IMPROVED BONE CANCER DIAGNOSIS: TRANSFER LEARNING INTEGRATION WITH U-NET FOR SEGMENTED HISTOPATHOLOGY IMAGE ANALYSIS

PHASE I REPORT

Submitted by

MADHULIKA G - 2116210701139

MAHALAKSHMI K - 2116210701143

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANNA UNIVERSITY, CHENNAI

NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this Report titled "**IMPROVED BONE CANCER DIAGNOSIS: TRANSFER LEARNING INTEGRATION WITH U-NET FOR SEGMENTED HISTOPATHOLOGY IMAGE ANALYSIS**" is the bonafide work of **MADHULIKA G (2116210701139), MAHALAKSHMI K (2116210701143)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. P. KUMAR, M.E., Ph.D.,
Professor and Head,
Department of Computer Science and Engineering,
Rajalakshmi Engineering College,
Thandalam,
Chennai - 602 105

Mr. Deepak Kumar K, M.E.,
Assistant Professor,
Department of Computer Science and Engineering,
Rajalakshmi Engineering College,
Thandalam,
Chennai - 602 105

Submitted to Project and Viva Voce Examination for the subject CS19711-

PROJECT PHASE–I held on _____.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S.MEGANATHAN, B.E, F.I.E.,** our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr.P.KUMAR, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Mr. Deepak Kumar K, M.E.,** Assistant Professor, Department of Computer Science and Engineering, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, **Dr.T.Kumaragurubaran**, **Ph.D.,** Assistant Professor (SG), Department of Computer Science Engineering for his useful tips during our review to build our project.

MADHULIKA G: 2116210701139

MAHALAKSHMI K: 2116210701143

ABSTRACT

Bone cancer is a rare but severe condition characterized by the uncontrolled growth of cells within the bone, leading to the destruction of normal bone tissue. The survival rate for bone cancer patients is generally low, but early detection significantly increases the chances of successful treatment. Traditional methods for detecting bone cancer involve radiologists examining medical images from MRIs, CT scans, and X-rays. These methods have limitations, including significant variability in diagnostic accuracy and a high potential for human error. Additionally, the manual review process is lengthy and often delays diagnosis, further complicating patient outcomes. Although Convolutional Neural Networks (CNNs) are proposed as effective algorithms for classifying and detecting cancer from images, they face challenges when applied to histopathological images. This study introduces a novel approach combining machine learning with cutting-edge image processing methods to increase the precision and effectiveness of diagnosis. The model retains several key features: U-Net is used for segmentation, and four different CNNs—ResNet50, VGG16, MobileNetV2, and DenseNet121—are employed for classification, along with data augmentation strategies. Our method aims to enhance the reliability of bone cancer detection.

Keywords: Bone Cancer, Machine Learning, U-Net, ResNet50, VGG16, MobileNetV2, DenseNet121, Image Processing, Data Augmentation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

1. **CNN -** Convolutional Neural Networks
2. **PET -** Positron Emission Tomography
3. **CT -** Computed Tomography
4. **ResNet -** Residual Neural Network
5. **VGG -** Visual Geometry Group
6. **DenseNet -** Densely Connected Convolutional Networks
7. **ReLU -** Rectified Linear Unit

CHAPTER

1. INTRODUCTION

1.1 GENERAL

Bone cancer particularly osteosarcoma is a rare but highly malignant form of tumor that occurs in bones arising from the osteoblasts which are involved in formation of bones and maintenance of bone tissue. The condition appears when these osteoblastic cells divide abnormally and start forming malignant tumors that tend to compromise the rigidity of the bone. Osteosarcoma is common with the bones in the length of the body or limbs, more specifically around the knee but it can also develop at the pelvic, shoulder or the jaws. One of the main peculiarities of osteosarcoma is that pain is generally located in one area, and it is intense, progressive, and unremitting. For instance, due to the disease, the affected bone develops brittleness, and persons are more likely to suffer from fractures in minor accidents.

Determination of osteosarcoma at an early stage is essential for a favorable outcome, and X-ray is usually the primary imaging technique applied if osteosarcoma is suspected. Radiological imaging shows destruction and sclerosis type of bone pathology and may also have both features simultaneously. Nevertheless, an X-ray is not enough for cancer confirmation. When the imaging suggests the presence of a tumor, a biopsy is done to remove a tissue sample for histopathological examination. Part of this tissue is then stained with Natural Black 1 and Eosin dyes that make it easier to discern the distribution and form of the tissue cells under a microscope. The most basic methods in cancer diagnosis, which include the examination of biopsied tissue under a light microscope, remain relevant and accurate. However, this method comes with the following draw back: The process is lengthy and strongly depends on the pathologist’s skills and his/her decision making which brings about human factor element.

Further, exploring large sets of tissue samples manually is extremely cumbersome and, in the process, relevant patterns essential for a correct diagnosis may be eliminated. Such difficulties arise when working with multi-parametric datasets or when the analyzed pathologies contain different types of cells, and it is impossible to determine whether they are malignant or benign. To address these weaknesses, there is a increasing call for companion strategies that can improve the sensitivity, specificity, precision, and efficiency of the osteosarcoma diagnosis.

Modern diagnostic technologies based on machine learning and image processing can conceptualize promising means for cancer detection. These systems can process large amounts of histological data and recognize various patterns and abnormal cells, which can often remain unnoticed at the best of times by a human analyst.

With the help of improved algorithms, these tools can help pathologists provide a better decision, minimize the risks of mistakes, and save considerable amount of time needed to make a definitive conclusion. In addition, the developed technologies can process big data faster, which can help expand screening activities and early diagnosis programs, which in turn can help increase patient survival rates in osteosarcoma. Also, there is the principle of using a multimodal treatment by oncologists, orthopedic surgeons, and radiologists, as well as pathologists. Further, advanced symptoms and elaborate treatment plans are crucial to enhancing the patients' survival. Today, the biggest ahead being immunotherapy and the directions towards the coming up with antigens and molecular and baseman's treatment, which endorse the immune system to fight cancerous cells. These techniques when combined with existing machine learning diagnostic tools present the prospect of changing the treatment of osteosarcoma by individualizing, accelerating, and increasing the efficacy of treatment.

Finally, traditional methods have not lost their relevance, but the use of automated systems in the diagnostics of osteosarcoma is a breakthrough that, on the continuation of traditional methods, offers a higher accuracy of diagnostics of osteosarcoma and more effective results in combating this dangerous disease.

1.2 OBJECTIVE

The present work will mainly focus on the development of modality to develop high accurately as well as time optimized, fully automated diagnostic system for bone cancer especially for osteosarcoma using complex Machine Learning tools. This research shall make a positive impact on the identification of cancerous cells in histopathological images by not only cutting-diminishing the time taken to identify these images but also reducing the intricacy whereby deep learning algorithm starts to discern small features in the imagery and anomaly in the cellular structure. In effect, these models will be trained to a degree where they can actually eliminate some of the drawbacks of having human-aided intervention and offer far more swift and accurate analysis of those images as well.

To overcome ordinary difficulties including small data occurrences during handling of medical imaging which include; little data and the absence of data balance, data augmentation techniques will be used. This is done by creating more training images by applying flip distortions to the images– the distortions such as image orientation, flips and zooms amongst others however when applied to the images are not capable of missing/capturing/occluding cancerous tissues. Consequently, the model will be able to suggest an expanded generic picture which covers other variations of histopathological images in the identification of what make up the benign and malignant cells. This augmentation process will be beneficial not only to reduce the drawbacks of small sample size problem but also to enhance explore the capability of the proposed model in terms of the detection of the existence of cancer under different scenarios, thereby, improve the reliability of the model.

In conclusion, the present study is interested in strengthening the detection of Bone cancer especially osteosarcoma, through the employment of deep learning and data augmentation to produce a trust worthy detection architecture from Histopathological images.

1.3 EXISTING SYSTEM

The current approach of identifying bone cancer is through pathological examination of histopathology images which is a tedious process that only requires expert pathologists. This traditional approach presents several challenges especially when applying algorithms on high resolution images which resolve even cellular patterns. The utilization of human interpretation enhances the risks of either missing or misinterpreting aspects as well as reducing uniformity. Moreover, the process of collecting the data is discrete and time-consuming, and most manifestly, it is challenging to implement in different areas with restricted medical professionals. This is made worse by the challenges of patient consent for use of data, and the requirement to obtain data from hospitals in real time, a process that consumes time hence delaying the diagnosis and treatment processes.

The current infrastructure used for cancer detection has experienced embracing deep learning approaches for better identification of the diseases. This work adopted models presented from convolutional neural networks (CNN) approaches and methods including dense CNNs for the purpose of classifying cancerous cells where features are extracted through data augmentation and image manipulation of rotation and edges. There are other combinations of approaches whereby architectures like MobileNetV2 and Max voting classifiers are used to enhance detection accuracy rates of bone cancer. Furthermore, other updated methods include the utilization of the deep convolutional transformer by replace Vit named ENMVit which will effectively segment the nuclei of osteosarcoma to deal with different painting styles. Recent enhancements have been conducted on lung nodule detection through models that build frames around lung nodules such as the SSD-VGG16 but challenges are realized when working with less clear images or low distortion images.

