On the dynamics and feasibility of transferred inference for diagnosis of invasive ductal carcinoma: A perspective

Harshvardhan GM (✉ harrshvardhan@gmail.com)

Kalinga Institute of Industrial Technology <https://orcid.org/0000-0003-3529-2931>

Aanchal Sahu

Kalinga Institute of Industrial Technology

Mahendra Kumar Gourisaria

Kalinga Institute of Industrial Technology

Vijander Singh

Manipal University - Jaipur Campus

Bunil Kumar Balabantaray

National Institute of Technology Meghalaya

Joel J.P.C. Rodrigues

Federal University of Piaui: Universidade Federal do Piaui

Research Article

Keywords: CNN, breast cancer, transfer learning, invasive ductal carcinoma

Posted Date: October 12th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-927704/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License. [Read Full License](#)

Abstract

It is generally noticed that increasing the number of convolutional layers in generic image classification procedures proves to be detrimental to model performance in terms of validation accuracy and loss. Apart from vanilla CNNs, we have state-of-the-art (SOTA) architectures such as ResNet50 (and its variants) which show that through the use of skip-connections, higher performance metrics are attainable through deeper architectures. However, most evaluative metrics converge on a log scale as we go deeper with diminishing gradient of the metrics' curves. Given these two contrasting speculations, in this paper, we implement various vanilla and SOTA CNNs for the diagnosis of one of the most common forms of breast cancer - invasive ductal carcinoma (IDC) - to examine and understand the feasibility of implementation of SOTA CNNs through transferred weights when juxtaposed with vanilla CNNs (and LeNet-5) of varying configurations in terms of their performance metrics and other parameters. In this paper, we solve the dual-objective of studying behavioural aspects of avant-garde CNN models (more specifically, VGG16, VGG19, ResNet50, ResNet50V2, MobileNetV2, and DenseNet121) and proper diagnosis of IDC through intermediate neural activations to critically evaluate and theorize the performance of different models. We notice that among all the models, only VGG16, VGG19, LeNet-5 and a selected vanilla CNN through an optimization procedure were the ones to attain the best metrics, shared amongst them.

1. Introduction

Deep Convolutional Neural Networks (CNNs) have interesting properties pertaining to the scalability of their feature capturing abilities. Generally, the depth of the deep CNN is decided by the number of features, and both are directly proportional to one another. With the natural tendency of capturing features of all different levels, i.e., low, medium, and high [1], CNNs have been put to great use for various applications [2–6], inclusive of medical applications [2, 7, 8–12]. One significant and relevant dataset to our discussion in this paper, the ImageNet [13], is a dataset which is used in the annually hosted ImageNet Very Large Scale Visual Recognition Challenge (ILSVRC) for both, object detection (correct localization of all objects present in an image) and object recognition (accurate identification of existence of objects in an image). The ImageNet is considered a standard benchmark for all SOTA models of object detection [14–17] and recognition. In this paper, we employ the techniques of transfer learning [18–21] for transferred inference of IDC. There are many categorizations of transfer learning as given by [20] such as instance-based, mapping-based, network-based and adversarial-based. Our implementation of transferred weights is a network-based approach where SOTA networks are pre-trained on ImageNet over a plethora of images. We re-use these pre-trained architectures barring the last few layers (and thus fine-tune the transferred model based upon our application) and compare those with fully, in-house trained vanilla CNNs to see how transferred learning affects model performance in the specific case of the detection of IDC for prognosis. Figure 1 describes how we use ImageNet pretrained SOTA models for transfer learning of feature extraction facilitative weights.

Generally, when deep networks converge, their accuracy, loss and other performance metrics also saturate. However, as observed [22–24], the level of this asymptotic saturation degrades when architectures' layers are increased. This phenomenon is not observed in ResNet [22] due to the utilization of skip-connections between layers. We investigate this phenomenon further in our implementations of vanilla CNNs in Sect. 5 and see which parameters affect this degradation most and through which adjustments in specific parameters it can be minimized. Many researchers work on such comparative studies on datasets such as the CIFAR10 [25], MNIST [26], etc. with the problem that working on these datasets only helps us understand model performance and not how they might perform on real-world application-based datasets. Keeping that in mind, we perform our experiments on clinical medical data to achieve a two-fold objective of understanding the dynamics and feasibility of transfer learning for several CNN models along with the creation of reliable models for the prediction of IDC.

Breast cancer (BCa) encompasses several diseases and involves the uncontrolled division of cells in the breast tissue. Around 80% of the cases of BCa are identified as IDC [27] and is also referred to as infiltrating ductal carcinoma since the terms *invasive* and *infiltrating* refer to the cancerous cells breaking out of their origin ducts or glands to invade new spaces or new breast tissue. Less common types of IDC are medullary ductal carcinoma (MDCa), mucinous ductal carcinoma (MDCb), tabular ductal carcinoma (TDC) and papillary carcinoma (PC). MDCa comprises only 3–5% of all BCa cases and is visible through X-ray imaging or mammograms. MDCb, also called colloid carcinoma, is the condition where cancerous cells secrete mucous (the inner surface lining of organs of the digestive tract, liver, lungs, etc. is made up of mucous) which surrounds the BCa cells. The mucin associates with these cells and eventually they form a tumour. However, the prognosis of pure MDCb is better than other forms of IDC.

TDC comprises 2% of the IDC cases and has an excellent prognosis as compared to other cases of IDC. The tumour formed by TDC appear tube-like when studied under a microscope. PC accounts for 0.5% of the total IDC cases [28] the cells in the PC condition appear finger-like (papillary, made of papules) projections and is more prominently observed in postmenopausal women over the age of 60. The cases of MDCa, MDCb, TDC and PC are viewed as histological classification of the more general IDC - only a quarter of all cases of IDC are histologically categorized based on the BCa cell shape, size and arrangement. IDC is also categorized into four major molecular subtypes: luminal A (HER2-/HR+), luminal B (HER2+/HR+), HR2-enriched (HR/HER2+), and basal-like (HR-/HER2-). Clinical approximations for molecular subtyping or categorizing types of BCa are often not crisp, a major reason being that there is noticed an overlap between different molecular subtypes [27, 29].

We enlist our contribution in this paper as follows -

- We implement avant-garde CNNs namely VGG16, VGG19, ResNet50, ResNet50V2, MobileNetV2, and DenseNet121. Along with these, we implement various traditional CNNs and LeNet-5 [26] and vary many different parameters to gather results and choose one best architecture

among them.

- We critically analyse the performance of all the models and study their nature of predictions in the context of the influence of transfer learning for inference, and additionally, the influence of tune-able parameters in traditional CNNs on their performance metrics.
- Through this process, display the important parameterizations to use along with the extent of feasibility of transfer learning while creating a model for effective diagnosis of IDC through classification.

The rest of the paper is organized as follows – Sect. 2 (Related Work) describes the related work which is divided into three different techniques used majorly for the detection of BCa, Sect. 3 (Methodology and Materials) we describe the nature of the data and the techniques used in this paper such as CNNs (and their architectures), transfer learning, etc., Sect. 4 (Evaluation Strategy) in which we define briefly all the metrics used for the evaluation of performance of all the models and also how we choose the best traditional CNN model for further considerations, Sect. 5 (Results) contains all the results in terms of the performance metrics, neural activations of intermediate chosen traditional CNN models, etc. In Sect. 6 (Discussion) we analyse performance of each model with each metric and understand the effect of transferred weights for inference, and finally in Sect. 7 (Concluding Remarks and Future Directions) we conclude the paper's findings and lay out the basis of work that can be done in this domain in the future.

2. Related Work

Machine learning and deep learning approaches have been vastly employed to solve various medical problems [30–37]. More specific use-cases are gene selection and classification and diagnosis of cancer [38, 39], or prediction of COVID-19 [40, 41]. There exist other optimization techniques such as social spider-inspired optimization for the detection of BCa [42]. Machine learning and deep learning methods are used even in non-medical fields such as credit card fraud detection [43], recommender systems [44], social distance monitoring in pandemics [45], cardiovascular risk prediction [46], etc. We divide this section into three broadly employed approaches for the detection of BCa, namely WSI segmentation-based, Region of Interest (ROI) based, and unsupervised deep learning-based approaches.

2.1 WSI-based segmentation approaches

Mostly, deep learning-based computer vision methods applied for the detection of BCa/IDC (also referred to as digital pathology) involve whole slide images (WSI) [47–49]. Cruz-Roa et al. (2014) [47] segmented the WSIs into various mini-regions, similar to what we do in this paper, and compared the performance of deep learning workflows with SOTA handcrafted feature methods namely Gray Histogram (GH) [50] Fuzzy Color Histogram (FCH) [51], HSV Color Histogram (HSVCH) [52], RGB (red, green, blue) Histogram (RGBH) [51], Haralick features [53], Graph-based features [53], MPEG7 Edge Histogram (M7Edge) [54], Local Binary Partition Histogram [55] and JPEG Coefficient Histogram [52]. They employed a very small, simplistic CNN architecture with two convolutional layers and a final fully connected dense layer. It was noticed that the CNN performed best based on balanced accuracy (BAC) and F1-score (71.8% and 84.2%) which was an improvement of 6% and 4% respectively over the next best handcrafted feature. Wang et al. 2016 used patch-based processing of the WSIs for detection of metastatic BCa through SOTA deep CNNs namely GoogLeNet [56], AlexNet [57], VGG16 [58], and FaceNet [59] and it was found that GoogLeNet and VGG16 attained maximum patch-based performance. Post classification, a tumour existence probability heatmap was generated which was used for computations of slide-based classification and lesion-based detection probabilities. Two interesting aspects of the work in [49] were the enrichment of the training set through inclusion of extra lymph node image data so as to help the models not misclassify such regions as BCa, and that to reduce computational costs, the WSIs were segmented by a threshold method that involved conversion of the image channels from RGB to HSV and application of Otsu's algorithm [60], and combination of the H and S mask images to get the final masks.

Janowczyk and Madabhushi (2016) [48] made use of deep learning approaches for seven different digital pathology tasks; one of these tasks was the correct segmentation of IDC from WSIs of breast tissue. The WSIs were divided into many different mini-patches (similar to our approach), but were resized to 32×32 and rotated for oversampling and tending to the problem of class imbalance. Using AlexNet with dropout and downsizing the patches, their model achieved an F1-score of 75.7% with a BAC of 84.23%, outperforming results obtained by [47] who considered patches of size 50×50. However, it was realised in [48] that using dropout did not improve results on the test set.

