IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 23, 2022, accepted March 7, 2022, date of publication March 14, 2022, date of current version March 24, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3159700

On the Dynamics and Feasibility of Transferred Inference for Diagnosis of Invasive Ductal Carcinoma: A Perspective

G. M. HARSHVARDHAN[1], AANCHAL SAHU[1], MAHENDRA KUMAR GOURISARIA[1], (Member, IEEE), PRADEEP KUMAR SINGH[2], (Senior Member, IEEE), WEI-CHIANG HONG[3], (Senior Member, IEEE), VIJANDER SINGH[4,5], AND BUNIL KUMAR BALABANTARAY[6], (Member, IEEE)

[1]School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha 751024, India
[2]Department of Computer Science and Engineering, KIET Group of Institutions, Delhi-NCR, Ghaziabad, Uttar Pradesh 201009, India
[3]Department of Information Management, Asia Eastern University of Science and Technology, Taipei 22064, Taiwan
[4]Department of Computer Science and Engineering, School of Computing and Information Technology, Manipal University Jaipur, Jaipur 302034, India
[5]ERCIM, Norwegian University of Science and Technology, Ålesund, 7491 Trondheim, Norway
[6]Department of Computer Science and Engineering, National Institute of Technology Meghalaya, Shillong 793003, India

Corresponding authors: Wei-Chiang Hong (samuelsonhong@gmail.com), Pradeep Kumar Singh (pradeep_84cs@yahoo.com), and Mahendra Kumar Gourisaria (mkgourisaria2010@gmail.com)

This work was supported by the Ministry of Science and Technology under Grant MOST 110-2410-H-161-001.

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.

INDEX TERMS CNN, breast cancer, transfer learning, invasive ductal carcinoma.

I. 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], [2]–[6], inclusive of medical applications [2], [7], [8]–[12], [128]. 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],

The associate editor coordinating the review of this manuscript and approving it for publication was Santosh Kumar.

[18], [20], [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. Fig. 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 Section V 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

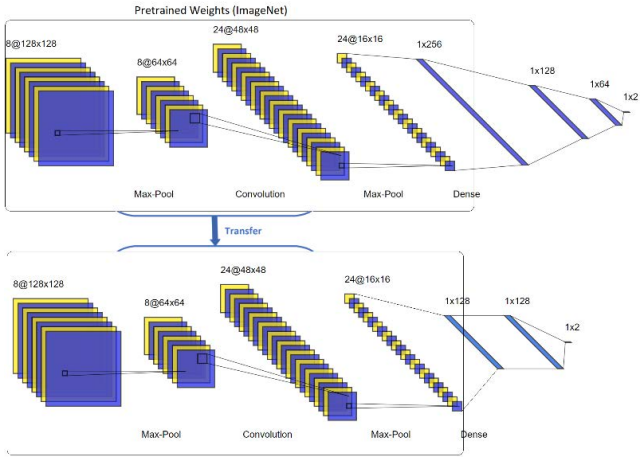

FIGURE 1. We make use of network-based transfer learning by using a portion of a fully trained CNN on the ImageNet dataset [13]. This portion comprises only the convolutional layers while the fully-connected dense layers are learnt in-house. Two vectors of 128 units are sequentially connected in the transferred model with a binary output layer at the right-most side.

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].

It has been found that the use of the more recent deep CNNs has been better than using traditional approaches, mainly those which involve the extraction of handcrafted features from the images over which machine learning models like random forest are applied (this is also discussed in [47]). Neural networks have been found to perform better not only in computer vision tasks but also in other applications like speech recognition, reinforcement learning, and generative modelling. When many samples are present, for a pathologist, it is a time-consuming and difficult task to check for IDC for all the samples, which is where using deep CNNs gives a significant advantage in that it can generalize the features better owing to large amounts of data and also save time by providing instantaneous predictions.

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 – Section II (Related Work) describes the related work which is divided into three different techniques used majorly for the detection of BCa, Section III (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., Section IV (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, Section V (Results) contains all the results in terms of the performance metrics, neural activations of intermediate chosen traditional CNN models, etc. In Section VI (Discussion) we analyse performance of each model with each metric and understand the effect of transferred weights for inference, and finally in Section VII (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.

II. 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], prediction of COVD-19 [40], [41], or detection of BCa through spider-inspired optimization [42]. Machine learning and deep learning methods are used even in non-medical fields [43]–[46]. 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.

A. 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.* [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 [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.* [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.* [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.

B. 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.* [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]. Dundar *et al.* [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.* [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, Yengec Tasdemir *et al.* [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.

C. 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.* [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.* [79] 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) [84], Blue Ratio (BR) thresholding [85], and Colour Deconvolution (CD) [86] in both qualitative and quantitative terms.

III. METHODOLOGY AND MATERIALS

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.

A. DATASET

We make use of 162 WSIs collected by [47] and [48] 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.

B. 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 [87], [88]. This is achieved through various ways such as whitening transforms [89], 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	$1/255$	-	-	✗

C. 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 [90], [91], action recognition [92], [93], scene labelling [94], 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). Fig. 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) [95] 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.

D. 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 [96] and [97].

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

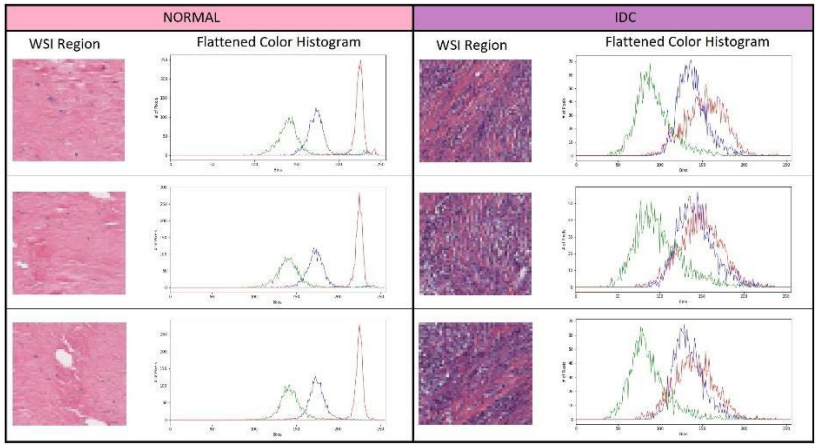

FIGURE 2. Flattened colour histogram over three channels (RGB) and 256 bins for normal and IDC cases.

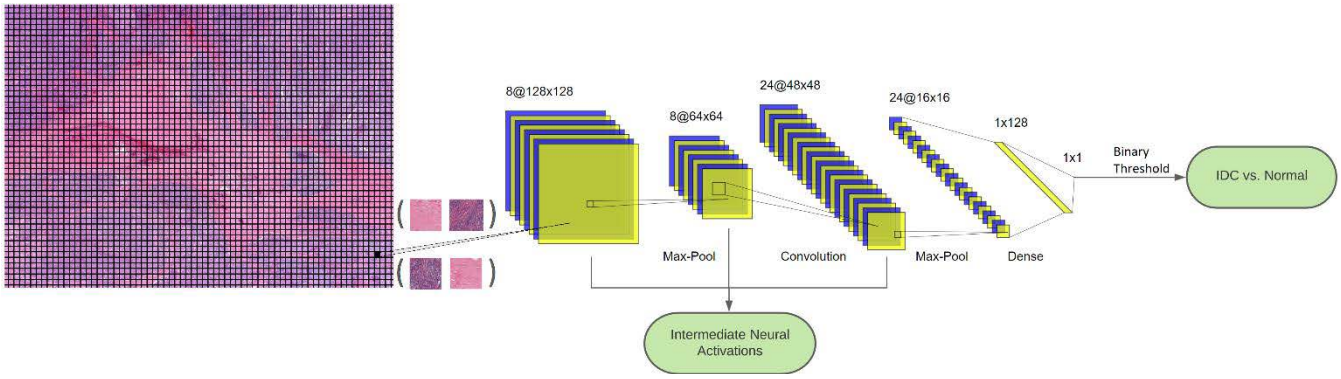

FIGURE 3. Patches of 50 × 50 are extracted from the WSI (left) and each patch is independently fed into the CNN (middle) to get the classification output (right). We also find the intermediate neural activations of different convolutional and max-pooling layers (lower-middle).

