Automatic Detection of Osteosarcoma Based on Integrated Features and Feature Selection Using Binary Arithmetic Optimization Algorithm

Priti Bansal (bansalpriti79@gmail.com)
Netaji Subhas Institute of Technology https://orcid.org/0000-0002-0806-7125

Kshitiz Gehlot
Netaji Subhas Institute of Technology: Netaji Subhas University of Technology

Abhishek Singhal
Netaji Subhas Institute of Technology: Netaji Subhas University of Technology

Research Article

Keywords: Osteosarcoma, Whole Slide Images, Arithmetic Optimization Algorithm, Integrated features, Deep Learning Model.

Posted Date: June 22nd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-525421/v1

Version of Record: A version of this preprint was published at Multimedia Tools and Applications on February 7th, 2022. See the published version at https://doi.org/10.1007/s11042-022-11949-6.

Automatic Detection of Osteosarcoma Based on Integrated Features and Feature Selection Using Binary Arithmetic Optimization Algorithm

Priti Bansal, , *Kshitiz Gehlot, *Abhishek Singhal

bansalpriti79@gmail.com, abhisheksinghal.18199@gmail.com, kshitizg99@gmail.com

Department of Information Technology,
Netaji Subhas University of Technology, Dwarka, New Delhi, India
*Department of Information Technology,
Netaji Subhas Institute of Technology, Dwarka, New Delhi, India

Corresponding Author: Priti Bansal, bansalpriti79@gmail.com, +91-9811614388

Abstract Osteosarcoma is one of the most common malignant bone tumor mostly found in children and teenagers. Manual detection of osteosarcoma requires expertise and is a labour-intensive process. If detected on time, the mortality rate can be reduced. With the advent of new technologies, automatic detection systems are used to analyse and classify images obtained from different sources. Here, we propose an automatic detection system Integrated Features-Feature Selection Model for Classification (IF-FSM-C) that detect osteosarcoma from the high-resolution whole slide images (WSIs). The novelty of the proposed approach is the use of integrated features obtained by fusion of features extracted using traditional handcrafted feature extraction techniques and deep learning models. It is quite possible that the integrated features may contain some redundant and irrelevant features which may unnecessarily increases the computation time and leads to wastage of resources. To avoid this, we perform feature selection (FS) before giving the integrated features to the classifier. To perform feature selection, we propose two binary variants of recently proposed Arithmetic Optimization Algorithm (AOA) known as BAOA-S and BAOA-V. The selected features are given to a classifier that classifies the WSIs into Viable tumor (VT), Non-viable tumor (NVT) and non-tumor (NT). Experiments are performed and the results prove the superiority of the proposed IF-FSM-C that uses integrated features and feature selection in classifying WSIs as compared to the classifiers which use handcrafted or deep learning features alone as well as state-of-the-art methods for osteosarcoma detection.

Keywords Osteosarcoma**,** Whole Slide Images, Arithmetic Optimization Algorithm, Integrated features, Deep Learning Model.

---

* Co-authors contributed equally to this research

1. Introduction

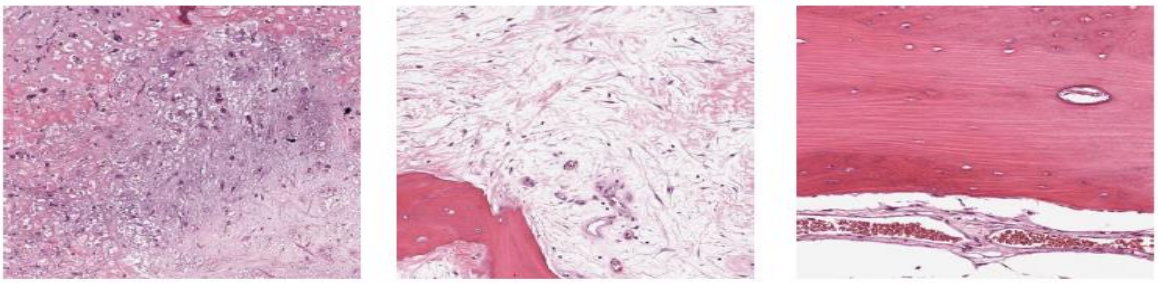
Osteogenic Sarcoma or Osteosarcoma is a form of bone cancer that typically starts its growth in long bones of legs and arms (Lindsey et al. 2017). However, it can grow in any bone in the bone cells and in rare cases can affect the soft tissues around the bones. It mainly affects people in their teens or young adults, but may infect other age groups as well. Osteosarcoma is the most usual form of malignant tumour occurring in youth and average age of diagnosis for the disease is 15 years. Till late adolescence i.e., 15-19 years, the boys and girls are prone to get infected almost equally and beyond 15 years of age boys are more generally affected and susceptible to be attacked in the bone cells. The cause of osteosarcoma is unknown and it typically do not run-in families, but much like many other forms of cancers it occurs due to a fault in the gene which can cause this disease. Also, it has been linked to cases wherein there is a family history of familial retinoblastoma, a childhood cancer of the eye (Kleinerman et al. 2005). Children with familial retinoblastoma have a high risk for osteosarcoma in adolescence. Classic symptoms of the disease include pain in the affected bone, with initially a lower degree of pain and becoming excruciating with time. One of the other symptoms is bone fracture, as the affected bone weakens due to infection and losing its strength.

The overall 5-year survival rate in case of osteosarcoma is 68% and 67% for children ages 0 to 14 years and teens ages 15 to 19 years respectively (Seigel et al. 2021). If osteosarcoma is detected and treated timely before it has spread outside the area where it started, the general 5-year survival rate for people of all ages is 74% (Mirabello et al. 2009). If it has spread outside of the bones and into surrounding tissues or organs and/or the regional lymph nodes, the 5-year survival rate decreases to 66% which further declines to 27% if spread to distant parts of the body (Mirabello et al. 2009). Looking at the low survival rate if not diagnosed on time, it can be inferred that it is important to diagnose and treat this disease as quickly as possible as early detection improves the chances of survival.

The task of osteosarcoma detection requires a lot of expertise, and is tiresome and time-consuming if done manually. There are some dissimilarities between the afflicted cells and the normal bone cells, with some striking characteristics differentiating the two. The cancerous bone cells are more densely populated with their nuclei being close to each other as compared to normal cells. The cancerous bone cells exhibit darker shades of blue and purple. Similar differences are also observed in texture and other properties. These cells are normally identified using Hematoxylin and Eosin (H&E) staining of the tissue taken from the affected areas (Goode et al. 2013). After staining, the specimens are put under a microscope for analysis by pathologists, by mounting them on glass slides. With the advent of new technology and faster processing systems, medical field has switched to automated detection systems which have the capability to predict the possibility of cancer being benign or malignant with some certainty. These systems have a general architecture of a feature extractor followed by a classifier which takes the features as input. Some of the systems may also include a pre-processing stage before the feature extractor, which is required to improve the quality of the images by cleaning the data being fed into the model. This pre-processing may have techniques such as: Noise Removal Filters, Contrast Enhancement, etc. All these techniques benefit in cases when the dataset is of poor quality and has blurred images as well. However, in this research the dataset used has sufficiently high-quality Whole Slide Images (WSIs) (Leavey et al. 2019), which are the digital form of conventional glass slides, and doesn't require any pre-processing. Here, WSIs are classified into Viable Tumor (VT), Non-Viable Tumor (NVT) and Non-Tumor (NT) as shown in Fig. 1.

Due to the presence of background and unwanted information such as: Muscles, Ink-markings, folded or severed tissues or blurred regions, it becomes of utmost need to remove such information before the data is fed into the feature extractor. For this task, segmentation is used which can assign labels to each pixel based on the properties (color intensity, texture or position), where each label represents a different region and the correlation between different regions is minimum. This helps us to differentiate between foreground and background attributes in the image and remove the unwanted information.

There are large number of segmentation methods used in literature each having a different basis for segmenting the regions. Some basic techniques are: edge-based (Wang and Oliensis 2010), clustering-based (Qureshi and Ahamad 2018)



(a) Viable Tumor (VT) (b) Non-Viable Tumor (NVT) (c) Non-Tumor (NT)

Fig. 1 Sample Images from the Dataset

thresholding techniques (Otsu 1979), watershed (Beucher and Meyer 1993) and region-based (Adams and Bischof 1994, Wu et al. 2015). Edge-based segmentation involves the application of an edge filter, followed by categorization of the pixels to edge and non-edge by the use of differential operators – Prewitt Operator, Sobel Operator etc., with assigning non-edge pixels to the same category. Clustering-based techniques such as K-means is a famous algorithm used for the purpose of segmentation that clusters the image into K clusters, with each cluster having similar characteristics and attributes. Thresholding such as Otsu thresholding, converts the image into discrete levels with each level indicating a different class of pixels. Watershed is another category of segmentation which makes use of topological interpretation of image, with the intensity levels of pixels treated as topology or elevation. Thus, in these we start by flooding the basins or areas having minima, and continue this process till each basin is full, and finally each basin is a separate region. It is quite similar to flood fill algorithms. Finally, we have the region-based segmentation techniques which can be seed-based and non-seed based. This categorization is done on the basis of the initial points taken as regions to start the process of segmentation.

The next step in automatic detection systems is feature extraction which can be handcrafted (HC) or deep learning (DL) features. Feature extraction is the process of reducing the dimensionality of the input data to represent the data in a more interpretable format which can be easily analysed (Samet 2006; Ding et al. 2002). These techniques are typically used for extraction of HC features which are specific properties of the image decided manually based on the attributes of the target space. HC features have been widely used by researchers due to their simplicity of extraction especially in those cases where the data sets are small. These features can be decided by consultation from experts in the field under consideration. HC features do not require any form of training set for their derivation and can be more easily visualized as compared to other features. At the same time, with the use of HC features we can assign different weights to each feature based on the role they play in carrying out the targeted task.