Innovative methods also pay attention to the improvement of medical image segmentation and classification using models at multiple scales and attention mechanisms. In order to address high-level tasks, including joint nuclei and contour segmentation, transfer learning, residual blocks, as well as channel attention modules

have been utilized. These models are further optimized with techniques like focal loss that helps in focusing on tough samples and prevent overfitting. Besides, the combination of spatial information originating from different image modalities such as PET and CT has been proposed for more accurate tumor identification and delineation. However, some issues include how to address the variation of images, data distribution skewness and how to minimize the consumption of computational resources which are main reason for continuous current research on these systems to make them much more precise for cancer detection.

Diagnostic methods and treatment plans of bone cancer, mostly osteosarcoma, have for a long time been based on medical imaging, histopathology and clinical assessment. However, these conventional ones pose various challenges in accuracy, efficiency, and reliability since the methods employ substantial use of expertise, are manual, and complex when dealing with the various pathological datasets. To be precise, Medical Imaging and Artificial Intelligence fields have brought change in diagnosis system by using deep learning and machine learning models.

Currently, many database systems used for medical image analysis and cancer diagnosis integrate such promising techniques as deep learning, transfer learning, multi-modal imaging, and efficient segmentation methods. Through employing CNN and ensemble classifiers such as MobileNetV2, deep learning models have high accuracy in diagnosis of cancerous tissues thus improving the diagnostic results.. Applying transfer learning procedure utilize existing models to deal with medical imaging tasks and address dataset fluctuations that enhance classification accuracy. Bimodal and multimodal imaging methods encompassing both PET and CT make it easier to localize tumors as well as segment them. Hence, techniques like UNet variations and attention-based networks overcome problems like limited resources, data distribution disparity, and computational load that lead to diverse medical applications efficiency. Altogether these models emphasize the evolution and advancement of procedures in existing systems and the models provide effective structures for enhanced diagnosis and analysis of diseases, including osteosarcoma.

1.4 PROPOSED SYSTEM

The proposed system is designed with an attempt to eliminate the drawbacks of the present day manual diagnosis with the help of machine learning. This involves the following steps:

1. Data Collection: Due to the limitation on the acquisition of real-time data, the study employed collected histopathological images and improved the data set using data augmentation. It also makes the data to grow and the learning ability of the model increases because it is not limited by a few data and so overfitting rarely happens. Moreover, the system also draws images from public medical image databases so the training data diversity exists to enhance generalization.

2. Data Augmentation: In order to avoid disturbing the image by adding augmented marks or similar, the rotations can be just flipping and changes in scale of the images and sometimes brightness and contrast. This also assists in class imbalance problems since instead of having limited samples for the underrepresented classes, there appear to have a larger number of samples. Besides these configurations, other augmentations including color jitter and random cropping are used to close the gap between simulated and real-world scenarios that the model will face.

3. Feature Extraction: Handling of the high-resolution histopathological images is done with the help of U-Net model, the Convolutional neural network specifically developed for the biomedical image segmentation. This means excluding all areas where the tissue is as yet healthy from the Region of Interest (ROI) in order to reduce the amount of noise and unnecessarily complex morphology within the image. To increase the accuracy of segmentation different post-processing methods are used, for instance, morphology operations helping to improve segmentation of cancerous areas.

4. Model Training: The segmented images are then classified using several CNN architectures among which are ResNet50, VGG16, MobileNetV2, DenseNet 121 . All these models are optimized, using the augmented data, to achieve better accuracy in detecting cancer cells. Furthermore, transfer learning is used so that current models can

be trained faster and with a better result than training from scratch and taking knowledge from large network sources such as ImageNet.

5. Performance Measure: The efficiency of each model is estimated with such parameters as accuracy, sensitivity, and specificity. This makes it possible to evaluate the accuracy of the models in diagnosing and categorizing cancerous cells. This is followed by accuracy and F1 scores, and confusion matrices and ROC-AUC curves are produced to offer further evaluation of the numerical and graphical aspects of the models to distinguish true positives from false ones and to introduce the question of threshold reliability.

2. LITERATURE SURVEY

Kumar et al. [1] presented a new method based on deep learning techniques to decision of bone cancer. Their approach was able to build data augmentation using features like feature extraction, edge detection, and rotation in order to enhance the capability of the model to handle other classes of data. The study showed how deep learning could be used to accurately identify cancerous tissues by using a dense CNN with three layers.

In the same manner, Walid et al. [2] used both MobileNetV2 and Max Voting classifiers for balanced identification of Bone cancer with an improved accuracy of 93.88%. The work underlined the importance of ensemble learning methods to enhance the diagnostic accuracy. Wu et al. [3] recently introduced the ENMVit based on the Conv-Transformer model for osteosarcoma nuclei precise detection from pathological images. The given approach compensated the specifics of staining within images, which resulted in improved diagnostic outcomes particularly in the least developed countries.

The SSD-VGG16 was used by Loraksa et al. [4] to detect lung nodules in osteosarcoma patients. While their method obtained a satisfactory accuracy of 75.97%, the model is not very good at identifying very faint or blurred nodules which further confirms the need to optimize the way of dealing with such complicated image data.

Further, similar to the above research, Saber et al. [5] introduced deep learning for classification of breast cancer using a hybrid of VGG-16 and CNN. This approach included better acquisition of data, iterating and finally doing a validation on the best local optimum which prevented over-fitting and provided gaols for future implementations of the similar methods for osteosarcoma diagnosis. More specific, Sun et al [6] applied transfer learning and multiple instance learning approach to divide the histopathological image into patches for easily classification of liver cancer images. The segmentation technique proved ways through which sophisticated machine learning tactics provided better cancer detection.

Weng et al. [7] proposed a NAS to solve the problem of medical image segmentation. In order to improve the efficiency of NAS architecture while addressing the problem of excessive GPU memory consumption, the study introduced NAS-Unet with downsampling and upsampling cells. Zeng et al. [8] extended this method in their RIC-Unet model with residual blocks, multi-scale inception modules and channel attention blocks. The presented technique was highly efficient in the separation of the abnormal cells and overlapping structures treatment, which is useful for osteosarcoma image examination.

Fu et al. Proposed an encoder decoder based spatial-focused attention module with PET and CT modalities for tumor localization in [9]. These findings highlighted the value of this innovative multimodal imaging application for improving segmentation precision and its possible utility in osteosarcoma diagnosis. Likewise, in [10], Ashwath et al., introduced a self-interpretable three-tier DCNN model, which incorporated global, attention, as well as fusion partitions to enhance the categorization of dispersed and irregular regions of clinical images.

J. Hu et al. [11] extended the idea of Mi-UNet and proposed the S-UNet model using cascading block approaches in order to segment the retinal vessels effectively. This method dealt with the issue of unbalanced data and fewer parameters were required during computations and it was flexible to other medical imaging conditions. E. Alabdulkreem et al. [12] proposed the OSADL-BCDC methods, which used the Owl Search Algorithm and have the pre-trained Inception v3 for the classification of X-rays. As for the feature and classification extraction their approach of using LSTM networks was unique.

In order to reconstruct the lost significant image features in convolutional networks, S. Alsubai, et al [13] proposed an attention-based artificial neural network with the aid of group teaching of hyperparameters. This technique was effective in enhancing the sequential image classification method while reducing computational overhead. To address the limitations of U-net pars in medical image segmentation, K. D. Shah et al. [14] proposed the Efficient Multi-Encoder-Decoder-based UNet (EMED-UNet). The

model provided effective compression of the networks regards to parameters, FLOPS, and memory without loss of accuracy, and therefore can be deemed useful in the circumstances of resource scarcity.

The author H. N. Dao et al. in [15] proposed a transfer learning using multimodal approach for medical image classification. This result also indicates that feature fusion is effective in merging the image and text vectors and in addressing the variation in datasets. N. Siddique et al. [16] have presented a bird's eye view emergence of U-net and especially its significance in image segmentation in the diagnosis of medical trials. Both highlighted issues related to data labeling, which is a crucial aspect for deploying deep learning into aiding diagnosing osteosarcoma.

A. Tabbakh et al. [17] used transfer learning and a transformer-based structure to diagnose plant diseases to demonstrate the framework's flexibility for medical uses. Four-stage framework, data augmentation, feature extraction were explaining the minimized computational time and improved classification accuracy by them. A. Anaya-Isaza et al [18] enhanced brain tumor segmentation using a Transformer-based structure with separable convolutions and cross-attention cubes. This method delivered close to 94% accuracy, marking the significance of complex architectures in medical image segmentation.

R. Zaitoon et al. [19] proposed a type of deep learning diagnostic model of brain tumor detection coupled with survival rate using the BraTS dataset. With the help of the modified CNMF, DBT-CNN classifiers, and RU-Net2+ segmentation, the model achieved new records for the rate of accuracy and prognosis. ResNet-50 based framework using Spatial Pyramid Pooling was suggested by K. Neamah et al. [20] for brain tumor classification. Therefore, the model based on such a high accuracy, precision, and recall can open the way to applying similar strategies for diagnosing osteosarcoma.

There are current systems deployed for the medical image analysis and cancer detection which utilise the state of the arts methods like deep learning, transfer learning, multi-modal imaging and efficient segmentation architectures. Convolutional neural networks and other classifiers, including ensemble classifier of MobileNetV2, have

reported accuracies in distinguishing malignant tissues and improving diagnostic delineation.

Using transfer learning entails fine-tuning models that provide solutions for other tasks and works splendidly in maintaining medical imaging functions adaptable to variations in the dataset and enhancing precision in classification. Modern techniques for imaging and investigations including functional PET and morphological CT help accurately specify tumours and separate them with a high degree of accuracy. It overcomes issues of limitations of resources, dataset scarcity and computational burden that are achieved by efficient architectures like the variations of UNets and networks based on attention mechanisms that are helpful in various medical applications. Together, such models demonstrate developments and advancements of the existing systems and provide sound frameworks for enhanced diagnostic and analytical capability of diseases, including osteosarcoma.

