diagnostics

MDPI

Article

A Novel Hybrid Approach for Classifying Osteosarcoma Using Deep Feature Extraction and Multilayer Perceptron

Md. Tarek Aziz [1], S. M. Hasan Mahmud [1,2,*], Md. Fazla Elahe [1,3], Hosney Jahan [1,4], Md Habibur Rahman [1,5], Dip Nandi [2], Lassaad K. Smirani [6], Kawsar Ahmed [7,8,*], Francis M. Bui [7] and Mohammad Ali Moni [9]

1 Centre for Advanced Machine Learning and Applications (CAMLAs), Bashundhara R/A, Dhaka 1229, Bangladesh; tarekiub17@gmail.com (M.T.A.); elahe.se@daffodilvarsity.edu.bd (M.F.E.); jahan@cse.mist.ac.bd (H.J.); habib@iu.ac.bd (M.H.R.)
2 Department of Computer Science, American International University-Bangladesh (AIUB), 408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh; dip.nandi@aiub.edu
3 Department of Software Engineering, Daffodil International University, Daffodil Smart City (DSC), Savar, Dhaka 1216, Bangladesh
4 Department of Computer Science & Engineering (CSE), Military Institute of Science and Technology (MIST), Mirpur Cantonment, Dhaka 1216, Bangladesh
5 Department of Computer Science and Engineering, Islamic University, Kushtia 7003, Bangladesh
6 The Deanship of Information Technology and E-learning, Umm Al-Qura University, Mecca 24382, Saudi Arabia; lksmirani@uqu.edu.sa
7 Department of Electrical and Computer Engineering, University of Saskatchewan, 57 Campus Drive, Saskatoon, SK S7N 5A9, Canada; francis.bui@usask.ca
8 Group of Biophotomatiχ, Department of Information and Communication Technology (ICT), Mawlana Bhashani Science and Technology University (MBSTU), Tangail 1902, Bangladesh
9 Artificial Intelligence & Digital Health, School of Health and Rehabilitation Sciences, Faculty of Health and Behavioural Sciences, The University of Queensland, St. Lucia, QLD 4072, Australia; m.moni@uq.edu.au
* Correspondence: hasan.swe@aiub.edu (S.M.H.M.); k.ahmed@usask.ca or kawsar.ict@mbstu.ac.bd (K.A.)

Citation: Aziz, M.T.; Mahmud, S.M.H.; Elahe, M.F.; Jahan, H.; Rahman, M.H.; Nandi, D.; Smirani, L.K.; Ahmed, K.; Bui, F.M.; Moni, M.A. A Novel Hybrid Approach for Classifying Osteosarcoma Using Deep Feature Extraction and Multilayer Perceptron. *Diagnostics* **2023**, *13*, 2106. https://doi.org/10.3390/diagnostics13122106

Academic Editor: Jae-Ho Han

Received: 3 May 2023
Revised: 10 June 2023
Accepted: 13 June 2023
Published: 18 June 2023

Abstract: Osteosarcoma is the most common type of bone cancer that tends to occur in teenagers and young adults. Due to crowded context, inter-class similarity, inter-class variation, and noise in H&E-stained (hematoxylin and eosin stain) histology tissue, pathologists frequently face difficulty in osteosarcoma tumor classification. In this paper, we introduced a hybrid framework for improving the efficiency of three types of osteosarcoma tumor (nontumor, necrosis, and viable tumor) classification by merging different types of CNN-based architectures with a multilayer perceptron (MLP) algorithm on the WSI (whole slide images) dataset. We performed various kinds of preprocessing on the WSI images. Then, five pre-trained CNN models were trained with multiple parameter settings to extract insightful features via transfer learning, where convolution combined with pooling was utilized as a feature extractor. For feature selection, a decision tree-based RFE was designed to recursively eliminate less significant features to improve the model generalization performance for accurate prediction. Here, a decision tree was used as an estimator to select the different features. Finally, a modified MLP classifier was employed to classify binary and multiclass types of osteosarcoma under the five-fold CV to assess the robustness of our proposed hybrid model. Moreover, the feature selection criteria were analyzed to select the optimal one based on their execution time and accuracy. The proposed model achieved an accuracy of 95.2% for multiclass classification and 99.4% for binary classification. Experimental findings indicate that our proposed model significantly outperforms existing methods; therefore, this model could be applicable to support doctors in osteosarcoma diagnosis in clinics. In addition, our proposed model is integrated into a web application using the FastAPI web framework to provide a real-time prediction.

Keywords: osteosarcoma; convolutional neural networks; transfer learning; feature extraction; feature selection; machine learning; MLP

1. Introduction

Osteosarcoma, also known as osteogenic sarcoma, is a primary mesenchymal tumor that is distinguished histologically by the formation of osteoid by malignant cells. Osteosarcoma affects people mostly between the ages of 10 and 30, making it the third most common cancer among children and adolescents. The United States reports approximately 1000 new cases every year [1,2] that illustrate osteosarcoma as a challenging issue. Osteosarcoma and Ewing sarcoma are the two malignant bone cancers that mostly affect children and adolescents, and they represents about 56% and 34% bone cancer, respectively. The most common sites for osteosarcoma are the femur (42%, 75% of which are in the distal femur), the tibia (19%, 80% of which are in the proximal tibia), and the humerus (10%, 90% of which are in the proximal humerus) [3]. Osteosarcoma signs typically start off as mild localized bone pain, warmth, and redness where the tumor is located [4]. Neoadjuvant chemotherapy (NAC) and surgery are current therapeutic modalities that have improved patient survival rates by almost five years. From 1975 to 2010, osteosarcoma patients experienced an increase in five-year survival rate from 40% to 76% for those under the age of 15 and from 56% to 66% for those aged between 15 and 19 [5]. However, the five-year survival rate for metastatic osteosarcoma is still under 20% [6]. Early osteosarcoma diagnosis and careful monitoring during the chemotherapy cycle can increase the overall survival rate [7].

For the majority of malignancies, including osteosarcoma, a biopsy (histology report) test is the best way to determine if a part of the body has cancer. In addition, non-invasive imaging methods such as MRI, CT, and PET imaging modalities have been used for quantitative analyses in osteosarcoma response monitoring and surgical planning [7]. Even though the approaches based on biopsy can successfully identify the malignancy, approaches such as histologically guided biopsies and similar techniques have limitations in detecting malignancy. Moreover, the process of preparing histology specimens takes time; e.g., to represent the surface of a substantial three-dimensional tumor at least 50 histology slides are required to detect osteosarcoma malignancy accurately [8]. While assessing cancer patients using biopsy-derived tissue slides, pathologists manually find the most affected areas and examine the nuclear morphology and cellular characteristics. This manual inspection and diagnosis using tissue slides, which may consist of huge number of cells, can be laborious and arbitrary. The whole slide image (WSI) analysis can increase the amount of information retrieved from tissue slides for making the decision and increase the reliability of analysis [9]. An automated method is expected to emerge for the histopathological slide classification of osteosarcoma because microscopic analysis of slides is difficult, time-consuming, tedious, and subject to bias [10]. The morphological and contextual cues present in the digital WSIs are used as features for tissue classification, which promotes the usage of image processing and analysis approaches [9,11].

Osteosarcoma is highly heterogeneous, and it is influenced by inter- and intra-observer differences. In osteosarcoma, the precursor cells and some types of tumor cells are both stained the same shade of blue, but the precursor cells are rounder, more closely spaced, and more regular than the tumor cells [12,13]. To accurately determine the percentage of necrosis, various histological regions must be taken into account, including hemorrhagic tumor, blood cells, growth plates, clusters of nuclei, fibrous tissues, osteoclasts, cartilage, osteoid, osteoblasts, and precursors [10]. Recent research based on medical data shows that CNN can be used for medical images to extract and analyze information [14,15] and has a very successful impact. Deep learning (DL) and machine learning (ML) methods achieved tremendous success and popularity in medical research for cancer classification. In this study, we propose a hybrid approach, combining CNN and ML algorithms to classify osteosarcoma malignancy by using whole slide images for three classes, specifically, a viable tumor, necrosis (including fibrosis, osteoid, and coagulative necrosis), and nontumor (cartilage, bone, other normal tissue). The key contribution of our proposed hybrid approach is the integration of different CNN (transfer learning) and ML techniques for data preprocessing, optimizer analysis, feature selection, and classification for lower computational costs and better performance. Most of the previously published works either

employed ML techniques or DL techniques where they only focused on classification tasks and did not explore enough other ML techniques that may influence accurate prediction. Initially, we performed normalization in the training phase to enable our model to learn faster with a zero-centered Gaussian distribution of data. We have also explored a total of five CNN models and five ML classifiers with different parameter settings to determine the best feature extractor and classifier integration. Furthermore, an optimizer analysis was conducted for the MLP classifier to ensure better optimization by selecting the most suitable optimizer that demonstrates improved convergence time and loss. We also analyzed the impact of RFE's criterion concerning the number of selected important features. Finally, a web application has been developed using our proposed framework for real-time prediction.

The remaining sections of this article are structured as follows: In Section 2, we provided a literature review of previous works that has been published on this dataset. In Section 3, we described our proposed methodology in detail, including the dataset, preprocessing feature extraction, selection, and classification. The experimental results from various experiments, with proper analysis, are illustrated in Section 4. Furthermore, finally, we added a discussion in Section 5.

2. Literature Review

Researchers have developed automatic systems to identify various malignancies and tumors that can evaluate and classify medical images such as X-rays, histology images, ultrasound imaging, CT scan, MRIs, etc. [16–25]. The use of digital histopathology has grown significantly and shown great potential recently. In 2014, Irshad et al. [11] presented a survey on histopathology images, specifically in H&E and immunohistochemical staining protocols, that discusses classification techniques, segmentation, feature computation, and the major trends of various nuclei detection. Their study involves techniques including image thresholding, morphological features, active contour models (ACMs), K-means clustering, and probabilistic models. In order to distinguish between different tumor regions on osteosarcoma histology slides, Arunachalam et al. [26] demonstrated multi-level Otsu thresholding and shape segmentation. Mandava et al. [7] proposed an automatic segmentation technique of osteosarcoma using MRI images. A dynamic clustering algorithm called DCHS was proposed in their work, and it is based on a combination of fuzzy c-means (FCM) and Harmony Search (HS). To designate the tumor volume by DCHS, they used pixel intensity values and a subset of Haralick texture features as feature space. Nasor et al. [27] presented an automatic segmentation technique for osteosarcoma using MRI images combined with image processing techniques that includes K-means clustering, iterative Gaussian filtering, Chan–Vese segmentation, and Canny edge detection. An enhanced graph-cut-based framework was introduced by Vandana et al. [28] to determine malignancy level in H&E-stained histopathology images. They used mathematical morphology, color-based clustering, and active contour for extracting feature, and analyzed those features for malignancy classification using a multiclass random forest (RF) classifier. Zhi et al. [6] proposed ML approaches to classify osteosarcoma patients using metabolomics data analysis. LR, RF, and SVM are applied in their studies to distinguish between tumor cases and healthy controls. Feng et al. [29] presented a four pseudogene classifier to identify prognostic pseudogene signatures of osteosarcoma using RNA-seq data. The cox-regression analysis was used to construct the signature model (univariate, multivariate, and lasso), and achieved 0.878 AUC value.

Due to the availability of enormous computing power, DL approaches have gradually taken the place of traditional histopathological image classification [14,15,30–32]. In order to increase effectiveness and accuracy of osteosarcoma classification, Mishra et al. [10] developed a convolutional neural network (CNN). They have used WSI in their work to classify tumor classes (necrosis, viable tumor) versus nontumor class. The accuracy of their proposed CNN model was 92%, and the model was compared with three existing CNN models AlexNet, LeNet, and VGGNet. The first fully automated tool to evaluate viable

and necrotic tumors in osteosarcoma is reported by Arunachalam et al. [33] that uses both DL and conventional ML techniques. Their intention was to classify the various tissue regions as viable tumor, necrotic tumor, or nontumor. They selected 13 different ML models in their study. Among them, the support SVM was the top performer, and a DL model was also developed to train on the same dataset. SVM, ensemble learner, and complex trees achieved an overall accuracy of 80.9%, 86.8%, and 89.9% respectively, and the overall accuracy for the deep learning model was 93.3% and 91.2% for patches and tiles of WSI's. Osteosarcoma classification using histopathological images using sequential region-based convolutional neural network (R-CNN) was proposed by Nabid et al. [5] that consisted of bidirectional gated recurrent units (GRU) and CNN. Performance of their proposed model compared with AlexNet, SVM models, ResNet50, LeNet, and VGG16 on the same dataset and shows an accuracy of 89%. D'Acunto et al. [34] applied a DL approach to discriminate between Mesenchymal Stromal Cells (MSCs) from osteosarcoma cells and to classify the cell populations. A faster R-CNN was adopted in their study via transfer learning. A deep Siamese network model (DS-Net) was designed by Yu et al. [35] to develop an automated system for identifying viable and necrotic tumor regions in osteosarcoma. DS-Net was developed using a fully connected convolutional network that is combined with an auxiliary supervision network (ASN) and a classification network. Their model achieved an average accuracy of 95.1%. In order to find best classifier and to identify necrotic images from non-necrotic tissues, Anisuzzaman et al. [4] adopted six well-known pre-trained transfer learning CNN models. In their study, they employed both multiclass and binary class classification, and among the six pre-trained models, VGG-19 achieved the highest accuracy of 96% . Recently, S. Gawade et al.[36] employed multiple supervised deep-learning models to classify osteosarcoma, where they utilized a transfer learning approach that modifies only the top layer (classifier) and achieved the highest accuracy of 90.36% using ResNet. A comparative methodological approach was proposed by I.A. Vezakis et al. [37] to investigate different deep learning models. They considered various pre-trained models with transfer learning to perform normalization and resize input images into different sizes based on individual model sizes and obtained the highest accuracy of 91.00% for the MobileNetV2 model.