However, in case of complex images these features become difficult to be decided upon and thus another form of feature extraction technique is employed namely deep Learning models (DLMs) as feature extractors. DLMs have gained much importance in the last decade due to the recent developments seen in the field of computational power with the introduction of faster and compact computers, enabling the professionals to train deeper networks with ease and in lesser time. Typically, in this case we employ Convolutional Neural Networks (CNNs) as the DLM for the task of feature extraction. These models have the ability to automatically learn the features from the input image, but require a large training set with high variation for sufficient quality of attribute derivation. They have the ability to obtain higher number of low-level features which can describe the image in greater detail, thus helping in further analysis. DLM provides the benefit of no significant pre-processing requirement as they have the ability to carry out the same task automatically. But the downside of such models are their high computational and data requirements. Thus, as can be seen that the features extracted from both techniques HC and DLM have their pros and cons. Therefore, to take the advantage of the strength of both the methods, we propose a model IF-FSM-C that uses integrated features obtained by fusion of HC and DL features for classification.

Feature extraction is an essential step in any image processing step as the performance of a classifier is highly dependent on the features. However, not all features extracted from HC and DLM are important and contribute equally towards the classification process. The presence of irrelevant and redundant features may unnecessarily degrade the performance of the classifier. Hence, it creates the need for feature selection which involves further reducing the dimensionality of the solution space by taking the features which may conversely affect the decision-making ability of the model. This problem of selecting an optimal set of features can be converted into an optimization problem where the objective function is to minimize the loss on the training dataset. Formally, it can be framed as: Find an optimal subset of features from the entire set of features which can minimize the loss on the training dataset and in turn maximize the testing accuracy.

One of the main downsides of feature selection methods is that it requires a lot of time to try out each possible combination. Typically, if there are n features, then the number of possible solutions is: $(2^n - 1)$. Thus, the number of solutions increase exponentially in powers of 2. Therefore, brute force methods of feature selection will require a lot of time and computational power in order to reach a sufficiently optimal solution. Metaheuristic algorithms are a class of algorithms which have the ability to explore the solution space in smaller time and give nearly optimal solution in much lesser time than brute force approaches. Recently, metaheuristic algorithms have been applied by researchers in many applications and obtained optimal solutions to the real-world problems which were difficult to solve or required a lot of processing time using conventional algorithms. They have the ability to come out of locally optimal solutions due to the randomness in the algorithm. This motivates us to use metaheuristic algorithm for feature selection. Here, we propose to use Arithmetic Optimization Algorithm (AOA) (Abualigah et al. 2021), a recently proposed metaheuristic algorithm, for the task of feature selection. AOA has been proved to be superior as compared to other metaheuristic algorithms in benchmark tests but it is aimed for continuous-valued problems. To tap into the strength of AOA, here we propose a binary variant of the original AOA known as Binary Arithmetic Optimization Algorithm (BAOA), so that it can be applied in binary optimization problems like feature selection problem. We apply two transfer functions namely S-shaped and V-shaped to make the continuous form of AAO applicable on binary search space. The respective binary variants are known as BAOA-S and BAOA-V. The main contributions of this paper are as follows:

1. An IF-FSM-C is proposed that uses integrated features obtained by the fusion of HC features and DLM features followed by feature selection to classify WSIs of osteosarcoma. Two deep learning models namely EfficientNet-B0 and Xception are trained as feature extractors and their performance is compared to each other in classifying WSIs.
2. Two binary variants of AAO known as BAOA-S and BAOA-V are proposed to solve feature selection problem.

The paper is organized as follows. Section 2 gives a brief overview of the related work. Section 3 gives a brief description of DLM used in this paper. In Section 4, AOA is explained in brief. In Section 5, the proposed binary variants of AOA namely BAOA-S and BAOA-V are explained. The proposed approach IF-FSM-C to classify WSIs of osteosarcoma is presented in Section 6. In Section 7, experimental results are presented and discussed to analyse the performance of the proposed approach. In section 8, we discuss conclusion and future work.

2. Related Work

With the advancement in technology, automatic detection systems have been proposed by researchers for analysing and classifying medical images (X-ray, ultrasound images, MRI, CT scan, histological images etc.) to detect various kind of cancers and tumors. Some automatic detection systems (Rahmawaty et al. 2016; Solmaz and Tareripour 2016; Bakheey 2017; Khan et al. 2019) classify medical images based on the features extracted using traditional handcrafted techniques whereas, others (Liu et al. 2015; Li et al. 2019; Bisla et al. 2019; Cao et al. 2019) use DLM to perform feature extraction and classification of medical images. Recently, it has been observed that researchers have proposed automatic detection systems (Yadav et al. 2018; Hasan et al. 2019; Almaraz-Damian 2020; Shankar and Perumal 2020) that use an amalgamation of both the techniques and these systems exhibit good performance as compared to their counter-parts.

Automatic detection systems for osteosarcoma detection are also proposed by researchers but as compared to other cancer detection systems, the

literature on osteosarcoma is relatively less. The dataset for osteosarcoma detection was initially proposed by Leavey et al. (2019), consisting of digital WSIs. Arunachalam et al. (2017), proposed a method to detect osteosarcoma by performing segmentation and handcrafted feature extraction on digital WSIs of osteosarcoma dataset. Mishra et al. (2017) propose a CNN architecture to classify the dataset images into VT, NVT and NT. Furthermore, Mishra et al. (2018) improved the CNN architecture by fine-tuning and augmenting the CNN to classify osteosarcoma WSIs dataset and then compared the results with AlexNet, VGGNet and LeNet. The image size was reduced to 128x128 patches, by cropping the original 1024x1024 image due to training limitations. An overall accuracy of 92.4% is obtained which outperformed AlexNet, VGGNet and LeNet. Arunachalam et al. (2019) combined machine learning and deep learning models to classify osteosarcoma WSIs. In the machine learning model, the segmentation was performed using K-means clustering followed by Otsu's multi-level thresholding to extract region of interests (ROIs) from the histological WSIs. Flood-fill algorithm is used to cluster the pixels and data analysis is performed to classify the WSIs into VT, NVT and NT. The segmented images are classified using a total of 13 machine learning models and the best accuracy was obtained on support vector machine (SVM). In deep learning, they built a custom CNN model based on AlexNet (Krizhevsky et al. 2012) and LeNet (LeChun 1989). The number of samples in the dataset are increased using various data augmentation techniques such as rotation, flipping etc. Two sets of experiments are performed, one with patch size of 128 X 128 and other with tiles of size 1024 X 1024. Overall accuracy achieved on SVM was 89.9% whereas in deep learning models, patches yielded 93.3% and tiles yielded 91.2%. Anisuzzaman et al. (2020) proposed a deep learning framework based on VGG19 and InceptionV3, trained using transfer learning. The input fed to these models were WSIs with no patches, in order to improve the classification accuracy. However, they images have to be resized to 375 X 375 due to memory limitation. The overall accuracy was 93.91% and 78.26% on VGG19 and InceptionV3 models respectively on multiclass classification.

3. Deep Learning Models

Convolution Neural Networks (CNNs) are generally used for automatic information extraction from image data. They comprise of convolution layers that extract neighbourhood relationships, and encode them to form the outputs of neurons. Generally, denser CNNs are able to learn higher amount of information. But increasing density introduces some problems like vanishing gradient. Thus, many CNN architectures are proposed that handle such problems in unique ways, and hence can be used according to the requirement. Here, we use two architectures namely EfficientNet-B0 (Tan and Le 2019) and Xception (Chollet 2016).

3.1 EfficientNet-B0

EfficientNet architectures are based on the idea of balanced scaling, to achieve the best possible results in available resources (Tan and Le 2019). While most architectures were developed at a fixed budget, and later scaled with more available resources, EfficientNet architectures are designed by uniform scaling in depth, width and resolution to provide an optimized architecture for given resources. Thus, they are a progressive set of architectures, with increasing complexity. In simplest architecture EfficientNet-B0, the feature extraction model is made up of a convolution layer followed by 16 MobileNet (Howard et al. 2017) inverted bottleneck MBConv layers of size 3x3 or 5x5, followed by convolution, pooling and fully connected layers. It has least number of trainable parameters (5.3 million), which makes training easier and efficient in EfficientNet-B0.

3.2 Xception

Xception architecture (Chollet 2016) is inspired from ideology of Inception (Szegedy et al. 2015) model, that says, cross channel correlation and the spatial correlation of N-dimensional data can be sufficiently decoupled so that they are not mapped jointly. Xception further improves it by completely decoupling cross channel and the spatial correlations. This is achieved using separable convolution layer. It also used shortcut connections between convolution layers, as proposed in ResNet (He et al. 2016) for identity mapping. The model consists of blocks and each block consists of at least two separable convolution layers of kernel size 3x3. The image size is reduced in initial blocks using MaxPooling of kernel 3x3, with convolution of kernel 3x3 with stride two in shortcut connections. The resultant model proved to be highly efficient for learning complex datasets.

4. Arithmetic Optimization Algorithm

Arithmetic Optimization Algorithm (AOA) is a metaheuristic algorithm recently proposed by Abualigah et al. (2021). It is based on the basic arithmetic operators, i.e., addition, subtraction, multiplication and division. Using these operators on a set of solutions, it can give the best element through mathematical optimization. The algorithm performs exploration using multiplication and division, as these operators can introduce change in large order. But these operators are unfit for local search due to high dispersion, so addition and subtraction operators are used to perform exploitation or local search.

AOA is a population-based algorithm, where initial solutions represented as $x_i = [x_i^1, x_i^2, \dots x_i^d]$, are generated randomly over a d-dimensional search space using Eq. 1.

$$x_i^j = x_{min}^j + r(x_{max}^j - x_{min}^j) \tag{1}$$

$$i = \{1, 2, \dots, N\}, j = \{1, 2, \dots, d\}$$

Where, N is the population size, x_i^j represents the jth dimension of the ith solution, x_{max}^j and x_{min}^j are the upper and lower bound in the search space for j^{th} dimension and r is a random number in the range 0 to 1. The initial solution X can also be represented by matrix as shown in Eq. (2).

$$X = \begin{bmatrix} x_1^1 & \cdots & x_1^d \\ \vdots & \ddots & \vdots \\ x_N^1 & \cdots & x_N^d \end{bmatrix} \tag{2}$$

A fitness function is defined which determines the quality of each solution in the population in an iteration. Here, the candidate solution with the highest fitness value in each iteration is considered as the most-optimal solution found so far. The decision regarding the selection of exploration and exploitation is based on Math Optimizer Accelerated (MOA) function calculated as shown in Eq. (3). It gives a coefficient based on current iteration C_{iter}, that is used in search phases.

$$MOA(C_{Iter}) = Min + C_{Iter} \times \left(\frac{Max - Min}{M_{Iter}}\right) \tag{3}$$

Where, C_{Iter} denotes the current iteration, M_{Iter} denotes the maximum number of iterations, Max and Min are constants, denoting the maximum and minimum possible values of MOA respectively. MOA is designed in such a way so as to favour exploration in the initial stages and exploitation in the later iterations. A random number $r_1 \in [0, 1]$ is generated and its value is compared with MOA. If $r_1 > MOA$, exploration is performed else exploitation.