3. SYSTEM DESIGN

3.1 GENERAL

3.1.1 SYSTEM FLOW DIAGRAM

The System Flow Diagram gives an overview of the processes and the flow of activities within the System. This tool reveals the step by step events, input, control decisions and results to provide a general understanding of how the system works. This diagram is very helpful when it comes to understanding how different piece of a picture as well as the different process depend on each other.

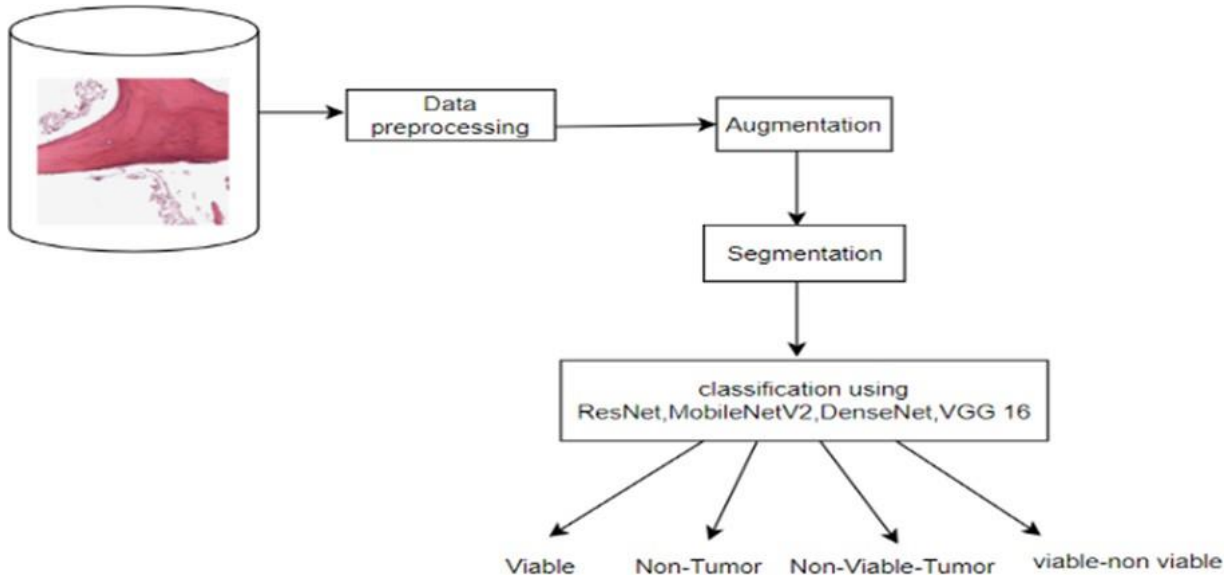


Figure 3.1.1 System flow diagram

The above Figure 3.1.1 demonstrates the flow of the system.The histopathological images including four classes which are stored in separate folders were preprocessed.During EDA the classes were found to be unbalanced.Data Augmentation was performed to improve the performance of the system.After augmentation feature extraction was performed to extract the important features and reduce the noise.Followed by the classification of classes using four pretrained models.

3.1.2 ARCHITECTURE DIAGRAM

The Opportunities Chart shows the development of the system and it's layout as seen in the Architecture Diagram. It offer an overview of the system parts, their relationships and ways these are utilized to create functionality of the system. This diagram is important when it comes to the design of the system, its scalability and its maintainability.

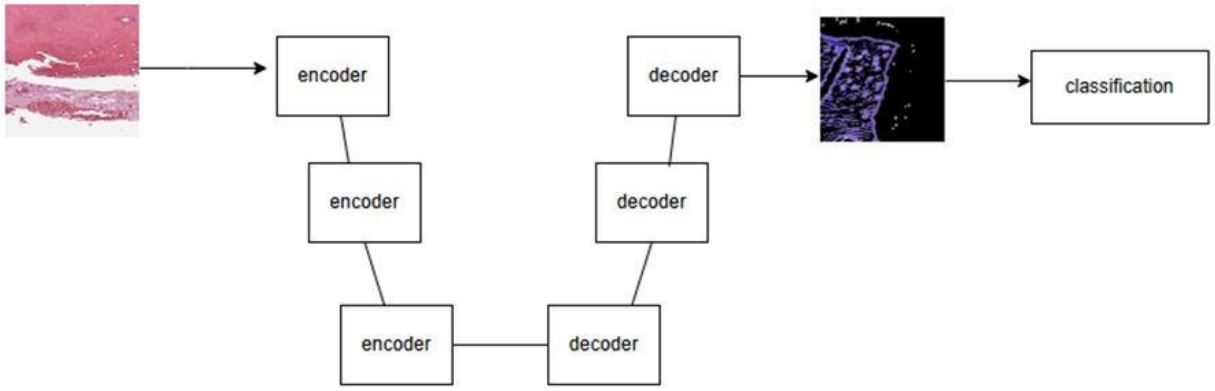


Figure 3.1.2 Architecture diagram

The figure 3.1.2 represents the architecture of the system. The histopathological images are preprocessed and fed to U-Net for feature extraction.The U-Net performs downsampling and upsampling with the help of encoder and decoder which contains convolutional layers.Initially the binary masks are generated for the original images using the predictions of the U-Net model. The combination mask images and original images are passed to generate the segmented image.

The important features extracted from the original image are represented as 1 and the less important features represented as 0 are converted into black regions. The segmented images are given to train the classification models which are pre trained using the weights of imagenet. A custom classification head is designed by adding the following layers GloabalAveragePoolingLayer2D,Dense layer which contains 1024 neurons with activation function as relu,Dense layer with 4 neurons with activation function as softmax.

3.1.3 USECASE DIAGRAM

The Use Case Diagram presents the behavior of the system based on requirements since it illustrates how the different actors engage with the use cases. It describes how actors (users or external systems) interact with the specific functions they employ. This diagram is good to note the dimension and working of the system.

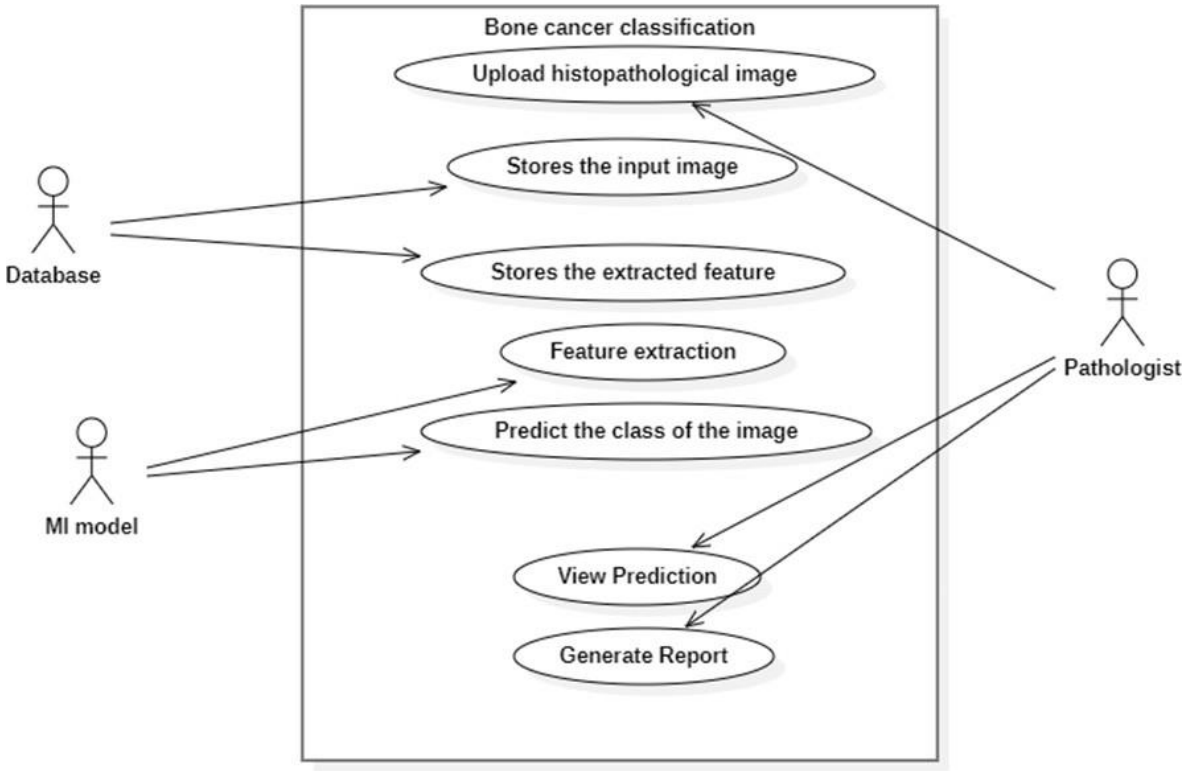


Figure 3.1.3 Use case diagram

The above diagram represents the Use Case diagram.The actors of the system include Database,Pathologist,Ml model. The database contains the histopathological image of four classes representing Non Viable tumor,Viable tumor, Non tumor, viable non-viable tumor. After preprocessing and segmentation the database stores the segmented images. The Ml model interacts with the system by extracting the features and predicting the classes.The pathologists use the system by giving input images and generating reports for diagnosis.