In recent years, ML based images processing approaches attracted a lot of interest and achieved a great success in the analysis of histopathological images of osteosarcoma. The literature survey motivated us to develop a hybrid model, combining DL and ML, to classify osteosarcoma using whole slide images. Firstly, a preprocessing technique was applied to the WSI cancer dataset to make the dataset more accurate format for analysis by the proposed method. We trained five cutting-edge CNN models to extract important features via transfer learning into a combined form of convolution and pooling from histopathological images. A decision tree-based RFE was developed to select the optimal number of features (e.g., 100, 200, . . . , 900) using a decision tree estimator from 1024 extracted features. Then, a modified MLP classifier was combined with different feature extractors with varying parameter settings for accurate prediction. Finally, we integrate the best data preprocessing, feature extractor, feature selector, and classifier to build our proposed model for predicting osteosarcoma. Here, we considered transfer learning with different hyperparameters that minimize the training time and provide more meaningful features. Moreover, feature selection techniques remove irrelevant features, thus reducing model complexity, and the modified classifier offers us more accurate classification results.

3. Methodology

This section describes dataset collection, image preprocessing, feature extraction, feature selection, and our proposed model. Figure 1 presents the schematic diagram of our proposed methodology.

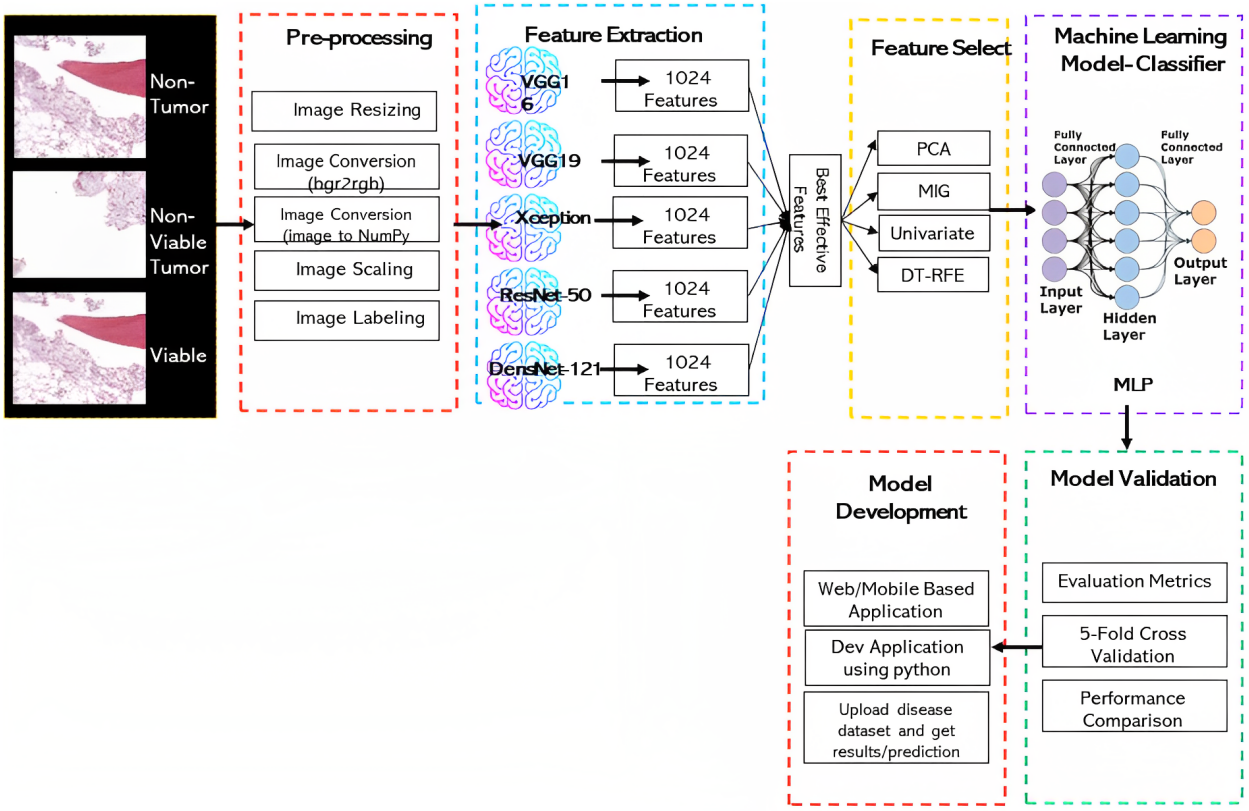


Figure 1. Proposed methodology.

The system Algorithm 1 of our proposed methodology is given as follows:

Algorithm 1 Proposed Algorithm

$E_p \leftarrow$ Number of Epochs
$W \leftarrow$ Transfer Learning Model Parameter
$\eta \leftarrow$ Learning Rate
$b_s \leftarrow$ Batch Size
$D \leftarrow$ Osteosarcoma Dataset
Output: The assessment metrics on the test dataset.
Dataset Prepossessing:
$X_{train} \leftarrow prepossessing(D)$
$X_{test} \leftarrow prepossessing(D)$
Initialise TL Models (VGG16, VGG19, ResNet50 Xception, DenseNet121)
Feature Extraction:
for *local epoch* $e_p \leftarrow$ *from 1 to* E_p **do**
  for $b_s = (x_s, y_s) \in$ *random batch from* X_{train} **do**
    Optimise model parameters
    $W_s \leftarrow W_s - \eta(\Delta(\mathcal{L}(W_s; b_s)))$
    $f_{train} \leftarrow ComputeFeatures(W_s, X_{train}, 1024)$
  end for
end for
Feature Selection :
$f_{best} \leftarrow DT - RFE(f_{train}, 900)$
Osteosarcoma Tumor Classification :
$TrainedModel \leftarrow MLP(f_{best}, y_{train})$
$Pred \leftarrow TrainedModel(X_{test})$
$Evaluation\ metrics \leftarrow ComputeMetrics(Pred, y_{test})$

Initially, the input whole slide images (WSI) were pre-processed as described in Section 3.2 and then fed into the feature extractor. We have employed five different pre-trained CNN models as our feature extractors and extracted 1024 features from every feature extractor with transfer learning techniques. Then we applied four different feature selection

techniques on those 1024 extracted features, including principal component analysis (PCA), recursive feature elimination (RFE), mutual information gain (MIG), and univariate analysis, respectively, to select the significant features. Different numbers of features (e.g., 100, 200..., 900) are chosen for each feature selector before being fed into the classifier to determine the optimal number of features. Five different ML-based classifiers, including decision tree (DT), random forest (RF), XGBoost, multi layer perceptron (MLP), and light gradient-boosting machine (LGBM), respectively, are employed as classifiers. The model was tested for binary and multiclass classification using a variety of performance metrics. To assure real-time prediction for osteosarcoma malignancy using whole slide images, we developed a web application by integrating our proposed model as well.

3.1. Dataset

Data on osteosarcomas from the work of Leavey et al. [38] were used in our study. Tumor samples were collected from Children's Medical Center, Dallas, which consists of 50 patients's pathology reports of osteosarcoma resection who were treated from 1995 to 2015. A total of 40 WSI (whole slide images) were selected where every WSI represents different sections of the microscopic slide. The WSI represents tumor heterogeneity and response properties as well. At 10X magnification factor, thirty 1024 × 1024 pixel image tiles from each WSI were randomly selected. After removing irrelevant tiles such as those falling in non-tissue, ink-mark regions, and blurry images, 1144 image tiles were selected from the resulting 1200 tiles. Each image tile is annotated by pathologists in a CSV (Comma Separated Value) file with Tile Identification Number (TIN) and its corresponding classification results. Viable tumor, nontumor, and necrotic tumor are the three main regions used in classification tasks. Among 1144 image tiles 47% (536) are nontumor tiles, 23% (263) are non-viable tumor (necrosis) tiles, and 30% (345) are viable tiles. Figure 2 illustrates sample images of the dataset. For our experiments and investigation, we have taken 80% of the data for training and 20% for testing from the dataset.

3.2. Data Pre Processing

The size of original images in our dataset was 1024 × 1024 pixels. The input for the ImageNet-based pre-trained models is less than or equal to 224 × 224. If we use transfer learning then the inputs must be suited to the pre-trained model, therefore we have resized all of our input images to 224 × 224 pixels. We transformed resized images into tensors to work with image intensity values. Then we perform normalization on our images by using this formula:

$$x = (x - mean)/std \quad (1)$$

Here, x stands for the input image that is converted into a tensor representing the pixel intensity, $mean$ is the the average pixel intensity of all images that exist in our dataset and std stands for standard deviation. Normalization enables data distribution that resembles a zero-centered Gaussian curve. By applying normalization, the gradient does not go out of control and makes convergence faster while training. Since we are using RGB images, we have used mean and standard deviation values of 0.5, 0.5, and 0.5 for the red, green, and blue channels, respectively. This resulted in image intensity values in the range of $(-1, 1)$. The preprocessing steps help us to train faster and reduce computational expenses. Figure 3 illustrates our preprocessing process.

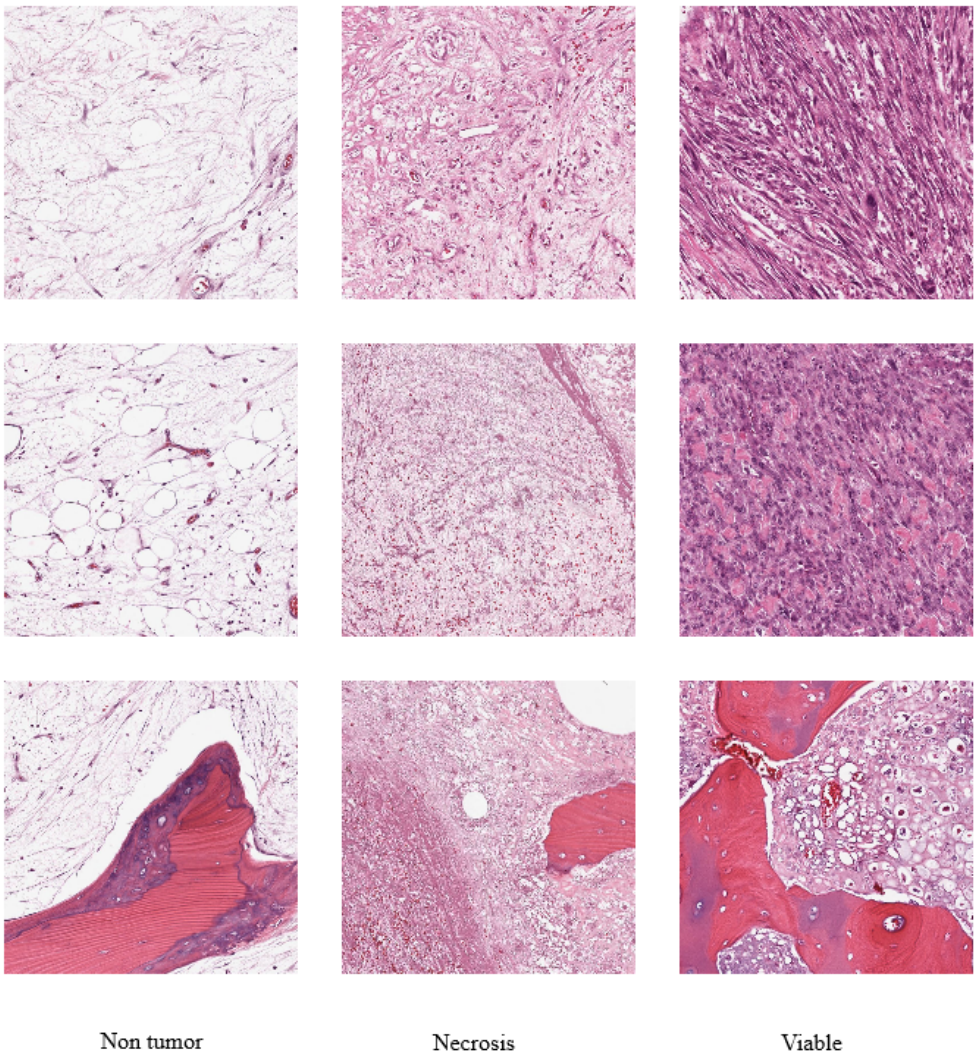


Figure 2. Sample images from our dataset [38].