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 $\mathcal{T}_a$ having domain $\mathcal{D}_a$, we can use a source domain $\mathcal{D}_s$ with a well defined $\mathcal{T}_s$. Through latent knowledge transfer from $\mathcal{D}_s$ and $\mathcal{T}_s$, an attempt at improving the predictive function $f_a(.)$ is made (which is a component of learning task $\mathcal{T}_a$) where $\mathcal{D}_a \neq \mathcal{D}_s$ or $T_a \neq T_s$. If we denote the sizes of domains $\mathcal{D}_a$ and $\mathcal{D}_s$ 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.

IV. 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),

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

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),

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

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,

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

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,

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

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,

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

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

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

Finally, we use the Matthews' Correlation Coefficient (MCC) [98] 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,

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

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,

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

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 Section V 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 $M_{\theta_i;\varphi_i}$ with parameters θ_i and implementation information φ_i, we denote a set $C = \{M_{\theta_1;\varphi_1}, M_{\theta_2;\varphi_2}, \ldots, M_{\theta_i;\varphi_i}, \ldots, M_{\theta_{15};\varphi_{15}}\}$ 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 Section V. Now, mathematically, the optimization function $\mathbb{O}\left(M_{\theta_i;\varphi_i}\right)$ is given by (9), if we denote $\max(x)$ and $\min(x)$ by $\psi(x)$ and $\omega(x)$ respectively,

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

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

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

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

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

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 (50×50)
Xception [100]	✗
VGG16 [58]	✓
VGG19 [58]	✓
ResNet50 [22]	✓
ResNet50V2 [101]	✓
InceptionV3 [102]	✗
MobileNetV2 [103]	✓
DenseNet121 [63]	✓
NASNetMobile [104]	✗
NASNetLarge [104]	✗

TABLE 4. Description of abbreviations used in Table 5.

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

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

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

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

We remark that the range of $\mathbb{O}\left(M_{\theta_i;\varphi_i}\right)$ varies between $[0, \infty)$.

V. RESULTS AND DISCUSSION

In this section, we look at the implementation details of all the fifteen vanilla and SOTA+LeNet-5 CNNs and results achieved by them.[1] As discussed in Section IV, we also calculate important metrics such as precision P, sensitivity S_n, specificity S_p, F1-score F, RAC and BAC. Moreover, through selection of the best vanilla CNN (further denoted as C_{best}) using optimization function $\mathbb{O}\left(M_{\theta_i;\varphi_i}\right)$ 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.

The fifteen traditional CNN models $M_{\theta_i;\varphi_i}$ 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 $\boldsymbol{A} = \{A_1, A_2, A_3, \ldots, A_{15}\}$ and $\boldsymbol{L} = \{L_1, L_2, L_3, \ldots, L_{15}\}$ 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 $\omega(\boldsymbol{L})$ and MVA as $\psi(\boldsymbol{A})$. 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 $\tau(M_{\theta_i;\varphi_i})$ in (13), as shown at the bottom of the next page. We present results of vanilla CNNs in Table 5, in which we find C_{best} through the maximum value of $\mathbb{O}\left(M_{\theta_i;\varphi_i}\right)$. 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 $\mathbb{O}$, hence $C_{best} \leftarrow M_{\theta_{11};\varphi_{11}}$. 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) [104], 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

[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.

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.

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.

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 Section IV. 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).

Fig. 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 Section VI further.

VI. 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 Section II. 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 [13], [106]–[111]. 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 Section V how Table 8 portrays the trends of T_EA and T_EL to have a higher gradient of improvement for models

$$\mathbb{O}\left(M_{\theta_i;\varphi_i}\right) = \underbrace{argmax}_{M_{\theta_i};\varphi_i} \left(\frac{\psi\left(\alpha\left(M_{\theta_i;\varphi_i}\right)\right)}{\omega\left(\frac{\tau\left(M_{\theta_i;\varphi_i}\right)-\omega(\tau(C))}{\psi(\tau(C))-\omega(\tau(C))}\right) + \omega\left(-\frac{1}{n}\sum_{i=1}^{n} y_i \log\left(p\left(y_i\right)\right) + \left(1-y_i\right)\log\left(1-p\left(y_i\right)\right)\right)} \right) \quad \forall M_{\theta_i;\varphi_i} \in C \tag{13}$$

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

SNo	CL	AL	Regularizations				FD	KS	PS	S	LVBCEL	MVA	TT	$\mathbb{O}(M_{\theta_i;\varphi_i})$
			L1	L2	BN	DO								
1	2	4	×	×	×	×	(64,32)	(9,3)	(4,2)	(1,1)	0.3749	0.8363	860	0.608262
2	2	4	×	×	✓	✓	(64,32)	(9,3)	(4,2)	(2,1)	0.4649	0.8044	491	0.7765751
3	2	4	×	×	✓	×	(64,32)	(9,3)	(4,2)	(2,1)	0.4209	0.8121	606	0.7215132
4	2	4	×	×	×	✓	(64,32)	(9,3)	(4,2)	(2,1)	0.6893	0.7161	562	0.7761760
5	2	2	×	×	×	×	(64,32)	(9,3)	(4,2)	(2,2)	0.3867	0.8377	694	0.7017813
6	2	3	×	×	✓	✓	(64,32)	(9,3)	(4,2)	(2,1)	0.414	0.8158	696	0.6668834
7	3	4	×	×	✓	×	(128,64,32)	(2,1,1)	(2,1,1)	(2,2,1)	0.5284	0.7743	568	0.6512934
8	3	5	×	✓	×	×	(128,64,32)	(3,3,1)	(2,1,1)	(2,1,1)	0.3817	0.827	768	0.6487683
9	4	4	✓	×	×	✓	(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	✓	✓	×	×	(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	×	×	×	×	(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	×	×	×	×	(64,32,32,1 6)	(3,2,1,1)	(2,2,2, 2)	(2,1,1,1)	0.3718	0.8338	768	0.6592225
13	5	5	×	×	×	×	(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	×	×	×	×	(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	×	×	✓	✓	(64,32)	(9,3)	(4,2)	(2,1)	0.4649	0.8044	491	0.608262

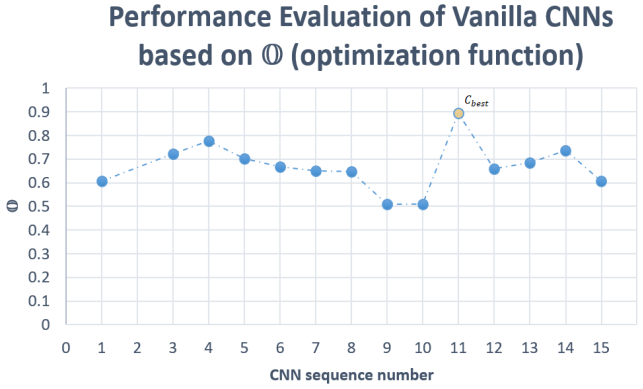

FIGURE 4. Performance of of C_{best} as opposed to other vanilla CNNs based on optimization function $\mathbb{O}\left(M_{\theta_i;\varphi_i}\right)$ based on Table 5.

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)

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 [109] 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 Section VI (A), and discuss a few other aspects in the following sub-sections.

A. 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 [97] 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 Section III (D). Let there be predictive learners $f_1(\cdot)$ and $f_2(\cdot)$ trained on $\mathcal{D}_a$ and the latter on $(\mathcal{D}_s + \mathcal{D}_a)$. 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 $\mathcal{D}_s$ 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 [109]. Intra-class variability, in other words, means a high variance between different 50 $\times$ 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 $\mathcal{D}_a$ has very specific features – which is not the case in IDC detection.