3.1.4 ACTIVITY DIAGRAM

The Activity Diagram shows dynamic features of the system that are the sequences of activities and actions. It shows how the control is established, and how decisions are made and activities performed in a system.

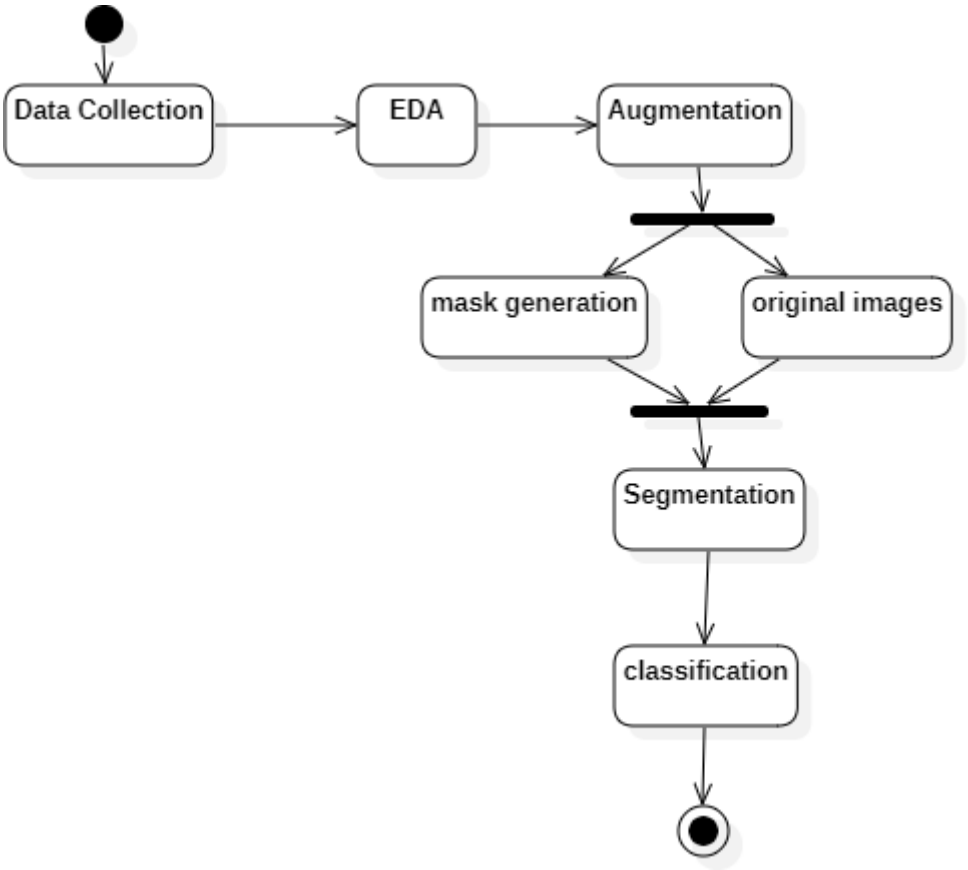


Figure 3.1.4 Activity diagram

The above diagram shows the flow of the activities involved in the system. Data Collection activity collects the data. Followed by EDA to analyze the collected data.Augmentation to generate images and improve the performance.Then fork is used to concurrently pass the mask images and original images to generate the segmented images by using join, followed by classification.Augmentation and segmentation are time events because they process more images.

3.1.5 CLASS DIAGRAM

The analysed Class Diagram captures and reflects the essential aspect of the static characteristics of the system as its classes and their attributes and methods, interrelations between them. There is a key component of object-oriented design, allowing for the purpose of describing the structure of data, as well as its interactions.

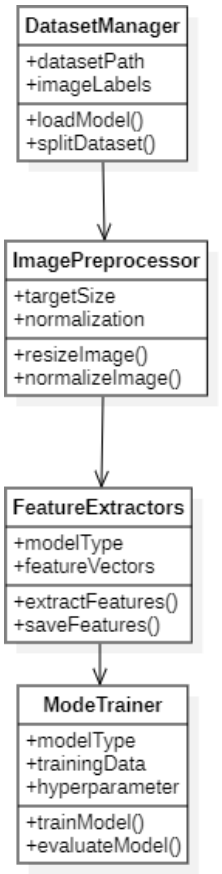


Figure 3.1.5 Class diagram

The process shown in figure 3.1.5 is perhaps one of the most modular machine learning pipeline for image data, envisioned to incorporate four major stages. DatasetManager is responsible for loading and splitting of image datasets while ImagePreprocessor enforces image size and color normalization, FeatureExtractors processes and saves features and ModelTrainer trains models and evaluates their performance using optimal hyperparameters.

3.1.6 SEQUENCE DIAGRAM

The Sequence Diagram promotes the temporal aspect of interaction of objects in the system. How objects work together and achieve a particular goal or function along with documentation of the messages exchanged in the process.

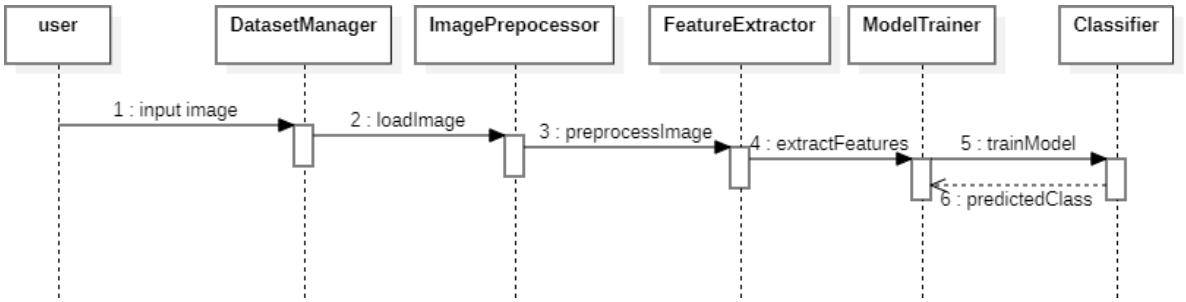


Figure 3.1.6 Sequence diagram

The process involved in an image classification pipeline can be shown as outlined in Figure 3.1.6 mentioned above. The User starts the workflow by entering an input image (Step 1) which then the DatasetManager loads and processes (Step 2). The ImagePreprocessor resizes and normalizes the image because the input data should be of the standard dimensions (Step 3). The FeatureExtractor extracts key features from the preprocessed image: the image itself is in a format that is the most appropriate to feed it to the model (Step 4). The ModelTrainer uses the above extracted features to train the classification model with an aim of achieving the right estimates in the output (Step 5). Last of all, the trained model employs the Classifier attribute thus furnishing the typical output to the user (Step 6). This pipeline makes it easier and structured when it comes to handling image classification tasks.

It shows that there is moderation of functions since every part executes specific functions, thus making it scalable. Yet it should be also flexible enough to make it possible to update, replace or swap components such as feature extraction or the training model while considering the rest of the process.

3.1.7 COMPONENT DIAGRAM

The Component Diagram offers a system perspective in terms of modular by depicting physical and logical things such as DBs, servers, and modules. It emphasises the relations between those components and how they are assembled to form the final system.

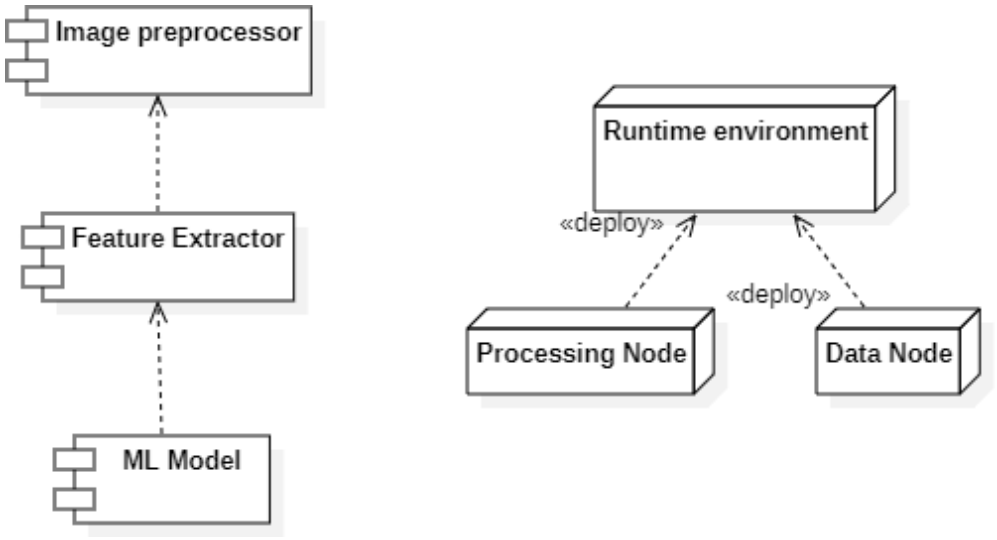


Figure 3.1.7 Component diagram

In the figure 3.1.7 above, there is provided an example of the machine learning pipeline deployment and execution framework. Image Preprocessor, Feature Extractor, and the ML Model work in a pipeline where Image Preprocessor performs the prerequisites or pre-processing of images, Feature Extractor extracts features from predefined images and the ML Model makes predictions. These components are implemented within a Runtime Environment that furnishes the execution platform. The Runtime environment communicates with a Processing Node for calculation purpose and communicates with a Data Node for data retrieval purpose. This framework makes the pipeline easily scalable, and organizes the process in a manner that makes the most sense, and is the most effective to implement.