Exploring the depth-wise separable convolution methodology in CNNs, Alghodhaifi et al. (2019) [61] compared the performance of a standard CNN against a depth-wise separable CNN for the diagnosis of IDC through 50×50 patches extracted from a total of 162 WSIs. Depth-wise separable CNNs work by applying convolution to each separate channel (in this case, there are only three channels: red, green and blue) and then combine the resulting output channels through pointwise convolution. It was noticed in [61] that the standard CNN performed marginally better in terms of specificity, F1-score, and accuracy, however the precision and sensitivity scores for both models were nearly same. Interestingly, they found that application of Gaussian noise to both the models had contrasting effects: the accuracy of the depth-wise separable CNN diminished by more than half (85.9% vs. 33.4%) while the standard CNN still held similar accuracy (87.1% vs. 77.4%). Using network-based transfer learning principles, Celik et al. (2020) [62] used two pre-trained SOTA CNNs that are included in the implementations of this paper namely DenseNet161 [63] and ResNet50 for the detection of IDC over patches of WSIs. They employed one-cycle policy [64] in which a tiny learning rate is chosen initially for training which is incremented after every mini-batch. This increment occurs until a proper learning rate along with the exploding loss value are reckoned. The main drawback of [62] is that they do not mention on which images or dataset the models were pre-trained on. This can be very crucial in the intelligibility of the model's

outputs and behaviour. Moreover, we notice that in the literature there is seldom any work on comparison-based analysis of the performance of numerous SOTA CNNs which make use of transfer learning in the field of detection of BCa.

2.2 Region of Interest (ROI)-based approaches

Subclinical diagnosis of BCa on whole images of full-field digital mammography (FDDM) through the use of deep learning techniques is a challenging task since the region of interest (ROI, where the BCa can be detected) is very small in comparison to the dimensions of the original FDDM image. To curb this issue, Shen et al. (2019) [65] pretrained a fully convolutional classifier on local patch-based WSIs embedded with annotations to incorporate ROI information. This pretrained classifier's weights were leveraged to initialize training of the same classifier on whole FDDM images to improve detection of BCa without the need of ROI annotations. They employed two different classifier SOTA CNN network designs which are also used in our paper namely VGG16 [58] and ResNet [22]. Dunder et al. (2011) [66] distinguished Usual Ductal Hyperplasia (UDH) from atypical ductal hyperplasia (ADH) and ductal carcinoma in situ (DCIS) over WSIs (manually identified ROIs) through multiple instance learning by making use of the large margin principle [67, 68].

Tackling the issues of automatic localization of ROIs for BCa from WSIs and classification of five different diagnostic varieties of ductal proliferations, Gecer et al. (2018) [69] used Fully Convolutional Networks (FCN) [70] for semantic segmentation of the WSIs to obtain ROIs from four different levels of magnifications. They showed that many redundant features are eliminated as the features are extracted from lower to higher magnifications. A deeper FCN was used for the classification of WSIs from five different diagnostic ductal proliferations namely non-proliferative changes, proliferative changes, IDC, ADH and DCIS. The morale behind usage of a deeper CNN for this task was to extract more features per WSI owing to visually similar proliferations. The performance of their model on the quin-classification task was not satisfactory (achieving an accuracy of 39.04%), so, in their last contribution they showed the fusion of the ROI and classifier outputs for WSI-level diagnosis helped improving accuracy. In more traditional mannerisms of extraction of features from digital mammography imaging, Yengce Tasdemir et al. (2019) [71] detected abnormal areas in a mammography by features extracted by Histogram of Oriented Gradients (HOG) [72] and Haralick features [73] to detect ROIs for presence of BCa. The mammography was segmented into smaller ROIs of size 73×68 and then converted into a two dimensional Discrete Wavelet Transform (2D-DWT) for multi-resolution decomposition of the ROIs [74]. On this 2D-DWT, Haralick and HOG features were extracted which was followed by a feature selection stage before classification by random forest, support vector machine (SVM) and AdaBoost.

2.3 Unsupervised deep learning-based approaches

More recently, researchers have looked into unsupervised methods of deep learning for the detection of BCa and components of histopathology tissue [75–79]. [75] made use of FusionNet [80], a form of a Convolutional Autoencoder (CAE), that made use of very long skip connections between the encoder and decoder subnets to generate images - similar to those done by generative models in machine learning [81]. As done predominantly elsewhere, they used patches of WSIs for detection of IDC by only training the encoder network of the FusionNet and running a softmax classifier to obtain binary outputs. Autoencoders are used for pre-training deep learning models but are also very useful for mapping high dimensional data into a latent space, thus acting as a powerful feature extractor. This feature extraction property is exploited by CAEs for image retrieval tasks. When we consider tabular data for BCa risk prediction, Belciug et al. (2010) [82] compared the performance of supervised and unsupervised deep learning approaches namely Multilayer Perceptron (MLP), Radial Basis Function (RBF) and Probabilistic Neural Networks (PNN) as supervised networks and Kohonen's self-organizing map (SOM) [83] as the unsupervised network. The SOM performed equally well as its supervised counterparts and outperformed PNN by a 5% difference of testing accuracy. It was noticed that the p-value between the average portions of correct classifications (through the z-test) was higher than the significant value (where $p > 0.05$) for RBF and SOM, indicating no significant statistical difference in their positive classifications. The p-value was lower than the significant value ($p < 0.05$) for SOM vs. MLP and SOM vs. PNN meaning that significant statistical difference did exist for their positive classifications. This concluded that unsupervised deep learning methods performed similar to their supervised counterparts in neural networks. Self-supervised approaches have also been employed as done by Xu et al. (2016) [84] through the use of stacked sparse autoencoders (SSAE) for automatic detection of nuclei in breast histopathology. The SSAE framework outperformed other techniques such as Expectation Maximization (EM) [85], Blue Ratio (BR) thresholding [86], and Colour Deconvolution (CD) [87] in both qualitative and quantitative terms.

3. Materials And Methodology

In this section, we describe the data used for our experiments and the preprocessing techniques applied on them to bring them into suitable form. Further, we briefly explain the architecture of the models used in our experiments and finally we present a formal explanation of network-based transfer learning employed in our approach.

3.1 Dataset

We make use of 162 WSIs collected by [48] and [47] scanned at 40x magnification. For our experiments, as mentioned earlier, instead of taking the WSIs, we use a sliding window technique and extract 277,524 patches having dimensions 50×50 characterized by a binary attribute to determine the existence of IDC. The binary class distribution is given by Table 1. We use a 9:1 train-test split ratio.

Table 1
IDC presence distribution in extracted patch specimens

IDC presence (class 1)	IDC absence (class 0)
78.786×10^3	198.738×10^3

After an initial screening of these patches, we noticed that a presence of IDC was attributed by darker shades of pink, i.e. tending to be purple. To understand this better, we plot a flattened colour histogram over three channels for normal and IDC patches as shown by Fig. 2. The x-axis contains the bin count (we take 256 bins) and y-axis depicts the number of pixels. Since each component (RGB, for red, green, and blue) represented has intensities varying $\in [0, 255]$, suitably, we take 256 bins to account for each intensity count. It is noticed from Fig. 2 that for normal cases, the R and B component are divergent; further apart, as opposed to IDC cases where R and B components almost converge to overlap. The shift is observed more in the R component which, in IDC, is pushed back to the native region of the B component. This is because R component has lower intensities for IDC WSI regions as opposed to normal WSI regions.

3.2 Image Augmentation

Deep learning models require abundance of data to train properly. Usually, image datasets of such scales are too space-intensive to maintain or transport for different applications. Image augmentation is a technique applied to the base dataset for the diversification of input images in terms of count and quality [88, 89]. This is achieved through various ways such as whitening transforms [90], rotations, shifts, shearing, zooming, rescaling, etc. The augmentation parameters we used in our implementations are given by Table 2. Rescaling is applied by multiplying data points with the given argument on the images after all other transformations are applied. Shear range represents the shear intensity which is the shear angle in counter-clockwise direction in degrees. Zoom range is the upper limit for a range used to sample random values lying within to zoom the image. Horizontal flip randomly flips the images horizontally. Only rescaling is applied to the testing set.

Table 2: Image augmentation parameter settings

	Rescale	Shear range	Zoom range	Horizontal Flip
Training set	$1/_{255}$	0.2	0.2	✓
Testing set	$1/_{255}$	-	-	✗

3.3 CNN and Activations

In this subsection we give a brief overview of the working of a CNN and how we use different CNNs in the implementation. CNNs, introduced by [26], have proven to be the backbone of modern deep computer vision technologies such as face detection [91, 92], action recognition [93, 94], scene labelling [95], etc. The convolution operation, most popularly used in signal processing, between two functions $p(t)$ and $q(t)$ can be defined as $p(t) * q(t) \triangleq \int_{-\infty}^{\infty} p(\tau) q(t - \tau) d\tau$.

This operation is performed on pixel values by various convolutional layers to extract features through the use of 2D matrices known as kernels or filters. This convolution step preserves the spatial relationships and representations in the image. The number of parameters are reduced approximately by 75% in the next step of max-pooling which only extracts the maximum counts of convolved values in a fixed sliding window. After a series of combinations of convolutional and max-pooling layers, finally, a flattened vector is obtained which is fed into an Artificial Neural Network (ANN) acting as a feed-forward network which learns to output the correct classes. This step is usually called the full connection (FC). Figure 3 pictorially depicts the methodology used for detecting IDC in patches of a WSI.

As seen from Fig. 3, we extract the intermediate activations of different convolutional and max-pooling layers to better understand the features detected by subsequent layers for the interpretation of how inputs are transformed. Due to existence of three channels, we visualize these activations channel-wise - independently - and plot the inputs decomposed into the different learned filters of the layers. Further, generation of class activation maps can be done through Global Average Pooling (GAP) [96] which obtains the spatial average of the feature map of all units of the convolutional layer at the end whose weighted sum is taken for the final activation maps (see Appendix D). Application of class activations is not feasible in this setup as we automatically classify 50×50 regions of a WSI which results in seeming like a low resolution class activation map. Further, we discuss the transfer learning methodology used in our implementations.

3.4 Weight Transfer

Deep learning frameworks require a lot of data to train effectively. Hence, fetching sufficient data can sometimes be a tedious prospect. This problem is largely solved in the literature and in real world applications through the use of readily available weights to initialize or kick-start the training of any CNN. The features learned by successive layers in a CNN for any task may be generalizable for use in a different task. We use nomenclatures used by [97] and [98].

Let there be a domain D given by $D = \{X, P(X)\}$ where X is the feature space and $P(X)$ is the marginal probability distribution, having $X = \{x_1, x_2, \dots, x_n\} \in X$. The data space of any task T is represented by X with $P(X)$ denoting the marginal probability distribution of a particular

learning sample. Task T is given by $T = \{Y, f(x)\}$ where Y is the label space containing the targets and $f(x)$ being the target probability function which may be written as a conditional probability $f(x) = P(y|x)$. Over the course of training, parameters of $f(x)$ are adjusted to optimize and minimize distances between outputs of predictive function $f(x)$ and $P(X)$. The predictive function $f(x)$ comprises tuples (x_i, y_i) where $x_i \in X, y_i \in Y$. Finally, before being able to define transfer learning, we take two instances a and s .