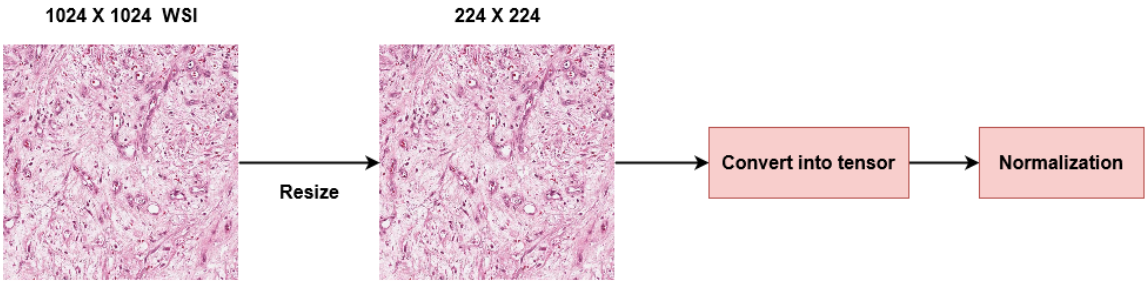


Figure 3. Data preprocessing stages where images were resized then performed normalization.

3.3. Model Selection

3.3.1. Feature Extraction (Deep Learning-Based Feature Extraction)

Feature extraction is the process by which we can extract meaningful information from an input image. DL-based feature extraction mainly uses CNN to extract features from images [39–42]. In CNN, convolution combined with pooling is utilized as a feature extractor. This study uses five pre-trained CNN models named VGG-16, VGG-19, Xception, ResNet-50, and DenseNet-121 as feature extractors via transfer learning. These pre-trained models were implemented by PyTorch [43] and Keras [44] on ImageNet [45] validation set. The base results of those models on ImageNet validation set is illustrated in Table 1.

Here top-1 accuracy is the conventional accuracy (the one with the highest probability), and top-5 accuracy means the model's top 5 highest probability answers that match with the expected answer (considers a classification is correct of any of the five predictions matches with the ground truth/target label).

Table 1. Parameters and accuracy of pre-trained models (acc. stands for accuracy).

CNN Model	Parameters	Top-1 acc.	Top-5 acc.
VGG-16	138.4M	71.592%	90.382%
VGG-19	143.7M	72.376%	90.876%
Xception	22.9M	79.0%	94.5%
ResNet-50	25.6M	76.13%	92.862%
DenseNet-121	8M	74.434%	91.972%

3.3.2. VGG-16 and VGG-19

Karen Simonyan and Andrew Zisserman from the University of Oxford proposed VGG Net [46], which took first and second place in the object detection and classification categories of the 2014 ImageNet challenges. VGG Net architecture has two variants in terms of layers, and the variations are VGG-16 and VGG-19.

VGG-16 is a deep CNN model which consists of 16 layers (roughly twice as deep as AlexNet [47]), constructed by stacking uniform convolution, which enhances the network performance. Without using relatively large respective fields in the first convolution layer (e.g., 11×11 with stride 4 in Krizhevsky et al. [47] or 7×7 with stride 2 in Zeiler and Fergus et al. [48]; Sermanet et al. [49]), they used very small (3×3) respective fields throughout the network. A stack of respective small filters (3×3) has been used instead of large $(7 \times 7$ or $11 \times 11)$ respective filters because respective small filters make the decision function more discriminative and reduce the number of parameters, allowing for less computational complexity. The 16 in VGG-16 stands for 16 weighted layers known as learnable parameter layers. A total of 21 layers make up VGG-16: 13 convolutional layers, 5 Max Pooling layers, and 3 Dense layers. This model uses ReLU as the activation function following convolution. In the pooling layer, a max pool layer of 2×2 filter with stride 2 has been used throughout the whole architecture. A stack of convolutions is followed by three fully connected layers, the third one having 1000 channels for classification and the first two each have 4096 channels. The dropout value is set to 0.5 for regularization, and Softmax is used as the activation function for classification. The model's default input tensor size is 224×224 with 3 RGB channels.

VGG-19 is deeper than VGG-16 as it has 19 layers. It has 16 convolution layers, 5 Max Pooling layers, 3 dense layers, which is a total of 24 layers that make up VGG-19. The 3rd, 4th, and 5th convolution of VGG-19 has an extra layer over VGG-16 and the other architectures are the same as VGG-16 i.e., kernel size, stride, padding, pooling, dropout probability, and activation function. It has a much larger number of parameters than VGG-16.

3.3.3. Xception

Francois Chollet introduced Xception from Google research [50]. The architecture is inspired by Inception and entirely based on depthwise separable convolution, where depthwise separable convolution has been used in place of the Inception module [51]It is based on a solid hypothesis that performs 1×1 convolution to map cross-channel correlations and separately map the spatial correlations of every output channel. This model performs channel-wise spatial convolution followed by a 1×1 convolution to achieve depth-wise separable convolution. The network contains 36 convolutional layers, which form the feature extraction base, and a logistic regression layer is used after the convolutional base

for classification. Xception has 14 modules that are made up of 36 convolutional layers, and all of them have a linear residual connection around them except the first and last modules. A global average pooling layer is used at the top layer of this architecture to produce a 1×2048 vector, and several fully-connected layers are kept optional. Here, ReLu has been used as the activation function for non-linearity, and a dropout layer of rate 0.5 has been used before the logistic regression layer in this network. The architecture of this network reduces the number of connections by using depth-wise separable convolution and thus reduces the number of parameters, making it computationally more efficient.

3.3.4. ResNet-50

ResNet was proposed by Kaiming He from Microsoft research [52]. ResNet's architecture is based on residual learning and is substantially deeper than previous models. Instead of learning unreferenced functions, ResNet explicitly reformulates the layers as learning residual functions. Deeper networks often face a notorious problem of vanishing gradients that hamper convergence [53,54]. ResNet addressed this phenomenon by normalizing initialization and utilizing intermediate normalization layers that enable networks with more layers to converge with backpropagation. ResNet also introduces a deep residual learning approach to overcome this degradation issue. Instead of using a few layers directly, this network uses a residual mapping to fit the underlying mapping by reformulating the residual function $F(x) := H(x) - x$ into $F(x) + x$ (where $H(x)$ is underlying mapping, x is input). To formulate $F(x) + x$, a shortcut connection (skipping one or more layers) has been used. The shortcut connections perform identity mapping, then add their results with the outputs from the stacked layers. ResNet-50 is a variant of ResNet that is a modified version of ResNet-34 with a bottleneck architecture and 50 layers. A bottleneck block contains a stack of 3 layers, which are 1×1, 3×3, and 1×1 convolutions. This architecture uses a batch normalization layer between the convolution and activation layers. ReLU has been used as an activation function, and the dropout layer has not been considered. A global average pooling layer and a fully connected layer of 1000 nodes with softmax are used at the end of this network.

3.3.5. DenseNet-121

DensNet was proposed by Gao Huang et al. [55], and this model enhances feature reuse capabilities based on ResNet in its architecture. It has L(L + 1)/2 direct connections, whereas traditional CNN has L layers with L connections. In DensNet, feature maps are combined using concatenation instead of summing before passing into a layer, and all previous layer's feature maps are used as input for any specific layer. The Dense Block is the main structure of DensNet, consisting of convolutional layers. DensNet-121 is one of the variants of the DensNet architecture, having a 121-connected convolutional layer with a final output layer. DensNet-121 contains four dense blocks, and there is a transition layer between each dense block. This network's dense connectivity for x_0, x_1, x_{l-1} inputs, where the lth layer receives feature maps from all preceding layers, can be defined as $X_l = H_l([x_0, x_1, ..., x_{l-1}])$. H_l is a composite function that contains three operations: batch normalization (BN), rectified linear unit (ReLU), and a 3×3 convolution. Each dense block of DenseNet-121 has two convolutions, 1×1 and 3×3, which are repeated differently in each block. A transition layer contains a 1×1 convolutional layer and an average pooling layer with a stride of 2. Before sending all feature maps to the fully connected layer for classification, this network performs a 7×7 global average pooling layer. This network has fewer parameters than ResNet and is more computationally efficient.

3.4. Decision Tree Based Recursive Feature Elimination (DT-RFE)

Recursive feature elimination (RFE) is a wrapper-type feature selection technique that uses different types of machine learning algorithms in its core, and the algorithms help to select features [56–58]. RFE fits a model and removes the least significant feature (or features) until the desired(selected) number of features is obtained. The coef or feature

importances properties of the model are used to rank the features [59], and RFE attempts to eliminate interdependence and correlation that may exist in the model by recursively eliminating a small number of features per cycle. The goal of RFE is to maximize generalization performance by eliminating the least significant features whose elimination will have the least impact on training errors and select smaller sets of features recursively [60]. There are two major steps that must be taken to implement RFE. Firstly, we need to choose an algorithm (also known as a classifier or estimator) that will give us feature importance, and then we need to specify the number of features we want to select. We have used the decision tree algorithm [61] as our estimator, and different numbers of features were selected for our experiment, i.e., 100, 200, 300, 400, 500, 600, 700, 800, and 900. Decision tree is a tree-based classifier that offers a variety of significance features and performs relatively well. Decision tree algorithms employ information gain to split a node, and for calculating information gain, different criteria can be used [62]. In our experiment two popular criteria have been employed, namely Gini index and entropy to determine which criterion provides better performance based on the data. Mathematically Gini index and entropy can be defined as follows:

$$Gini = 1 - \sum_{i=1}^{n} p^2(c_i) \tag{2}$$

$$Entropy = \sum_{i=1}^{n} -p(c_i) log_2(p(c_i)) \tag{3}$$

where $p(c_i)$ is the probability of class c_i in a node. The range of the Gini Index is [0, 0.5], while the range of the entropy is [0, 1]. We have applied multi layer perceptron (MLP) separately on selected features for our final classification.

3.5. Multi Layer Perceptron (MLP)

Multi layer perceptron, in short MLP, is a unique variety of an artificial neural network (ANN) [63]. MLP is a feed-forward multilayer network of artificial neurons, and each layer contains a finite number of units (often called neurons) [64–66]. Each layer's unit is connected to each layer's preceding (and consequently succeeding) unit via a network of connecting lines. Typically, these connections are referred to as links or synapses [67]. Information transmits from one layer to the next layer (thus the term feed-forward). For $x_1, x_2, ..., x_n$ inputs, the model predicts output as $y^1, y^2, ..., y^n$ with lh hidden nodes or units (h is the number of nodes). In this study, MLP model works as follows:

1. The input layer produces output of its jth node as x_{oj}.
2. The output x_{ij} from each jth node of the $(i-1)$th layer is sent to the kth node of the ith layer. Then the values of x_{ij} are multiplied by some constants (referred to as weights) w_{ijk}, and the resulting products are summed.
3. A shift b_{ik}(referred to as bias) and then a fixed mapping σ(referred to as activation function) are applied to the above sum, and the resulting value represents the output $x_{i+j,k}$ of this kth node of the ith layer. This can be formulated as follows:

$$x_{i+1,k} = \sigma(\sum_{j} w_{ijk} x_{ij} + b_{ik}) \tag{4}$$

With the above procedure, for input $x = (x_1, x_2, ..., x_n)$, we can write the output $\hat{y}$ of a single hidden layer perceptron model with q nodes in the hidden layer as follows:

$$\hat{y} = \sum_{i=1}^{q} w_i^2 . \sigma(\sum_{j=1}^{n} w_{ij}^1 x_j + b_i) \tag{5}$$

Here, w_{ij}^1 is the weight of jth unit of the input and ith unit in the hidden layer, b_i is the bias at the ith unit of the hidden layer, and w_i^2 is the weight between the ith unit of the hidden layer and the output.

Another step is to determine the values of weights w_{ij} and bias b_i in a way that the model behaves well on a given set of inputs and corresponding outputs. This process is called learning or training, and the MLP model uses backpropagation as the basic learning algorithm. Backpropagation is a gradient descent algorithm and mathematically it can be represented as,

$$repeat\ unti\ convergence : w_j := w_j - \alpha . \frac{\delta}{\delta w_j} . J(w_0, w_1, ..., w_n) \tag{6}$$

where w_j is weights, α is learning rate, and J is the cost function. Cost function basically quantifies the error between the predicted value and the true value of inputs, and mathematically it can be represented as follows:

$$J(w_0, w_1) = \frac{1}{2m} . \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 \tag{7}$$

y_i is the actual value, $\hat{y}_i$ is the predicted value, and m is the number of data samples.