3.1.8 COLLABORATION DIAGRAM

Collaboration Diagram targets the flow of messages between objects and how these objects are related to each other. It offers a clear insight of the communication and the messages that passes through the system. This diagram can be used to comprehend a system in terms of how it contributes to attaining a particular result.

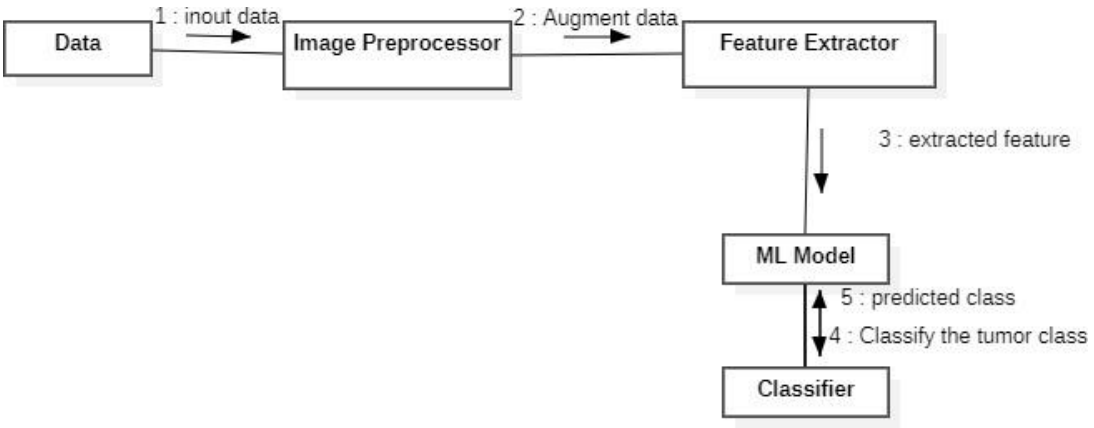


Figure 3.1.8 Collaboration diagram

Figure 3.1.8 above is a collaboration diagram showing how the various parts of a tumor classification system interact with each other. The process begins with Data, where raw input data (Step 1: (input data) where by raw information is provided for analysis. The Image Preprocessor performs data augmentation (Step 2: use augment data) to add more and enrich the given data set and also make the model more reliable. The processed data is then passed to the Feature Extractor, which extracts meaningful features (Step 3: extracted feature) significant for classification. These features are fed into the ML Model, which classifies the tumor into specific categories (Step 4: classify the tumor class). Finally, the Classifier outputs the predicted class (Step 5: probabilistic or categorical output which is equivalent to the predicted class that give the final verdict in a particular case. This approach of work organization guarantees effective and precise classification of tumors.

4. PROJECT DESCRIPTION

4.1 METHODOLOGIES

The methodology involves data acquisition and preparation for formation of histopathology images for learning and evaluation. The datasets for the intent of identifying bone cancer were collected from public domains where the images had tags. To ensure that the input is uniform in size and form, the photos were preprocessed by methods such as resizing and standardization.

To prevent over fitting and broaden the training data more data augmentation techniques were used whereby the data was scaled, flipped alongside random rotations. These strategies enhance the extent to which the model is able to generalize by replicating the standard variances that may be observed in actual medical imaging data.

The collected data is put into four CNN models, which includes ResNet50, VGG16, MobileNetV2, DenseNet121 and U-Net to improve the performance of feature extraction on medical images at the same time as sample segmentation and classification. To extract features for each of the four models, and to classify in each case, the segmented images produced through the U-net network were utilised.

This approach ensure that in identification of fine patterns, different strengths of each model are utilized and therefore improves the diagnosis of bone cancer. In evaluating how successful the model is in distinguishing between bone cancer and the values got from histopathological test results the parameters used were precision, recall, f1 score and accuracy.

4.1.1 DATA COLLECTION:

Data collection is the first and important process when starting any machine learning project, specifically in case of medical images. This means that for deeper models the quality of training data, which has to be labeled, increases proportionally and determines the quality of projections. In particular, the bone cancer dataset for use in this study was obtained from online archives that house histopathological images of cancerous tissues. For example, the data contains several varieties of bone cancer images, all of which are labeled carefully so each sample is accurately labeled as non-tumor, non-viable-tumor, viable tumor or viable_non-viable tumor.

After the collection of the dataset, data preprocessing was performed to prepare the input data that we wanted to feed to the machine learning models. Matched-pair t-tests were used to reduce picture dimensions to a standard size and to normalize pictures to enhance the performance of the model. This was important because changes in image size and intensity can affect the training and specifically for CNNs which are quite sensitive to such changes.

Furthermore, the dataset is also divided in four classes depending of the presence or not of a tumor. This division into categories was helpful for the training and subsequent evaluation as it outlined more clearly just what was expected of the model in terms of classification.

To enhance the reliability of the evaluation of the model, the data set was split into 80/20 training set and test set. This ratio has been coined famous for its ability to strike a balance that is adequate training with a view that the model will adequately be tested on new data.

4.1.2 EXPLORATORY DATA ANALYSIS:

Once we gathered the data, it is analyzed some more to look for more patterns to make the system perform at its best. The dataset for bone cancer classification comprises four distinct classes: Rather, we can distinguish non-tumor, non-viable tumor, viable tumor, and viable_non-viable tumor. An initial analysis of the data distribution also highlighted the existence of a large number of instances corresponding to the class labeled as non-tumor while other classes are scarcer. However, this still presents an issue because the quality and distribution of the training data largely determines the model's ability to perform the identification.

Since most deep learning models are trained to classify pixels as either having or not having the defined objects or characteristics of interest, undersampled classes like non-viable tumor, viable tumor, and viable_non-viable tumor are likely to be ignored by the model during training, hence the trained model is likely to be a biased model that does not generalize easily well for minority classes. One of the most common approaches when working with imbalanced datasets is undersampling the largest class or oversampling the smallest classes (SMOTE or data augmentation, for example) and although effective, they also pose certain problems. Over sampling on the majority class might lead to oversampling of non informative or irrelevant features, and thereby has negative impact on the diverseness of training data set sampled. However, oversampling when not well managed might result to over fitting or duplicity of data in the data set.

Hence, it is crucial to gain a balance of the classes without distorting the richness of the dataset and its general representativeness. It also maintains dimensions of similarity across the classes to avoid distorting the required distinctions sharply with the help of the model. In this, it is seen that the class distribution of the dataset is rather imbalanced, as shown in figure 4.1.1.1 below, which should justify the intention to implement appropriate measures for improving the training and performance of the models under study.

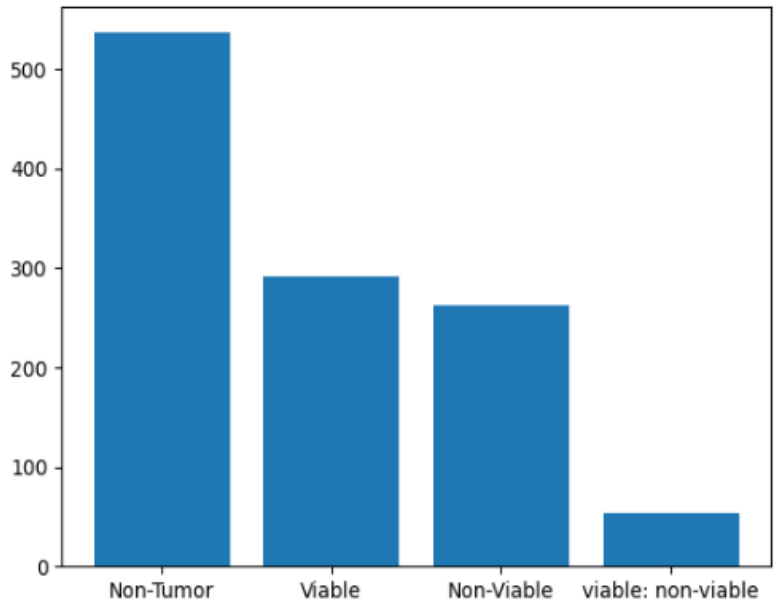


Figure 4.1.2.1: Class distribution in the dataset

4.1.3 DATA AUGMENTATION:

A range of data augmentation procedures was utilized to reduce overfitting by effectively increasing the training dataset's size and improving the model's generalizability. In medical imaging tasks, especially for bone cancer diagnosis, data availability is often limited due to the complexity and cost of acquiring annotated datasets. This scarcity makes augmentation not just an enhancement but a necessity for training robust models that can handle real-world variability.

By augmenting the dataset, we aimed to recreate the variations typically encountered in medical scans, such as differences in orientation, scale, and structure. These augmentations help the model adapt to the subtle differences in imaging conditions, patient positioning, or scan quality, thereby making it more reliable in clinical applications. Among the augmentation techniques applied were random rotations, which mimic different scan orientations often observed during the imaging process. Rotating the images ensures that the model does not become overly reliant on specific orientations, thereby improving its ability to detect tumors regardless of how the scan was performed.