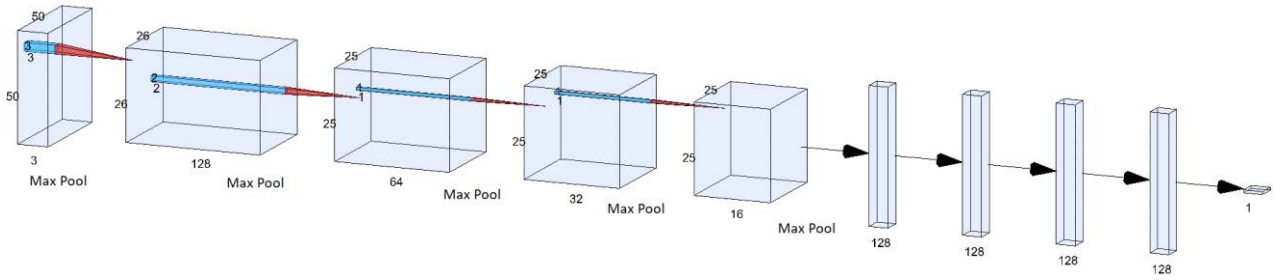

FIGURE 5. The NN-SVG [105] representation of C_best. The window size 50 × 50 diminishes as input progresses through the network. Full connection (right) is made before getting an output at the last layer with one neuron. This output ∈ [0, 1] denoting the probability of existence of IDC.

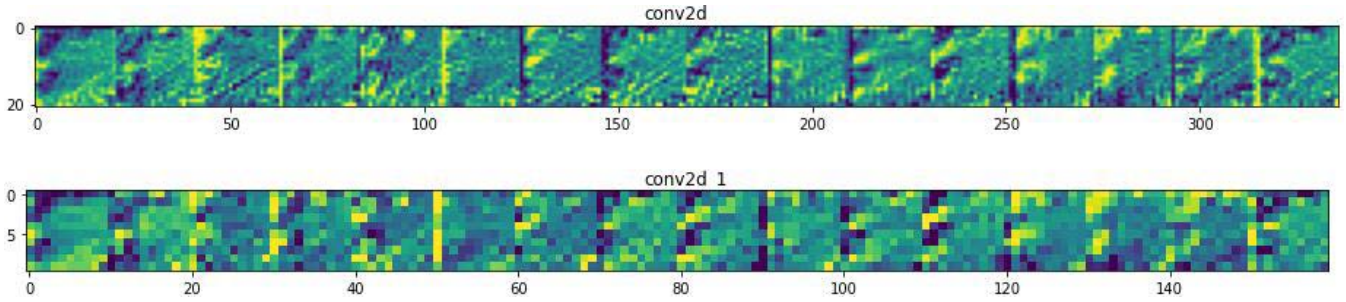

FIGURE 6. Convolutional neural network activations for LetNet-5.

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

There are many intra-class and lesser inter-class variations, making the effects of using transferred inference largely neutral.

B. 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 almost never be considered alone – the reasons being how they can vary in their behaviour when predicting positive or negative cases, as we shall see. Further, at times, false negatives may be more important to reduce.

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 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 tea and tel 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	0.3966
	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	0.4711
	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	0.8292
	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	0.7812
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	0.431
	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	0.5447
	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	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	0.7385
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	0.3272	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	0.6405
	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	0.8631
	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	0.7161
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	0.4371
	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	0.4835
	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	0.8017
	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	0.7701
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	0.3988
	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**	0.4939
	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	0.8281
	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**	0.7696
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	0.4078
	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**	0.5105
	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	0.8247
	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**	0.7406
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	0.4049
	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	0.4424
	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	0.8231
	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	0.7942	**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	0.3003
	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	0.4929
	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	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	0.7929

C. 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 Section IV 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 [112]. Nonetheless, there are works available in the literature that compare attained accuracy to of those achieved in the past [62], [106], [107]. 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 Section IV). 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

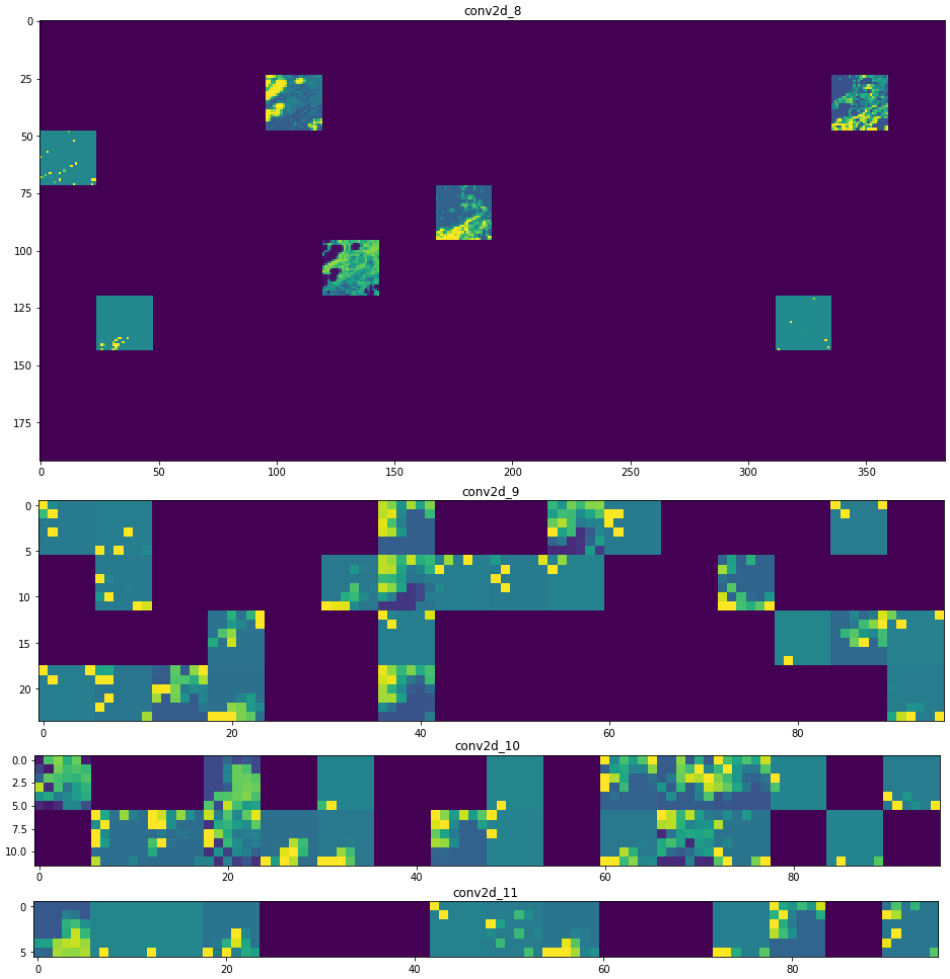

FIGURE 7. Convolutional neural activations for C_{best}.

TABLE 9. Comparison of models based on P, S_n, S_np, 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

accuracy as high as 95%, to which we argue that the dataset in consideration plays a major role.

The aspects discussed in sub-sections VI (A), VI (B), and VI (C) 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

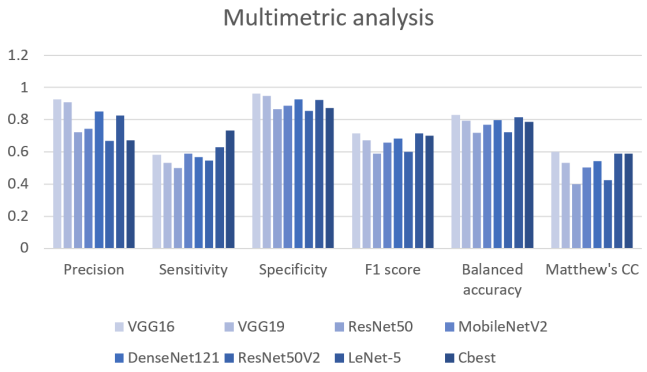

FIGURE 8. Multi-metric analysis of all models based on the performance metrics defined in Section IV.