Transfer learning may be defined as follows – if we are given a learning task T_a having domain D_a , we can use a source domain D_s with a well defined T_s . Through latent knowledge transfer from D_s and T_s , an attempt at improving the predictive function $f_a(\cdot)$ is made (which is a component of learning task T_a) where $D_a \neq D_s$ or $T_a \neq T_s$. If we denote the sizes of domains D_a and D_s by n_a and n_s respectively, then, we may say that usually $n_s \gg n_a$. For the learning task of training all successive convolutional layers in any SOTA CNN used in our implementation (except LeNet-5), we make use of a source domain of ImageNet by using network-based transfer learning to use weights of pre-trained models. We do this by freezing the parameter learning process of the convolutional part of the networks and learning only parameters of the fully connected (FC) layers. This process has been shown in Fig. 1.

4. Evaluation Strategy

In this section we enlist descriptions of all the terminologies associated with our evaluation strategy for all the models. For a binary classification task, we have cases of true positivity (TP), true negativity (TN), false positivity (FP), and false negativity (FN). TP indicates a correctly classified positive, i.e. in our case, a correctly classified case of IDC. Similarly, TN indicates a correctly classified negative, FP a falsely classified positive and FN a falsely classified negative. Based on these four terms, we define precision, sensitivity (or recall), specificity, F1-score and balanced accuracy. These metrics are widely used in the literature for classification tasks. Precision P is the ratio of TP to all the labels predicted as positive and is given by (1),

$$P = \frac{TP}{(TP+FP)} \quad (1)$$

P helps answering to what extent the model correctly classifies positive cases. Further, sensitivity S_n (or recall) is the ratio of TP to the number of positives in reality, given by (2),

$$S_n = \frac{TP}{(TP+FN)} \quad (2)$$

S_n gives the measure of how many correct predictions of positive cases were made out of total positive cases. Specificity S_p can be seen as an opposite of S_n because it gives the measure of correctly labelled negatives (TN) out of the total population of the real distribution of negatives. Mathematically,

$$S_p = \frac{TN}{(TN+FP)} \quad (3)$$

F1-score F takes a combination of P and S_n which presents the harmonic mean between these two variables. It is given by (4) as,

$$F = \frac{2S_nP}{(S_n+P)} \quad (4)$$

In this paper, we use two different types of accuracy metrics: regular accuracy (RAC) and balanced accuracy (BAC). RAC will be used when we describe the test set validation accuracy of different models. However, once a confusion matrix of classifications is generated for all the models, we will calculate a BAC that will better represent model performance. BAC is required when there is a high class imbalance and can be mathematically expressed for binary classification tasks as,

$$BAC = \frac{\left[\frac{TP}{(TP+FP)} + \frac{TN}{(TN+FP)} \right]}{2} \quad (5)$$

While, RAC can be mathematically expressed by (6) as,

$$RAC = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (6)$$

Finally, we use the Matthews' Correlation Coefficient (MCC) [99] for in-depth analysis of each model. MCC (also known as the phi coefficient) lies in the range $\in [-1, 1]$ where -1 and 1 respectively mean total disagreement between observation and prediction, and perfect prediction. A value of 0 indicates that the model is as efficient as a random classifier. Most importantly, it is a balanced metric, meaning that class imbalance does not perturb the ease of its interpretation. Mathematically,

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TN+FN)(TN+FP)(TP+FN)}} \quad (7)$$

A binary cross entropy loss (BCE) is calculated for the training of all the models. This BCE loss is taken into consideration when we calculate an optimization function (that we describe in this section later) and also by the neural net itself for the adjustment of weights and biases. BCE is

expressed mathematically as,

$$H(v) = -\frac{1}{n} \sum_{i=1}^n y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i)) \quad (8)$$

In (8), the distribution of data labels is given by y making $p(y_i)$ the model's prediction on data label i . True data distribution is represented by v with n being the total number of samples. Given (8), we are now able to tract an optimization function used to select the best trained-from-scratch traditional CNN. In our experiments, we train fifteen CNNs by changing various parameters such as number of layers, neurons, regularizations, etc. that we shall describe in Sect. 5 more. As mentioned earlier, to determine how feasible transfer learning is in our application, we must compare it to some baselines, and hence we use vanilla CNNs for this comparison. Selection of a 'best' CNN can be tricky due to three metrics that all play a pivotal role in describing performance, namely, validation accuracy (or RAC), validation BCE loss, and training time. Here, validation refers to the calculation of metrics on the validation or test set (we use validation set and test set interchangeably in this paper, although their meanings in detail are not exactly same). Ideally, it is desirable to maximize RAC, minimize BCE loss and minimize training time, as we do in (9). Given a classifier model $M_{\theta_i; \phi_i}$ with parameters θ_i and implementation information ϕ_i we denote a set $C = \{M_{\theta_1; \phi_1}, M_{\theta_2; \phi_2}, \dots, M_{\theta_i; \phi_i}, \dots, M_{\theta_{15}; \phi_{15}}\}$ that contains all the traditional CNN models used for experimentation. The implementation information ϕ_i can be thought of as an m -tuple where m is the number of hyper-parameters (and other architectural information) that we vary over all our experiments. The cardinality and elements of this m -tuple will be clearly shown in Sect. 5. Now, mathematically, the optimization function $O(M_{\theta_i; \phi_i})$ is given by (9), if we denote $\max(x)$ and $\min(x)$ by $\psi(x)$ and $\omega(x)$ respectively,

$$O(M_{\theta_i; \phi_i}) = \frac{\psi(\alpha(M_{\theta_i; \phi_i}))}{\omega(\tau(M_{\theta_i; \phi_i})) + \omega(H_{M_{\theta_i; \phi_i}}(v))}, \forall M_{\theta_i; \phi_i} \in C \quad (9)$$

In (9), $\alpha(\cdot)$ denotes the validation RAC, $\tau(\cdot)$ denotes the training time, and $H_{M_{\theta_i; \phi_i}}(v)$ denotes the BCE loss for given model $M_{\theta_i; \phi_i}$. The objective is to maximize $O(M_{\theta_i; \phi_i})$ given by $\operatorname{argmax}_{M_{\theta_i; \phi_i}} (O(M_{\theta_i; \phi_i}))$. This procedure yields us a single model $M_{\theta_i; \phi_i}$ that we regard as the 'best' vanilla CNN to be compared with other SOTA implementations. Hence, maximizing $O(M_{\theta_i; \phi_i})$ transforms (9) as,

$$O(M_{\theta_i; \phi_i}) = \operatorname{argmax}_{M_{\theta_i; \phi_i}} \left(\frac{\psi(\alpha(M_{\theta_i; \phi_i}))}{\omega(\tau(M_{\theta_i; \phi_i})) + \omega(H_{M_{\theta_i; \phi_i}}(v))} \right), \forall M_{\theta_i; \phi_i} \in C \quad (10)$$

It is important to note that we had to normalize values of the function $\tau(M_{\theta_i; \phi_i})$ because of the huge difference in the scale of the values yielded by $\tau(M_{\theta_i; \phi_i})$ as compared to $\alpha(M_{\theta_i; \phi_i})$ and $H_{M_{\theta_i; \phi_i}}(v)$ – the latter two being restricted in the range $\in [0, 1]$. Typically, $\tau(M_{\theta_i; \phi_i})$ yields values of units of seconds (s) which, due to hardware-related limitations, can never lie in $[0, 1]$. Thus, we apply a normalized $\tau(M_{\theta_i; \phi_i})$ in our final optimization function, this function being denoted as $N(\tau(M_{\theta_i; \phi_i}))$,

$$O(M_{\theta_i; \phi_i}) = \operatorname{argmax}_{M_{\theta_i; \phi_i}} \left(\frac{\psi(\alpha(M_{\theta_i; \phi_i}))}{\omega(N(\tau(M_{\theta_i; \phi_i}))) + \omega(H_{M_{\theta_i; \phi_i}}(v))} \right), \forall M_{\theta_i; \phi_i} \in C \quad (11)$$

The normalization function $N(x)$ is defined by (12) as,

$$N(X) = \frac{x_i - \omega(X)}{\psi(X) - \omega(X)} \quad \forall x_i \in X = \{x_1, \dots, x_n\} \quad (12)$$

Using (8) and (12) in (11), we get,

$$\begin{aligned} \mathcal{O}(M_{\theta_i; \phi_i}) = & \quad (13) \\ \underset{M_{\theta_i; \phi_i}}{\operatorname{argmax}} & \left(\frac{\psi(\alpha(M_{\theta_i; \phi_i}))}{\omega\left(\frac{\tau(M_{\theta_i; \phi_i}) - \omega(\tau(C))}{\psi(\tau(C)) - \omega(\tau(C))}\right)} + \omega\left(-\frac{1}{n} \sum_{i=1}^n y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i))\right) \right) \\ & \forall M_{\theta_i; \phi_i} \in C \end{aligned}$$

We remark that the range of $\mathcal{O}(M_{\theta_i; \phi_i})$ varies between $[0, \infty)$.

5. Results

In this section, we look at the implementation details of all the fifteen vanilla and SOTA + LeNet-5 CNNs and results achieved by them¹. As discussed in Sect. 4, we also calculate important metrics such as precision P , sensitivity S_{tr} , specificity S_{pr} , F1-score F , RAC and BAC. Moreover, through selection of the best vanilla CNN (further denoted as C_{best}) using optimization function $\mathcal{O}(M_{\theta_i; \phi_i})$ we compare performance of transferred inference of SOTA models against C_{best} +LeNet-5 and also inspect the intermediate neural activations of the latter two models. Further, we remark that only select SOTA models (out of a pool of all models shown in Table 3) were implementable due to the limitation of certain models having a fixed minimum input dimension size. Since our patches were of dimensions 50×50 , all SOTA models (as we had originally planned) could not be implemented.

Table 3
Original pool of SOTA models to be implemented for IDC detection. Models having suitable minimum input dimensions (denoted by ticks) were used for experiments.

SOTA CNN	Minimum input size $\leq (50 \times 50)$
Xception [100]	□
VGG16 [58]	□
VGG19 [58]	□
ResNet50 [22]	□
ResNet50V2 [101]	□
InceptionV3 [102]	□
MobileNetV2 [103]	□
DenseNet121 [63]	□
NASNetMobile [104]	□
NASNetLarge [104]	□

The fifteen traditional CNN models $M_{\theta_i; \phi_i}$ are characterized by implementation information θ_i which is given by an m -tuple, $\theta_i = \{CL, AL, L1, L2, BN, DO, FD, KS, PS, S, LVBCel, MVA, TT\}$ where $m = 13$. Descriptions of each of these elements in the 13-tuple are given in Table 4. The number of neurons in each AL is taken to be 128. Tuples such as (128, 64, 32) in FD represent the number of filters being used in each successive CL; this description follows for stride S as well. However, tuples in KS represent the square root of the sizes of kernels used in each successive CL. For example, a KS of (9, 3) represents size of first kernel taken as 9×9 in the first CL and 3×3 in the second. This description follows for pooling layer sizes PS as well. LVBCel and MVA are the result of minimum loss and maximum accuracy encountered at any epoch. Let there be vectors $\mathbf{A} = \{A_1, A_2, A_3, \dots, A_{15}\}$ and $\mathbf{L} = \{L_1, L_2, L_3, \dots, L_{15}\}$ which store the RAC or accuracy A_j and BCE loss L_j for each epoch $j \in [1, 15]$. Then, LVBCel is defined as $\omega(\mathbf{L})$ and MVA as $\psi(\mathbf{A})$. We noticed that 15 epochs for these vanilla CNNs were enough for proper convergence. Lastly, TT has the SI unit of seconds (s) and is represented by $\tau(M_{\theta_i; \phi_i})$ in (13). We present results of vanilla CNNs in Table 5, in which we find C_{best} through the maximum value of $\mathcal{O}(M_{\theta_i; \phi_i})$. Descriptions of L1, L2, BN and DO (which are the regularizations) are given in Appendix A, Appendix B, and Appendix C. In Table 5, a tick mark represents the use of the corresponding regularization, and a cross represents that the regularization was not used. These regularizations have been used randomly (in regard to their position in the network) for all the models.