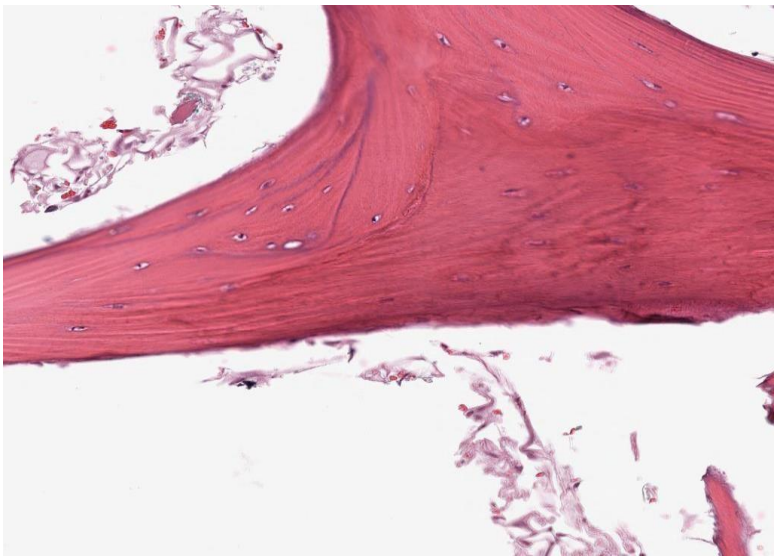


Figure 4.1.3.1: Before augmentation

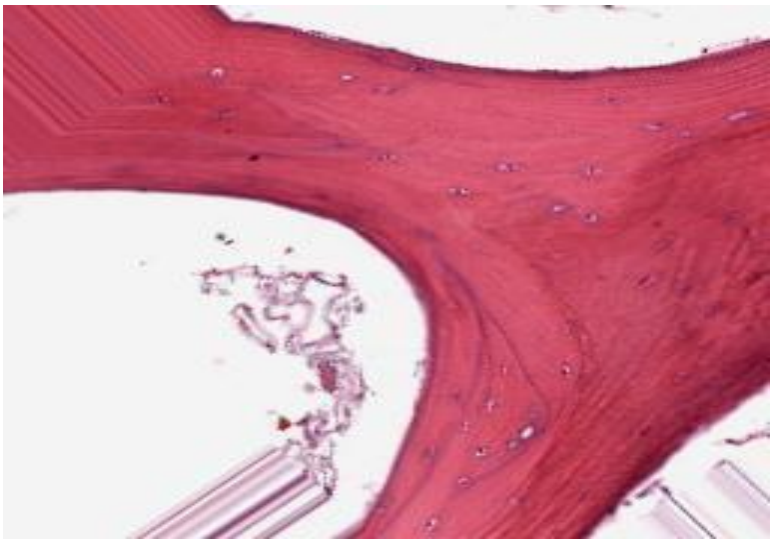


Figure 4.1.3.2: After augmentation

Before augmentation the medical images are in their raw form and sometimes do not possess enough variation for proper training. After augmentation, they are further adjusted by random rotations, scales, and flip operations to acquire the diversity that may reflect the real-varying medical scans.

4.2 MODEL ARCHITECTURE:

4.2.1 U-Net:

An encoder-decoder structure is used which will not only generate the class labels but generates masks that are of same dimensions as input images, by eliminating need for a fully connected network making it useful for the identification of what is in the image.The encoder framework consists of four encoder blocks,each including an activation function of ReLU after two layers of convolution with a valid padding and 3x3 kernel size.The result is sent into a 2x2 kernel-sized max pooling layer, which splits learnt spatial dimensions into half and lowers the training cost of the model.

The bottom layer, known as the bottleneck layer, is composed of two convolutional layers, subsequent to ReLU, and is situated between the encoding and decoding networks. The final aspect of feature map illustration is created by this layer. U-Net's skip connections along with decoding frameworks are its key strengths in segmenting images.

Decoder network up samples the provided feature maps to imitate input image’s size. This network uses the Feature Map associated with bottleneck layer to produce segmentation masks utilizing skip connections. It assists in determining where the object is located in the image. Each of the four decoder blocks in it has a transpose convolution layer with a 2x2 kernel size. Following this output's concatenation with the encoder block's associated skip layer connection, the network applies a Relu activation function and a pair of convolutional layers with kernel sizes of 3 by 3.

By using skip connections, we can more effectively create our segmentation map using the contextual feature data that was gathered in the encoder blocks. The idea is to project the bottleneck layer's output, or feature map using high-resolution features that have been learnt from the encoder blocks (via skip connections).

The last decoder block is immediately followed by a 1x1 convolution with sigmoid activation, resulting in the output of a segmentation mask with pixel-wise

classification. Thus, by employing a U-Net, it is possible to obtain both the feature information and the localization.

4.2.2 DenseNet121:

DenseNet121 is a densely connected CNN. It has a dense block ,transition layer which reduces the number of channels without losing the feature. In this model every layer gets input from the preceding layer.Each layer shares the knowledge gained which reduces the duplication of the calculation and makes it easy for the model to learn. The reuse of features makes it pass information efficiently through the dense layers.

4.2.3 ResNet:

Residual Network is a deep learning model that was aimed to help train deeper neural networks. It does so by exploiting some pathways called “skip connections” which enable information to leap across some layers.One of the issues, which can occur in deep networks, can be fixed, for instance, one can lose information as it passes through layers.

4.2.4 VGG 16:

VGG 16 is a deep learning model, used for image classification and it consists of a systematic design structure with multiple convolution and pooling layers consecutively. It accepts only images of a fixed size but then extracts hierarchical features by utilizing small convolutional filters and using pooling layers that keep reducing the image spatial size. Following the multiple convolutional blocks the feature maps are flattened and fed to fully connected layers, the ones which do the reasoning and classification.

4.2.5 MobileNetV2:

MobileNetV2 is a deep neural network architecture suitable for use in low power platforms Depthwise separable convolutions are applied which decrease load while increasing accuracy. Its main idea is an inverted residual building block that allows carefully preserving the features of the data during the transformation.

4.3 COMPARATIVE STUDY:

In this paper, we present the result of a comparative analysis of four state-of-art deep learning architectures namely DenseNet121, MobileNetV2, VGG16, and ResNet50 on a dataset. Each model uses transfer learning where their first layers are set to be non-trainable and then gradually trained on the dataset. The DenseNet121 architecture allowed efficient parameterization, facilitated by dense connections improving feature reuse, and a high accuracy was reached. Another effective model is MobileNetV2 which is narrow and deep both speeding up and delivering results at a needed performance level for mobile use.

Although VGG16 is larger and slower, depth makes it a good feature extractor. Last but not the least ResNet50 comes with residual connections which seems to have proved effective in mitigating vanishing gradient problems which makes training to be stable and fairly high accurate especially after fine-tuning. All models have different advantages and disadvantages concerning accuracy, computational time, and inference time where DenseNet121 was the most efficient model with respect to Accuracy, Precision, Recall and F1-score. MobileNetV2 capacitated all parameters very closely while VGG16 and ResNet50 stood lower.

5. CONCLUSIONS AND WORK SCHEDULE FOR PHASE II

As we performed Exploratory Data Analysis (EDA) for this project, it became quite noticeable to us that the given dataset has a severe class imbalance problem. The above imbalance posed a serious unfairness issue since it was likely to result into models that favored the majority class while giving minimal attention to the minority classes. To counter this, data augmentation techniques such as rotation, flipping and zooming were adopted with an aim of increasing the variety of the datasets obtained and to reduce risks of over-fitting the models. These augmentations proved to facilitate augmenting the variability of data from medical imaging thus making the model to better perform when applied on data not used during training.

Firstly, histopathology images were selected and processed to provide contrastive features through a U-net segmentation model which is currently considered accurate among the current models. The segmented regions of interest were then input into many classifiers including ResNet 50, VGG 16, Mobile Net V2, Dense Net 121. Applying these models, we intended to measure their suitability for the assessment of accuracy, precision, recall, and F1 score. They say that each of these models has its own advantages and disadvantages which helped the authors decide it is suitable for bone cancer detection.

The selection of these models was made consciously because they are well known for their performance on image classification tasks. ResNet50 incorporating residual connections are supposed to journey deep architectures by addressing the vanishing gradient problem. Therefore, VGG16 uses a deep and very simple architecture that has the same number of filters in each layer. MobileNetV2 is specially designed for lightweight applications with a major concern in computational costs, thus it would be very suitable for usage in medical applications with constrained hardware capabilities. Lastly, DenseNet121 adopts dense connections between layers and this significantly

improve gradient flow and feature reusing, which is important for processing high resolution medical images.

In the experiment phase the models have shown the fluctuation in their performs and some models performed well than others. Finally, DenseNet121 turned out to be the most effective model, with the accuracy estimation of 79.88%, which is just a little higher compared to MobileNetV2 with the accuracy estimation of 75.93%. That DenseNet121 performs better than other models is due to its characteristic of extracting and reusing features to a great extent, which is particularly suitable for medical imaging applications with exceedingly delicate details. The model was also counterpoise in other parameters like precision, recall besides F1 score thereby forming a reliable model for this task.

For the trained model, both MobileNetV2 and DenseNet121 models were evaluated; while DenseNet121 provided slightly better results, MobileNetV2's result was also was good, thus making it ideal for use in a scenario with limited network and computational resources. The described model had a relatively simple structure, which enabled it to work with the data quickly, albeit slightly sacrificing the accuracy, which makes it an effective solution for cases that require less accuracy but more operating speed.