4.1 Exploration Phase

During this phase, the solution space is explored using division and multiplication operators. exploration behaviour. For exploration, one of the division or multiplication operators are chosen randomly with equal probabilities. The new solution is calculated as shown in Eq. 4.

$$x_i^j(C_{Iter} + 1) = \begin{cases} B(x^j) \div (MOP + \epsilon) \times \left((x_{max}^j - x_{min}^j) \times \mu + x_{min}^j\right), r_2 < 0.5 \\ B(x^j) \times MOP \times \left((x_{max}^j - x_{min}^j) \times \mu + x_{min}^j\right), otherwise \end{cases} \tag{4}$$

Where, $x_i^j(C_{Iter} + 1)$ represents the jth position of the ith solution in the next iteration, $B(x^j)$ represents the jth position in the current best solution, ϵ is a small non-zero number, μ is a control parameter to adjust the search process and its value is set to 0.5, and r_2 is a random number between [0,1] and MOP is a Math Optimizer Function which is calculated in each iteration using Eq. 5.

$$MOP(C_{Iter}) = 1 - \frac{C_{Iter}^{1/\alpha}}{M_{Iter}^{1/\alpha}} \tag{5}$$

Where, α is the sensitivity parameter whose value is set to 5.

4.2 Exploitation Phase

This phase represents the deep search of solution, i.e., the search of optimal solution near the best solution. Thus, the operators used are addition and subtraction operator. Similar to the exploration, the probability of selection of operators in exploitation is also equal. The new solutions are calculated as shown in Eq. 6.

$$x_i^j(C_{Iter} + 1) = \begin{cases} best(x_j) - MOP \times \left((x_{max}^j - x_{min}^j) \times \mu + x_{min}^j\right), r_3 < 0.5 \\ best(x_j) + MOP \times \left((x_{max}^j - x_{min}^j) \times \mu + x_{min}^j\right), \ otherwise \end{cases} \tag{6}$$

Where, r_3 is a random number between [0,1].

5. Binary Arithmetic Optimization Algorithm (BAOA-S/BAOA-V)

In this section, we present the proposed binary variants BAOA-S and BAOA-V of AOA. As mentioned by Kennedy and Eberhart (1997), transfer function (TF) is an effective way of converting a continuous version of an algorithm into binary. The TF is used to convert given real values vector to binary vector. It defines the probability of changing the element in a solution vector to 0 or 1 based on the value of step vector. Here we apply two TFs: S-shaped (sigmoid) and V-shaped (hyperbolic tangent) as shown in Fig. 2 to transform AOA into BAOA-S and BAOA-V respectively using Eq. 7 and Eq. 8.

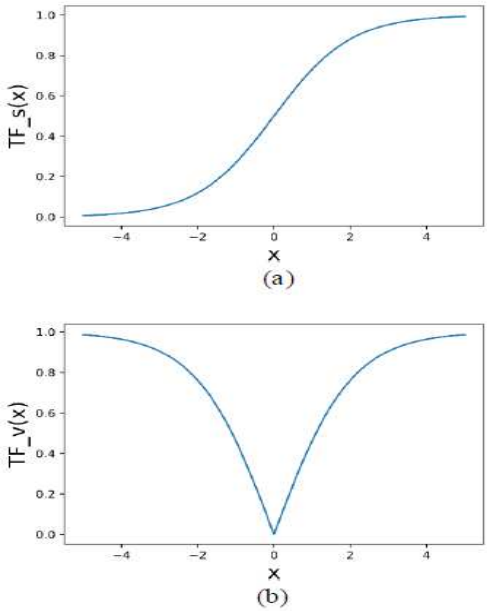


Fig. 2 Transfer Function (a) S-shaped and (b)V-shaped

$$TF_S(x_i^j) = \frac{1}{1+e^{x_i^j}} \quad (7)$$

$$TF_V(x_i^j) = \left|\frac{2}{1+e^{x_i^j}} - 1\right| \quad (8)$$

Where, x_i^j represents the jth dimension of the ith solution. In BAOA-S/BAOA-V, the position of the ith solution is updated using Eq. 4 and Eq. 6 depending on the values of MOA, r_2 and r_3. The generated solution act as a step function and consists of real values. To convert this into a binary vector, first we apply either TF_S or TF_V on the generated solution. Now, a random number r belonging to standard uniform distribution is generated and based on the value of the TF and random number r, the real values in the solution vectors are mapped to either 0 or 1 as shown in Eq. 9 and Eq. 10 for S-shaped and V-shaped TF respectively.

$$x_i^{j,b}(C_{Iter}+1) = \begin{cases} 1, & r < TF_S(x_i^j(C_{Iter}+1) \\ 0, & r \geq TF_S(x_i^j(C_{Iter}+1) \end{cases} \quad (9)$$

$$x_i^{j,b}(C_{Iter}+1) = \begin{cases} \sim x_i^j, & r < TF_S(x_i^j(C_{Iter}+1) \\ x_i^j, & r \geq TF_S(x_i^j(C_{Iter}+1) \end{cases} \quad (10)$$

Where, $x_i^{j,b}(C_{Iter}+1)$ is the binary vector obtained by mapping the step or real value solution vector $x_i^j(C_{Iter}+1)$ using TF.

6. The Proposed Model: IF-FSM-C

The proposed model is shown in Fig. 3 and the various steps are discussed in the following sections.

6.1 Image Segmentation

The original WSIs contain lot of information out of which some of the information is not important and do not contribute to the classification of the histological images. It is also possible that they may hamper the prediction of the classifier and is thus termed as “background or useless elements”. In order to remove this form of information from the images, we apply segmentation as a pre-processing stage and its output is the segmented image with all the unwanted information removed. The various forms of background information to be removed are: muscles, ink-markings, folded or severed tissues or blurred regions. The approach for the segmentation is similar to that proposed by Arunachalam et al. (2017). We apply hue normalization to the image by converting the RGB image into HSV, followed by multi-level thresholding in the hue channel. This image is then converted back into the RGB form. The RGB image are subjected to K-Means clustering segmentation for K=3 (pink, blue and white clusters). The pink and blue clusters are normal and required for further feature extraction, while the white cluster is the background information and needs to be removed from the image. For the removal of the white cluster, the pixels closest to the white centre are identified and are converted to white colour. The final output of this step is a modified image which has all the background information removed from it. Segmentation is important when HC feature extraction techniques are used for feature extraction. In case of DLM, segmentation is not mandatory as they have the ability to segment and identify the region of interest in the image automatically.

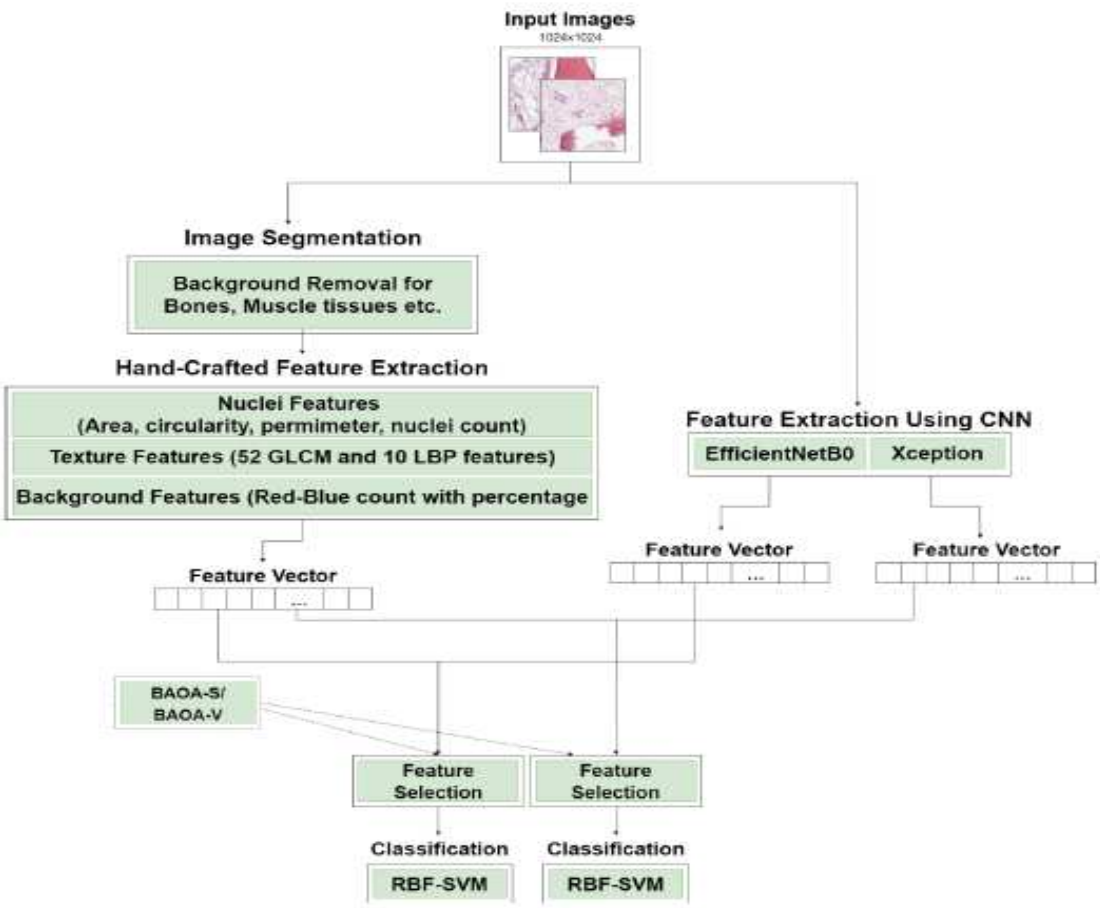


Fig. 3 The proposed IF-FSM-C

6.2 Feature Extraction

The performance of any classifier depends on the quality of features supplied to it. Thus, it is of utmost importance that sufficient and high-quality features are extracted from the image which are the closest representative of the attributes of the original image. Generally, the features extracted from the images are HC but with the advancement in the DLM, the use of CNNs has gained much importance showing significant improvements over the manual HC extraction. However, the use of HC features is still considered essential due to the freedom to select the attributes based on the properties of the image which the experts can observe and does not need any form of training for extraction process. In this research, we propose an integrated approach where features are extracted using both HC techniques and DLMs.