Different activation functions can be used in different layers on MLP. In our experiment we have used ReLU as an activation function in the hidden layer. Mathematically ReLU can be defined as follows for input x:

$$f(x) = max(0, x) \tag{8}$$

In the output layer, we have used different activation functions for binary and multi-class classification, respectively. For binary classification we used logistic sigmoid activation function in the output layer, and mathematically it can be defined as follows for input x:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{9}$$

Additionally, for multi class classification we have used softmax as activation function of the output layer. Softmax can be defined as follows for input x:

$$\sigma(x_i) = \frac{e_i^x}{\sum_{k=1}^{N} e^{x_k}} \tag{10}$$

In our proposed MLP model, the inputs $(x_1, x_2, ..., x_n)$ have 1024 features after extracting the feature, so there are 1024 nodes for each input and 100 nodes in the hidden layer. The output layer contains 2 nodes for binary classification and 3 nodes (as we have 3 classes) for multiclass classification. We have used ReLU as the activation function in the hidden layer, logistic sigmoid as the activation function for binary classification and softmax for multiclass classification in the output layer. Furthermore, Adam was used as an optimizer, and it is stochastic gradient-based. $beta_1 = 0.9$, $beta_2 = 0.999$, $epsilon = 1 \times 10^{-8}$ has been used for the Adam optimizer. We applied L2 regularization with a value of $\alpha = 0.0001$ with a learning rate of 0.001; we trained our model for 200 epochs.

4. Experimental Results

The experiments were conducted in the Google Colaboratory environment that includes the NVIDIA Tesla K80 graphics card, 12.68 GB RAM, and 107.72 GB disk space. To implement our proposed model, the Python 3.8.3 programming language with PyTorch, Keras, Scikit-learn frameworks, and various libraries such as Numpy, Pandas, Matplotlib, etc. has been utilized.

4.1. Evaluation

The main objective of our proposed model was to classify the osteosarcoma images into one of the three tumor phases (nontumor, necrosis, and viable tumor) as mentioned in the earlier section. In this study, we employed various performance metrics to evaluate our proposed model. Moreover, the impact of the feature selection technique is also analyzed by a comparison of the results before and after applying it based on various performance metrics. Here, accuracy, ROC curve, specificity, sensitivity (recall), precision, F1 score, Matthews correlation coefficient (MCC), and confusion matrix were all considered in the evaluation. Confusion matrix is a table that describes how well a classification algorithm performs, and it visualizes and summarizes the prediction results for a classification problem. In a confusion matrix where true positive (*TP*) stands for a value that is correctly predicted as positive, true negative (*TN*) stands for a value that is correctly predicted as negative, false positive (*FP*) indicates a value incorrectly predicted as positive, and false negative (*FN*) indicates a value incorrectly predicted as negative. Mathematically all these evaluation metrics can be written as follows:

$$Accuracy = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) \times 100\% \tag{11}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{12}$$

$$Specificity = \frac{TN}{TN + FP} \tag{13}$$

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$F1\,score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{15}$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{16}$$

We have also considered ROC curves, which represent two-dimensional charts that are frequently used to evaluate and assess the performance of classifiers. It simply illustrates a classifier's sensitivity or specificity for all possible classification thresholds and indicates how effectively the model can distinguish between different categories. The true positive rate is plotted on the *y*-axis and the false positive rate is plotted in the *x*-axis in this graph, and an AUC close to 1 implies a predicted model does well at class label separability, while an AUC close to 0 indicates a poor predicted model.

4.2. Feature Extractor and Classifier Selection

To determine the best feature extractor, we employed five different CNN models, including VGG-16, VGG-19, ResNet-50, DenseNet-121, and Xception. Here, we considered a transfer learning approach rather than training a CNN model from the scratch where pre-trained weights of those models are utilized. The Fully Connected Layer (used as the classifier of a model) of every CNN model was discarded and replaced with five different classifiers based on the ML algorithm. DT, RF, XGBoost, LGBM, and MLP are the algorithms that have been used as classifiers individually with every CNN. The purpose of this experiment with a combinational approach is to investigate and compare the performance of each feature extractor with different ML classifiers. This experiment also allows us to determine the best classifier among all of the ML based classifiers mentioned earlier in this section. A dataset containing tumor samples from patients with osteosarcoma is used to evaluate each combined model; a description of this dataset is given above (see Section 3.1). The investigation was evaluated on the test set, and from this investigation, we select the

best feature extractor and the best classifier, as well as the best combination based on their performance. Table 2 represents the experimental results of every combination.

Table 2. Experimental results of the performance of feature extractor with different classifier.

Extractor	Classifier	ACC	AUC	MCC	SP	SN
VGG-16	Decision Tree	0.680	0.768	0.497	0.802	0.650
	Random Forest	0.627	0.877	0.437	0.743	0.580
	XGBoost	0.811	0.921	0.705	0.890	0.791
	LGBM	0.759	0.901	0.631	0.845	0.700
	MLP	0.908	0.974	0.855	0.950	0.890
VGG-19	Decision Tree	0.684	0.814	0.505	0.805	0.660
	Random Forest	0.737	0.912	0.602	0.836	0.700
	XGBoost	0.838	0.930	0.747	0.906	0.820
	LGBM	0.768	0.913	0.649	0.857	0.720
	MLP	0.895	0.975	0.838	0.946	0.890
Xception	Decision Tree	0.636	0.786	0.418	0.766	0.610
	Random Forest	0.781	0.916	0.649	0.861	0.720
	XGBoost	0.803	0.928	0.684	0.880	0.760
	LGBM	0.803	0.934	0.685	0.882	0.760
	MLP	0.820	0.938	0.713	0.895	0.800
Resnet-50	Decision Tree	0.776	0.762	0.648	0.871	0.739
	Random Forest	0.715	0.892	0.553	0.823	0.680
	XGBoost	0.842	0.948	0.755	0.916	0.830
	LGBM	0.803	0.943	0.699	0.879	0.760
	MLP	0.877	0.973	0.812	0.938	0.870
DenseNet-121	Decision Tree	0.825	0.797	0.725	0.901	0.820
	Random Forest	0.825	0.944	0.812	0.930	0.860
	XGBoost	0.895	0.957	0.839	0.941	0.900
	LGBM	0.829	0.955	0.740	0.897	0.814
	MLP	0.934	0.989	0.913	0.966	0.940

From the experimental results shown in Table 2, we can see that DenseNet-121 combined with all five classifiers achieved the highest average accuracy of 86.16% among all of the feature extractors. VGG-16 obtained the lowest average accuracy among all of them, which is 75.7%. The three other feature extractors, including VGG-19, Xception, and ResNet-50, obtained an average accuracy of 78.44%, 76.86%, and 80.26%, respectively, which are 7.72%, 9.3%, and 5.9% lower than DenseNet-121, respectively. The highest average AUC score is also achieved by DenseNet-121, which is 92.84%. The average AUC scores of DenseNet-121 are 4.02%, 1.96%, 2.8%, and 2.48% higher than the VGG-16, VGG-19, Xception, and ResNet-50 models, respectively. This extractor also achieved the highest average score for other evaluation metrics, including MCC, specificity, and sensitivity, which are 80.58%, 92.7%, and 86.68%, respectively. The average MCC scores of DenseNet-121 are 18.08%, 13.76%, 17.6%, and 11.24% higher than the VGG-16, VGG-19,

Xception, and ResNet-50 models, respectively. The specificity and sensitivity are also 8.1%, 5.7%, 7.02%, and 4.16% and 14.46%, 10.88%, 13.68%, and 9.1% higher than the other four feature extractors. When compared to other extractors, these differences in results are quite significant, and DenseNet-121 achieved the highest score in every evaluation metric aspect. DenseNet-121 also achieved the third highest top-5 accuracy of 91.97% on the ImageNet validation dataset and used much fewer parameters than others among all of the mentioned feature extractors (a descriptive overview is provided in Section 3). In our case, DenseNet-121 outperforms all other CNN models on the test dataset for every evaluation metric. Therefore, we have chosen DenseNet-121 as our feature extractor.

From Table 2, we can also see that the MLP classifier achieved the highest average accuracy of 88.68%, which is 16.66%, 14.98%, 4.9%, and 9.44% higher than four other classifiers, including DT, RF, XGBoost, and LGBM, respectively. The MLP classifier also achieved the highest average AUC score of 96.98%, which is 18.44%, 6.16%, 3.3%, and 4.06% higher than the other four mentioned classifiers. The DT, RF, XGBoost, and LGBM obtained average MCC scores of 55.86%, 61.06%, 74.6%, and 68.08%, respectively, while MLP achieved the highest average score of 82.62%. The MCC scores of the other four classifiers are lower than the MLP classifier. The MLP classifier also achieved the highest average score for other evaluation metrics, including specificity and sensitivity, which are 93.9% and 87.8%, respectively. The second-highest average scores for ACC, AUC, MCC, SP, and SN are obtained by the XGBoost classifier, which is 4.9%, 3.3%, 8.02%, 3.24%, and 5.78% lower compared with the MLP classifier; the differences are quite significant. From our investigation, we found that the MLP classifier outperformed all other classifiers that we used in our experiment on our test data. These findings allow us to select MLP as the best classifier. Figure 4 illustrates the ROC-AUC curve for each feature extractor combined with five different classifiers.

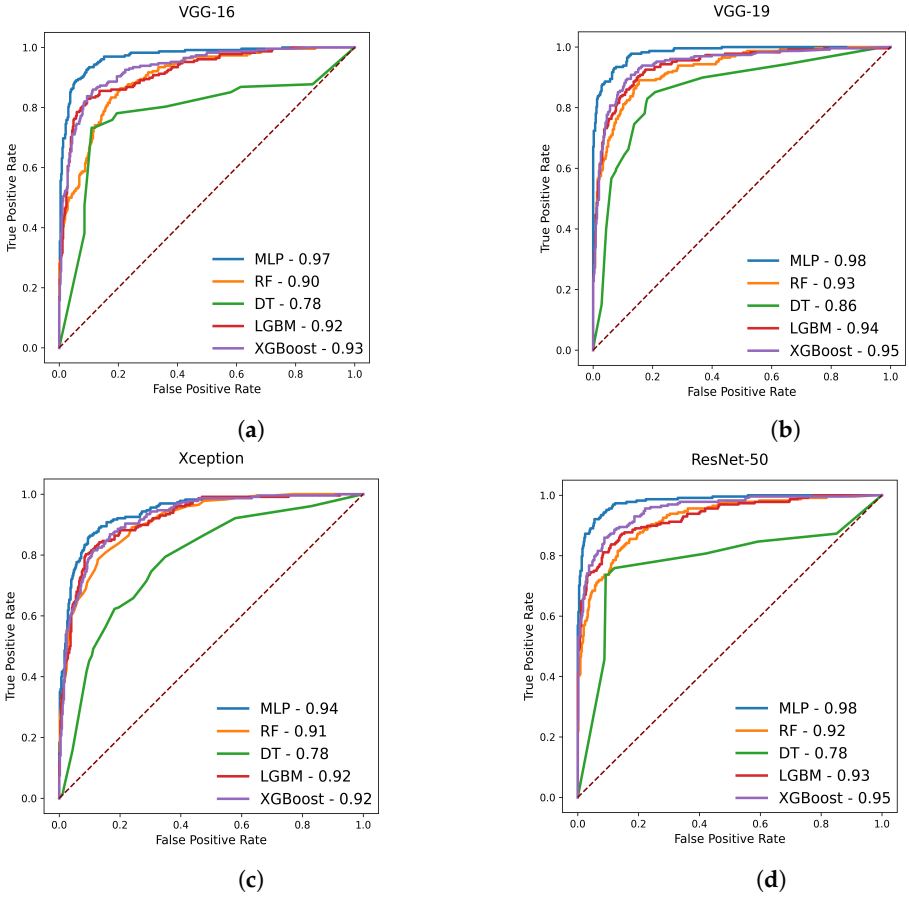
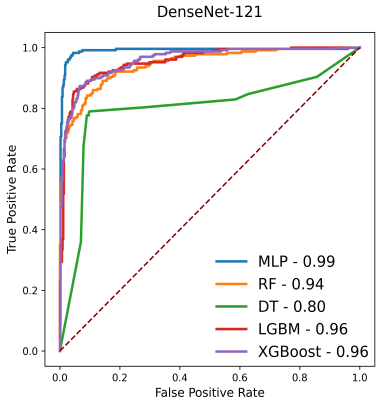


Figure 4. *Cont.*



(**e**)

Figure 4. ROC-AUC for five feature extractor. (**a**) ROC-AUC for VGG-16; (**b**) ROC-AUC for VGG-19; (**c**) ROC-AUC for Xception; (**d**) ROC-AUC for ResNet-50; (**e**) ROC-AUC for DenseNet-121.