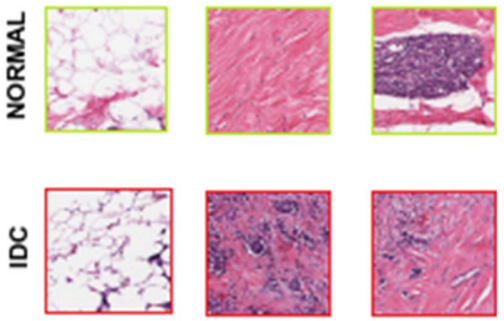

FIGURE 9. Intra-class variability in the dataset, eliminating the case of negative transfer. (Left): Patch having a majority of white regions. (Middle): Typical patch with different colour schemes for both the classes (as seen in Fig. 2). (Right): Mixture of the colour schemes in the same patch, making the detection process difficult.

TABLE 10. Summary of metric-based analysis.

Metrics	Top performing models
RAC	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	VGG16, VGG19, DenseNet121, LeNet-5, C_{best}
MCC	LeNet-5, C_{best}

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. Our work is different from that done in the literature because we try to analyze the applicability of transfer learning from a generic source domain in the detection of IDC. To the best of our knowledge, there has been no direct comparison of the performance of vanilla CNNs and larger pre-trained models from ImageNet for the detection of IDC. Through this work, we hope to inform the readers in the scenario where a choice is given to them – whether to use vanilla, trained-from-scratch CNNs or use an ImageNet pre-trained large CNN model.

VII. 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. To put these results into perspective, pre-training large CNN models over a generic source domain such as ImageNet does not provide a significant increase in various aforementioned performance metrics when compared to smaller, trained-from-scratch vanilla CNNs comprising only a few layers. Given the higher complexity and time involved in training larger models, it would almost be better to always use a vanilla CNN rather than use large models pre-trained on a generic source domain. 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 [113], deep belief networks [114],

autoencoders, and etc. to explore and open doors for more open comparisons between the efficacy of different varieties of techniques for IDC detection. Future work in detection of IDC is very important when deep CNNs area concerned because the large models, when trained from scratch, are very slow to train given the sheer amount of data needed to properly train neural networks to detect IDC. For this, faster methods that use the deep CNN methodology must be developed that can be trained faster and provide similar performance benefits. Lastly, 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.

CONFLICTS OF INTEREST/COMPETING INTERESTS

The authors hereby declare that there is no conflict of financial/personal interest or belief that could affect their objectivity.

APPENDIX A
L1 & L2 NORMALIZATION

Regularizations, such as the L1 and L2 regularization [115], [116] are used to decrease model complexity by penalizing significantly larger weights and reducing them to avoid overfitting. Given a loss function $L(\boldsymbol{x}, \boldsymbol{y})$ with input vector $\boldsymbol{x}$ and model outputs $\boldsymbol{y}$, we can define the model's predictions by a function $f : \boldsymbol{x} \rightarrow \boldsymbol{y}$ given as,

$$f\left(x_i\right) = \sum_{i=0}^{n} w_i x_i^i = w_0 + w_1 x_1 + w_2 x_2^2 + \cdots + w_n x_n^n \quad (14)$$

In (14), w_i denotes the weights where $f(x_i)$ takes n input variables. Hence, $L(x, y)$ can be given as,

$$L\left(\boldsymbol{x}, \boldsymbol{y}\right) = \sum_{i=1}^{n} \left(\left(y_i - f\left(x_i\right)\right)^2 = \sum_{i=1}^{n} \left(y_i - \sum_{i=0}^{n} w_i x_i^i\right)^2 \right. \quad (15)$$

L1 regularization (sometimes also called as Lasso regularization) introduces a regularization term to (15) with a regularization parameter λ that determines the extent of penalty applied on weights as shown in (16).

$$L\left(\boldsymbol{x}, \boldsymbol{y}\right) = \sum_{i=1}^{n} \left(y_i - \sum_{i=0}^{n} w_i x_i^i\right)^2 + \lambda \sum_{i=1}^{n} |w_i| \quad (16)$$

On the other hand, L2 regularization (also known as Ridge regularization) is given by (17) as,

$$L\left(\boldsymbol{x}, \boldsymbol{y}\right) = \sum_{i=1}^{n} \left(y_i - \sum_{i=0}^{n} w_i x_i^i\right)^2 + \lambda \sum_{i=1}^{n} w_i^2 \quad (17)$$

These regularizations have been used widely in literature and are a part of standard regularization mechanisms in place for deep neural networks [117]–[122].

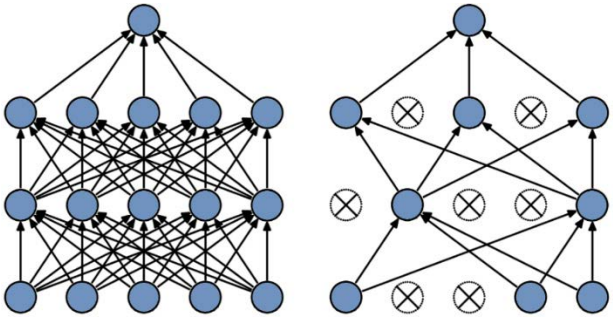

FIGURE 10. (Left): A normal deep neural network. (Right): Dropout applied on each layer (except the top-most layer). Image adapted from [24].

APPENDIX B
BATCH NORMALIZATION

Batch normalization (also termed as batch norm) proposed by Ioffe & Szegedy [123] speeds up convergence of a deep neural network by normalizing each d dimensions in each layer. Having two learnable parameters p and q, the technique takes a mini-batch $\mathcal{B}$ of size k, $\mathcal{B} \leftarrow \{x_{1,\ldots,k}\}$, and normalizes each input x_i into $\hat{x}_i$ to be plugged into a linear transformation to restore the representation power of the model.

$$\hat{x}_i = \frac{x_i - M_{\mathcal{B}}}{\sqrt{V_{\mathcal{B}}^2 + c}} \quad (18)$$

In (18), $M_{\mathcal{B}}$ is the mean of the values of the batch, $V_{\mathcal{B}}$ is the variance and c is a constant added for stability. The mean and the variance are defined by (19) and (20) respectively as,

$$M_{\mathcal{B}} = \frac{1}{k} \sum_{i=1}^{k} x_i \quad (19)$$

$$V_{\mathcal{B}} = \sqrt{\frac{1}{k} \sum_{i=1}^{k} (x_i - M_{\mathcal{B}})^2} \quad (20)$$

In final transformation step, the learnable parameters p and q are used as,

$$y_i = p\hat{x}_i + q \quad (21)$$

Batch normalization has been used in various applications [124]–[127].

APPENDIX C
DROPOUT

Dropout, proposed by Srivastava *et al.* [24] is another technique used in deep neural networks to reduce the extent of overfitting. It does this by randomly ignoring a percentage of units (or nodes) on the layer it is applied on. By "ignoring", it is implied that these nodes are deactivated and do not fire or propagate any values. The randomization is obtained by sampling the Bernoulli distribution. Fig. 10 shows the simplistic nature of dropout.