6.2.1 Handcrafted Feature Extraction

Arunachalam et al. (2019) proposed the use of various feature extraction techniques to obtain HC features such as: Nuclei Count, along with their shape and size, textural features using Gray-level co-occurrence matrix (GLCM) and red-blue pixel count and percentage. In this paper, we propose to extract local textural features of WSIs using rotation-invariant Local Binary Patterns (LBP) (Ojala et al. 2002).

Rotation-Invariant LBP allows us to analyse the local textural properties of an image. This is considered to be of significance with respect to the typical characteristics of texture in images belonging to different categories. In case of VT, research have shown a higher density of nuclei, which can be detected by a higher value of the LBP operator. On the other hand, the density of the nuclei in cases of NVT and NT are lower with former having a relatively higher density. In the segmented image, the local textures are negligible in the white regions obtained by removing bones in the original image and all LBP values for the pixels would be similar and have slight to no variation at all. Thus, the use of textural properties derived from rotation-invariant LBP allows us to distinguish the images on

the core feature of density of nuclei. A total of 10 features are extracted using (8,1) neighbourhood.

Global texture features are extracted using GLCM for four different orientations 0°, 45°, 90° and 135°, each giving a total of 13 features. Thus, the number of global texture features are 52.

Red-blue pixel count feature is used to represent the number of red and blue pixels in the image. In case of VT, the number of red and blue pixels are very high with a significant difference in their numbers. Here, there are more darker regions and less lighter regions. In case of NT, the pixels count is low and the difference between the count of red and blue pixels is also less. On the other hand, the pixel count is moderate in NVT but the difference between the counts is significant. Unlike VT, the high difference here signifies the presence of a greater number of lighter regions and less darker regions.

Finally, we extract the nuclei characteristics such as: Nuclei Count, shape and size. The shape and size are measured using Area, Perimeter and Circularity of the different nuclei as shown in Eq. 11 – Eq. 14. In order to extract these features, it is imperative to identify and isolate the nuclei from the segmented images. For nuclear identification, region growing algorithm like watershed/flood fill is used, which marks the nuclei contours. The marked nuclei are used to extract the four features mentioned before.

$$Perimeter = \sum_{i,j=0}^{X,Y} boundary(i,j) \quad (11)$$

Where, X and Y are dimensions of the image and $boundary(i,j)$ is calculated as:

$$boundary\ (i,j) = \begin{cases} 1, if\ Connectivity\ of\ P_{i,j} \geq 2 \\ 0, otherwise \end{cases} \quad (12)$$

$$Area = \sum_{n_i=0}^{N} Number\ of\ pixels\ in\ n_i \quad (13)$$

Where, N is the number of nuclei regions.

$$Circularity = 4\pi * \frac{Area}{Perimeter} \quad (14)$$

A total of 70 features are extracted for each WSI using HC methods.

6.2.2 DLMs as Feature Extractor

The use of DLMs for the purpose of classification has been carried out numerous times and has also given significant results and improvements over the conventional models. The basic principle of the DLM is to break down the image into low-level features in initial layers and then combine those features to determine complex and high-level properties of an image. The classification from these features is carried out by using a dense layer to make predictions. The ability of the DLMs to learn automatically from the training data acts as an added advantage over manual methods. This capability of these models to convert the image into a set of features can be exploited to use them as feature extractors. In addition to the HC features, we propose the extraction of features using DLM also.

For the purpose of extracting features using a DLM, the first step is to train the neural network. In this paper, we use EfficientNet-B0 and Xception as feature extractors. Each of these models are initialized with pre-trained ImageNet (Deng et al. 2009) weights. To train these deep learning models, the dataset is divided into 3 subsets – training (60%), validation (20%) and testing (20%) set. After the model, we append a global max pooling layer which finds the maximum value in the spatial dimension of the image at the last layer of the DLM. The output of the pooling layer is then flattened and fed into a dense layer of size same as the number of classes. The dense layer has SoftMax activation function due to multi-class classification problem.

Each network is trained on the input images rescaled to a size of 512x512 due to memory limitations. The images fed into the network are original WSIs without segmentation. This is because the DLM being an automatic feature extractor has the ability to itself segment and identify areas of interests. Thus, these networks can work on original images without the need for pre-processing of any form. In case of EfficientNet-B0, the output is of dimension 16x16x1280 which is then subjected to global max pooling, converting the output to size 1280. Thus, a total of 1280 features are obtained when EfficientNet-B0 is used as feature extractor as shown in Fig. 4. On the other hand, the output of Xception is of dimensions 32x32x2048, which results in 2048 features after the application of global pooling as shown in Fig. 5.

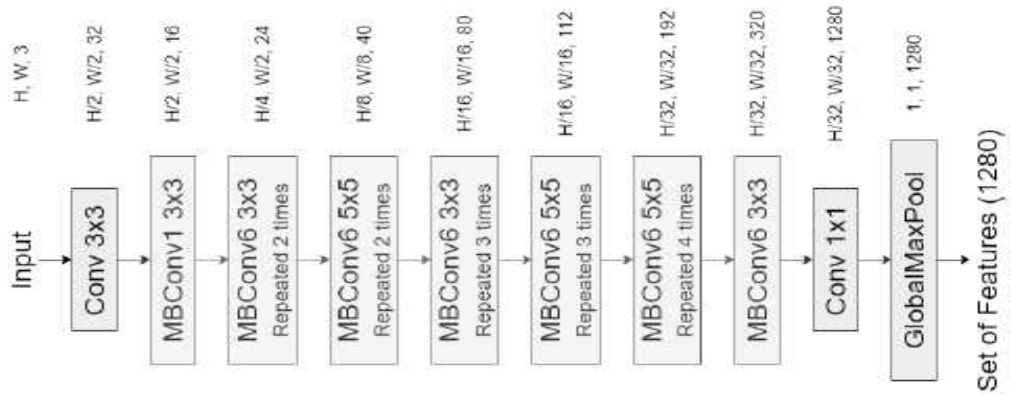


Fig. 4 Architecture of EfficientNet-B0 as Feature Extractor

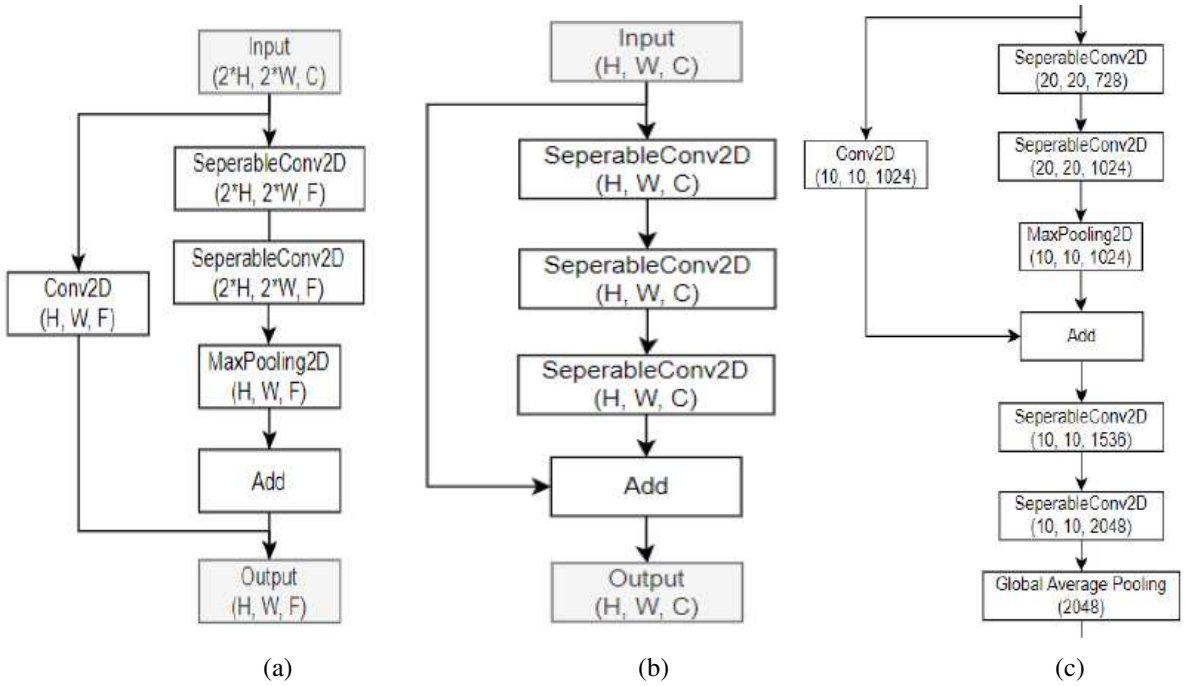


Fig. 5 Architecture of Xception as Feature Extractor (a) Initial Block to reduce image size by half, repeated 3 times, (b) Middle block, repeated 8 times, and (c) Final Layers

6.3 Feature Selection

The features identified through the two aforementioned methods are combined together for each image and two sets of feature vectors are obtained. The first set contains feature vectors obtained by the fusion of features extracted from HC methods and EfficientNet-B0. The size of feature vector obtained is 1350. The second set contains feature vectors obtained by the fusion of features extracted from HC methods and Xception. The size of feature vector obtained is 2118. This results in huge pool of features for each image. But it is quite possible that not all of these features contribute to the final classification decision. So, it is important to select significant features from the pool of extracted features. Here, we employ the proposed BAOA-S and BAOA-V explained in Section 5 to perform feature selection.