The experimental results and our investigation also illustrate that the MLP classifier combined with every feature extractor achieved the highest ACC, AUC, MCC, SP, and SN scores among all other combinations. On the other hand, every classifier combined with DenseNet-121 obtained the highest scores compared with other feature extractors. Finally, we can see that DenseNet-121 combined with the MLP classifier achieved the highest accuracy of 93.4%, which is much higher than all other combinations. By this finding, we have chosen DensNet-121 as a feature extractor and MLP as a classifier, and we have applied this combination to develop our proposed model.

4.3. Optimizer Algorithms, Loss, and Convergence Analysis of MLP

The effectiveness and efficiency of optimization algorithms have a significant impact on the implementation of ML models. They generate gradients and try to minimize the loss function that leads to more accurate results. There are many different optimization algorithms that can be implemented to minimize loss in a ML or DL model for supervised, unsupervised, semi-supervised, and reinforcement learning. In our study, three different optimization algorithms have been employed to determine which optimization algorithm works better on our data for MLP, and those are named Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam), and Limited-memory BFGS (Lbfgs), respectively. Table 3 represents the optimization algorithms performance on our data with learning rate, number iteration to convergence, loss, and execution time for both multiclass and binary class classification.

Our experimental results show that SGD takes a higher number of iterations to converge than Adam and Lbfgs. After a several number of experiments we found that SGD takes around 900 iterations to convergence while Adam and Lbfgs takes 500 and 300 iterations, respectively. To investigate the loss value and execution time for each optimization algorithm, we set a certain number of iterations for each of them for both multiclass and binary classification with an initial learning rate of 0.001. The experimental results indicate that Adam and Lbfgs produce both lower loss value and execution time than SGD. In multiclass classification, Adam, Lbfgs, and SGD produce loss values of 0.002621, 0.020567, and 0.000057 with execution times of 3.98 s, 57.15 s, and 2.40 s, respectively. Binary classification's experimental results also indicate that SGD takes higher execution time, loss value, and number of iteration than Adam and Lbfgs. This investigation motivates us to use Adam as our optimization algorithm for our proposed model as it is more computationally efficient and produces less loss value. Furthermore, we plotted the optimizer analysis for both multiclass and binary class classification based on the no. of iteration, loss, and execution time that is illustrated in Figure 5.

Table 3. Experimental results of optimizer analysis.

Classification	Optimizer Name	LR	Iteration	Loss	Time (s)
Multi class	Adam	0.001	500	0.002621	3.98
	SGD	0.001	900	0.020567	57.15
	Lbfgs	0.001	300	0.000057	2.40
Nontumor vs. Necrosis + Viable	Adam	0.001	500	0.001610	6.09
	SGD	0.001	900	0.020836	27.80
	Lbfgs	0.001	300	0.000153	4.54
Nontumor vs. Necrosis	Adam	0.001	500	0.002714	9.42
	SGD	0.001	900	0.064409	73.59
	Lbfgs	0.001	300	0.000080	3.95
Necrosis vs. Viable	Adam	0.001	500	0.002406	9.18
	SGD	0.001	900	0.040208	38.84
	Lbfgs	0.001	300	0.000114	5.22
Nontumor vs. Viable	Adam	0.001	500	0.003227	4.94
	SGD	0.001	900	0.051609	40.33
	Lbfgs	0.001	300	0.000098	3.821

4.4. Impact of Feature Selection Techniques

Feature selection (FS) is an effective strategy for choosing the most appropriate feature subset in pattern recognition and medical image processing. This technique helps us eliminate irrelevant features that allow us to build a more straightforward and faster model with higher prediction capability. In recent studies, various feature selection techniques have been widely used in medical image processing and bioinformatics. To investigate the effectiveness of FS techniques, we employed four different feature selectors, namely PCA, RFE, MIG, and univariate, to determine the most effective one that works best in our dataset. We evaluate our proposed model on the test set, which contains 224 images belonging to three different classes. Here, 100, 200, 300, 400, 500, 600, 700, 800, and 900 features were selected individually for each FS technique to determine which number of features yielded the best prediction scores across all evaluation metrics. Both multiclass and binary-class classifications were performed to ensure each FS technique's impact. Table 4 illustrates the data samples distributed for different classification tasks.

Table 4. Number of samples in each class in our test dataset (our evaluation based on this data). NT = Nontumor, Nec. = Necrosis, Via. = Viable.

Classification	NT	Nec.	Via.	Nec. + Via.	Total
Multiclass	118	56	54	-	228
NT vs. Nec. + Via.	118	-	-	110	228
NT vs. Nec.	118	56	-	-	174
NT vs. Via.	118	-	54	-	172
Nec. vs. Via.	-	56	54	-	110

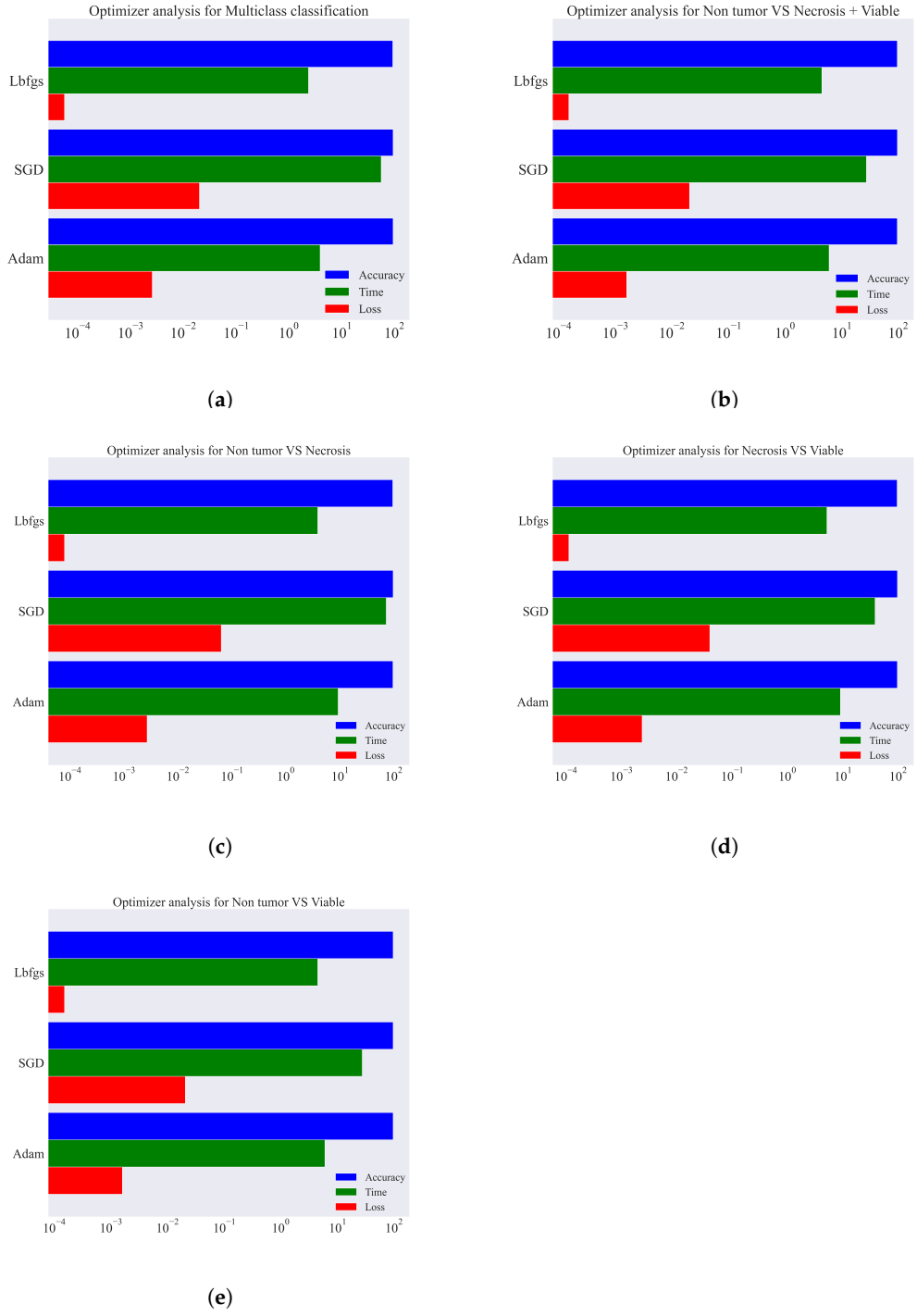


Figure 5. Optimizer analysis for both multiclass and binary class classification based on no. of iteration, loss, and execution time. (**a**) Analysis for multiclass. (**b**) Analysis for NT vs. Nec. + Via. (**c**) Analysis for NT vs. Nec. (**d**) Analysis for Nec. vs. Via. (**e**) Analysis for NT vs. Via.

Table 4 shows that we performed a single multiclass classification and four different binary class classifications. Necrosis and viable samples were combined into a class for nontumor versus necrosis + viable binary classification and added nontumor as another class. This binary classification aims to investigate how well our proposed model can classify a tumorous and a nontumorous sample. We also conducted three class-specific binary classifications to ensure that our proposed model can discriminate between two classes in all possible combinations. Initially, multiclass and binary class classification were per-

formed without FS techniques where applied the MLP classifier to 1024 features extracted from DenseNet-121. The results are illustrated in Table 5.

Table 5. Experimental classification results without feature selection.

Classification	No. of Features	ACC	AUC
Multiclass	1024	0.934	0.985
NT vs. Nec. + Via.	1024	0.947	0.993
NT vs. Nec.	1024	0.954	0.983
NT vs. Via.	1024	0.971	0.997
Nec. vs. Via.	1024	0.936	0.978

The experimental results of four different feature selection techniques that we used to classify osteosarcoma malignancy for both multiclass and binary class classification are shown in Table 6. This table includes the experimental results for those feature dimensions that achieved the highest performance.

Table 6. Experimental results of feature selection using different feature selection techniques.

Classification	Algorithm	No. of Feat.	ACC	AUC	SP	SN	MCC	Prec.	F1 Score
Multi class	PCA	100	0.943	0.986	0.970	0.930	0.907	0.930	0.930
	RFE	900	0.952	0.987	0.973	0.943	0.922	0.950	0.950
	MIG	400	0.943	0.986	0.986	0.936	0.908	0.940	0.940
	Univariate	700	0.952	0.987	0.973	0.943	0.922	0.950	0.950
Nontumor vs. Nec. + Via.	PCA	200	0.969	0.993	0.969	0.970	0.939	0.970	0.970
	RFE	100	0.969	0.990	0.969	0.970	0.939	0.974	0.970
	MIG	600	0.969	0.993	0.969	0.970	0.939	0.970	0.970
	Univariate	200	0.969	0.993	0.961	0.970	0.939	0.970	0.970
Nontumor vs. Necrosis	PCA	500	0.960	0.976	0.947	0.945	0.907	0.960	0.950
	RFE	100	0.966	0.979	0.956	0.955	0.921	0.960	0.960
	MIG	700	0.966	0.988	0.956	0.955	0.921	0.960	0.960
	Univariate	500	0.966	0.979	0.952	0.950	0.921	0.970	0.960
Nontumor vs. Viable	PCA	700	0.994	1.000	0.996	0.995	0.987	0.990	0.990
	RFE	600	0.994	0.998	0.996	0.995	0.987	0.990	0.990
	MIG	700	0.994	0.998	0.996	0.995	0.987	0.990	0.990
	Univariate	300	0.988	0.999	0.991	0.990	0.978	0.980	0.990
Necrosis vs. Viable	PCA	400	0.955	0.986	0.954	0.950	0.909	0.950	0.950
	RFE	300	0.955	0.977	0.954	0.950	0.909	0.950	0.950
	MIG	800	0.945	0.981	0.945	0.945	0.891	0.950	0.950
	Univariate	400	0.955	0.977	0.954	0.950	0.909	0.950	0.950

From Table 6, we can see that the average accuracy of five different classification tasks without FS technique is 94.84%, which is 1.58%, 1.88%, 0.5%, and 1.76% lower than the average accuracy of PCA, RFE, MIG, and univariate. This observation shows that FS techniques can improve prediction results significantly on our dataset. The four FS techniques, including PCA, RFE, MIG, and univariate, obtained an average accuracy

of 96.42%, 96.72%, 96.34%, and 96.60%, respectively. Among all of the FS techniques, RFE achieved the highest average accuracy, which is 0.30%, 0.38%, and 0.12% higher than PCA, RFE, MIG, and univariate, respectively. The RFE also achieved the highest average score for MCC, precision, and sensitivity, and those scores are 93.56%, 96.48%, and 96.26%, respectively. The highest average AUC achieved by the MIG technique is 98.72%. The RFE obtained an average AUC score of 98.62%, which is slightly lower than the MIG (0.1%). The specificity and F1 scores are more or less the same for every FS technique. The experimental results show that some FS techniques achieved the highest average accuracy on a single evaluation metric, some prediction results have a slight difference from each other, and some predictions are the same for all of the techniques. However, based on various evaluation metrics, we discovered that the RFE FS technique consistently outperformed all of the others. We also investigate the performance based on the number of features. PCA, RFE, MIG, and univariate used an average of 380, 400, 640, and 420, respectively, to obtain their best prediction results. Though PCA uses a smaller average number of features for the best prediction, it does not provide better results than RFE. From the experimental results, we can see that for multiclass classification, it uses only 100 features to predict the best results, but its accuracy is 0.9% lower than the RFE technique. PCA also uses a higher number of features than RFE in all four binary classifications, and its performance is also significantly lower. RFE and univariate use a higher average number of features than RFE to obtain their best prediction results. In the binary classification, we can see that RFE uses a smaller number of features than all other FS techniques except in the nontumor versus viable tumor classification. We implemented DT-based RFE using the Scikit-Learn (sk-learn) library, where DT has been used as an estimator that has been discussed in Section 3.2. As this library offers two different criteria for the DT estimator, we also analyze its criterion based on the execution time for a more sophisticated version of DT-RFE that works on our dataset. Table 7 represents the execution time during the experiment using Gini and entropy criterion.