According to Table 5 and Fig. 4, CNN 11 can be regarded as C_{best} since it attains the highest value for optimization function $\mathcal{O}$, hence

$C_{best} \leftarrow M_{\theta_{11}; \phi_{11}}$. The architecture of C_{best} is visually represented in Fig. 4. Additionally, for our experiments, we remark that a batch size of 32

images was used and the activation function for each FC layer was taken to be rectified linear unit (ReLU) [105], except the last layer, which had

sigmoid activation for vanilla CNN experiments and softmax for SOTA models. To have a better idea of number of parameters that each architecture learns as opposed to other models, we specify the number of total number of parameters along with the count of trainable and non-trainable parameters for each model. The number of parameters is calculated at each CL and added up. If a CL has n filters of size $p \times q$ with bias b and the number of channels c , then the number of trainable parameters at this CL can be calculated as $(n \times p \times q \times c) + b$. In the case of FC layers, the adjustable weight matrices along with the biases are taken to be its parameters. We remark that for all models except LeNet-5 and C_{best} through network-based transfer learning, only the FC layers' parameters are learned with all the CL parameters frozen. Through this dual nature of experimentation it becomes possible to learn and understand the feasibility of transfer learning in our application as a comparison can be made between models that had transferred weights against the models that did not. Table 5 describes the composition of the FC layers of all the models.

Table 4
Description of abbreviations used in Table 6.

Abbreviation	Description
CL	Convolutional layers or number of convolutional layers.
AL	Artificial Neural Network layers or the number of AL layers.
L1	L1 regularization
L2	L2 regularization
BN	Batch normalization
DO	Dropout
FD	Features detected or the number of filters
KS	Kernel sizes
PS	Pooling sizes
S	Stride of the CL
LVBCEL	Least validation binary cross entropy loss
MVA	Maximum validation accuracy
TT	Time taken

Table 5
Implementation information θ_i (13-tuple) for each vanilla CNN $M_{\theta_i; \phi_i}$ and calculation of optimization function O tractable by (13) to find C_{best}

SNo.	CL	AL	Regularizations				FD	KS	PS	S	LVBCEL	MVA	TT	$O(M_{\theta_i; \phi_i})$
			L1	L2	BN	DO								
1	2	4	0	0	0	0	(64,32)	(9,3)	(4,2)	(1,1)	0.3749	0.8363	860	0.608262
2	2	4	0	0	0	0	(64,32)	(9,3)	(4,2)	(2,1)	0.4649	0.8044	491	0.7765751
3	2	4	0	0	0	0	(64,32)	(9,3)	(4,2)	(2,1)	0.4209	0.8121	606	0.7215132
4	2	4	0	0	0	0	(64,32)	(9,3)	(4,2)	(2,1)	0.6893	0.7161	562	0.7761760
5	2	2	0	0	0	0	(64,32)	(9,3)	(4,2)	(2,2)	0.3867	0.8377	694	0.7017813
6	2	3	0	0	0	0	(64,32)	(9,3)	(4,2)	(2,1)	0.414	0.8158	696	0.6668834
7	3	4	0	0	0	0	(128,64,32)	(2,1,1)	(2,1,1)	(2,2,1)	0.5284	0.7743	568	0.6512934
8	3	5	0	0	0	0	(128,64,32)	(3,3,1)	(2,1,1)	(2,1,1)	0.3817	0.827	768	0.6487683
9	4	4	0	0	0	0	(128,64,32,16)	(2,2,1,1)	(2,1,1,1)	(2,1,1,1)	0.5974	0.7161	696	0.5090629
10	4	5	0	0	0	0	(128,64,32,16)	(3,2,1,1)	(2,1,1,1)	(2,1,1,1)	0.5967	0.7161	694	0.5101602
11	4	5	0	0	0	0	(128,64,32,16)	(3,2,1,1)	(2,1,1,1)	(2,2,1,1)	0.3835	0.8319	471	0.8933879
12	4	5	0	0	0	0	(64,32,32,16)	(3,2,1,1)	(2,2,2,2)	(2,1,1,1)	0.3718	0.8338	768	0.6592225
13	5	5	0	0	0	0	(128,64,32,32,16)	(2,2,2,2,1)	(2,2,2,1,1)	(2,1,1,1,1)	0.3829	0.8306	712	0.6859888
14	5	6	0	0	0	0	(128,64,32,32,16)	(2,2,1,1,1)	(2,2,1,1,1)	(2,1,1,1,1)	0.3549	0.8401	674	0.7378224
15	2	4	0	0	0	0	(64,32)	(9,3)	(4,2)	(2,1)	0.4649	0.8044	491	0.608262

Table 6
Composition of FC (AL) layers for different implemented models.

Models	AL/FC composition
VGG16	(4096, 4096, 2)
VGG19	(4096, 4096, 2)
ResNet50	(1000, 2)
MobileNet	(1024, 2)
DenseNet121	(1024, 2)
ResNet50V2	(1000, 2)
LeNet-5	(120, 84, 10, 2)
C_{best}	(128, 128, 128, 2)

Table 7

Total number of parameters for each model and the distribution of trainable vs. non-trainable parameters for pre-trained models. F/T ratio is the proportion of frozen (non-trainable) parameters to the total number of parameters.

Models	Total # params	Trainable # params	Non-trainable # params	F/T ratio
VGG16	33,605,442	18,890,754	14,714,688	0.437
VGG19	38,915,138	18,890,754	20,024,384	0.514
ResNet50	31,782,714	8,195,002	23,587,712	0.742
MobileNet	4,280,514	1,051,650	3,228,864	0.754
DenseNet121	8,089,144	1,051,650	7,037,504	0.869
ResNet50V2	31,759,802	8,195,002	23,564,800	0.741
LeNet-5	206,028	206,028	0	0
C_{best}	146,162	146,162	0	0

We freeze certain layers (which have a certain number of parameters) – these layers are pre-trained from ImageNet data. In Table 7, the models which are pre-trained have non-zero number of non-trainable parameters. We calculate a ratio to understand the extent of the proportion of parameters that we freeze. Metrics of training accuracy denoted as T_{RA} , testing accuracy as T_{EA} , training loss as T_{RL} and testing loss as T_{EL} in Table 8 over 15 epochs for each model are calculated.

Table 8. Train accuracy (T_{RA}), test accuracy (T_{EA}), train loss (T_{RL}) and test loss (T_{EL}) over 15 epochs for all models. The maximum and minimum in the T_{EA} and T_{EL} sequences respectively are emphasized in bold.

Models	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
VGG16	T_{RL}	0.4709	0.4384	0.4285	0.4233	0.4185	0.4146	0.4121	0.4097	0.4064	0.4048	0.4026	0.401	0.4006	0.3988
	T_{EL}	0.4911	0.5015	0.5427	0.5197	0.5307	0.5135	0.4986	0.491	0.5332	0.4564	0.4657	0.464	0.5226	0.4687
	T_{RA}	0.7925	0.811	0.8163	0.8181	0.8192	0.8204	0.8226	0.8233	0.8225	0.8255	0.8253	0.8271	0.8268	0.8283
	T_{EA}	0.7782	0.7693	0.7437	0.7575	0.7508	0.7593	0.7687	0.7718	0.7471	0.7892	0.7832	0.7848	0.7523	0.7806
VGG19	T_{RL}	0.5264	0.487	0.4739	0.4646	0.4589	0.4544	0.4492	0.4473	0.4441	0.4402	0.4391	0.4369	0.4341	0.4323
	T_{EL}	0.5179	0.5594	0.5518	0.5574	0.5584	0.5579	0.5658	0.5636	0.5518	0.5578	0.5474	0.5119	0.5507	0.5535
	T_{RA}	0.7513	0.7793	0.7863	0.7918	0.7963	0.7987	0.8012	0.8019	0.8032	0.8059	0.8068	0.8082	0.8089	0.81
	T_{EA}	0.7467	0.7203	0.7304	0.729	0.7293	0.7323	0.7283	0.7299	0.7342	0.7314	0.7363	0.7563	0.7365	0.734
ResNet50	T_{RL}	0.4242	0.366	0.3532	0.3473	0.3428	0.3404	0.3362	0.3358	0.3331	0.3306	0.3298	0.3272	0.3256	0.3233
	T_{EL}	0.601	0.6029	0.615	0.5997	0.606	0.6258	0.6164	0.623	0.6182	0.6245	0.6365	0.6581	0.644	0.6503
	T_{RA}	0.8261	0.8463	0.851	0.8539	0.8545	0.8556	0.8575	0.8586	0.8584	0.8604	0.8613	0.8619	0.8617	0.8629
	T_{EA}	0.7161	0.7161	0.7161	0.7162	0.7161	0.7161	0.7161	0.7161	0.7161	0.7161	0.7161	0.7154	0.7161	0.7159
MobileNetV2	T_{RL}	0.5235	0.4763	0.4638	0.459	0.4536	0.4516	0.4477	0.4457	0.4428	0.4449	0.4436	0.4406	0.4402	0.4388
	T_{EL}	0.497	0.4943	0.4941	0.4858	0.4841	0.4782	0.4835	0.4872	0.4952	0.4808	0.482	0.4924	0.4763	0.489
	T_{RA}	0.7616	0.7831	0.7886	0.7915	0.7936	0.7952	0.797	0.7976	0.7992	0.7979	0.7983	0.8006	0.8001	0.801
	T_{EA}	0.7615	0.7591	0.7593	0.769	0.7726	0.7786	0.7691	0.7633	0.757	0.777	0.776	0.7609	0.7792	0.7597
DenseNet121	T_{RL}	0.4656	0.4268	0.4186	0.4157	0.4122	0.411	0.409	0.4059	0.4044	0.4036	0.4043	0.4024	0.4007	0.3993
	T_{EL}	0.4884	0.4807	0.4833	0.4826	0.4809	0.4823	0.4813	0.4846	0.4839	0.4875	0.4877	0.4886	0.4843	0.4801
	T_{RA}	0.7944	0.8156	0.8193	0.8198	0.8229	0.8226	0.8227	0.8252	0.8261	0.8263	0.8255	0.8256	0.8265	0.8275
	T_{EA}	0.7703	0.7664	0.7647	0.7667	0.7681	0.7678	0.77	0.7684	0.7692	0.7682	0.7686	0.7689	0.7704	0.7732
ResNet50V2	T_{RL}	0.4982	0.4495	0.4356	0.4282	0.425	0.4217	0.4195	0.4193	0.4164	0.4135	0.4133	0.4112	0.4111	0.4107
	T_{EL}	0.5259	0.5246	0.5145	0.5146	0.513	0.5167	0.5105	0.5109	0.5042	0.5065	0.5091	0.5083	0.5065	0.5036
	T_{RA}	0.7867	0.8084	0.8133	0.8166	0.8163	0.8187	0.8189	0.8199	0.8212	0.8217	0.8217	0.8232	0.8244	0.823
	T_{EA}	0.739	0.7321	0.7421	0.7428	0.7385	0.7373	0.7406	0.74	0.7448	0.7432	0.7394	0.7445	0.7426	0.7471
LeNet-5	T_{RL}	0.4561	0.4222	0.4129	0.4143	0.4136	0.4144	0.413	0.4102	0.4086	0.4086	0.4126	0.4136	0.4075	0.4065
	T_{EL}	0.5746	0.5457	0.4418	0.4987	0.5988	0.5253	0.5609	0.4409	0.5404	0.4893	0.5079	0.5227	0.4331	0.461
	T_{RA}	0.7971	0.8128	0.8169	0.818	0.8178	0.8177	0.8187	0.82	0.82	0.8205	0.8188	0.8184	0.8206	0.8218
	T_{EA}	0.6947	0.725	0.8106	0.7622	0.6751	0.7539	0.7219	0.7925	0.7347	0.7721	0.7715	0.7508	0.81	0.8116
C_{best}	T_{RL}	0.4021	0.3646	0.3527	0.3422	0.3346	0.33	0.3228	0.3195	0.3151	0.3129	0.3108	0.3088	0.3052	0.3024
	T_{EL}	0.4715	0.3884	0.4171	0.4435	0.3679	0.4525	0.3849	0.3937	0.417	0.4541	0.6891	0.4315	0.6012	0.8488
	T_{RA}	0.8223	0.8427	0.8491	0.8543	0.8585	0.8613	0.8655	0.8664	0.8686	0.8695	0.8701	0.8713	0.872	0.8735
	T_{EA}	0.7901	0.835	0.8165	0.8009	0.8374	0.8027	0.8308	0.8228	0.811	0.8061	0.7857	0.8185	0.7956	0.7743