APPENDIX D
CLASS ACTIVATION MAP (CAM)

For an input image $\mathcal{I}$, we take the activation of unit p of the last convolutional layer spatially located at (x, y) and denote this activation by $f_p(x, y)$. Let $G(x)$ be the GAP function and hence when we plug $f_p(x, y)$ in $G(x)$ we get,

$$G\left(f_p\left(x, y\right)\right) = \sum_p f_p\left(x, y\right) = G_p \tag{22}$$

Taking weight w_p^c for some class c and unit p, the input to the softmax S^c is given by,

$$S^c = \sum_p w_p^c G_p \tag{23}$$

Thus, using (22) substituted in (23), we have the softmax class score,

$$\begin{aligned} S^c &= \sum_p w_p^c \sum_{x,y} f_p(x, y) \\ &= \sum_{x,y} \sum_p w_p^c f_p(x, y) \end{aligned} \tag{24}$$

The prominence of activation at spatial vicinity of (x, y) is characterized by a parameter $\beta(c; x, y)$ which decomposes (24) as,

$$S^c = \sum_{x,y} \beta(c; x, y) \tag{25}$$

From (24) and (25) we infer that,

$$\beta\left(c; x, y\right) = \sum_p w_p^c f_p(x, y) \tag{26}$$

The class prominence parameter defined in (26) allows to have a visual region-wise representation for the predicted class c.

REFERENCES

[1] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision—ECCV 2014* (Lecture Notes in Computer Science), vol. 8689, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014.

[2] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, A. Gertych, and R. S. Tan, "A deep convolutional neural network model to classify heartbeats," *Comput. Biol. Med.*, vol. 89, pp. 389–396, Oct. 2017, doi: 10.1016/j.compbiomed.2017.08.022.

[3] T. Guo, J. Dong, H. Li, and Y. Gao, "Simple convolutional neural network on image classification," in *Proc. IEEE 2nd Int. Conf. Big Data Anal. (ICBDA)*, Mar. 2017, pp. 721–724, doi: 10.1109/ICBDA.2017.8078730.

[4] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 39–48, doi: 10.1109/CVPR.2017.12.

[5] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4031–4039, doi: 10.1109/CVPR.2017.429.

[6] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1222–1228, doi: 10.1109/ICACCI.2017.8126009.

[7] U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, and R. S. Tan, "Deep convolutional neural network for the automated diagnosis of congestive heart failure using ECG signals," *Int. J. Speech Technol.*, vol. 49, no. 1, pp. 16–27, Jan. 2019, doi: 10.1007/s10489-018-1179-1.

[8] K. R. Kruthika and H. D. Maheshappa, "CBIR system using capsule networks and 3D CNN for Alzheimer's disease diagnosis," *Informat. Med. Unlocked*, vol. 14, pp. 59–68, 2019, doi: 10.1016/j.imu.2018.12.001.

[9] U. Raghavendra, H. Fujita, S. V. Bhandary, A. Gudigar, J. H. Tan, and U. R. Acharya, "Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images," *Inf. Sci.*, vol. 441, pp. 41–49, May 2018, doi: 10.1016/j.ins.2018.01.051.

[10] P. Rajpurkar, J. Irvin, A. Bagul, D. Ding, T. Duan, H. Mehta, B. Yang, K. Zhu, D. Laird, R. L. Ball, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng, "MURA: Large dataset for abnormality detection in musculoskeletal radiographs," 2017, *arXiv:1712.06957*.

[11] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng, "CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning," 2017, *arXiv:1711.05225*.

[12] P. Sharma, K. Bora, K. Kasugai, and B. Kumar Balabantaray, "Two stage classification with CNN for colorectal cancer detection," *Oncologie*, vol. 22, no. 3, pp. 129–145, 2020.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.

[14] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[15] W. Liu *et al.*, "SSD: Single shot MultiBox detector," in *Computer Vision—ECCV 2016* (Lecture Notes in Computer Science), vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[18] H. Chang, J. Han, C. Zhong, A. M. Snijders, and J.-H. Mao, "Unsupervised transfer learning via multi-scale convolutional sparse coding for biomedical applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1182–1194, May 2018, doi: 10.1109/TPAMI.2017.2656884.

[19] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 136–144.

[20] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning—ICANN 2018* (Lecture Notes in Computer Science), vol. 11141, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds. Cham, Switzerland: Springer, 2018.

[21] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2415–2421.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[23] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5353–5360, doi: 10.1109/CVPR.2015.7299173.

[24] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks: Training very deep networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2015.

[25] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Univ. Toronto, Toronto, ON, Canada, 2009.

[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[27] *Breast Cancer Facts & Figures 2019–2020*, American Cancer Society, Atlanta, GA, USA, 2020.

[28] S. K. Pal, S. K. Lau, L. Kruper, U. Nwoye, C. Garberoglio, R. K. Gupta, B. Paz, L. Vora, E. Guzman, A. Artinyan, and G. Somlo, "Papillary carcinoma of the breast: An overview," *Breast Cancer Res. Treatment*, vol. 122, no. 3, pp. 637–645, Aug. 2010, doi: 10.1007/s10549-010-0961-5.