Table 7. Experimental results of DT-RFE criteria analysis.

Classification	Criterion	Exec. Time
Multi class	Gini (500)	5.23 m
	entropy	8.41 m
NT vs. Nec. + Via.	Gini (100)	8.61 m
	entropy	12.18 m
NT vs. Nec.	Gini (600)	3.35 m
	entropy	4.03 m
Nec. vs. Via.	Gini (300)	4.95 m
	entropy	5.46 m
NT. vs. Via.	Gini (600)	1.93 m
	entropy	2.24 m

In this table, it is shown that the entropy criterion takes much more time than the Gini index to select the most prominent features for both multiclass and binary class classification. In multiclass classification, the Gini index takes around 5.23 min while entropy takes 8.41 min, and in other binary classifications including nontumor vs. necrosis and viable, nontumor vs. necrosis, necrosis vs. viable, and nontumor vs. viable entropy takes more execution time than the Gini index. Less execution time is more computationally efficient, which motivates us to use the Gini index as our criterion for the decision tree estimator in the RFE feature selection technique. We also plotted a violin plot for our selected feature selector that is shown in Figure 6.

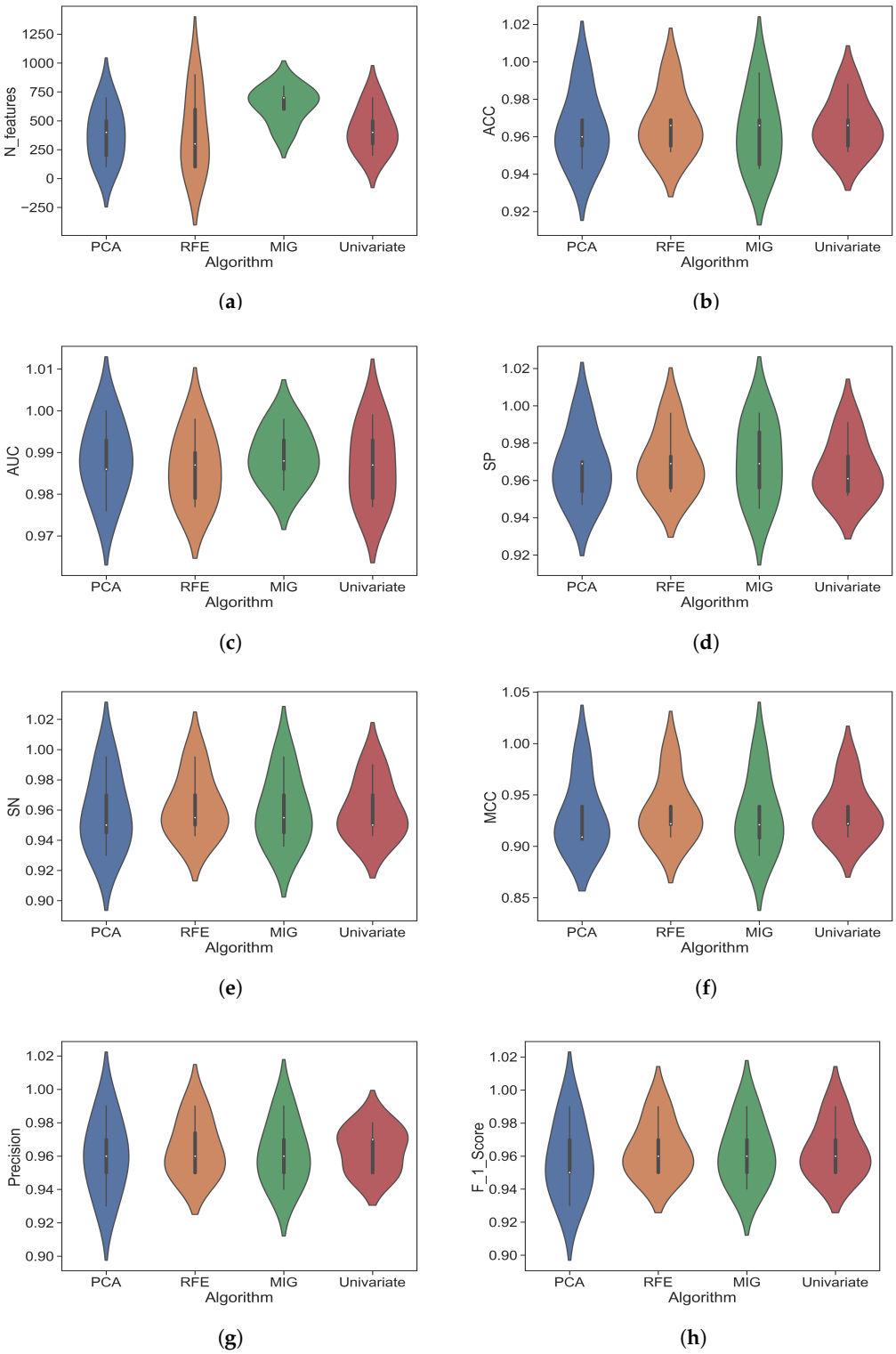


Figure 6. Violin plot the experimental results after applying 4 different feature selection techniques. (**a**) For number of features. (**b**) For accuracy. (**c**) For AUC. (**d**) For specificity. (**e**) For sensitivity. (**f**) For Matthew's correlation coefficient. (**g**) For precision. (**h**) For F1 score.

In addition to our study, we applied gradient-weighted class activation mapping (Grad-CAM) [68] to further analyze and explain the feature extractor of our proposed model. All convolutional layers in a CNN retain their respective spatial information that is lost in the FC layer. Grad-CAM uses the gradient information flowing into the last convolutional layer

of the CNN to assign importance values to each neuron for a particular decision of interest, as this last layer contains high-level semantics and detailed spatial information of an input image. This method computes the gradient of the target class score with respect to the last layer's feature maps, weights computed gradients by average pooling for the importance of each feature map channel, and finally combines weighted gradients to generate a heat map that illustrates the feature importance. We have overlayed this generated heatmap with input images to obtain the Grad-CAM images. Figure 7 represents some sample Grad-CAM images from our dataset that have been utilized as input images. By visualizing those images, we can investigate the decision-making process of our proposed model, where the red regions are the affected areas considered by the model.

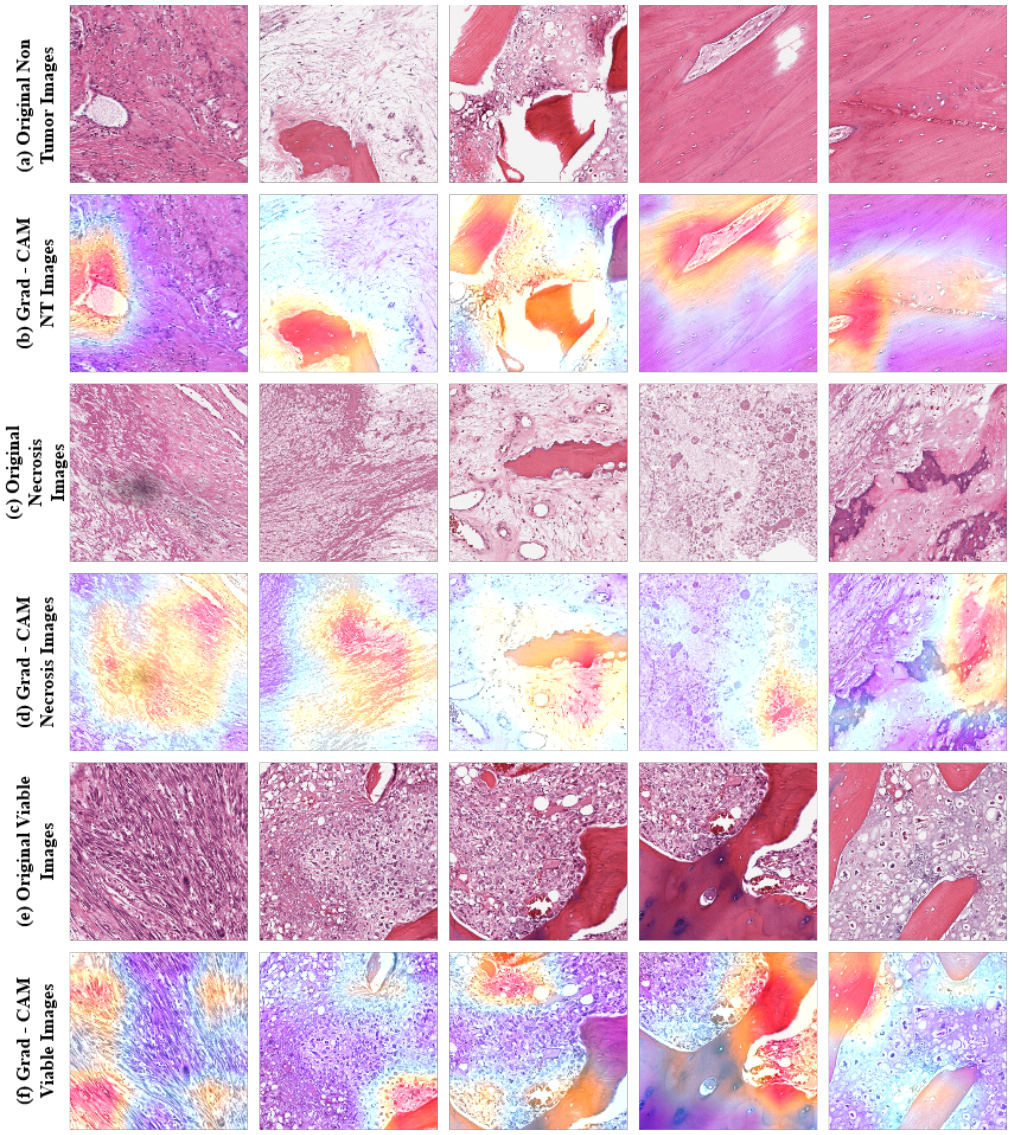


Figure 7. Example of some input images and their Grad-CAM images generated by proposed model, where red regions indicated the affected area.

4.5. Web Application for Osteosarcoma Classification

A web application is developed using our proposed model with real-time validation to classify osteosarcoma using whole slide images as input. A modern, fast (high-performance) web framework named FastAPI [69] has been used to develop our web application on the Python 3.8.3 version. FastAPI is used for building APIs and backend development, and we have used HTML, CSS, and JavaScript for frontend development. The workflow of our web application is given in Figure 8.

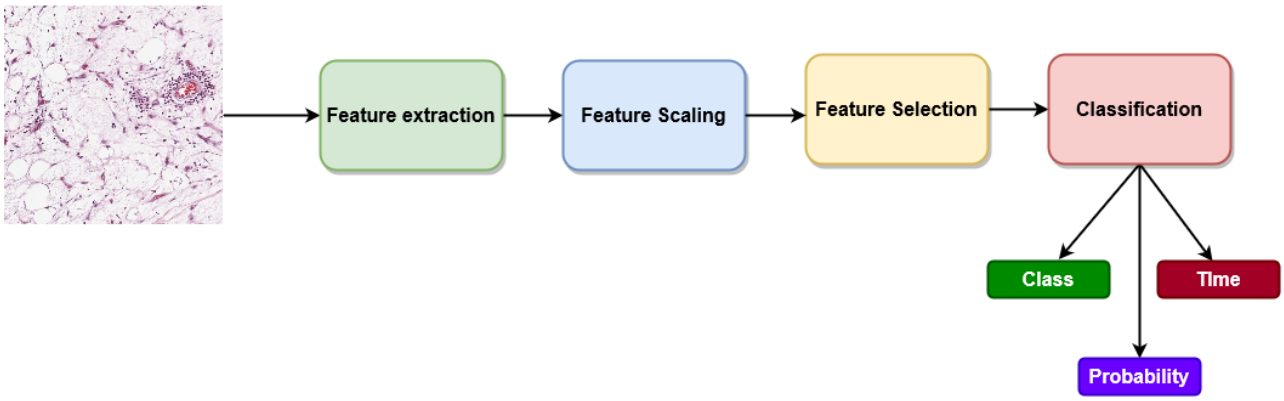


Figure 8. Workflow diagram of our developed web application.