From Table 8, we notice that transferred inference can have a diminishing effect on T_{EA} since the pre-trained SOTA models of VGG16, VGG19, ResNet50, MobileNetV2, DenseNet121 and ResNet50V2 attain a maximum T_{EA} of 78.9%, 74.6%, 71.6%, 77.9%, 77.3% and 74.7% respectively over 15 epochs. On the other hand, LeNet-5 and C_{best} attain T_{EA} maxima as high as 81.1% and 83.7% respectively. Even with minimum T_{EL} we have 0.433 and 0.367 for LeNet-5 and C_{best} which are the lowest among all other models. These are the first evidences that only training the FC component for SOTA models keeping ImageNet weights for all convolutions is not a competent approach when compared to training smaller CNNs from scratch. It may be possible to have better performance with SOTA models by freezing less number of parameters and let those be learned. However, the biggest drawback in doing this is the computationally intensive nature of such training-from-scratch procedures for all SOTA models, making possession of advanced hardware a necessity. The high number of parameters to be learned, as seen in Table 7, for all SOTA models disallowed us to test their

efficacy with a F/T ratio of 0. Another striking difference noticed in Table 8 is the general trend of T_{EA} and T_{EL} for all models having F/T ratio > 0 vs. the improving trend of LeNet-5 and C_{best} for these metrics. There is no improvement T_{EA} and T_{EL} for pre-trained models indicating that adjustment of weights and biases of the FC components hardly makes any difference for the same features detected by all lower convolutional operations. Due to frozen weights and biases of all convolutions, there is no improvement or change in the higher level features detected by the final layers. Remarkably, it is possible that if LeNet-5 and C_{best} had been trained for more epochs, their maximum T_{EA} and minimum T_{EL} may have differed to be even better. To visualize the higher level features detected by LeNet-5 and C_{best} we plot their intermediate neural activations for all convolutional layers given in Fig. 6 and Fig. 7 respectively.

It is evident from Fig. 6 and Fig. 7 that as we go deeper with the convolutions, the features selected are more abstract in nature. This aspect is more pronounced when activations of max pooling layers are included as well. Further, we review the confusion matrices obtained by all the models under our observations and find the metrics given by (1), (2), (3), (4), (5), and (7) as defined in Sect. 4. In Table 9, we denote each model's respective confusion matrix (CM) using the following convention - $TN \leftarrow (0, 0)$, $FN \leftarrow (0, 1)$, $FP \leftarrow (1, 0)$ and $TP \leftarrow (1, 1)$, where we have the abscissa and ordinate of the CM as (x, y) .

Figure 8 plots all the metrics given in Table 9 for each model for better visual interpretation of the attained metrics. It is noticed that VGG16 and VGG19 perform relatively better than all other models when evaluated using said metrics. For better understanding, however, we discuss the performance of each model based on each metric in Sect. 6 further.

Table 9. Comparison of models based on P, S_n, S_p, F, BAC, MCC , and confusion matrix CM

Model	P	S_n	S_p	F	BAC	MCC	CM	
VGG16	0.926514	0.580655	0.96186	0.713902	0.830621	0.598944	1460	25272
							579	7300
VGG19	0.906714	0.531587	0.9486517	0.6702317	0.7949843	0.5322836	13579	6295
							735	7144
ResNet50	0.7207767	0.5001321	0.8658373	0.5905168	0.7175887	0.399076	14198	5676
							2200	5679
MobileNetV2	0.7418454	0.5880873	0.8858201	0.6560781	0.7679238	0.5039268	5845	4094
							2034	15780
DenseNet121	0.8511232	0.5670077	0.9263469	0.6806049	0.7967249	0.5411612	6706	5121
							1173	14753
ResNet50V2	0.6687397	0.5444307	0.8556017	0.6002164	0.723446	0.4228135	15465	4409
							2610	5269
LeNet-5	0.8271354	0.6276002	0.9215844	0.7136834	0.8162798	0.5893997	16007	3867
							1362	6517
C_{best}	0.6702627	0.7338799	0.8736197	0.7006302	0.7869528	0.5904638	17959	1915
							2598	5281

[1] The experiments were performed on a 64-bit workstation with 4 GB RAM having an Intel i5-4460 @ 3.20 GHz processor on Windows 10 Home OS. Python 3 was used as the programming language for experimentation.

6. Discussion

Detection of IDC, and hence BCa, is a problem that has profound clinical importance for facilitating the development of AI-driven techniques in modern day medical practices. Faster and more accurate diagnoses may be possible with augmented AI systems supervised by experts or clinicians, making their job easier and less intensive. Detection of IDC is an active area of research with numerous developments on different fronts for the diagnosis of BCa as we saw in Sect. 2. Of the techniques that employ CNNs, many have used transferred inference on models such as the VGG16, VGG19, ResNet50, etc. mainly based on ImageNet weights [107–110, 13, 111, 112]. The brunt of the results of our work are given by Table 5, Table 8 and Table 9. However, Table 8 and Table 9 do not clearly give away any single model being superior to the other. This is because, we discussed in Sect. 5 how Table 8 portrays the trends of T_{EA} and T_{EL} to have a higher gradient of improvement for models without transferred weights (LeNet-5 and C_{best}) along with better T_{EA} and T_{EL} . However, in Table 9 (and thus in Fig. 8) we notice that pre-trained models such as the VGG16 and VGG19 perform comparably well, if not better, than LeNet-5 and C_{best} in terms of P, S_p, F and BAC. This observation is noticed in [110] as well, where it was seen that pre-trained networks trained on non-medical images surprisingly performed comparable to those pre-trained on a medical image domain. We discuss this effect in detail in Sect. 6.1, and discuss a few other aspects in the following sub-sections.

6.1 Absence of Negative Transfer

The fact that there is no clear superior model when pre-trained models are put against those trained from scratch in our instance means that negative transfer [98] does not play a major role in the application of transferred inference for detection of IDC when using datasets collected by [47, 48]. To define negative transfer formally, let us consider the nomenclature used in Sect. 3.4. Let there be predictive learners $f_1(\cdot)$ and $f_2(\cdot)$ trained on D_a and

the latter on $(D_s + D_a)$. Then, negative transfer is the condition where $f_1(\cdot)$ performs better than $f_2(\cdot)$. The comparable performance of both schemes of training is surprising in this case because the source domain D_s is the ImageNet, which consists of images very different to those of breast histopathology. One of the reasons that negative transfer does not impact model performance here could be the intra-class variability in IDC datasets, as also discussed in [110]. Intra-class variability, in other words, means a high variance between different 50×50 patches of the same class. To demonstrate this variance, we show three different types of patches which belong to the same class in Fig. 9. Transfer learning provides a case for many different variations in the image to be detected. However, it can be detrimental when task domain D_a has very specific features – which is not the case in IDC detection. There are many intra-class and lesser inter-class variations, making the effects of using transferred inference largely neutral.

6.2. Metric-based Analysis

From Table 7 we noted that LeNet-5 and C_{best} performed best in terms of RAC / T_EA maxima being 81.1% and 83.7% respectively. However, especially in medical domains, testing accuracy alone should never be considered – the reasons being how they can vary in their behaviour when predicting positive or negative cases, as we shall see further. In terms of precision P , from Fig. 8 it is noticeable that VGG16 and VGG19 performed better than the rest. This means that these two pre-trained models are better at correctly predicting the positive IDC cases, i.e. having less number of FP. However, the same does not apply to sensitivity S_n , where the two trained-from-scratch models LeNet-5 and C_{best} perform better than all other SOTA models, meaning that they are better at predicting the positive cases out of all the positive cases in the test split of the dataset. In other words, LeNet-5 and C_{best} have minimal number of FN. Again, this does not hold for specificity S_p which is same as S_n but for negative cases. VGG16 and VGG19 having top values for S_n means that they are better at predicting the negative cases out of all the negative cases in the test split of the dataset. Presenting a harmonic mean between P and S_n we have F1-score F which is attained best by the models VGG16, LeNet-5 and C_{best} . In terms of balanced accuracy BAC, the five models, VGG16, VGG19, DenseNet121, LeNet-5, and C_{best} perform equally well. Finally, when we look at the Matthew's Correlation Coefficient MCC, it is noticed that LeNet-5 and C_{best} outperform other SOTA models. All these results are summarized in Table 10.