6.3.1 Generation of Initial Population

BAOA-S/BAOA-V starts with the generation of initial population of binary feature vectors (FV) as shown in Fig. 6. Since the number of features chosen

1	0	1	0	1	.....	1	0	0

Fig. 6 Binary Feature Vector

should also be uniformly distributed, a greedy method is employed to generate the set of initial solutions. The pseudo code of the greedy method used to generate initial population is shown in Fig. 7. A '1' in the feature vector shows that the corresponding feature is included during classification and a '0' shows that the corresponding feature is excluded.

1. Repeat steps 1.1 to 1.3 for each feature vector FV.

    1.1. Select a random integer R$\in U[1, NF]$, where, NF is the total number of features extracted

    1.2. Create a binary vector, with R number of 1's, rest 0's, i.e.,
    $FV_i = [1, 1, \ldots (\mathrm{R}\ times), 0, 0, \ldots (NF - \mathrm{R}\ times)]$.

    1.3. Shuffle the vector FV, to obtain a random sequence of 0s and 1s.

2.. The result will be a binary matrix, with each row representing a solution.

Fig. 7 Pseudo code of greedy method for population initialization

6.3.2 Fitness Function

In order to evaluate the quality of the feature vectors, the fitness of each FV is calculated. For this, the RBF-SVM classifier is trained with a set of features selected by that individual and then tested on the same set of features, but with the test set. Here, the objective is to maximize the classification accuracy and minimize the number of features. So, the fitness function is defined as shown in Eq. 14.

$$Fitness\ (FV_i) = \alpha\ E(C) + \beta \frac{|F|}{|N|} \qquad (14)$$

Where, $Fitness\ (FV_i)$ is the fitness of the ith solution vector, $E(C)$ is the classification error rate of SVM classifier, $|F|$and $|N|$ are the number of features selected and the total number of features respectively, $\alpha\epsilon[0,1]$ and $\beta = (1 - \alpha)$ are adopted from literature (Mafarja et al. 2019; Emary et al. 2016)

The aim of applying BAOA-S and BAOA-V is to explore and exploit the solution space properly so as to obtain a set of features that minimizes the fitness function and increases the classification accuracy.

6.4 Classification

The final stage in the proposed approach is the classification of the WSIs into VT, NVT and NT. For this, once an optimal set of features are obtained, we select those features and feed it into a SVM classifier with RBF kernel. Once the training is completed, the test set is fed into the trained RBF-SVM classifier to obtain the testing accuracy and other metrics for comparison.

To evaluate the performance of a model, the metrics used are: Accuracy, Sensitivity, Specificity and Precision. In addition to these, we employ ROC Curve and area under the curve (AUC). All these metrics are calculated using the values of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN).

7. Results and Discussion

Here, we use a publicly available dataset that contains Hematoxylin and Eosin (H&E) stained histological images of osteosarcoma (Leavey et al. 2019). The dataset comprises of 345 VT, 263 NVT and 536 NT images making a total of 1144 images. The images are of size 1024 X 1024.

Two sets of experiments are performed to compare the performance of various combinations of feature extraction techniques and to assess the impact of feature selection on RBF-SVM classifier's performance. The various combinations of features obtained by using different feature extraction techniques are a) only HC features, b) features extracted using EfficientNet-B0 only, c) features extracted using Xception only, d) combination of HC and EfficientNet-B0 features and e) combination of HC and Xception features. The dataset is divided into two sets- training set (80%) and testing set (20%). The proposed model IF-FSM-C is implemented in Jupyter notebook in Python and 3.7.10. Python libraries OpenCV, SciKit Learn and Numpy are used for image processing, and Tensorflow with Keras framework for CNN training. The scripts are executed on Google Colab platform running on dual core Intel(R) Xeon(R) CPU @ 2.30GHz, and TPU acceleration for CNN training using Google Cloud TPUv2 (octa core). The dataset was stored and retrieved using TFRecords on Google Cloud Storage bucket for faster processing.

In the first set of experiments, we compare the performance of RBF-SVM when features are extracted using different combinations of feature extraction techniques and are given to it to classify WSIs. Here, feature selection is not applied on the extracted features. The results are shown in Table 1.

In the second set of experiments, we apply feature selection on the features extracted using different combinations of feature extraction techniques before giving it to RBF-SVM for classification. BAOA-S and BAOA-V are randomized algorithms, so we run these algorithms 30 times on each combination of features and report the best and average results obtained over 30 runs. The confusion matrix of RBF-SVM generated using the features obtained after applying BAOA-S for feature selection in the best case is shown in Fig.8. The corresponding performance metrics and ROC-AUC curves are shown in Table 2 and Fig. 9 respectively. The confusion matrix of RBF-SVM generated using the features obtained after applying BAOA-V in the best case is shown in Fig. 10. The corresponding performance metrics and ROC-AUC curves are shown in Table 3 and Fig. 11 respectively. It is to be noted here that feature selection techniques are not applied in the case where HC features are used alone for classification. Finally, the average number of features selected over 30 runs of BAOA-S and BAOA-V and the number of features on which the best accuracy is obtained are shown in Table 4 and Table 5 respectively.

Table 1. Performance metrics of RBF-SVM for classification of WSIs into VT, NVT and NT using various combinations of feature extraction techniques

Class	Performance Metrics	Feature Extraction Techniques				
		HC	**EfficientNet-B0**	**Xception**	**EfficientNet-B0 + HC**	**Xception + HC**
VT	**Specificity**	0.945	0.9809	0.9814	0.9747	**1. 0**
	Sensitivity	0.8636	0.9516	**1. 0**	0.9836	0.9839
	Precision	0.9406	0.9516	0.9508	0.9375	**1.0**
	Accuracy	0.9041	0.9726	0.9863	0.9772	**0.9954**
NVT	**Specificity**	0.9515	0.9645	0.9643	0.9825	**0.9826**
	Sensitivity	0.8889	0.96	**0.9608**	0.9583	0.9362
	Precision	0.8571	0.8889	0.8909	**0.9388**	0.9362
	Accuracy	0.9361	0.9635	0.9635	**0.9772**	0.9726
NT	**Specificity**	0.939	0.9643	**0.9817**	**0.9817**	0.9633
	Sensitivity	0.9455	0.9252	0.9182	0.9455	**0.9727**
	Precision	0.8387	0.9612	0.9806	**0.9811**	0.964
	Accuracy	0.9406	0.9452	0.9498	0.9635	**0.968**
	Overall Accuracy	0.8904	0.9406	0.9497	0.9589	**0.9608**

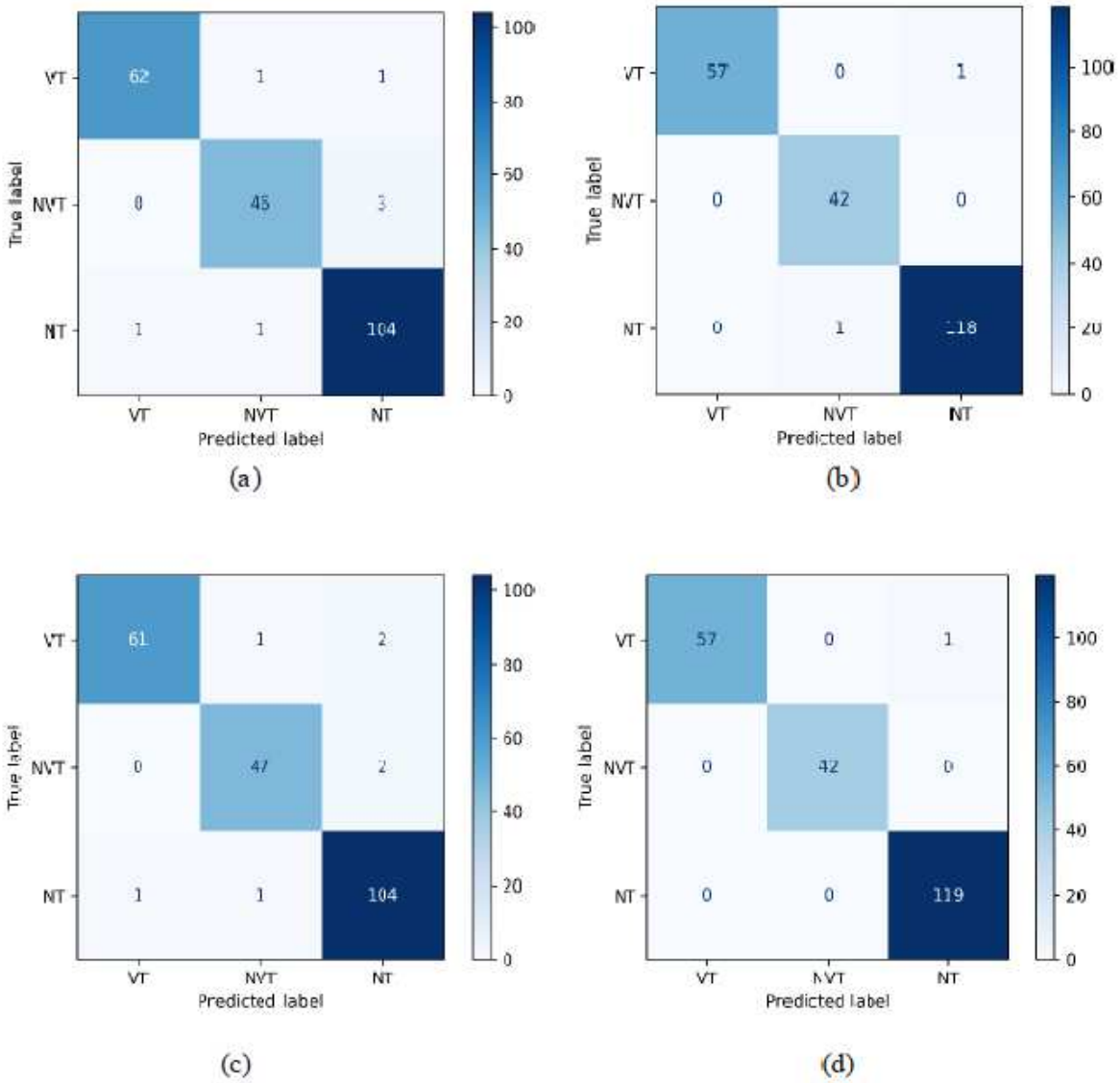


Fig. 8 Confusion matrix of classification of WSIs into VT, NVT and NT by RBF-SVM classifier using various combinations of feature extraction techniques and feature selection using BAOA-S (a) EfficientNet-B0, (b) Xception, (c) EfficientNet-B0 + HC, and (d) Xception + HC