Initially, the user needs to select an image from the user interface as input to see the classification result. The input image will be preprocessed based on what has been used in our training phase, then fed into the pre-trained CNN to extract features. The features will be scaled through the loaded feature scaler and fed into the RFE feature selector. The selected features will be fed into the loaded classifier for prediction. We have performed all five classifications including multiclass and binary with saved classifiers, then max-voted the predicted class from all classifiers, and selected the most frequent class as the predicted class. This max-voting process ensures the reliability of our model for web applications.

After developing the web application, it has been deployed on a cloud platform as a service named Render [70]. Render provides a publicly accessible URL by which any user can access web applications that have been deployed on this platform. The home page that takes inputs (Figure 9a) and the output page (Figure 9b–d) are presented in Figure 9. Users need to click on the select an image file box to upload an image, which prompts up their local store where they can select the input image. By clicking on the submit button, the user will be able to see the output results for a given image including predicted class, class probability, and inference time. Some random images are given as input to evaluate the robustness of our proposed model using the web applications for real-time validation, and the result is shown in Figure 9.

4.6. Comparison with Existing Models

In this section, we compared our proposed hybrid model with existing state-of-the-art models that were developed to identify osteosarcoma malignancy on the dataset. To ensure the effectiveness and robustness of the proposed model, the comparison was performed with different performance metrics including accuracy, precision, recall, and F1 score. Our proposed model has been compared with Mishra, Rashika et al. [71], Mishra et al. [10], Arunachalam, Harish Babu et al. [33], Nabid et al. [5], and Anisuzzaman et al. [4] in terms of mentioned performance metrics listed in Table 8.

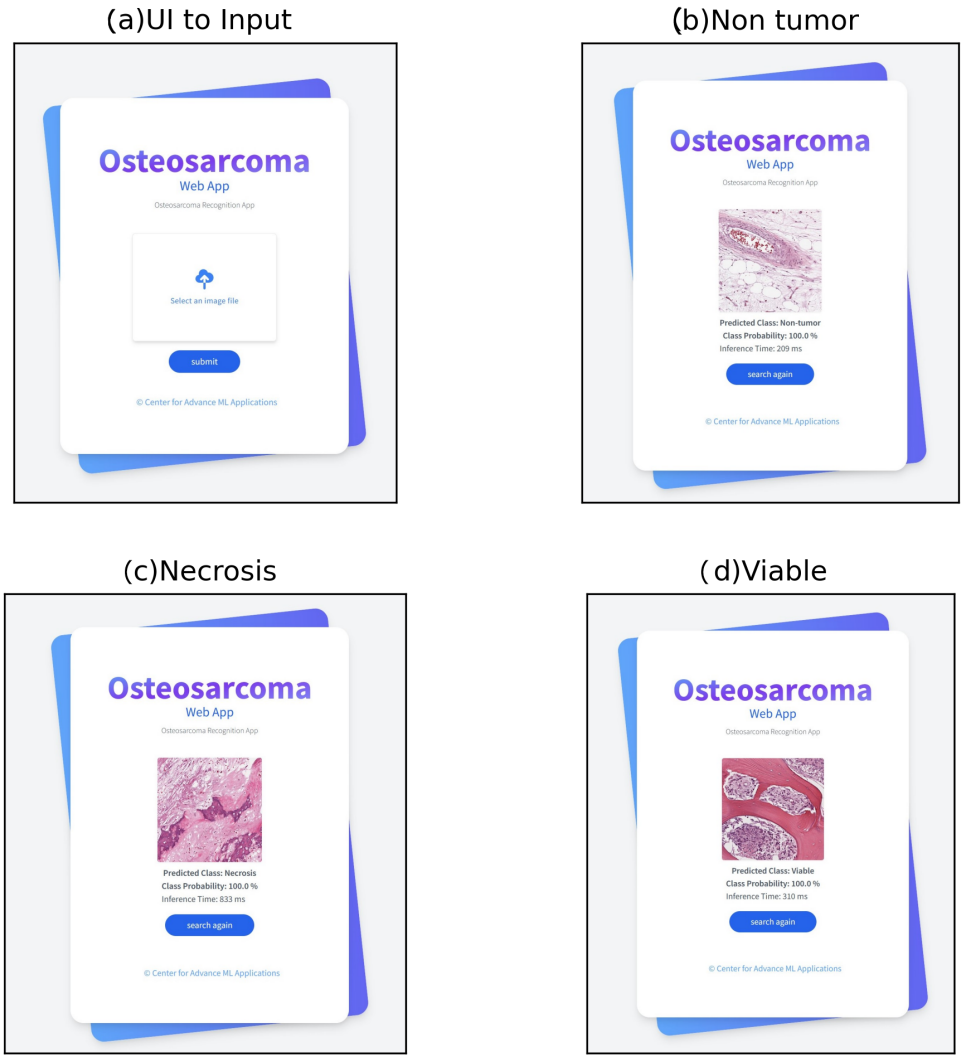


Figure 9. Sampleinput and output from our web application.

Table 8. Comparison of our proposed model with existing state-of-the-art models.

Article Author	Highest Acc. (%)	Precision	Recall	F1 Score
Mishra, Rashika et al. [71]	84	0.89	0.84	0.86
Mishra et al. [10]	92	0.97	0.94	0.95
Arunachalam, Harish Babu et al. [33]	89.9/91.2	-	-	-
Nabid et al. [5]	89	0.88	0.89	0.88
Anisuzzaman et al. [4]	96	0.95	0.95	0.95
Proposed model	**99.4**	**0.99**	**1.00**	**0.99**

As shown in Table 8, Mishra, Rashika et al. [71], Mishra et al. [10], and Nabid et al. [5] reported low accuracy, precision, recall, and F1 score. Arunachalam, Harish Babu et al. [33] obtained the highest accuracy of 89.9% for the ML approach by using SVM. They also employed deep learning approaches where they obtained the highest accuracy of 91.2% for tiles (WSI). Both of their accuracy results refer to class-specific accuracy, and they did not talk about precision, recall, and F1 score. Anisuzzaman et al. [4] achieved the highest accuracy of 96% for binary class classification using VGG-19 via transfer learning; this is 3.4% lower than ours. Furthermore, the model's precision, recall, and F1 score are 4%,

5%, and 4%, respectively, which are all lower than our proposed model. Our proposed model achieved the highest accuracy of 99.4%, which is higher than all existing models. Furthermore, the precision, recall, and F1 score of our model are 0.99, 1.00, and 0.99, which are also higher than all existing models. From this investigation and comparison, we can clearly see that our proposed model outperforms all the existing models in the literature so far based on various evaluation metrics. The robustness and high performance of the proposed model are achieved due to the techniques we have developed and implemented for classifying osteosarcoma malignancy. Initially, normalization techniques were performed in the preprocessing step to enable our model to learn faster during the training phase with a zero-centered Gaussian distribution of data. We have also explored various CNN models and ML classifiers to select the best feature extractor and classifier by conducting huge experiments on the dataset. A total of 25 different combinations of CNN models and ML classifiers were evaluated with different parameter settings to determine the best integration (as shown in Table 2). Furthermore, an optimizer analysis was conducted for the MLP classifier to ensure better optimization by selecting the most suitable optimizer for classifying osteosarcoma that demonstrates improved convergence time and loss. In addition to the above techniques, we have investigated various feature selection techniques where DT-based RFE is selected based on performance. We also analyzed the impact of RFE's criterion concerning time and the number of selected features, aiming for enhanced performance as well as reduced computational cost. Moreover, the integration of CNN and ML with feature selectors leverages the advantages of each approach. This integration makes our proposed model more robust and outperforms all the existing state-of-the-art models to classify osteosarcoma on this dataset.

5. Conclusions

Classification of osteosarcoma malignancy using histological biopsy by pathologists is quite challenging, tedious, and time-consuming. In this paper, we proposed a hybrid model that combines DL and ML to classify osteosarcoma malignancy that will help pathologists with a computer-aided system. First, it extracts relevant features from whole slide images using DenseNet- 121, then performs feature selection using DT-RFE to select the most significant features, and, finally, the MLP classifier is applied to those features chosen for osteosarcoma classification. However, we utilized transfer learning (pre-trained CNN models) for feature extraction rather than building a CNN model from scratch, as it requires a large amount of data and a higher training time. Feature selection techniques have been applied in our model to reduce feature dimensions. Transfer learning, DenseNet- 121, and the feature selection DT-RFE technique reduce computational costs and make our model faster. Moreover, from the five well-known ML algorithms, we selected MLP for classification as the best-performing algorithm based on the performance of our dataset. The experimental results illustrate that our proposed model has higher prediction performance than existing state-of-the-art models developed for osteosarcoma malignancy classification on the same dataset. We also developed a web application of our proposed model that can be used in clinics for early diagnosis of osteosarcoma. After applying feature selection techniques, the accuracy has increased 1.8% for multiclass classification. For binary classification, it has been increased by 2.2%, 1.2%, 2.3%, and 1.6% for nontumor vs. necrosis + viable, nontumor vs. necrosis, nontumor vs. viable, and necrosis vs. viable, respectively. We believe our proposed hybrid model is not only applicable to osteosarcoma classification, but also it can be applied to other histopathological image classifications. In the future, we plan to integrate uncertainty mining and pLOF techniques into our model to make our predicted results more trustworthy.

Author Contributions: Conceptualization, S.M.H.M. and M.T.A.; methodology, S.M.H.M. and M.F.E.; software, M.T.A.; validation, S.M.H.M., M.F.E. and K.A.; writing, M.T.A., S.M.H.M., H.J. and M.F.E.; writing—review and editing, D.N., M.H.R., L.K.S., K.A., M.A.M. and F.M.B.; supervision, S.M.H.M. and K.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All developed models and implemented codes are available at: https://github.com/taareek/osteosarcoma_classification.

Acknowledgments: The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: 23UQU4340560DSR12.

Conflicts of Interest: The authors declare no conflicts of interest.

Sample Availability: The following link will take you to the public website that was created for this work: https://osteosacoma-classificatio.onrender.com. For testing purpose, we provided some sample images for three different class in this link.