From Table 10 it is evident that there is no single superior model, however it becomes clear that ResNet50, MobileNetV2, and ResNet50V2 do not perform as well as the other models, since they do not appear in the top performing models list.

Table 10
Summary of metric-based analysis.

Metrics	Top performing models
RAC (regular accuracy)	LeNet-5, C_{best}
P (precision)	VGG16, VGG19
S_n (sensitivity)	LeNet-5, C_{best}
S_p (specificity)	VGG16, VGG19
F (F1-score)	VGG16, LeNet-5, C_{best}
BAC (balanced accuracy)	VGG16, VGG19, DenseNet121, LeNet-5, C_{best}
MCC (Matthew's correlation coefficient)	LeNet-5, C_{best}

6.3. The Conundrum of IDC Detection Accuracy

In machine learning applications, the validation set or test set accuracy plays a major role in determining model performance. However, such is not the case with clinical applications. Along with the accuracy, other metrics such as those discussed in Sect. 4 also play a major role. Some suggest that MCC is the most informative single score for a binary classifier through which a 2×2 confusion matrix can be attained [113]. Nonetheless, there are works available in the literature that compare attained accuracy to of those achieved in the past [62, 107, 108]. The problem with this is that in the case of detection of BCa, accuracy and other metrics are highly dependent on the dataset. Many characteristics may be attributed to a dataset such as the size, class balance ratio, intra-class variance, inter-class variance, sample dimensions, etc. All these factors make comparing attained metrics to those done in the past by other groups of researchers a futile strategy. To the best of our knowledge, there are no other works in literature that compare all the models used in this implementation with the same dataset as comprehensively as we have, taking into consideration all the different metrics that we use for evaluation (as discussed in Sect. 4). Hence, we do not provide a comparison-based analysis of our work as opposed to other works done in the past. Enforcing this ideology, we notice that the detection accuracy attained by [47] and F1-score by [48] (the two sources of the dataset that we use) are in agreement to what we achieved in this paper. Hence, there may be researchers achieving test accuracy as high as 95%, to which we argue that the dataset in consideration plays a major role.

The aspects discussed in sub-Sect. 6.1, 6.2 and 6.3 help us understand the dynamics and feasibility of transferred inference for the diagnosis of IDC. Transfer learning only either has positive or no effect to the detection of IDC as discussed in 6.1. In this paper, we notice that on certain metrics, pre-

trained models perform better than trained-from-scratch alternatives, and vice-versa (Table 10). It may be possible for SOTA models to outperform trained-from-scratch models when they are pre-trained on domains closely resembling the data distribution being used for comparison, unlike here, as we use ImageNet pre-trained weights for all SOTA models. However, one obvious advantage of making use of transfer learning in this application is that a lot of time and computation can be saved for training models having a large number of parameters.

7. Concluding Remarks And Future Directions

In this paper, we explore the dynamics and feasibility of transferred inference for the detection of invasive ductal carcinoma (IDC). We use pre-trained models namely VGG16, VGG19, ResNet50, MobileNetV2, DenseNet121 and ResNet50V2 along with LeNet-5 and a custom CNN architecture C_{best} chosen by comparing various traditional small-scale CNNs through maximization of an optimization function. For all models except LeNet-5 and C_{best} , transferred ImageNet weights were used and we tested the efficacy of both non pre-trained and pre-trained schemes on various metrics such as precision, sensitivity, specificity, F1-score, balanced accuracy and Matthew's correlation coefficient. We noticed that although LeNet-5 and C_{best} performed slightly better in terms of testing accuracy, transferred inference did not have a pronounced impact when all other metrics were taken into account as a whole. The best results for metrics were shared between largely VGG16, VGG19, LeNet-5 and C_{best} (Table 10). Due to the significant difference between the source domain of transferred weights (ImageNet) and the data distribution of the dataset of IDC, pre-trained models may not have been tested with their full potential. It may be possible to do so by using pre-trained models trained on a similar source distribution. Nonetheless, training models from scratch, as time and computationally intensive as it may be, promises to be a worthy alternative when proper source domains for transfer of weights are not available.

Admittedly, it is a challenging feat to achieve clinician-level accuracy for deep learning methods in the detection of IDC due to high intra-class variance in the datasets. Future directions for the detection of IDC may involve a mixture of detection of breast cancer (BCa) through whole slide images (WSI) using models trained to classify only patches of the WSI, as done in [65]. Models such as Fast R-CNN [14], Faster R-CNN [17], You Only Look Once (YOLO) [16], and Single Shot Detection (SSD) [15] may be used to localize the exact regions of the carcinoma in WSI. More emphasis may be given to tackle IDC detection using unsupervised deep learning methods such as extraction of high level features through restricted Boltzmann machines [44] and deep Boltzmann machines [114], deep belief networks [115], autoencoders, and etc. to explore and open doors for more open comparisons between the efficacy of different varieties of techniques for IDC detection. WSI-based patch dataset creators may consider addition of two more classes, namely 'sparse-normal', and 'sparse-IDC' to tackle with patches having a majority of the regions empty (white) as seen in Fig. 9 (left) to help CNN-based techniques better identify classes and reduce the intra-class variance in IDC datasets.

Declarations

Acknowledgements

The author would like to thank all the co-authors for their help and useful comments.

Funding: This work is partially funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the Project UIDB/50008/2020; and by Brazilian National Council for Scientific and Technological Development – CNPq, via Grant No. 313036/2020-9.

Conflict of Interest: There are no conflicts of interests.

Availability of data and material: N/A

Code availability: N/A

References

1. Zeiler, M.D., Fergus, R.: (2014). Visualizing and Understanding Convolutional Networks. arXiv:1311.2901v3 [cs.CV] 28 Nov 2013. Comput. Vision–ECCV 2014. DOI: 10.1007/978-3-319-10590-1_53
2. Acharya, U.R., Oh, S.L., Hagiwara, Y., Tan, J.H., Adam, M., et al.: (2017). A deep convolutional neural network model to classify heartbeats. Computers in Biology and Medicine. DOI: 10.1016/j.combiomed.2017.08.022
3. Guo, T., Dong, J., Li, H., Gao, Y.: (2017). Simple convolutional neural network on image classification, 2017 IEEE 2nd International Conference on Big Data Analysis, ICBDA 2017. DOI: 10.1109/ICBDA.2017.8078730
4. Rocco, I., Arandjelovic, R., Sivic, J.: (2017). Convolutional neural network architecture for geometric matching. Proceedings – 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. DOI: 10.1109/CVPR.2017.12
5. Sam, D.B., Surya, S., Babu, R.V.: (2017). Switching convolutional neural network for crowd counting. Proceedings – 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. DOI:10.1109/CVPR.2017.429
6. Vinayakumar, R., Soman, K.P., Poornachandran, P.: (2017). Applying convolutional neural network for network intrusion detection. 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017. DOI:10.1109/ICACCI.2017.8126009

7. Acharya, U.R., Fujita, H., Oh, S.L., Hagiwara, Y., Tan, et al.: (2019). Deep convolutional neural network for the automated diagnosis of congestive heart failure using ECG signals. *Applied Intelligence*, DOI:10.1007/s10489-018-1179-1
8. Kruthika, K.R., Rajeswari, Maheshappa, H.D.: (2019). CBIR system using Capsule Networks and 3D CNN for Alzheimer's disease diagnosis. *Informatics in Medicine Unlocked*. DOI: 10.1016/j.imu.2018.12.001
9. Raghavendra, U., Fujita, H., Bhandary, S.V., Gudigar, A., Tan, J.H., et al.: (2018). Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images. *Inf. Sci.* DOI:10.1016/j.ins.2018.01.051
10. Rajpurkar, P., Irvin, J., Bagul, A., Ding, D., Duan, et al.: MURA: Large dataset for abnormality detection in musculoskeletal radiographs. *arXiv* (2017)
11. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, et al.: (2017). CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning. *arXiv*
12. Sharma, P., Bora, K., Kasugai, K., Balabantaray, B.K.: Two Stage Classification with CNN for Colorectal Cancer Detection. *Oncologie* **22**(3), 129–145 (2020)
13. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., et al.: (2015). ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vision*. DOI:10.1007/s11263-015-0816-y
14. Girshick, R.: (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*. DOI: 10.1109/ICCV.2015.169
15. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., et al.: (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. DOI: 10.1007/978-3-319-46448-0_2
16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. DOI: 10.1109/CVPR.2016.91
17. Ren, S., He, K., Girshick, R., Sun, J.: *Faster R-CNN*. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017)
18. Chang, H., Han, J., Zhong, C., Snijders, A.M., Mao, J.H.: (2018). Unsupervised Transfer Learning via Multi-Scale Convolutional Sparse Coding for Biomedical Applications. *IEEE Trans. Pattern Anal. Mach. Intell.* DOI:10.1109/TPAMI.2017.2656884
19. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. *Advances in Neural Information Processing Systems* (2016)
20. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., et al.: (2018). A survey on deep transfer learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. DOI:10.1007/978-3-030-01424-7_27
21. Zhu, H., Long, M., Wang, J., Cao, Y.: (2016). Deep hashing network for efficient similarity retrieval. *30th AAAI Conference on Artificial Intelligence*, AAAI 2016
22. He, K., Zhang, X., Ren, S., Sun, J.: (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. DOI: 10.1109/CVPR.2016.90
23. He, K., Sun, J.: (2015). Convolutional neural networks at constrained time cost. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. DOI:10.1109/CVPR.2015.7299173
24. Srivastava, R.K., Greff, K., Schmidhuber, J.: (2015). Highway NetWorks: Training Very Deep Networks *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems*
25. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. *Cs.Toronto.Edu* (2009)
26. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: (1998). Gradient-based learning applied to document recognition. *Proceedings of IEEE*. DOI:10.1109/5.726791
27. American Cancer Society: Breast Cancer Facts & Figs, pp. 2019–2020. *Am. Cancer Soc* (2019)
28. Pal, S.K., Lau, S.K., Kruper, L., Nwoye, U., Garberoglio, C., et al.: (2010). Papillary carcinoma of the breast: An overview. *Breast Cancer Research and Treatment*. DOI: 10.1007/s10549-010-0961-5
29. Cheang, M.C.U., Martin, M., Nielsen, T.O., Prat, A., Voduc, D., et al.: (2015). Defining Breast Cancer Intrinsic Subtypes by Quantitative Receptor Expression. *Oncologist*. DOI: 10.1634/theoncologist.2014-0372
30. Anand, A., Anand, H., Rautaray, S.S., Pandey, M., Gourisaria, M.K.: Analysis and prediction of chronic heart diseases using machine learning classification models. *International Journal of Advanced Trends in Computer Science and Engineering* **9**, 8479–8487 (2020)
31. Das, S., Sharma, R., Gourisaria, M.K., Rautaray, S.S., Pandey, M.: Heart disease detection using core machine learning and deep learning techniques: A comparative study. *International Journal of Emerging Technology* **11**, 531–538 (2020)
32. Dey, S., Gourisaria, M.K., Rautaray, S.S., Pandey, M.: (2021). Segmentation of nuclei in microscopy images across varied experimental systems. *Advances in Intelligent Systems and Computing*. DOI: 10.1007/978-981-15-5679-1_9
33. GM, H., Gourisaria, M.K., Rautaray, S.S., Pandey, M.: Pneumonia detection using CNN through chest X-ray. *Journal of Engineering Science and Technology* **16**(1), 861–876 (2021)
34. Mishra, S., Pandey, M., Rautaray, S.S., Gourisaria, M.K.: A Survey on Big Data Analytical Tools & Techniques in Health Care Sector. *International Journal of Emerging Technology* **11**, 554–560 (2020)