Table 2. Performance metrics of classification of WSIs into VT, NVT and NT by RBF-SVM using various combinations of feature extraction techniques and feature selection using BAOA-S

Class		Specificity				Sensitivity			
		Efficient Net-B0	**Xception**	**Efficient NetB0+HC**	**Xception + HC**	**Efficient Net-B0**	**Xception**	**Efficient NetB0+HC**	**Xception + HC**
VT	avg	0.9847	0.992	0.9797	0.9938	0.984	0.9983	0.9838	**1.0**
	best	0.9872	0.9938	0.9809	0.9938	0.9841	**1.0**	0.9839	**1.0**
	std	0.0042	0.0028	0.0025	0. 0	0.0002	0.0052	0.0001	**0.0**
NVT	avg	0.983 0	0.9938	0.9876	1.0	0.9565	0.9537	0.9553	**1.0**
	best	0.9882	1.0	0.9882	1.0	0.9583	0.9767	0.9592	**1.0**
	std	0.0017	0.004	0.0017	0.0	0.0055	0.02	0.0113	**0.0**
NT	avg	0.9819	0.983	0.9811	1.0	0.9612	0.9833	0.9612	**0.9917**
	best	0.9820	0.99	0.982	1.0	0.963	0.9916	0.9717	**0.9917**
	std	0.0001	0.0077	0.0025	0.0	0.0035	0.0052	0.0053	**0.0**

Class		Precision				Accuracy			
		Efficient Net-B0	**Xception**	**Efficient NetB0+HC**	**Xception + HC**	**Efficient Net-B0**	**Xception**	**Efficient NetB0+HC**	**Xception + HC**
VT	avg	0.9625	0.9776	0.9500	**0.9828**	0.9845	0.9936	0.9808	**0.9954**
	best	0.9688	**0.9828**	0.9531	**0.9828**	0.9863	**0.9954**	0.9817	**0.9954**
	std	0.0104	0.0079	0.0062	**0.0**	0.003	0.0022	0.0018	**0.0**
NVT	avg	0.9408	0.9738	0.9571	**1.0**	0.9772	0.9858	0.9804	**1.0**
	best	0.9592	**1.0**	0.9592	**1.0**	0.9772	0.9954	0.9817	**1.0**
	std	0.0061	0.0167	0.0061	**0.0**	0.0	0.0059	0.0029	**0.0**
NT	avg	0.9811	0.9857	0.9802	**1.0**	0.9717	0.9831	0.9712	**0.9954**
	best	0.9811	0.9916	0.9811	**1.0**	0.9726	0.9909	0.9726	**0.9954**
	std	**0.0**	0.0066	0.0028	**0.0**	0.0018	0.005	0.0021	**0.0**
					Overall Accuracy	0.9908	0.9680	0.9680	**0.9954**

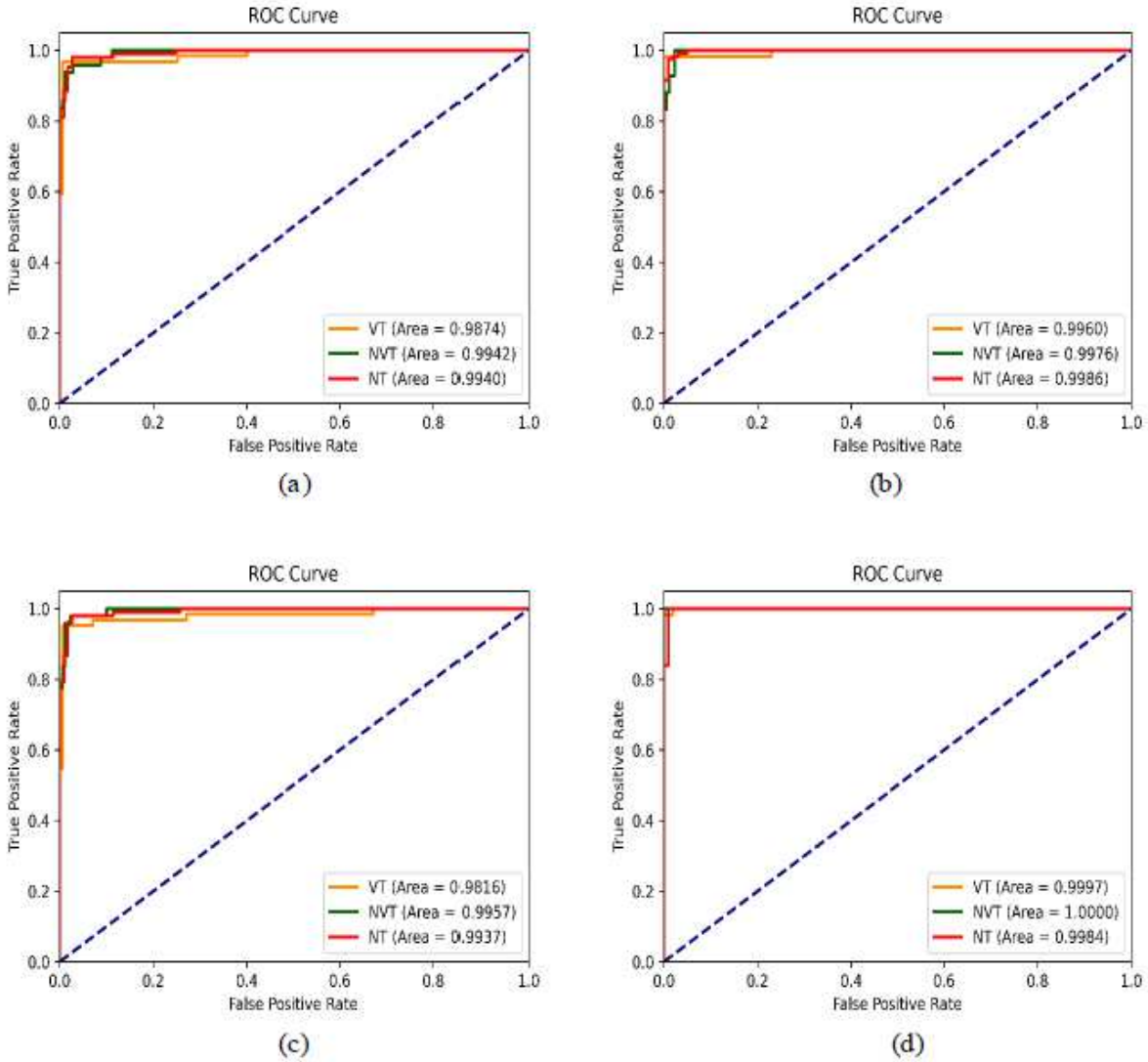


Fig. 9 ROC-AUC of classification of WSIs into VT, NVT and NT by RBF-SVM classifier using various combinations of feature extraction techniques and feature selection using BAOA-S (a) EfficientNet-B0, (b) Xception, (c) EfficientNet-B0 + HC, and (d) Xception + HC