[29] M. C. U. Cheang, M. Martin, T. O. Nielsen, A. Prat, and D. Voduc, "Defining breast cancer intrinsic subtypes by quantitative receptor expression," *Oncologist*, vol. 20, no. 5, pp. 474–482, May 2015, doi: 10.1634/theoncologist.2014-0372.
[30] A. Anand, H. Anand, S. S. Rautaray, M. Pandey, and M. K. Gourisaria, "Analysis and prediction of chronic heart diseases using machine learning classification models," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, pp. 8479–8487, Mar. 2020.
[31] S. Das, R. Sharma, M. K. Gourisaria, S. S. Rautaray, and M. Pandey, "Heart disease detection using core machine learning and deep learning techniques: A comparative study," *Int. J. Emerg. Technol.*, vol. 11, pp. 531–538, Jan. 2020.
[32] S. Dey, M. K. Gourisaria, S. S. Rautray, and M. Pandey, "Segmentation of nuclei in microscopy images across varied experimental systems," in *Intelligent Data Engineering and Analytics* (Advances in Intelligent Systems and Computing), vol. 1177, S. Satapathy, Y. D. Zhang, V. Bhateja, and R. Majhi, Eds. Singapore: Springer, 2021.
[33] G. M. Harshvardhan, M. K. Gourisaria, S. S. Rautaray, and M. Pandey, "Pneumonia detection using CNN through chest X-ray," *J. Eng. Sci. Technol.*, vol. 16, pp. 861–876, Jul. 2021.
[34] S. Mishra, M. Pandey, S. S. Rautaray, and M. K. Gourisaria, "A survey on big data analytical tools & techniques in health care sector," *Int. J. Emerg. Technol.*, vol. 11, pp. 554–560, May 2020.
[35] S. S. Rautaray, S. Dey, M. Pandey, and M. K. Gourisaria, "Nuclei segmentation in cell images using fully convolutional neural networks," *Int. J. Emerg. Technol.*, vol. 11, pp. 731–737, Sep. 2020.
[36] S. S. Rautaray, M. Pandey, M. K. Gourisaria, and R. S. Sharma Das, "Paddy crop disease prediction–A transfer learning technique," *Int. J. Recent Technol. Eng.*, vol. 8, pp. 1490–1495, Apr. 2020, doi: 10.35940/ijrte.f7782.038620.
[37] R. Sharma, "ECG classification using deep convolutional neural networks and data analysis," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 5788–5795, Aug. 2020.
[38] P. Johri, V. S. Saxena, and A. Kumar, "Rummage of machine learning algorithms in cancer diagnosis," *Int. J. E-Health Med. Commun.*, vol. 12, no. 1, pp. 1–15, Jan. 2021, doi: 10.4018/IJEHMC.2021010101.
[39] S. H. Shah, M. J. Iqbal, I. Ahmad, S. Khan, and J. J. P. C. Rodrigues, "Optimized gene selection and classification of cancer from microarray gene expression data using deep learning," *Neural Comput. Appl.*, to be published, doi: 10.1007/s00521-020-05367-8.
[40] G. Jee, G. Harshvardhan, and M. K. Gourisaria, "Juxtaposing inference capabilities of deep neural models over posteroanterior chest radiographs facilitating COVID-19 detection," *J. Interdiscipl. Math.*, vol. 24, no. 2, pp. 299–325, Feb. 2021, doi: 10.1080/09720502.2020.1838061.
[41] I. Saha, M. K. Gourisaria, and G. Harshvardhan, "Distinguishing pneumonia and COVID-19: Utilizing computer vision to mimic clinician efficacy," in *Proc. Int. Conf. Artif. Intell. Smart Syst. (ICAIS)*, Mar. 2021, pp. 834–841, doi: 10.1109/ICAIS50930.2021.9395961.
[42] R. Selvanambi and N. Jaisankar, "Healthcare: Prediction of breast cancer stage using social spider-inspired optimization algorithm," *Int. J. E-Health Med. Commun.*, vol. 10, no. 2, pp. 63–85, Apr. 2019, doi: 10.4018/IJEHMC.2019040104.
[43] A. Sahu, H. Gm, and M. K. Gourisaria, "A dual approach for credit card fraud detection using neural network and data mining techniques," in *Proc. IEEE 17th India Council Int. Conf. (INDICON)*, Dec. 2020, pp. 1–7, doi: 10.1109/INDICON49873.2020.9342462.
[44] G. Harshvardhan, M. K. Gourisaria, S. S. Rautaray, and M. Pandey, "UBMTR: Unsupervised Boltzmann machine-based time-aware recommendation system," *J. King Saud Univ.-Comput. Inf. Sci.*, to be published, doi: 10.1016/j.jksuci.2021.01.017.
[45] I. Ahmed, M. Ahmad, J. J. P. C. Rodrigues, G. Jeon, and S. Din, "A deep learning-based social distance monitoring framework for COVID-19," *Sustain. Cities Soc.*, vol. 65, Feb. 2021, Art. no. 102571.
[46] A. Sahu, G. M. Harshvardhan, M. K. Gourisaria, S. S. Rautaray, and M. Pandey, "Cardiovascular risk assessment using data mining inferencing and feature engineering techniques," *Int. J. Inf. Technol.*, vol. 13, pp. 2011–2023, Apr. 2021.
[47] A. Cruz-Roa, A. Basavanhally, F. González, H. Gilmore, and M. Feldman, "Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks," *Proc. SPIE*, vol. 9041, Mar. 2014, Art. no. 904103, doi: 10.1117/12.2043872.
[48] A. Janowczyk and A. Madabhushi, "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases," *J. Pathol. Informat.*, vol. 7, no. 1, p. 29, 2016, doi: 10.4103/2153-3539.186902.
[49] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep learning for identifying metastatic breast cancer," 2016, *arXiv:1606.05718*.
[50] J. Caicedo, "A prototype system to archive and retrieve histopathology images by content," M.S. thesis, Nat. Univ. Colombia, Bogotá, Colombia, 2008.
[51] J. Han and K.-K. Ma, "Fuzzy color histogram and its use in color image retrieval," *IEEE Trans. Image Process.*, vol. 11, no. 8, pp. 944–952, Aug. 2002, doi: 10.1109/TIP.2002.801585.
[52] M. Lux and S. A. Chatzichristofis, "Lire: Lucene image retrieval: An extensible Java CBIR library," in *Proc. ACM Int. Conf. Multimedia, Co-Located Symp. Workshops*, 2008, pp. 1085–1088, doi: 10.1145/1459359.1459577.
[53] A. Basavanhally, S. Ganesan, M. Feldman, N. Shih, C. Mies, J. Tomaszewski, and A. Madabhushi, "Multi-field-of-view framework for distinguishing tumor grade in ER+ breast cancer from entire histopathology slides," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 8, pp. 2089–2099, Aug. 2013, doi: 10.1109/TBME.2013.2245129.
[54] D. S. Messing, P. Van Beek, and J. H. Errico, "The MPEG-7 colour structure descriptor: Image description using colour and local spatial information," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2001, pp. 670–673, doi: 10.1109/icip.2001.959134.
[55] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006, doi: 10.1109/TPAMI.2006.244.
[56] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
[57] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 84–90, doi: 10.1061/(ASCE)GT.1943-5606.0001284.
[58] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015.
[59] D. Wang, C. Otto, and A. K. Jain, "Face search at scale," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1122–1136, Jun. 2017, doi: 10.1109/TPAMI.2016.2582166.
[60] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979, doi: 10.1109/TSMC.1979.4310076.
[61] H. Alghodhaifi, A. Alghodhaifi, and M. Alghodhaifi, "Predicting invasive ductal carcinoma in breast histology images using convolutional neural network," in *Proc. IEEE Nat. Aerosp. Electron. Conf. (NAECON)*, Jul. 2019, pp. 374–378, doi: 10.1109/NAECON46414.2019.9057822.
[62] Y. Celik, M. Talo, O. Yildirim, M. Karabatak, and U. R. Acharya, "Automated invasive ductal carcinoma detection based using deep transfer learning with whole-slide images," *Pattern Recognit. Lett.*, vol. 133, pp. 232–239, May 2020, doi: 10.1016/j.patrec.2020.03.011.
[63] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.
[64] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 464–472, doi: 10.1109/WACV.2017.58.
[65] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. McBride, and W. Sieh, "Deep learning to improve breast cancer detection on screening mammography," *Sci. Rep.*, vol. 9, no. 1, p. 12495, Dec. 2019, doi: 10.1038/s41598-019-48995-4.
[66] M. M. Dundar, S. Badve, G. Bilgin, V. Raykar, R. Jain, O. Sertel, and M. N. Gurcan, "Computerized classification of intraductal breast lesions using histopathological images," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 7, pp. 1977–1984, Jul. 2011, doi: 10.1109/TBME.2011.2110648.
[67] M. M. Dundar, S. Badve, V. C. Raykar, R. K. Jain, O. Sertel, and M. N. Gurca, "A multiple instance learning approach toward optimal classification of pathology slides," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2732–2735, doi: 10.1109/ICPR.2010.669.
[68] S. R. Sain and V. N. Vapnik, "The nature of statistical learning theory," *Technometrics*, vol. 38, no. 4, p. 409, 1996, doi: 10.2307/1271324.

[69] B. Gecer, S. Aksoy, E. Mercan, L. G. Shapiro, D. L. Weaver, and J. G. Elmore, "Detection and classification of cancer in whole slide breast histopathology images using deep convolutional networks," *Pattern Recognit.*, vol. 84, pp. 345–356, Dec. 2018, doi: 10.1016/j.patcog.2018.07.022.

[70] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440, doi: 10.1109/CVPR.2015.7298965.

[71] S. B. Yengec Tasdemir, K. Tasdemir, and Z. Aydin, "ROI detection in mammogram images using wavelet-based Haralick and HOG features," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 105–109, doi: 10.1109/ICMLA.2018.00023.

[72] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893, doi: 10.1109/CVPR.2005.177.

[73] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973, doi: 10.1109/TSMC.1973.4309314.

[74] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989, doi: 10.1109/34.192463.

[75] N. Brancati, G. De Pietro, M. Frucci, and D. Riccio, "A deep learning approach for breast invasive ductal carcinoma detection and lymphoma multi-classification in histological images," *IEEE Access*, vol. 7, pp. 44709–44720, 2019, doi: 10.1109/ACCESS.2019.2908724.

[76] A. A. Cruz-Roa, J. E. A. Ovalle, A. Madabhushi, and F. A. G. Osorio, "A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2013* (Lecture Notes in Computer Science), vol. 8150, K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, Eds. Berlin, Germany: Springer, 2013.

[77] Y. Feng, L. Zhang, and Z. Yi, "Breast cancer cell nuclei classification in histopathology images using deep neural networks," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 13, no. 2, pp. 179–191, Feb. 2018, doi: 10.1007/s11548-017-1663-9.

[78] L. Hou, V. Nguyen, A. B. Kanevsky, D. Samaras, T. M. Kurc, T. Zhao, R. R. Gupta, Y. Gao, W. Chen, D. Foran, and J. H. Saltz, "Sparse autoencoder for unsupervised nucleus detection and representation in histopathology images," *Pattern Recognit.*, vol. 86, pp. 188–200, Feb. 2019, doi: 10.1016/j.patcog.2018.09.007.