35. Rautaray, S.S., Dey, S., Pandey, M., Gourisaria, M.K.: Nuclei segmentation in cell images using fully convolutional neural networks. *International Journal of Emerging Technology* **11**, 731–737 (2020)
36. Rautaray, S.S., Pandey, M., Gourisaria, M.K., Sharma, R., Das, S.: Paddy Crop Disease Prediction- A Transfer Learning Technique. *International Journal of Recent Technology and Engineering*. **8**, 1490–1495 (2020). DOI:10.35940/ijrte.f7782.038620
37. Sharma, R., Gourisaria, M.K., Rautaray, S.S., Pandey, M., Patra, S.S.: ECG Classification using Deep Convolutional Neural Networks and Data Analysis. *International Journal of Advancing Trends in Computer Science and Engineering* **9**, 5788–5795 (2020)
38. Johri, P., Saxena, V.S., Kumar, A.: Rummage of Machine Learning Algorithms in Cancer Diagnosis. *International Journal of E-Health and Medical Communications (IJEHMC)*. **12**(1), 1–15 (2021). DOI:10.4018/IJEHMC.2021010101
39. Shah, S.H., Iqbal, M.J., Ahmad, I., Khan, S., Rodrigues, J.J.: (2020). Optimized gene selection and classification of cancer from microarray gene expression data using deep learning. *Neural Computing and Applications*, 1–12
40. Jee, G., GM, H., Gourisaria, M.K.: Juxtaposing inference capabilities of deep neural models over posteroanterior chest radiographs facilitating COVID-19 detection. *Journal of Interdisciplinary Mathematics*. **24**(2), 299–325 (2021). DOI:10.1080/09720502.2020.1838061
41. Saha, I., Gourisaria, M.K., GM, H.: (2021). Distinguishing Pneumonia and COVID-19: Utilizing Computer Vision to Mimic Clinician Efficacy. 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 834–841. DOI: 10.1109/ICAIS50930.2021.9395961
42. Selvanambi, R., Jaisankar, N.: Healthcare: Prediction of Breast Cancer Stage Using Social Spider-Inspired Optimization Algorithm. *International Journal of E-Health and Medical Communications (IJEHMC)*. **10**(2), 63–85 (2019). DOI:10.4018/IJEHMC.2019040104
43. Sahu, A., GM, H., Gourisaria, M.K.: (2020). A Dual Approach for Credit Card Fraud Detection using Neural Network and Data Mining Techniques. 2020 IEEE 17th India Council International Conference (INDICON), 1–7. DOI: 10.1109/INDICON49873.2020.9342462
44. GM, H., Gourisaria, M.K., Rautaray, S.S., Pandey, M.: !2021). UBMTR: Unsupervised Boltzmann machine-based time-aware recommendation system. *Journal of King Saud University-Computer and Information Sciences*
45. Ahmed, I., Ahmad, M., Rodrigues, J.J., Jeon, G., Din, S.: A deep learning-based social distance monitoring framework for COVID-19. *Sustainable Cities and Society* **65**, 102571 (2021)
46. Sahu, A., GM, H., Gourisaria, M.K., Rautaray, S.S., Pandey, M.: (2021). Cardiovascular risk assessment using data mining inferencing and feature engineering techniques. *International Journal of Information Technology*, 1–13
47. Cruz-Roa, A., Basavanhally, A., González, F., Gilmore, H., Feldman, M., et al.: (2014). Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks. *Medical Imaging 2014: Digital Pathology*. DOI:10.1117/12.2043872
48. Janowczyk, A., Madabhushi, A.: (2016). Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of Pathology Informatics*. DOI:10.4103/2153-3539.186902
49. Wang, D., Khosla, A., Gargeya, R., Irshad, H., Beck, A.H.: 2016. Deep Learning for Identifying Metastatic Breast Cancer. arXiv preprint arXiv:1606.05718
50. Caicedo, J.: A prototype system to archive and retrieve histopathology images by content. National University of Colombia (2008)
51. Han, J., Ma, K.K.: (2002). Fuzzy color histogram and its use in color image retrieval. *IEEE Trans. Image Process*. DOI:10.1109/TIP.2002.801585
52. Lux, M., Chatzichristofis, S.A.: (2008). LIRe: Lucene image retrieval - An extensible java CBIR library. MM'08 - Proceedings of the 2008 ACM International Conference on Multimedia, with Co-Located Symposium and Workshops. DOI: 10.1145/1459359.1459577
53. Basavanhally, A., Ganesan, S., Feldman, M., Shih, N., Mies, C., et al.: (2013). Multi-field-of-view framework for distinguishing tumor grade in ER + breast cancer from entire histopathology slides. *IEEE Trans. Biomed. Eng*. DOI:10.1109/TBME.2013.2245129
54. Messing, D.S., Van Beek, P., Errico, J.H.: (2001). The MPEG-7 colour structure descriptor: Image description using colour and local spatial information. *IEEE International Conference on Image Processing*. DOI: 10.1109/icip.2001.959134
55. Ahonen, T., Hadid, A., Pietikäinen, M.: (2006). Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell*. DOI:10.1109/TPAMI.2006.244
56. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., et al.: (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. DOI:10.1109/CVPR.2015.7298594
57. Krizhevsky, A., Sutskever, I., Hinton, G.E.: (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*. DOI: 10.1061/(ASCE)GT.1943-5606.0001284
58. Simonyan, K., Zisserman, A.: (2015). Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings
59. Wang, D., Otto, C., Jain, A.K.: (2017). Face Search at Scale. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. DOI: 10.1109/TPAMI.2016.2582166
60. Otsu, N.: (1979). THRESHOLD SELECTION METHOD FROM GRAY-LEVEL HISTOGRAMS. *IEEE Transactions on System Man Cybernetics*. DOI: 10.1109/tsmc.1979.4310076
61. Alghodhaifi, H., Alghodhaifi, A., Alghodhaifi, M.: (2019). Predicting Invasive Ductal Carcinoma in breast histology images using Convolutional Neural Network. *Proceedings of IEEE National. Aerospace Electronics Conference NAECON 2019*, 374–378. DOI: 10.1109/NAECON46414.2019.9057822

62. Celik, Y., Talo, M., Yildirim, O., Karabatak, M., Acharya, U.R.: (2020). Automated invasive ductal carcinoma detection based using deep transfer learning with whole-slide images. *Pattern Recogn. Lett.* DOI:10.1016/j.patrec.2020.03.011
63. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: (2017). Densely connected convolutional networks. *Proceedings – 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017.* DOI: 10.1109/CVPR.2017.243
64. Smith, L.N.: (2017). Cyclical learning rates for training neural networks. *Proceedings – 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017.* DOI: 10.1109/WACV.2017.58
65. Shen, L., Margolies, L.R., Rothstein, J.H., Fluder, E., McBride, R., Sieh, W.: (2019). Deep Learning to Improve Breast Cancer Detection on Screening Mammography. *Scientific Reports.* DOI: 10.1038/s41598-019-48995-4
66. Dundar, M.M., Badve, S., Bilgin, G., Raykar, V., Jain, R., et al.: (2011). Computerized classification of intraductal breast lesions using histopathological images. *IEEE Trans. Biomed. Eng.* DOI:10.1109/TBME.2011.2110648
67. Dundar, M.M., Badve, S., Raykar, V.C., Jain, R.K., Sertel, O., et al.: (2010). A multiple instance learning approach toward optimal classification of pathology slides. *Proceedings - International Conference on Pattern Recognition.* DOI: 10.1109/ICPR.2010.669
68. Sain, S.R., Vapnik, V.N.: (1996). The Nature of Statistical Learning Theory. *Technometrics.* DOI: 10.2307/1271324
69. Gecer, B., Aksoy, S., Mercan, E., Shapiro, L.G., Weaver, D.L., et al.: (2018). Detection and classification of cancer in whole slide breast histopathology images using deep convolutional networks. *Pattern Recogn.* DOI:10.1016/j.patcog.2018.07.022
70. Long, J., Shelhamer, E., Darrell, T.: (2015). Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* DOI: 10.1109/CVPR.2015.7298965
71. Yengec Tasdemir, S.B., Tasdemir, K., Aydin, Z.: (2019). ROI Detection in Mammogram Images Using Wavelet-Based Haralick and HOG Features. *Proceedings – 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018.* DOI: 10.1109/ICMLA.2018.00023
72. Dalal, N., Triggs, B.: (2005). Histograms of oriented gradients for human detection. *Proceedings – 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005.* DOI: 10.1109/CVPR.2005.177
73. Haralick, R.M., Dinstein, I., Shanmugam, K.: (1973). Textural Features for Image Classification. *IEEE Transactions on System Man Cybernetics.* DOI:10.1109/TSMC.1973.4309314
74. Mallat, S.G.: (1989). A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis Machine Intelligence.* DOI:10.1109/34.192463
75. Brancati, N., De Pietro, G., Frucci, M., Riccio, D.: (2019). A Deep Learning Approach for Breast Invasive Ductal Carcinoma Detection and Lymphoma Multi-Classification in Histological Images. *IEEE Access.* DOI:10.1109/ACCESS.2019.2908724
76. Cruz-Roa, A.A., Ovalle, A., Madabhushi, J.E., González Osorio, A., F. A. (2013). A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* DOI: 10.1007/978-3-642-40763-5_50
77. Feng, Y., Zhang, L., Yi, Z.: (2018). Breast cancer cell nuclei classification in histopathology images using deep neural networks. *Int. J. Comput. Assist. Radiol. Surg.* DOI:10.1007/s11548-017-1663-9
78. Hou, L., Nguyen, V., Kanevsky, A.B., Samaras, D., Kurc, T.M., et al.: (2019). Sparse autoencoder for unsupervised nucleus detection and representation in histopathology images. *Pattern Recognition.* DOI: 10.1016/j.patcog.2018.09.007
79. Xu, J., Xiang, L., Liu, Q., Gilmore, H., Wu, J., et al.: (2016). Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. *IEEE Trans. Med. Imaging.* DOI:10.1109/TMI.2015.2458702
80. Quan, T.M., Hildebrand, D.G.C., Jeong, W.-K.: (2016). FusionNet: A deep fully residual convolutional neural network for image segmentation in connectomics. *arXiv:1612.05360*
81. GM, H., Gourisaria, M.K., Pandey, M., Rautaray, S.S.: A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review.* **38**, 100285 (2020). DOI:10.1016/j.cosrev.2020.100285
82. Belciug, S., Gorunescu, F., Gorunescu, M., Salem, A.B.M.: (2010). Assessing performances of unsupervised and supervised neural networks in breast cancer detection. *INFOS2010–2010 7th International Conference on Informatics and Systems*
83. Kohonen, T.: (1997). Exploration of very large databases by self-organizing maps. *IEEE International Conference on Neural Networks - Conference Proceedings.* DOI: 10.1109/ICNN.1997.611622
84. Xu, J., Xiang, L., Liu, Q., Gilmore, H., Wu, J., et al.: (2016). Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. *IEEE Trans. Med. Imaging.* DOI:10.1109/TMI.2015.2458702
85. Fatakdawala, H., Xu, J., Basavanahally, A., Bhanot, G., Ganesan, S., Feldman, M., et al.: (2010). Expectation-maximization-driven geodesic active contour with overlap resolution (EMaGACOR): Application to lymphocyte segmentation on breast cancer histopathology. *IEEE Trans. Biomed. Eng.* DOI:10.1109/TBME.2010.2041232
86. Chang, H., Han, J., Borowsky, A., Loss, L., Gray, J.W., et al.: (2013). Invariant delineation of nuclear architecture in glioblastoma multiforme for clinical and molecular association. *IEEE Transactions in Medical Imaging.* DOI:10.1109/TMI.2012.2231420
87. Ruifrok, A.C., Johnston, D.A.: Quantification of histochemical staining by color deconvolution. *Analytical and Quantitative Cytology and Histology* (2001)

