

31. Lim, B. Y., Dey, A. K. & Avrahami, D. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *SIGCHI* (2009).

32. Edwards, L. & Veale, M. Slave to the algorithm: Why a right to an explanation is probably not the remedy you are looking for. *Duke L. & Tech. Rev.* **16**, 18 (2017).

33. Guidotti, R. *et al.* A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* **51**, 93 (2018).

34. Lipton, Z. C. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490* (2016).

35. Montavon, G., Samek, W. & Müller, K.-R. Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.* **73**, 1–15 (2018).

36. Selvaraju, R. R. *et al.* Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 618–626 (2017).

37. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013).

38. Zeiler, M. D. & Fergus, R. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833 (Springer, 2014).

39. Adebayo, J., Gilmer, J., Goodfellow, I. & Kim, B. Local explanation methods for deep neural networks lack sensitivity to parameter values. *arXiv preprint arXiv:1810.03307* (2018).

40. Kindermans, P.-J. *et al.* The (un) reliability of saliency methods. *arXiv preprint arXiv:1711.00867* (2017).

41. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119 (2013).

42. Kim, B., Gilmer, J., Wattenberg, M. & Viégas, F. Tcav: Relative concept importance testing with linear concept activation vectors. (2018).

43. https://github.com/DataSystemsGroupUT/DC-classification. [Online; accessed 1-January-2022].

44. http://www.andrewjanowczyk.com/use-case-6-invasive-ductal-carcinoma-idc-segmentation/. [Online; accessed 1-January-2022].

45. Cruz-Roa, A. *et al.* Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks. In *Medical Imaging 2014: Digital Pathology*, vol. 9041, 904103 (International Society for Optics and Photonics, 2014).

46. Kumar, N. *et al.* A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE transactions on medical imaging* **36**, 1550–1560 (2017).

47. https://www.leicabiosystems.com/digital-pathology/manage/aperio-imagescope/. [Online; accessed 1-January-2022].

48. Rakhlin, A., Shvets, A., Iglovikov, V. & Kalinin, A. A. Deep convolutional neural networks for breast cancer histology image analysis. In *International Conference Image Analysis and Recognition*, 737–744 (Springer, 2018).

49. Abney, S. Bootstrapping. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (2002).

50. Sun, W. *et al.* Using undiagnosed data to enhance computerized breast cancer analysis with a three stage data labeling method. In *Medical Imaging 2014: Computer-Aided Diagnosis*, vol. 9035, 90350T (International Society for Optics and Photonics, 2014).

51. https://github.com/Microsoft/nni. [Online; accessed 1-January-2022].

52. Kim, B. *et al.* Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279* (2017).

53. Graziani, M., Andrearczyk, V. & Müller, H. Regression concept vectors for bidirectional explanations in histopathology. In *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, 124–132 (Springer, 2018).

54. Alain, G. & Bengio, Y. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644* (2016).

55. Raghu, M., Gilmer, J., Yosinski, J. & Sohl-Dickstein, J. Svcca: Singular vector canonical correlation analysis for deep understanding and improvement. *stat* **1050**, 19 (2017).

56. Szegedy, C. *et al.* Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

57. Beck, A. H. *et al.* Systematic analysis of breast cancer morphology uncovers stromal features associated with survival. *Sci. translational medicine* **3**, 108ra113–108ra113 (2011).

58. Chang, H. *et al.* Invariant delineation of nuclear architecture in glioblastoma multiforme for clinical and molecular association. *IEEE transactions on medical imaging* **32**, 670–682 (2012).

59. Elston, C. W. & Ellis, I. O. Pathological prognostic factors in breast cancer. i. the value of histological grade in breast cancer: experience from a large study with long-term follow-up. cw elston & io ellis. histopathology 1991; 19; 403–410: Author commentary. *Histopathology* **41**, 151–151 (2002).

60. Janowczyk, A. & Madabhushi, A. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *J. pathology informatics* **7** (2016).

61. Cho, H., Lim, S., Choi, G. & Min, H. Neural stain-style transfer learning using gan for histopathological images. *arXiv preprint arXiv:1710.08543* (2017).

62. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

63. Haralick, R. M., Shanmugam, K. *et al.* Textural features for image classification. *IEEE Transactions on systems, man, cybernetics* 610–621 (1973).

Acknowledgements

This work is funded by the European Regional Development Funds via the Mobilitas Plus programme (grant MOBTT75).

Author contributions statement

Conceptualization, S.S and R.E; methodology, S.S and R.E; software, K.K.; validation, S.S., K.K. and R.E.; formal analysis, K.K. and R.E.; K.K. conducted the experiments; investigation, R.E.; resources, K.K.; data curation, R.E and S.S.; writing—original draftprepa- ration, R.E. and S.S; writing—review and editing, R.E. and S.S; visualization, K.K.; supervision, S.S.