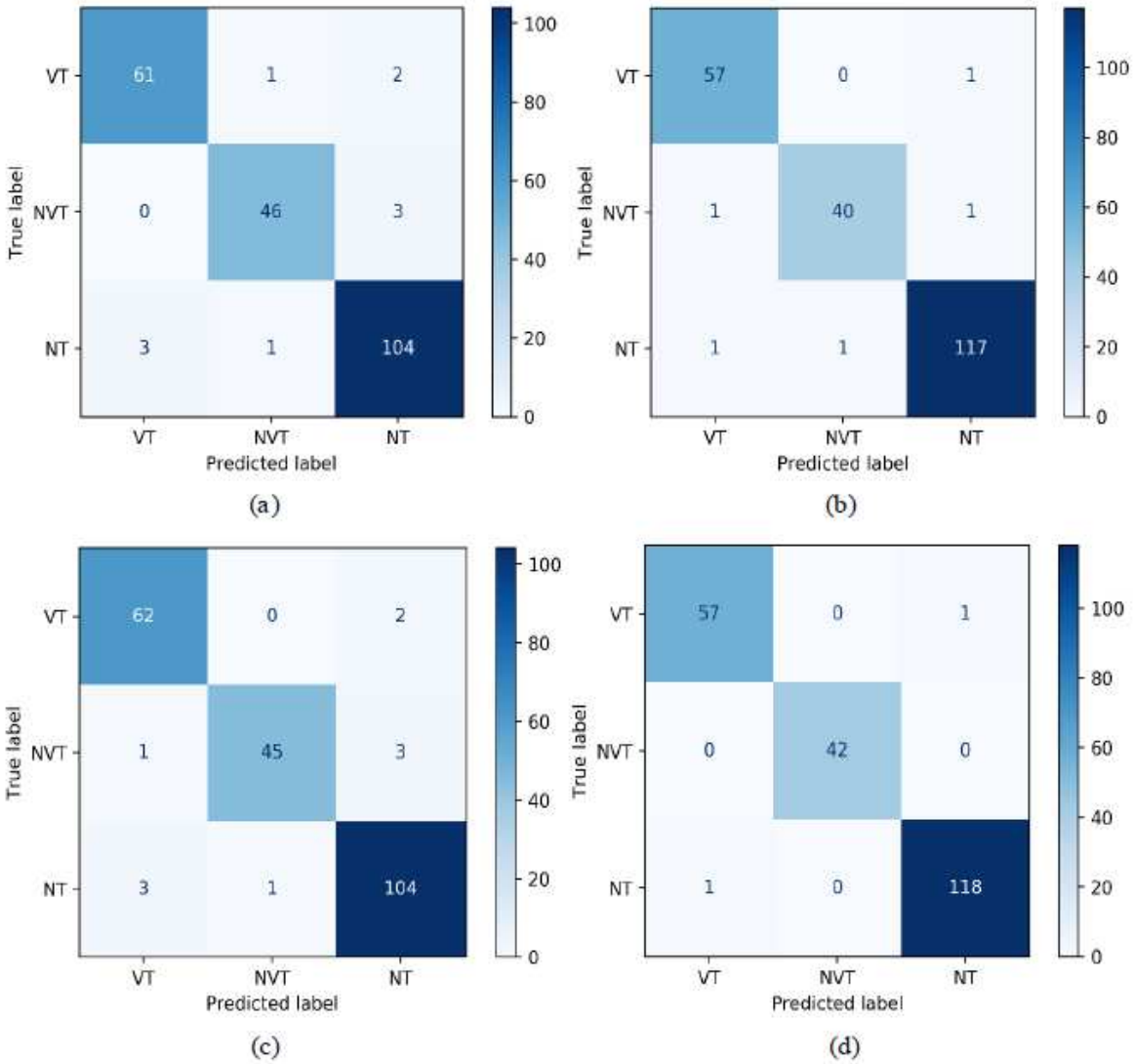


Fig. 10 Confusion matrix of classification of WSIs into VT, NVT and NT by RBF-SVM classifier using various combinations of feature extraction techniques and feature selection using BAOA-V (a) EfficientNet-B0, (b) Xception, (c) EfficientNet-B0 + HC, and (d) Xception + HC

Table 3. Performance metrics of classification of WSIs into VT, NVT and NT by RBF-SVM using various combinations of feature extraction techniques and feature selection using BAOA-V

Class		Specificity				Sensitivity			
		Efficient Net-B0	**Xception**	**Efficient NetB0+HC**	**Xception + HC**	**Efficient Net-B0**	**Xception**	**Efficient NetB0+HC**	**Xception + HC**
VT	avg	0.9602	0.9745	0.9614	0.99	0.9482	0.9793	0.9388	0.9916
	best	0.9646	0.9800	0.9643	0.99	0.9541	0.9832	0.9541	0.9916
	std	0.0044	0.0032	0.0034	0.0	0.0072	0.0049	0.0061	0.0
NVT	avg	0.9735	0.9852	0.9716	1.0	0.9561	0.9548	0.9682	1.0
	best	0.9828	0.9888	0.9771	1.0	0.9583	0.9756	0.9783	1.0
	std	0.0093	0.0042	0.0077	0.0	0.0058	0.0072	0.0104	0.0
NT	avg	0.9773	0.9847	0.9812	0.9938	0.9511	0.986	0.9387	0.9828
	best	0.9809	0.9938	0.9871	0.9938	0.9538	0.9611	0.9394	0.9828
	std	0.0052	0.0054	0.0065	0.0	0.0022	0.0081	0.0005	0.0

Class		Precision				Accuracy			
		Efficient Net-B0	**Xception**	**Efficient NetB0+HC**	**Xception + HC**	**Efficient Net-B0**	**Xception**	**Efficient NetB0+HC**	**Xception + HC**
VT	avg	0.9625	0.9752	0.9539	0.9916	0.9456	0.977	0.9535	0.9909
	best	0.963	0.9832	0.963	0.9916	0.9593	0.9817	0.9593	0.9909
	std	0.0004	0.0086	0.0103	0.0	0.0083	0.0062	0.0067	0.0
NVT	avg	0.9322	0.9472	0.9165	1.0	0.9775	0.9834	0.9735	1.0
	best	0.9388	0.9524	0.9183	1.0	0.9775	0.9863	0.9774	1.0
	std	0.0065	0.0072	0.0012	0.0	0.	0.004	0.0045	0.0
NT	avg	0.9511	0.9801	0.9647	0.9801	0.9774	0.9821	0.9712	0.9909
	best	0.9538	0.9828	0.9688	0.9828	0.9729	0.9863	0.9729	0.9909
	std	0.0008	0.0004	0.0067	0.0011	0.0011	0.0083	0.0011	0.0
					Overall Accuracy	0.9547	0.9771	0.9547	0.9908

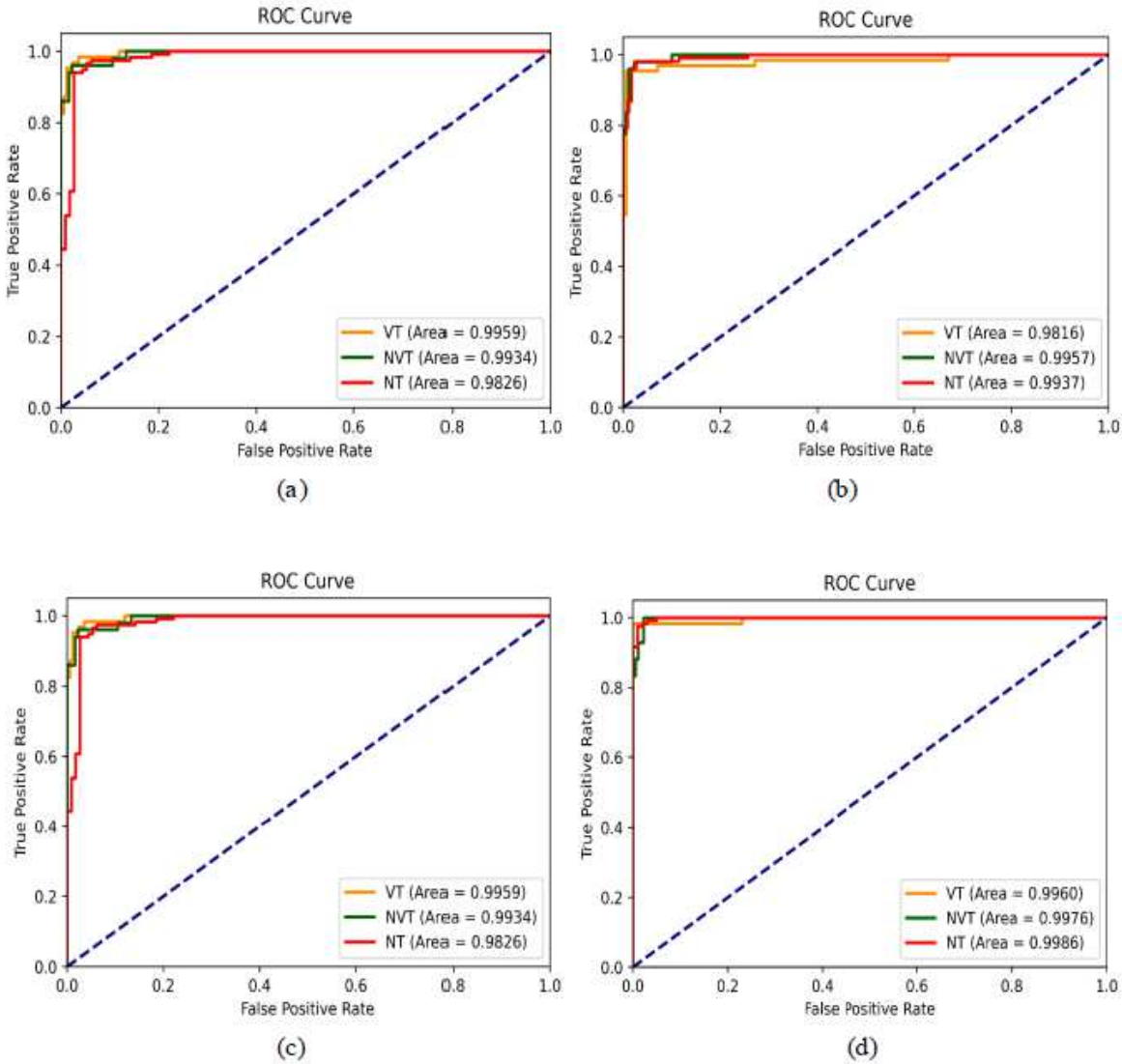


Fig. 11 ROC-AUC of classification of WSIs into VT, NVT and NT by RBF-SVM classifier using various combinations of feature extraction techniques and feature selection using BAOA-V (a) EfficientNet-B0, (b) Xception, (c) EfficientNet-B0 + HC, and (d) Xception + HC

Table 4. Number of features selected using BAOA-S when features are extracted using a combination of different feature extraction techniques

		EfficientNet-B0	Xception	EfficientNetB0+HC	Xception + HC
Number of Features	avg	152.1	149.600	156.1	18.8
	best	799	1023	879	188
	std	304.6869	303.7618	315.2924	56.4

Table 5. Number of features selected using BAOA-V when features are extracted using a combination of different feature extraction techniques

		EfficientNet-B0	Xception	EfficientNetB0+HC	Xception + HC
Number of Features	avg	238.5	341.3	265.2	47.8
	best	815	1543	912	211
	std	334.686	406.56	387.78	146.769

When features extracted using DLM are used for classification, the performance of the classifier is better as compared to when HC features are used alone as can be seen from Table 1. However, when integration of HC and DLM features are used for classification, the classifier exhibits higher performance than when these features are used alone. In case of DLM, the features extracted using Xception leads to better classification of WSIs as compared to EfficientNet-B0 features because Xception is a combination of ResNet and Inception networks, combining the strengths of both the networks, which leads to better identification of low-level features and consequently better results. On the other hand, EfficientNet majorly focuses on the efficient learning of the data and with this aim the accuracy may be compromised. Without feature selection, the best performance is exhibited by integrated Xception and HC features with an overall accuracy of 96.08%. The classifier doesn't perform well on HC features because the deep learning models, being automatic feature extractors, have the ability to identify low-level features which sometimes are skipped when HC features are used. This may lead to better performance on DL features as compared to HC features only.