At the same time, such models as VGG16 and ResNet 50 had lower results when compared with the specified metrics. Having such accuracy of 73.96% the result was still lower than the outcomes given by DenseNet121 and MobileNetV2. This might have resulted to this gap, due to its poor ability in capturing intricate features in the data set. ResNet50 fared worst, with an accuracy of 57.50% the lowest among the tested models, and suits3275955 was the most suitable model to differentiate between fake and real data. This could be attributed to its less ability to learn due to the highly sensitive high resolution medical images as well as the nature of the dataset which was imbalanced with much lesser percentage of positive samples.

To improve the performance even more, we looked at the transformer-based architectures which were more promising and newer in image classification. We were

able to pay attention to specific areas in the histopathology images using the techniques of fine-tuning and attention mechanisms to make the differentiation between the classes accurate. Vision Transformers (ViT) and their derivatives were even more beneficial in harnessing self-attention to model the long-range connections in the images.

The analysis provided here confirmed that among all the models DenseNet121 architecture was superior to the others across all the measures, making it the most suitable for further application of bone cancer detection and segmentation in the framework of this study. Its high pro metric which depicts the precision and high recall that represents the reduction of false positives / false negatives common in medical diagnosis. Likewise, the subsequent developments in transformer-based architectures and fine-tuning efficiency described here created the common foundation for future improvements in processing similar radiology images, providing an optimal preliminary solution to the future studies.

Table 5.1: Performance metrics

Model / Metrics	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
MobileNetV2	75.93	76.10	75.93	75.96
DenseNet121	80.01	79.96	79.88	79.85
VGG16	73.96	73.79	73.96	73.44
ResNet50	57.50	60.59	57.50	58.25

The best performers in this investigation, DenseNet121 and MobileNetV2, showed their ability to run time for applying in clinical practice. DenseNet121 provided precise and accurate feature learning from the histopathology images with maximum recall values along with high accuracy whereas MobileNetV2 has less computational overhead compared to others and hence provides fast completion of procedure.

These models used in the study achieved image processing speeds of less than 0.5 seconds and can therefore be useful in diagnostic applications. Although

DenseNet121 performed well in feature extraction, MobileNetV2 had an efficient architecture with appropriate performance and speed.

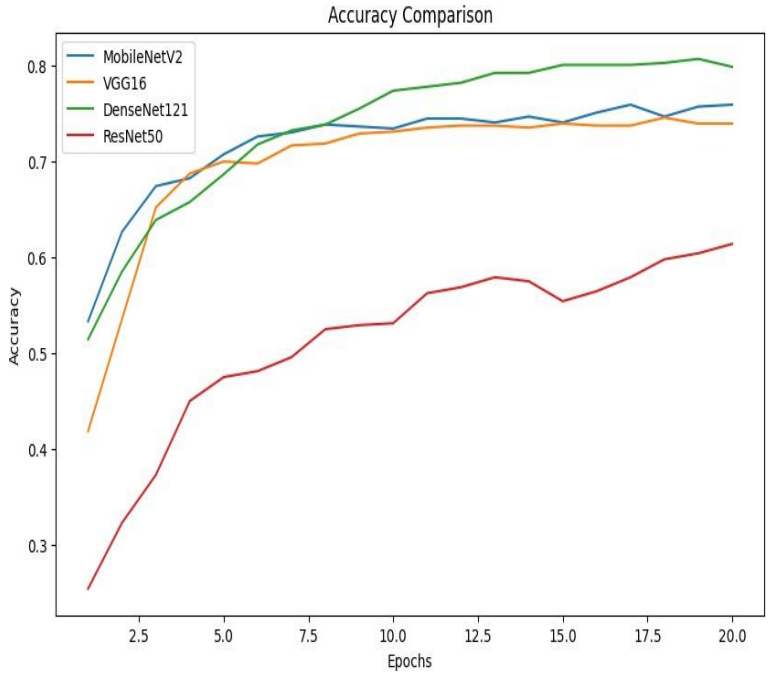


Figure 5.1: Accuracy comparison

In order to improve bone cancer diagnosis, MobileNetV2's efficiency would provide the ideal balance, while DenseNet121 appeared to perform better than other models on the dense feature extraction front. By highlighting these aspects, these models can improve diagnosis speed and accuracy.

The model's real-time processing capability suggests its potential for clinical applications, enhancing efficiency in diagnostic workflows.These enhanced detection methods discussed in this paper could greatly improve the clinical practice by offering quicker and more reliable diagnosis of bone cancer.With increased accuracy and precision, the model can aid medical professionals in identifying early-stage tumors, which could lead to better patient outcomes and fewer diagnostic errors.

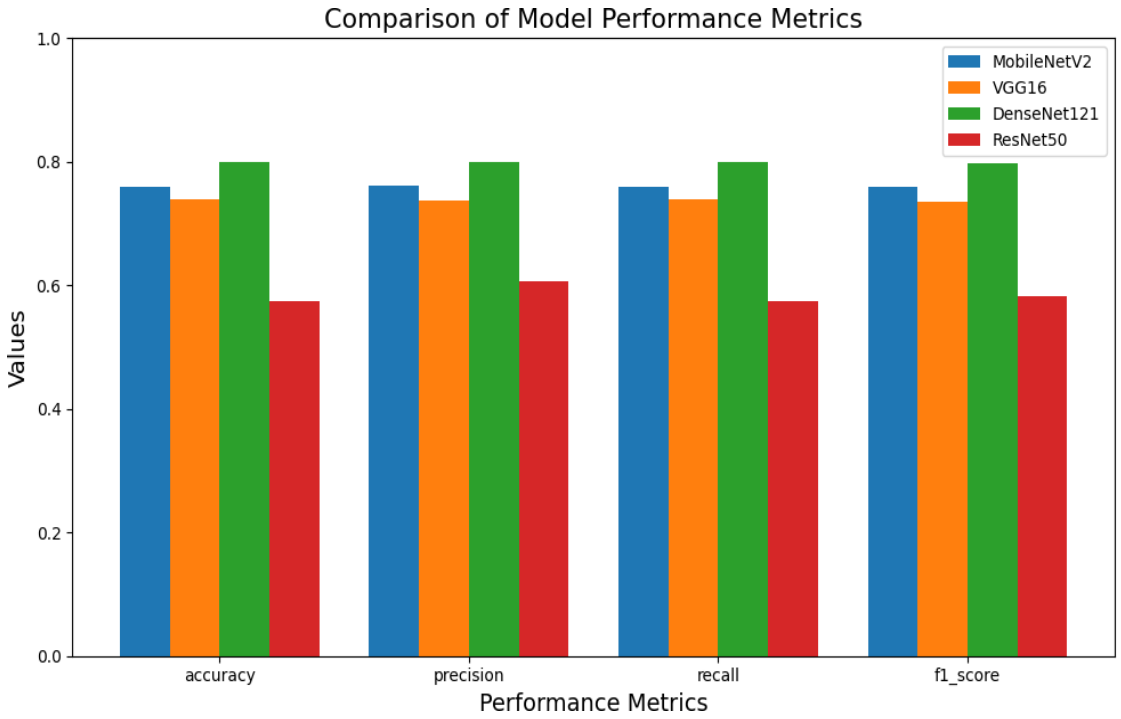


Figure 5.2: Performance metrics

However, the study faced limitations, including the dataset size, as the availability of labeled histopathology images for bone cancer remains limited. Although augmentation techniques were employed to expand the effective training size, this constraint poses a challenge. Additionally, the model required substantial computational resources, which can be a drawback in implementing the model in conditions of limited computational resources.

Future Work:

For future work, several promising directions could be explored to further enhance the performance of the U-Net and Vision Transformer (ViT) pipeline in bone cancer detection. One area for improvement is incorporating advanced attention mechanisms within the U-Net architecture to refine focus on critical regions during segmentation. This would help the model distinguish more accurately between tumor and non-tumor regions, especially in cases where the cancerous areas are small or surrounded by noise. Additionally, integrating attention mechanisms more deeply into the ViT could fine-tune the transformer model for better performance by allowing it to prioritize important image patches and discard irrelevant information more effectively. This fine-tuning could involve multi-head attention adjustments or hierarchical attention layers, enabling the model to capture intricate relationships between different regions of the image and further boost its classification accuracy.

Another potential area for improvement is fine-tuning the entire pipeline through extensive experimentation with different ViT configurations, such as adjusting the number of layers or attention heads to balance computational cost with accuracy. Optimizing the model's hyperparameters could also lead to improved results by making the ViT more sensitive to smaller and more nuanced details in the input data. Furthermore, introducing additional layers of fine-grained attention could help the model better capture subtle differences between healthy and cancerous tissues, leading to more accurate predictions. By focusing on enhancing the attention mechanisms within both U-Net and ViT, future research can improve the system’s overall capability for precise and reliable bone cancer detection.

APPENDIX-I

PUBLICATION STATUS

PUBLICATION STATUS OF PHASE I PAPER

TITLE: **Improved Bone Cancer Diagnosis: Transfer Learning Integration with U-Net for Segmented Histopathology Image Analysis**

AUTHOR: **Mr. Deepak Kumar K, M.E.,**
Dr. S. Senthil Pandi, M.E., Ph.D.,
Dr. P. Kumar, M.E., Ph.D.,
Madhulika G,
Mahalakshmi K.