[79] J. Xu, L. Xiang, Q. Liu, H. Gilmore, J. Wu, J. Tang, and A. Madabhushi, "Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images," *IEEE Trans. Med. Imag.*, vol. 35, no. 1, pp. 119–130, Jan. 2016, doi: 10.1109/TMI.2015.2458702.

[80] T. Minh Quan, D. G. C. Hildebrand, and W.-K. Jeong, "FusionNet: A deep fully residual convolutional neural network for image segmentation in connectomics," 2016, *arXiv:1612.05360*.

[81] H. Gm, M. K. Gourisaria, M. Pandey, and S. S. Rautaray, "A comprehensive survey and analysis of generative models in machine learning," *Comput. Sci. Rev.*, vol. 38, Nov. 2020, Art. no. 100285, doi: 10.1016/j.cosrev.2020.100285.

[82] S. Belciug, F. Gorunescu, M. Gorunescu, and A. B. M. Salem, "Assessing performances of unsupervised and supervised neural networks in breast cancer detection," in *Proc. 7th Int. Conf. Inform. Syst.*, Mar. 2010, pp. 1–8.

[83] T. Kohonen, "Exploration of very large databases by self-organizing maps," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Jun. 1997, pp. PL1–PL6, doi: 10.1109/ICNN.1997.611622.

[84] H. Fatakdawala, J. Xu, and A. Basavanhally, "Expectation–maximization-driven geodesic active contour with overlap resolution (EMaGACOR): Application to lymphocyte segmentation on breast cancer histopathology," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 7, pp. 1676–1689, Jul. 2010, doi: 10.1109/TBME.2010.2041232.

[85] H. Chang, J. Han, A. Borowsky, L. Loss, J. W. Gray, P. T. Spellman, and B. Parvin, "Invariant delineation of nuclear architecture in glioblastoma multiforme for clinical and molecular association," *IEEE Trans. Med. Imag.*, vol. 32, no. 4, pp. 670–682, Apr. 2013, doi: 10.1109/TMI.2012.2231420.

[86] A. C. Ruifrok and D. A. Johnston, "Quantification of histochemical staining by color deconvolution," *Anal. Quant. Cytolol. Histol.*, vol. 23, no. 4, pp. 291–299, 2001.

[87] A. Mikolajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *Proc. Int. Interdiscipl. PhD Workshop (IIPhDW)*, May 2018, pp. 117–122, doi: 10.1109/IIPHDW.2018.8388338.

[88] C. Wigington, S. Stewart, B. Davis, B. Barrett, B. Price, and S. Cohen, "Data augmentation for recognition of handwritten words and lines using a CNN-LSTM network," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 639–645, doi: 10.1109/ICDAR.2017.110.

[89] A. C. Koivunen and A. B. Kostinski, "The feasibility of data whitening to improve performance of weather radar," *J. Appl. Meteorol.*, vol. 38, no. 6, pp. 741–749, 1999, doi: 10.1175/1520-0450(1999)038<0741:TFODWT>2.0.CO;2.

[90] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An all-in-one convolutional neural network for face analysis," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Washington, DC, USA May 2017, pp. 17–24, doi: 10.1109/FG.2017.137.

[91] X. Yin and X. Liu, "Multi-task convolutional neural network for pose-invariant face recognition," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 964–975, Feb. 2018, doi: 10.1109/TIP.2017.2765830.

[92] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 579–583, doi: 10.1109/ACPR.2015.7486569.

[93] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013, doi: 10.1109/TPAMI.2012.59.

[94] P. O. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 82–90.

[95] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929, doi: 10.1109/CVPR.2016.319.

[96] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.

[97] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, p. 9, May 2016, doi: 10.1186/s40537-016-0043-6.

[98] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochim. et Biophys. Acta (BBA)-Protein Struct.*, vol. 405, no. 2, pp. 442–451, 1975, doi: 10.1016/0005-2795(75)90109-9.

[99] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.

[100] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision—ECCV 2016* (Lecture Notes in Computer Science), vol. 9908, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016.

[101] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[102] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[103] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710, doi: 10.1109/CVPR.2018.00907.

[104] A. Fred Agarap, "Deep learning using rectified linear units (ReLU)," 2018, *arXiv:1803.08375*.

[105] A. LeNail, "NN-SVG: Publication-ready neural network architecture schematics," *J. Open Source Softw.*, vol. 4, no. 33, p. 747, Jan. 2019, doi: 10.21105/joss.00747.

[106] E. Deniz, A. Şengür, Z. Kadiroğlu, Y. Guo, V. Bajaj, and Ü. Budak, "Transfer learning based histopathologic image classification for breast cancer detection," *Health Inf. Sci. Syst.*, vol. 6, no. 1, Dec. 2018, doi: 10.1007/s13755-018-0057-x.

[107] S. Guan and M. Loew, "Breast cancer detection using transfer learning in convolutional neural networks," in *Proc. IEEE Appl. Imag. Pattern Recognit. Workshop (AIPR)*, Oct. 2017, pp. 1–8, doi: 10.1109/AIPR.2017.8457948.

[108] S. Khan, N. Islam, Z. Jan, I. Ud Din, and J. J. P. C. Rodrigues, "A novel deep learning based framework for the detection and classification of breast cancer using transfer learning," *Pattern Recognit. Lett.*, vol. 125, pp. 1–6, Jul. 2019, doi: 10.1016/j.patrec.2019.03.022.
[109] B. Kieffer, M. Babaie, S. Kalra, and H. R. Tizhoosh, "Convolutional neural networks for histopathology image classification: Training vs. Using pre-trained networks," in *Proc. 7th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Nov. 2017, pp. 1–6, doi: 10.1109/IPTA.2017.8310149.
[110] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "A dataset for breast cancer histopathological image classification," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 7, pp. 1455–1462, Jul. 2016, doi: 10.1109/TBME.2015.2496264.
[111] M. Talo, "Convolutional neural networks for multi-class histopathology image classification," 2019, *arXiv:1903.10035*.
[112] D. Chicco, "Ten quick tips for machine learning in computational biology," *BioData Mining*, vol. 10, no. 1, p. 35, Dec. 2017, doi: 10.1186/s13040-017-0155-3.
[113] R. Salakhutdinov and G. Hinton, "Deep Boltzmann machines," *J. Mach. Learn. Res.*, vol. 5, no. 2, pp. 448–455, Jan. 2009.
[114] R. M. Neal, "Connectionist learning of belief networks," *Artif. Intell.*, vol. 56, no. 1, pp. 71–113, 1992, doi: 10.1016/0004-3702(92)90065-6.
[115] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for L1 regularization: A comparative study and two new approaches," in *Machine Learning: ECML 2007* (Lecture Notes in Computer Science), vol. 4701, J. N. Kok, J. Koronacki, R. L. Mantaras, S. Matwin, D. Mladenič, and A. Skowron, Eds. Berlin, Germany: Springer, 2007.
[116] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 regularization for learning kernels," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 109–116.
[117] R. Ma, J. Miao, L. Niu, and P. Zhang, "Transformed ℓ_1 regularization for learning sparse deep neural networks," *Neural Netw.*, vol. 119, pp. 286–298, Nov. 2019, doi: 10.1016/j.neunet.2019.08.015.
[118] X. Ni, L. Fang, and H. Huttunen, "AdaptiveReID: Adaptive L2 regularization in person re-identification," 2020, *arXiv:2007.07875*.
[119] X. Qian, H. Huang, X. Chen, and T. Huang, "Efficient construction of sparse radial basis function neural networks using L_1-regularization," *Neural Netw.*, vol. 94, pp. 239–254, Oct. 2017, doi: 10.1016/j.neunet.2017.07.004.
[120] C. Yang, Z. Yang, A. M. Khattak, L. Yang, W. Zhang, W. Gao, and M. Wang, "Structured pruning of convolutional neural networks via l1 regularization," *IEEE Access*, vol. 7, pp. 106385–106394, 2019, doi: 10.1109/ACCESS.2019.2933032.
[121] S. Zeng, B. Zhang, Y. Zhang, and J. Gou, "Collaboratively weighting deep and classic representation via l2 regularization for image classification," 2018, *arXiv:1802.07589*.
[122] Y. Zhai, W. Deng, Y. Xu, Q. Ke, J. Gan, B. Sun, J. Zeng, and V. Piuri, "Robust SAR automatic target recognition based on transferred MS-CNN with L2-regularization," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–13, Nov. 2019, doi: 10.1155/2019/9140167.
[123] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[124] A. Atanov, A. Ashukha, D. Molchanov, K. Neklyudov, and D. Vetrov, "Uncertainty estimation via stochastic batch normalization," in *Advances in Neural Networks—ISNN 2019* (Lecture Notes in Computer Science), vol. 11554, H. Lu, H. Tang, and Z. Wang, Eds. Cham, Switzerland: Springer, 2019.
[125] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognit.*, vol. 80, pp. 109–117, Aug. 2018, doi: 10.1016/j.patcog.2018.03.005.
[126] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville, "Recurrent batch normalization," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2017.
[127] S.-H. Wang, K. Muhammad, J. Hong, A. K. Sangaiah, and Y.-D. Zhang, "Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization," *Neural Comput. Appl.*, vol. 32, no. 3, pp. 665–680, Feb. 2020, doi: 10.1007/s00521-018-3924-0.
[128] S. Chandra, M. K. Gourisaria, H. Gm, D. Konar, X. Gao, T. Wang, and M. Xu, "Prolificacy assessment of spermatozoan via state-of-the-art deep learning frameworks," *IEEE Access*, vol. 10, pp. 13715–13727, 2022.