When feature selection is applied on features extracted using HC techniques and DLM, it is observed from Table 2 and Table 3 that application of feature selection results in improvement in the performance of the classifier in case of both BAOA-S and BAOA-V. This is attributed to the fact that FS will keep only those features which are important and have a role in the classification process. The curse of redundant features may lead to decrease in the accuracy of a model and by the application of FS, the redundant or dependent features are removed, which leads to better results. It is also evident from Table 2 and Table 3, that BAOA-S results in selection of better features as compared to BAOA-V. The best overall accuracy of 99.54% and higher specificity and sensitivity is observed when integrated Xception and HC features after applying BAOA-S feature selection is used by the classifier. Also, the number of average features selected by BAOA-S is less as compared to BAOA-V as evident from Table 4 and Table 5, which indicates usage of lower computational resources and faster processing. These models can therefore, be applied in real time systems, where the results are expected in a fraction of time. The number of integrated Xception and HC features selected by applying BAOA-S on which the classifier shows the best performance is 188 as against 211 selected by applying BAOA-V.

The comparison of the proposed IF-FSM-C with the results reported in literature is shown in Table 6. It is clear from Table 10 that the proposed IF-FSM-C outperforms the existing state-of-the-art approaches.

7. Conclusion and Future Work

In this paper, we proposed an automatic detection system Integrated Features-Feature Selection Model for Classification (IF-FSM-C) that detect osteosarcoma from the high-resolution whole slide images (WSIs). Here, a fusion of features extracted using handcrafted techniques and deep learning models namely EfficientNet-B0 and Xception are used for classification by RBF-SVM. A feature selection step is applied before giving the integrated features to the classifier in order to remove

redundant and irrelevant features. Two binary variant BAOA-S and BAOA-V of recently proposed AOA are proposed and applied for feature selection. The results indicates that integrated features along with feature selection by BAOA-S results in significant improvement in the performance of the classifier as compared to when HC or deep learning features are used alone, with or without feature selection. In case of classifiers using integrated features, the performance of classifier on integrated Xception and HC features is better than that of integrated EfficientNet-B0 and HC features.

In future, we plan to explore other deep learning models which has a performance comparable to the current models and is computationally cheaper. This may help to further reduce the computational load and increase the speed of the system.

Table 10. Comparison of Class-wise and Overall Accuracy of State-of-the-Art Approaches with the Proposed Approach

Class	Accuracy in %					
	Mishra et al.'s (2018) Deep Learning Model	Arunachalam et al.'s (2019) Deep Learning Model		Arunachalam et al.'s (2019) Expert-guided and CellProfiler Features (SVM)	Anisuzzaman et al.'s (2020) Deep Learning Model (VGG19)	IF-FSM-C Model (Xception +HC)
		Patch	Tile			
VT	92	95.3	92.6	91	96.09	**99.54**
NVT	90	92.7	91.5	87	95.65	**100**
NT	95	91.9	89.5	91	96.09	**99.54**
Overall Accuracy	92.4	93.3	91.2	89.9	93.91	**99.54**

Declarations

Funding: Not Applicable

Conflicts of interest/Competing interests: There is no conflict of interest

Availability of data and material: Data available on request from the authors

Code Availability: Code available on request from the authors

References

Abualigah L, Diabat A, Mirjalili S, Elaziz MA, Gandomi AH (2021) The Arithmetic Optimization Algorithm. Computer Methods in Applied Mechanics and Engineering 376, 113607

Adams R, Bischof L (1994) Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(6): 641-647.

Almaraz-Damian J-A, Ponomaryov V, Sadovnychiy S, Castillejos-Fernandez H (2020) Melanoma and Nevus Skin Lesion Classification Using Handcraft and Deep Learning Feature Fusion via Mutual Information Measures. Entropy 22, 484.

Anisuzzaman DM, Barzekar H, Tong L, Luo J, Yu Z (2020) A Deep Learning Study on Osteosarcoma Detection from Histological Images. https://arxiv.org/pdf/2011.01177.pdf.

Arunachalam HB, Mishra R, Armaselu B, Daescu O, Martinez M, Leavey PJ, Rakheja D, Cederberg KB, Sengupta AL, Ni'suilleabhain M (2017) Computer aided image segmentation and classification for viable and non-viable tumor identification in osteosarcoma. Pacific Symposium on Biocomputing 195-206.

Arunachalam HB, Mishra R, Daescu O (2019) Viable and necrotic tumor assessment from whole slide images of osteosarcoma using machine-learning and deep-learning models. PLoS One 14(4).

Bakheet S (2017) An svm framework for malignant melanoma detection based on optimized hog features. Computation 5(1), 4.

Bisla D, Choromanska A, Berman RS, Stein JA, Polsky D (2019) Towards automated melanoma detection with deep learning: Data purification and augmentation. In Proceedings - 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2019, 2720-2728.

Cao Z, Duan L, Yang G (2019) An experimental study on breast lesion detection and classification from ultrasound images using deep learning architectures. BMC Med Imaging 19, 51.

Chollet F (2016) Xception: Deep learning with depthwise separable convolutions. arXiv preprint arXiv:1610.02357.

Deng J, Dong W, Socher R, Li Li-Jia, Li K, Fei-Fei L (2009) ImageNet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition 248-255. doi: 10.1109/CVPR.2009.5206848.

Ding C, He X, Zha H, Simon HD (2002) Adaptive Dimension Reduction for Clustering High Dimensional Data. Proceedings of International Conference on Data Mining.

Emary E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. Neurocomputing 172:371–381.

Goode A, Gilbert B, Harkes J, Jukic D, Satyanarayanan M (2013) OpenSlide: A vendor-neutral software foundation for digital pathology. Journal of Pathology informatics 4-27.

Hasan AM, Jalab HA, Meziane F, Kahtan H, Al-Ahmad AS (2019) Combining deep and handcrafted image features for MRI brain scan classification IEEE Access 7:79959-79967,

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In Proceedings of CVPR, 770–778.

Howard A, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. In CVPR.

Kennedy J, Eberhart RC (1997) A discrete binary version of the particle swarm algorithm. In 1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation 5:4104-4108. IEEE

Khan MQ et al. (2019) Classification of Melanoma and Nevus in Digital Images for Diagnosis of Skin Cancer. IEEE Access 7:90132-90144.

Kleinerman RA, Tucker MA, Tarone RE, et al (2005) Risk of new cancers after radiotherapy in long-term survivors of retinoblastoma: an extended follow-up. J Clin Oncol 23:2272–2279.

Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems.

Leavey P, Sengupta A, Rakheja D, Daescu O, Arunachalam HB, Mishra R (2019) Osteosarcoma data from UT Southwestern/UT Dallas for Viable and Necrotic Tumor Assessment [Data set]. The Cancer Imaging Archive.

LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation 1 (4): 541–551.

Li Y, Deng L, Yang X (2019) Early diagnosis of gastric cancer based on deep learning combined with the spectral-spatial classification method. Biomed Opt Express 10(10): 4999-5014.

Lindsey BA, Markel JE, Kleinerman ES (2017) Osteosarcoma Overview. Rheumatology and Therapy 4(1):25–43.

Liu D, Gan T, Rao N, Xu G, Zeng B, Li H (2015) Automatic Detection of Early Gastrointestinal Cancer Lesions Based on Optimal Feature Extraction from Gastroscopic Images. Journal of Medical Imaging and Health Informatics 5(2).

Mafarja M, Aljarah I, Faris H, Hammouri AI, Ala'M AZ, Mirjalili S (2019) Binary grasshopper optimisation algorithm approaches for feature selection problems. Expert Systems with Applications 117:267-286.

Mirabello L, Troisi RJ, Savage SA (2009) Osteosarcoma incidence and survival rates from 1973 to 2004: data from the Surveillance, Epidemiology, and End Results Program. Cancer 115(7):1531-1543

Mishra R, Daescu O, Leavey P, Rakheja D, Sengupta A (2017) Histopathological Diagnosis for Viable and Non-viable Tumor Prediction for Osteosarcoma Using Convolutional Neural Network. International Symposium on Bioinformatics Research and Applications, Springer, Cham.

Mishra R, Daescu O, Leavey P, Rakheja D, Sengupta A (2018) Convolutional neural network for histopathological analysis of osteosarcoma. Journal of Computational Biology 25(3): 313-325.

Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence 24(7):971-987.

Otsu N (1979) A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics 9(1): 62-66.

Qureshi MN, Ahamad MV (2018) An Improved Method for Image Segmentation Using K-Means Clustering with Neutrosophic Logic. Procedia Computer Science 132: 534-540.

Rahmawaty M, Nugroho HA, Triyani Y, Ardiyanto I, Soesanti I (2016) Classification of breast ultrasound images based on texture analysis. International Conference on Biomedical Engineering (iBioMed) 1-6.

Samet H (2006) Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann.

Seigel R et al. (2021) Cancer Statistics 2021. CA: A Cancer Journal for Clinicians 71(1):7–33.

Serge Beucher, Fernand Meyer (1993) The morphological approach to segmentation: the watershed transformation. In Mathematical Morphology in Image Processing (Ed. E. R. Dougherty) 433–481.

Shankar K, Perumal E (2020) A novel hand-crafted with deep-learning features based fusion model for COVID-19 diagnosis and classification using chest X-ray images, Complex Intell. Syst.

Solmaz A, Tajeripour F (2016) Detection of Brain Tumor in 3D MRI Images using Local Binary Patterns and Histogram Orientation Gradient. Neurocomputing 219.

Szegedy C, Liu W, Jia Y et al. (2015) Going deeper with convolutions. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Tan M, Le QV (2019) EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of International Conference on Machine Learning (ICML).

Wang H, Oliensis J (2010) Generalizing edge detection to contour detection for image segmentation. Computer Vision and Image Understanding 114: 731–744.

Wu L, Wang Y, Long J, Liu Z (2015) A Non-seed-based Region Growing Algorithm for High Resolution Remote Sensing Image Segmentation. In: Zhang YJ. (eds) Image and Graphics. ICIG 2015. Lecture Notes in Computer Science, vol 9217. Springer, Cham.

Yadav D, Kohli N, Agarwal A, Vatsa M, Singh R, Noore A (2018) Fusion of handcrafted and deep learning features for large-scale multiple iris presentation attack detection. In IEEE Conference on Computer Vision and Pattern Recognition Workshops