88. Mikołajczyk, A., Grochowski, M., et al.: (2018). Data augmentation for improving deep learning in image classification problem. 2018 International Interdisciplinary PhD Workshop, IIPHDW 2018. DOI: 10.1109/IIPHDW.2018.8388338
89. Wigington, C., Stewart, S., Davis, B., Barrett, B., Price, B., et al.: (2017). Data Augmentation for Recognition of Handwritten Words and Lines Using a CNN-LSTM Network. Proceedings of the International Conference on Document Analysis and Recognition, ICDAR. DOI: 10.1109/ICDAR.2017.110
90. Koivunen, A. C., Kostinski, A. B. (1999). The feasibility of data whitening to improve performance of weather radar. Journal of Applied Meteorology. DOI: 10.1175/1520-0450(1999)038<0741:TFODWT>2.0.CO;2
91. Ranjan, R., Sankaranarayanan, S., Castillo, C.D., Chellappa, R.: (2017). An All-In-One Convolutional Neural Network for Face Analysis. IEEE 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017) - Washington, DC, DC, USA (2017.5.30-2017.6.3)] 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), 17–24. DOI: 10.1109/FG.2017.137
92. Yin, X., Liu, X.: (2018). Multi-Task Convolutional Neural Network for Pose-Invariant Face Recognition. IEEE Trans. Image Process. DOI:10.1109/TIP.2017.2765830
93. Du, Y., Fu, Y., Wang, L.: (2016). Skeleton based action recognition with convolutional neural network. Proceedings – 3rd IAPR Asian Conference on Pattern Recognition, ACPR 2015. DOI: 10.1109/ACPR.2015.7486569
94. Ji, S., Xu, W., Yang, M., Yu, K.: (2013). 3D Convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis Machine Intelligence. DOI:10.1109/TPAMI.2012.59
95. Pinheiro, P.O., Collobert, R.: (2014). Recurrent convolutional neural networks for scene labeling. 31st International Conference on Machine Learning, ICML 2014
96. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: (2016). Learning Deep Features for Discriminative Localization. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. DOI: 10.1109/CVPR.2016.319
97. Pan, S.J., Yang, Q.: (2010). A survey on transfer learning. IEEE Transactions Knowledge Data Engineering. DOI:10.1109/TKDE.2009.191
98. Weiss, K., Khoshgoftaar, T.M., Wang, D.D.: (2016). A survey of transfer learning. Journal of Big Data. DOI:10.1186/s40537-016-0043-6
99. Matthews, B.W.: (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. BBA - Protein Structure. DOI: 10.1016/0005-2795(75)90109-9
100. Chollet, F.: 2017. Xception: Deep learning with depthwise separable convolutions. Proceedings – 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. DOI: 10.1109/CVPR.2017.195
101. He, K., Zhang, X., Ren, S., Sun, J.: (2016). Identity mappings in deep residual networks. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). DOI: 10.1007/978-3-319-46493-0_38
102. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: (2016). Rethinking the Inception Architecture for Computer Vision. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. DOI: 10.1109/CVPR.2016.308
103. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. DOI: 10.1109/CVPR.2018.00474
104. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: (2018). Learning Transferable Architectures for Scalable Image Recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. DOI: 10.1109/CVPR.2018.00907
105. Agarap, A.F.M.: (2018). Deep Learning using Rectified Linear Units (ReLU). arXiv
106. LeNail, A.: (2019). NN-SVG: Publication-Ready Neural Network Architecture Schematics. Journal Open Source Software. DOI:10.21105/joss.00747
107. Deniz, E., Şengür, A., Kadiroğlu, Z., Guo, Y., Bajaj, V., et al.: (2018). Transfer learning based histopathologic image classification for breast cancer detection. Health Information Science and Systems. DOI:10.1007/s13755-018-0057-x
108. Guan, S., Loew, M.: (2018). Breast cancer detection using transfer learning in convolutional neural networks. Proceedings - Applied Image Pattern Recognition Workshop 2017-October, 1–8. DOI: 10.1109/AIPR.2017.8457948
109. Khan, S.U., Islam, N., Jan, Z., Ud Din, I., Rodrigues, J.J.P.C.: (2019). A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. Pattern Recogn. Lett. DOI:10.1016/j.patrec.2019.03.022
110. Kieffer, B., Babaie, M., Kalra, S., Tizhoosh, H.R.: (2018). Convolutional neural networks for histopathology image classification: Training vs. Using pre-trained networks. Proceedings of the 7th International Conference on Image Processing Theory, Tools and Applications, IPTA 2017. DOI: 10.1109/IPTA.2017.8310149
111. Spanhol, F.A., Oliveira, L.S., Petitjean, C., Heutte, L.: (2016). A Dataset for Breast Cancer Histopathological Image Classification. IEEE Trans. Biomed. Eng. DOI:10.1109/TBME.2015.2496264
112. Talo, M.: Convolutional neural networks for multi-class histopathology image classification. arXiv (2019)
113. Chicco, D.: (2017). Ten quick tips for machine learning in computational biology. BioData Mining. DOI: 10.1186/s13040-017-0155-3
114. Salakhutdinov, R., Hinton, G.: Deep Boltzmann machines. Journal of Machine Learning Research (2009)
115. Neal, R.M.: (1992). Connectionist learning of belief networks. Artif. Intell. DOI:10.1016/0004-3702(92)90065-6
116. Schmidt, M., Fung, G., Rosales, R.: (2007). Fast optimization methods for L1 regularization: A comparative study and two new approaches. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). DOI:

117. Cortes, C., Mohri, M., Rostamizadeh, A.: (2009). L2 regularization for learning kernels. Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI 2009
118. Ma, R., Miao, J., Niu, L., Zhang, P.: (2019). Transformed ℓ_1 regularization for learning sparse deep neural networks. Neural Netw. DOI:10.1016/j.neunet.2019.08.015
119. Ni, X., Fang, L., Huttunen, H.: (2020). AdaptiveReID: Adaptive L2 Regularization in Person Re-Identification. arXiv
120. Qian, X., Huang, H., Chen, X., Huang, T.: (2017). Efficient construction of sparse radial basis function neural networks using L1-regularization. Neural Networks. DOI: 10.1016/j.neunet.2017.07.004
121. Yang, C., Yang, Z., Khattak, A.M., Yang, L., Zhang, W., et al.: (2019). Structured Pruning of Convolutional Neural Networks via L1 Regularization. IEEE Access. DOI:10.1109/ACCESS.2019.2933032
122. Zeng, S., Zhang, B., Zhang, Y., Gou, J.: (2018). Collaboratively weighting deep and classic representation via l2 regularization for image classification. arXiv
123. Zhai, Y., Deng, W., Xu, Y., Ke, Q., Gan, J., et al.: (2019). Robust SAR Automatic Target Recognition Based on Transferred MS-CNN with L2-Regularization. Computational Intelligence and Neuroscience. DOI: 10.1155/2019/9140167
124. Ioffe, S., Szegedy, C.: (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. 32nd International Conference on Machine Learning, ICML 2015
125. Atanov, A., Ashukha, A., Molchanov, D., Neklyudov, K., Vetrov, D.: (2019). Uncertainty Estimation via Stochastic Batch Normalization. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). DOI: 10.1007/978-3-030-22796-8_28
126. Li, Y., Wang, N., Shi, J., Hou, X., Liu, J.: (2018). Adaptive Batch Normalization for practical domain adaptation. Pattern Recognition. DOI: 10.1016/j.patcog.2018.03.005
127. Cozijmans, T., Ballas, N., Laurent, C., Gülçehre, Ç., Courville, A.: (2017). Recurrent batch normalization. 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings
128. Wang, S.H., Muhammad, K., Hong, J., Sangaiah, A.K., Zhang, Y.D.: (2020). Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization. Neural Comput. Appl. DOI:10.1007/s00521-018-3924-0

Figures

Figure 1

We make use of network-based transfer learning by using a portion of a fully trained CNN on the ImageNet dataset [13]. This portion comprises only the convolutional layers while the fully-connected dense layers are learnt in-house. Two vectors of 128 units are sequentially connected in the transferred model with a binary output layer at the right-most side

Figure 3

Patches of 50×50 are extracted from the WSI (left) and each patch is independently fed into the CNN (middle) to get the classification output (right). We also find the intermediate neural activations of different convolutional and max-pooling layers (lower-middle).

Performance Evaluation of Vanilla CNNs based on Θ (optimization function)

Figure 4

Performance of C_{best} as opposed to other vanilla CNNs based on optimization function $O(M\theta_i, \sigma_i)$ based on Table 5.

Figure 5

The NN-SVG [106] representation of C_{best} . The window size 50×50 diminishes as input progresses through the network. Full connection (right) is made before getting an output at the last layer with one neuron. This output $\varepsilon[01]$ denoting the probability of existence of IDC.

Figure 6

Convolutional neural activations for LeNet-5.

Figure 7

Convolutional neural activations for Cbest

Multimetric analysis

Figure 8

Multi-metric analysis of all models based on the performance metrics defined in Section 4.

Figure 9

Intra-class variability in the dataset, eliminating the case of negative transfer. (Left): Patch having a majority of white regions. (Middle): Typical patch with different colour schemes for both the classes (as seen in Fig. 2). (Right): Mixture of the colour schemes in the same patch, making the detection process difficult.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [Appendices.docx](#)