References

1. Wittig, J.C.; Bickels, J.; Priebat, D.; Jelinek, J.; Kellar-Graney, K.; Shmookler, B.; Malawer, M.M. Osteosarcoma: A multidisciplinary approach to diagnosis and treatment. *Am. Fam. Physician* **2002**, *65*, 1123. [PubMed]
2. Geller, D.S.; Gorlick, R. Osteosarcoma: A review of diagnosis, management, and treatment strategies. *Clin. Adv. Hematol. Oncol.* **2010**, *8*, 705–718. [PubMed]
3. Ottaviani, G.; Jaffe, N. The epidemiology of osteosarcoma. *Pediatr. Adolesc. Osteosarcoma* **2009**, *152* 3–13.
4. Anisuzzaman, D.M.; Barzekar, H.; Tong, L.; Luo, J.; Yu, Z. A deep learning study on osteosarcoma detection from histological images. *Biomed. Signal Process. Control* **2021**, *69*, 102931. [CrossRef]
5. Nabid, R.A.; Rahman, M.L.; Hossain, M.F. Classification of osteosarcoma tumor from histological image using sequential RCNN. In Proceedings of the 2020 11th International Conference on Electrical and Computer Engineering (ICECE), virtual, 17–19 December 2020; pp. 363–366.
6. Li, Z.; Soroushmehr, S.R.; Hua, Y.; Mao, M.; Qiu, Y.; Najarian, K. Classifying osteosarcoma patients using machine learning approaches. In Proceedings of the 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Jeju Island, Republic of Korea, 11–15 July 2017; pp. 82–85.
7. Mandava, R.; Alia, O.M; Wei, B.C.; Ramachandram, D.; Aziz, M.E.; Shuaib, I.L. Osteosarcoma segmentation in MRI using dynamic harmony search based clustering. In Proceedings of the 2010 International Conference of Soft Computing and Pattern Recognition, Cergy-Pontoise, France, 7–10 December 2010; pp. 423–429.
8. Arndt, C.A.; Crist, W.M. Common musculoskeletal tumors of childhood and adolescence. *Clin. Adv. Hematol. Oncol.* **1999**, *341*, 342–352. [CrossRef]
9. Kothari, Sonal and Phan, John H and Stokes, Todd H and Wang, May D, Pathology imaging informatics for quantitative analysis of whole-slide images. *J. Am. Med. Inform. Assoc.* **2013**, *20*, 1099–1108. [CrossRef]
10. Mishra, Rashika and Daescu, Ovidiu and Leavey, Patrick and Rakheja, Dinesh and Sengupta, Anita, Convolutional neural network for histopathological analysis of osteosarcoma. *J. Comput. Biol.* **2018**, *25*, 313–325. [CrossRef]
11. Irshad, Humayun and Veillard, Antoine and Roux, Ludovic and Racoceanu, Daniel, Methods for nuclei detection, segmentation, and classification in digital histopathology: A review—current status and future potential. *IEEE Rev. Biomed. Eng.* **2013**, *7*, 97–114. [CrossRef]
12. Fuchs, T.J.; Wild, P.J.; Moch, H.; Buhmann, J.M. Computational pathology analysis of tissue microarrays predicts survival of renal clear cell carcinoma patients. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, New York, NY, USA, 6–10 September 2008; pp. 1–8.
13. Yu, K.H.; Zhang, C.; Berry, G.J.; Altman, R.B.; R.é; C.; Rubin, D.L.; Snyder, M. Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features. *Nat. Commun.* **2016**, *7*, 12474. [CrossRef]
14. Litjens, Geert and Sánchez, Clara I and Timofeeva, Nadya and Hermsen, Meyke and Nagtegaal, Iris and Kovacs, Iringo and Hulsbergen-Van De Kaa, Christina and Bult, Peter and Van Ginneken, Bram and Van Der Laak, Jeroen, Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. *Sci. Rep.* **2016**, *6*, 26286. [CrossRef]
15. Spanhol, F.A.; Oliveira, L.S.; Petitjean, C.; Heutte, L. Breast cancer histopathological image classification using convolutional neural networks. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 2560–2567.
16. Mehmood, S.; Ghazal, T.M.; Khan, M.A.; Zubair, M.; Naseem, M.T.; Faiz, T.; Ahmad, M. Malignancy Detection in Lung and Colon Histopathology Images Using Transfer Learning with Class Selective Image Processing. *IEEE Access* **2022**, *10*, 25657–25668. [CrossRef]
17. Arafa, S.H.; Elbanna, K.; Osman, G.E.; Abulreesh, H.H. Candida diagnostic techniques: a review. *J. Umm Al-Qura Univ. Appl. Sci.* **2023**, *10*, 1–8. [CrossRef]

18. Chamekh, M.; Latrach, M.A.; Jday, F. Multi-step semi-analytical solutions for a chikungunya virus system. *J. Umm Al-Qura Univ. Appl. Sci.* **2023**, *8*, 1–9. [CrossRef]
19. Abu-Hashem, M.A.; Gutub, A.; Salem, O.; Shambour, M.K.; Shambour, Q.; Shehab, M.; Izzat, A.; Alrawashdeh, M.J. Discrepancies of remote techno-tolerance due to COVID-19 pandemic within Arab middle-east countries. *J. Umm Al-Qura Univ. Eng. Architecture* **2023**, *10*, 1–5. [CrossRef]
20. Taleb, N.; Mehmood, S.; Zubair, M.; Naseer, I.; Mago, B.; Nasir, M.U. Ovary Cancer Diagnosing Empowered with Machine Learning. In Proceedings of the 2022 International Conference on Business Analytics for Technology and Security (ICBATS), Dubai, United Arab Emirates, 16–17 February 2022; pp. 1–6.
21. Nadeem, M.W.; Goh, H.G.; Khan, M.A.; Hussain, M.; Mushtaq, M.F.; Ponnusamy, V.A. *Fusion-Based Machine Learning Architecture for Heart Disease Prediction*; Tech Science Press: Henderson, NV, USA, 2021.
22. Siddiqui, S.Y.; Athar, A.; Khan, M.A.; Abbas, S.; Saeed, Y.; Khan, M.F.; Hussain, M. Modelling, simulation and optimization of diagnosis cardiovascular disease using computational intelligence approaches. *J. Med. Imaging Health Inform.* **2020**, *10*, 1005–1022. [CrossRef]
23. Ahmed, U.; Issa, G.F.; Khan, M.A.; Aftab, S.; Khan, M.F.; Said, R.A.; Ghazal, T.M.; Ahmad, M. Prediction of diabetes empowered with fused machine learning. *IEEE Access* **2022**, *10*, 8529–8538. [CrossRef]
24. Nasir, M.U.; Khan, M.A.; Zubair, M.; Ghazal, T.M.; Said, R.A.; Al Hamadi, H. *Single and Mitochondrial Gene Inheritance Disorder Prediction Using Machine Learning;* Tech Science Press: Henderson, NV, USA, 2022.
25. Rahman, A.U.; Alqahtani, A.; Aldhafferi, N.; Nasir, M.U.; Khan, M.F.; Khan, M.A.; Mosavi, A. Histopathologic Oral Cancer Prediction Using Oral Squamous Cell Carcinoma Biopsy Empowered with Transfer Learning. *Sensors* **2022**, *22*, 3833. [CrossRef]
26. Arunachalam, H.B.; Mishra, R.; Armaselu, B.; Daescu, O.; Martinez, M.; Leavey, P.; Rakheja, D.; Cederberg, K.; Sengupta, A.; Ni'suilleabhain, M. Computer aided image segmentation and classification for viable and non-viable tumor identification in osteosarcoma. In Proceedings of the Pacific Symposium on Biocomputing 2017, Kohala Coast, HI, USA, 3–7 January 2017; pp. 195–206.
27. Nasor, M.; Obaid, W. Segmentation of osteosarcoma in MRI images by K-means clustering, Chan-Vese segmentation, and iterative Gaussian filtering. *IET Image Process.* **2021**, *15*, 1310–1318. [CrossRef]
28. Vandana, B.S.; Antony, P.J.; Alva, S.R. Analysis of malignancy using enhanced graphcut-based clustering for diagnosis of bone cancer. *Inf. Commun. Technol. Sustain. Dev.* **2020**, *933*, 453–462.
29. Liu, F.; Xing, L.; Zhang, X.; Zhang, X. A four-pseudogene classifier identified by machine learning serves as a novel prognostic marker for survival of osteosarcoma. *Genes* **2019**, *10*, 414. [CrossRef]
30. Sirinukunwattana, K.; Raza, S.E.; Tsang, Y.W.; Snead, D.R.; Cree, I.A.; Rajpoot, N.M. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE Trans. Med Imaging* **2016**, *35*, 1196–1206. [CrossRef] [PubMed]
31. O'Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Hern, ez, G.V.; Krpalkova, L.; Riordan, D.; Walsh, J. Deep Learning vs. Traditional Computer Vision. *Adv. Comput. Vis.* **2019**, *11*, 128–144.
32. Lecun, Y.; Bengio, Y. Hinton, G. Deep learning for AI. *Commun. ACM* **2021**, *64*, 58–65. [CrossRef]
33. Arunachalam, H.B.; Mishra, R.; Daescu, O.; Cederberg, K.; Rakheja, D.; Sengupta, A.; Leonard, D.; Hallac, R.; Leavey, P. Viable and necrotic tumor assessment from whole slide images of osteosarcoma using machine-learning and deep-learning models. *PLoS ONE* **2019**, *14*, e0210706. [CrossRef]
34. D'Acunto, M.; Martinelli, M.; Moroni, D. From human mesenchymal stromal cells to osteosarcoma cells classification by deep learning. *J. Intell. Fuzzy Syst.* **2019**, *37*, 7199–7206. [CrossRef]
35. Fu, Y.; Xue, P.; Ji, H.; Cui, W.; Dong, E. Deep model with Siamese network for viable and necrotic tumor regions assessment in osteosarcoma. *Med. Phys.* **2020**, *47*, 4895–4905. [CrossRef]
36. Gawade, S.; Bhansali, A.; Patil, K.; Shaikh, D. Application of the convolutional neural networks and supervised deep-learning methods for osteosarcoma bone cancer detection. *Healthc. Anal.* **2023**, *3*, 100153. [CrossRef]
37. Vezakis, I.A.; Lambrou, G.I.; Matsopoulos, G.K. Deep Learning Approaches to Osteosarcoma Diagnosis and Classification: A Comparative Methodological Approach. *Cancers* **2023**, *15*, 2290. [CrossRef]
38. Leavey, P.; Sengupta, A.; Rakheja, D.; Daescu, O.; Arunachalam, H.B.; Mishra, R. Osteosarcoma data from UT Southwestern/UT Dallas for Viable and Necrotic Tumor Assessment [Data set]. *Cancer Imaging Arch.* **2019**, *14*. [CrossRef]
39. Benkaddour, M.K.; Bounoua, A. Feature extraction and classification using deep convolutional neural networks, PCA and SVC for face recognition. *Trait. Du Signal* **2017**, *34*, 77–91. [CrossRef]
40. Liu, Y.H. Feature extraction and image recognition with convolutional neural networks. *J. Phys. Conf. Ser.* **2018**, *1087*, 062032. [CrossRef]
41. Jogin, Manjunath and Madhulika, MS and Divya, GD and Meghana, RK and Apoorva, S and others, Feature extraction using convolution neural networks (CNN) and deep learning. In Proceedings of the 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 18–19 May 2018; pp. 2319–2323.
42. Yang, B.; Shan, Y.; Peng, R.; Li, J.; Chen, S.; Li, L. A feature extraction method for person re-identification based on a two-branch CNN. *Multimed. Tools Appl.* **2022**, *81* 39169–39184. [CrossRef]
43. Models and Pre-Trained Weights. Available online: https://pytorch.org/vision/stable/models.html (accessed on 31 December 2022).

44. Keras Application. Available online: https://keras.io/api/applications/ (accessed on 31 December 2022).
45. Deng, J.; Dong, W.; Socher, R.; Li, L-J.; Li, K.; Li, F-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
46. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
47. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
48. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. *Eur. Conf. Comput. Vis.* **2014**, *8689*, 818–833.
49. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv* **2013**, arXiv:1312.6229.
50. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
51. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
52. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
53. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef]
54. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
55. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
56. Wang, Y.Y.; Li, J. Feature-selection ability of the decision-tree algorithm and the impact of feature-selection/extraction on decision-tree results based on hyperspectral data. *Int. J. Remote Sens.* **2008**, *29*, 2993–3010. [CrossRef]
57. Zeng, X.; Chen, Y.W.; Tao, C. Feature selection using recursive feature elimination for handwritten digit recognition. In Proceedings of the 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, Japan, 12–14 September 2009; pp. 1205–1208.
58. You, W.; Yang, Z.; Ji, G. Feature selection for high-dimensional multi-category data using PLS-based local recursive feature elimination. *Expert Syst. Appl.* **2014**, *41*, 1463–1475. [CrossRef]
59. sklearn-RFE. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html (accessed on 31 December 2022).
60. Lian, W.; Nie, G.; Jia, B.; Shi, D.; Fan, Q.; Liang, Y. An intrusion detection method based on decision tree-recursive feature elimination in ensemble learning. *Math. Probl. Eng.* **2020**, *2020*, 2835023. [CrossRef]
61. Song, Y.Y.; Ying, L.U. Decision tree methods: Applications for classification and prediction. *Shanghai Arch. Psychiatry* **2015**, *26*, 130.
62. sklearn-dt. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html (accessed on 31 December 2022).
63. Basu, S.; Das, N.; Sarkar, R.; Kundu, M.; Nasipuri, M.; Basu, D.K. An MLP based Approach for Recognition of Handwritten-Bangla'Numerals. *arXiv* **2012**, arXiv:1203.0876.
64. Zhai, X.; Ali, A.A.; Amira, A.; Bensaali, F. MLP neural network based gas classification system on Zynq SoC. *IEEE Access* **2016**, *4*, 8138–8146. [CrossRef]
65. Al Bataineh, A.; Manacek, S. MLP-PSO hybrid algorithm for heart disease prediction. *J. Pers. Med.* **2022**, *12*, 1208. [CrossRef]
66. Yang, J.B.; Shen, K.Q.; Ong, C.J.; Li, X.P. Feature selection for MLP neural network: The use of random permutation of probabilistic outputs. *IEEE Trans. Neural Netw.* **2009**, *20*, 1911–1922. [CrossRef]
67. Pinkus, Allan, Approximation theory of the MLP model in neural networks. *Acta Numer.* **1999**, *8*, 143–195. [CrossRef]
68. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
69. FastAPI. Available online: https://fastapi.tiangolo.com/ (accessed on 31 December 2022).
70. Cloud Application Hosting for Developers. Render. Available online: https://render.com/ (accessed on 31 December 2022).
71. Mishra, R.; Daescu, O.; Leavey, P.; Rakheja, D.; Sengupta, A. Histopathological diagnosis for viable and non-viable tumor prediction for osteosarcoma using convolutional neural network. *J. Comput. Biol.* **2018**, *25*, 313–325. [CrossRef] [PubMed]