G. M. HARSHVARDHAN is currently pursuing the B.Tech. degree with KIIT Deemed to be University, Odisha, India. He has authored over 20 papers which have been published in reputed conferences and SCI/ESCI journals. His research interests include unsupervised and self-supervised deep learning methods, clinical applications of machine learning for medicine, generative models in machine learning, recommender systems for content delivery, energy-efficient task scheduling in cloud computing, topic modeling in natural language processing using latent Dirichlet allocation, and modeling high dimensional clinical data in medicine.

AANCHAL SAHU is currently pursuing the Bachelor of Technology degree in computer science and communication engineering with KIIT Deemed to be University, Odisha, India. In the past, she has interned as a Technical Content Writer at Heu Technologies Private Ltd., HYD, India. She has also worked at a Texas-based software company HighRadius as a Product Engineer Intern. Her research interests include intersection of security, stability, and society. She has published papers covering various aspects of security and stability, like artificial intelligence for healthcare, financial security, and the environment. She has published five research papers and articles over the past year.

MAHENDRA KUMAR GOURISARIA (Member, IEEE) received the master's degree in computer application from Indira Gandhi National Open University, New Delhi, and the M.Tech. degree in computer science and engineering from the Biju Patnaik University of Technology, Rourkela. He is currently pursuing the Ph.D. degree with KIIT Deemed to be University, Bhubaneswar, Odisha. He is presently working as an Assistant Professor with the School of Computer Engineering, KIIT Deemed to be University. He has an experience of more than 18 years in academia and seven years in research. He has published more than 60 research papers in different book chapters, international journals, and conferences of repute. He has also served as the organizing committees members for various conferences and workshops. His research interests include cloud computing, machine learning, deep learning, data mining, soft computing, and internet and web technology. He is a member of IAENG and UACEE, and a Life Member of ISTE, CSI, and ISCA.

PRADEEP KUMAR SINGH (Senior Member, IEEE) is currently working as a Professor and the Head with the Department of CS, KIET Group of Institutions, Delhi-NCR, Ghaziabad, Uttar Pradesh, India. He has published nearly 125 research papers. He has received three sponsored research projects grant worth Rs. 25 Lakhs. He has edited a total 16 books from Springer and Elsevier and also edited several special issues for SCI and SCIE journals from Elsevier and IGI Global. He has Google scholar citations 1501, H-index 25, and i-10-index 48. He is an Associate Editor of *IJISMD* (indexed by Scopus & Web of Science), *IJAEC* (IGI Global USA, SPY, and Wiley), and *IJISC* from Romania. He is recently appointed as a Section Editor of *Discover IoT* (Springer) journal.

WEI-CHIANG HONG (Senior Member, IEEE) is currently a Professor with the Department of Information Management, Asia Eastern University of Science and Technology, Taiwan. He has Google scholar citations 7302, H-index 46, and i-10-index 80 in his account. His research interests include computational intelligence (neural networks and evolutionary computation), application of forecasting technology (ARIMA, support vector regression, and chaos theory), and machine learning algorithms. He serves as the Program Committee Member for various international conferences including premium ones such as IEEE CEC, IEEE CIS, IEEE ICNSC, IEEE SMC, IEEE CASE, and IEEE SMCia. In May 2012, his paper had been evaluated as Top Cited Article 2007–2011 by Elsevier Publisher (The Netherlands). In September 2012, once again, his paper had been indexed in ISI Essential Science Indicator database as highly cited papers, in the meanwhile, he also had been awarded as the Model Teacher Award by the Taiwan Private Education Association. He is a Senior Member of IIE. He is indexed in the list of Who's Who in the World (25th–30th Editions), Who's Who in Asia (2nd Edition), and Who's Who in Science and Engineering (10th and 11th Editions). He is currently appointed as the Editor-in-Chief of the *International Journal of Applied Evolutionary Computation*, in addition, he serves as a Guest Editor for *Energies*, and is appointed as an Editorial Board Member of *Applied Soft Computing*.

VIJANDER SINGH received the Ph.D. degree from Banasthali University, Banasthali, India, in April 2017, and the M.Tech. degree (Hons.) from Rajasthan Technical University, in 2019. He also qualified NET and GATE examinations, in 2012 and 2018, respectively. He is currently working as a Postdoctoral Fellow at the Cyber-Physical Systems Laboratory, Department of ICT and Natural Sciences, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology, Norway, and an Associate Professor with the Department of Computer Science and Engineering, Manipal University Jaipur, Jaipur, India. He has published more than 40 journal articles, 15 conference papers, ten book chapters, and two edited books. His research interests include machine learning, deep learning, precision agriculture, and networking. He is handling journals of international repute as a guest editor. He has organized several international conferences, FDPs, and workshops, as core team member of organizing committee.

BUNIL KUMAR BALABANTARAY (Member, IEEE) received the B.Tech. and M.Tech. degrees in computer science and engineering from BPUT, Rourkela, India, in 2005 and 2010, respectively, and the Ph.D. degree from the National Institute of Technology, Rourkela, India, in 2017. He is currently working as an Assistant Professor with the Department of Computer Science and Engineering, National Institute of Technology Meghalaya, India. He is having more than 15 years of teaching and research experience. After joining at NIT Meghalaya, he has developed the Robotics and Mechatronics Laboratory, the Machine Intelligence Laboratory, and the Robotics Club. Under his leadership, the Machine Intelligence Laboratory is having collaboration with NVIDIA, Robotics Laboratory with IIT Delhi and five industries, Robotics Club is the training partner of the Government of Meghalaya. He is having more than 40 research articles in journals and international conferences of repute. His research interests include robotic vision, computer vision and biomedical image processing, and machine intelligence. He is associated with more than 20 reputed international conferences as a technical program committee member and an advisor. He has conducted more than 30 workshops/FDPs in his research area. He is a member of IEEE Society for Engineering in Medicine and Biology and a Life Member of ISTE and CSI. He is associated with more than ten international journals of repute as a reviewer. He is also the Guest Editor of *Multimedia Tools and Application* and the *International Journal of Computer Applications in Technology* (Inderscience Publishers).