CONFERENCE: **International Conference On Emerging Research In Computational Science -2024**

MODE OF PUBLICATION: **Online**

STATUS: **Accepted**

Paper Acceptance E-mail:

Name of the conference: INTERNATIONAL CONFERENCE ON EMERGING RESEARCH IN COMPUTATIONAL SCIENCE - 2024
Id of submission: 305
Title of submission: Improved Bone Cancer Diagnosis: Transfer Learning Integration with U-Net for Segmented Histopathology Image Analysis
Status of submission: Accept

Improved Bone Cancer Diagnosis: Transfer Learning Integration with U-Net for Segmented Histopathology Image Analysis

Deepak Kumar K
Department of CSE
Rajalakshmi Engineering College
Chennai, India
kdeepak.srmit@gmail.com

Senthil Pandi S
Department of CSE
Rajalakshmi Engineering College
Chennai, India
mailtosenthil.ks@gmail.com

Kumar P
Department of CSE
Rajalakshmi Engineering College
Chennai, India
kumar@rajalakshmi.edu.in

Madhulika G
Department of CSE
Rajalakshmi Engineering College
Chennai, India
210701139@rajalakshmi.edu.in

Mahalakshmi K
Department of CSE
Rajalakshmi Engineering College
Chennai, India
210701143@rajalakshmi.edu.in

***Abstract*—Bone cancer, while rare, poses significant health challenges with often late diagnoses. Though CNN, proposed as a good algorithm for classifying and detecting cancer from images, it faces challenges when it is a histopathological image. This study introduces a novel approach combining machine learning with cutting-edge image processing methods to increase the precision and effectiveness of diagnosis. The model retains several key features. Utilizing U-Net for segmentation and four different CNNs, ResNet50, VGG16, MobileNetV2, and DenseNet121 for classification, along with data augmentation strategies, our method aims to enhance the reliability of bone cancer detection. Preliminary results demonstrate improved performance in accuracy and speed compared to traditional methods.**

Keywords—Bone Cancer, Machine Learning, U-Net, ResNet50, VGG16, MobileNetV2, DenseNet121, Image Processing, Data Augmentation.

I. INTRODUCTION

Bone Cancer is also called Osteosarcoma. The multiplication of osteoblastic cells helps in formation of bone. When this happens rapidly it leads to the onset of malignant disease bone cancer [1]. The symptoms of the disease include pain and fracture might happen when the bone becomes weak. It is detected that the majority of the formation of cancerous cells forms at the knee in the case of this type of cancer. The diagnosis of the disease is usually done with X Rays. In the case of tumour, a surgery called biopsy is done to examine how far the disease has affected the cells and damage in the cell is identified from the small tissue sample extracted from the body. The extracted tissue sample is processed for the staining stage .Natural black 1 and eosin are usually used for this stage of the study or examination of the tissue for cells that are cancerous. After staining, the process of diagnosis requires a equipped pathologists for examining. It also depends on the microscope used. The process is prone to errors due to external factors and it is time consuming [2]. The microscopic examination of tissue is difficult and it provides limited information about the pattern, key patterns might be missed, not appropriate for a very large amount of sample.

The diagnosis process can be improved with help of advanced machine learning algorithms in terms of precision and effectiveness. Deep neural networks are widely employed in the applications of processing and classifying images. The image processing [3] followed by classification is majorly based on the data from which it learns. Image for processing of bone cancer is not much available and it is difficult to get real time data. We need to approach the hospital and get the permission of the patient and hospital, which requires confidentiality and time. The efficiency of the model depends on the data, it needs to be preprocessed and duplicated if the data is less. The technique of data augmentation has been employed to expand the data, which improves the model's performance. It also enables the model to pick up on various aspects of the image thus making sure that key features are not missed. For classification the data should be segregated as a separate folder as classes.

The histopathology image is a high resolution image and the algorithm used should be able to decide the global features of the image. Segmentation of the images [4] in this study is done using the U-Net architecture, which is helpful in determining the feature of interest. The segmented dataset is then classified using four distinct convolutional neural network architectures: These are ResNet50, VGG16, MobileNetV2 and DenseNet121.Implementing the chosen data augmentation techniques the model improves its insensitivity to the specific data distribution and handles the class imbalance problem [5]. The incorporation of these models greatly helps in the enhancement of the detection and identification of the cancerous cells within the images so it will reduce time in terms of classification speed and gains time in diagnosis deeply the steps that are involved in the approach proposed in this paper are as follows:

The steps that are involved in the approach proposed in this paper are as follows:

979-8-3315-3496-7/24/$31.00 ©2024 IEEE

- Data Collection
- Data Augmentation
- Feature Extraction
- Model Training
- Performance measure

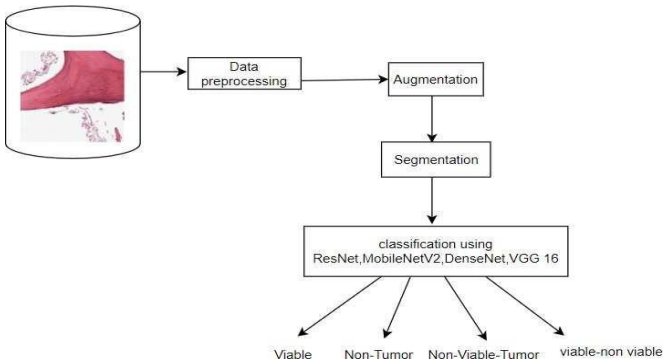


Figure. 1. Flow Diagram

II. Literature Survey

Jasper Gnana Chandran, J et al. [6] focused on identifying cancer type that occurs within the marrow of bone by following deep learning techniques which include data synthesis using augmentation by taking out the edges, features, rotating the picture and finally classified using three layers. They proposed dense convolutional neural network for determining cancer.

Walid et al. [7] conducted in order to build a model for the hard-voting, unbiased detection of bone cancer. The highest kappa and accuracy scores are 87.55% and 92.14% for MobileNetV2, respectively. Performing better, with an accuracy percentage of 93.88%, is the Max voting classifier.

Wu et al. [8] discussed about the accurate detection of nuclei in osteosarcoma using pathological images by considering the differences in styles of the stain and improving the diagnosis in underdeveloped regions with the help of conv-transformer architecture called ENMVit.

Loraksa et al [9]. proposed a module for detecting lung nodules in osteosarcoma using SSD-VGG16 which detects the nodules by constructing the bounding box around them and calculating the confidence scores with accuracy of 75.97%. The proposed model could not detect the lung nodules when the size of the nodules in the image is very tiny or blur.

Saber et al. [10] proposed a model for breast cancer classification using a pre-trained network combined with CNN by performing comparison with other four models, VGG-16 achieved high accuracy. The data is enhanced with the help of augmentation, fine tuning and cross validation is done to overcome overfitting.

Sun et al. [11] proposed a model where the histopathological images are segmented into patches using transfer learning before image level features are extracted using MIL and classified the liver cancer images into normal and abnormal. They used the approach of selecting and sorting the k top and bottom features for classification.

Weng et al. [12] employed NAS in medical image segmentation and in the NAS-Unet model, two supporting cells which are downsampling (DownSC) and the upsampling (UpSC) cells improve the model. To address issues related to small batch sizes, the architecture applies group normalization besides using a DAG-based mixed operation strategy for architecture search. During training, binary gating occurs by changing two paths at once in order to cut the GPU memory usage in half.

Zeng et al. [13] improved the actual efficiency in segmenting diseased cells, residual blocks, multi-scale inception modules, and channel attention modules were incorporated into the planned RIC-Unet architecture. This structure handles overlapping and various cell topologies by using both local and global context. This model uses focal loss to assist it focus on hard samples during the training phase and achieve joint nuclei and contour segmentation.

Fu et al. [14] created a subnetwork of the multifunctional spatial focused attention module, with an encoder-decoder core serving as its foundation. This model utilizes PET-spatial attention maps that guides tumor localization in CT feature maps, enhancing segmentation by integrating spatial information from both PET and CT modalities.

Ashwath et al. [15] combined three branches which were targeted for determining the type of medical images with scattered and irregular regions, a three-tier self-interpretable DCNN model was presented. Three branches are used: a global subdivision that uses the entire image to extract patterns, an attention subdivision that highlights important regions, along with a fusion subdivision that combines information from both to improve categorization.

III. Methodology

The methodology entails data collecting and preprocessing to create histopathology images for training and assessment. Datasets for the purpose of detecting bone cancer were obtained from publicly accessible sources that contained tagged images. For consistency in input size and format, the photos were preprocessed using techniques including scaling and normalization. In order to reduce overfitting and broad the amount of training data, additional techniques for augmenting data were applied, including scaling, flipping, and random rotations. By mimicking common variances found in real-world medical imaging, these strategies improve themodel's generalization.

The collected data is supported in four CNN models, namely ResNet50, VGG16, MobileNetV2, DenseNet121 and U-Net which are used to enhance feature extraction on medical images while enhancing the sample segmentation and classification. For each of the four models, the segmented images from the U-Net network were used as input in order to extract optimal features and to classify. This approach guarantees that different strengths of each model are used in identification of fine patterns and the accuracy of the diagnosis of bone cancer is enhanced. To assess the degree of success of the model to differentiate between bone cancer and results derived from histopathological findings measures including precision, recall, f1 scores and accuracy were used.

Data Collection: The collection of precise as well as valid data is essential for model's training in order to get the highest level of classification accuracy that makes the model more credible. This study uses the publicly accessible data set in which bone cancer images can be found as histopathological photos. After collecting data, preliminary processing is carried out to transform the images to a form which the model

may accept as input.Data is first divided into four folders illustrating four classes: non-tumor, non-viable-tumor, viable and viable_non-viable respectively. Afterwards, data is separated into train and test data with a proportion of 80:20.

Exploratory Data Analysis: After collecting the data it is further analyzed for better performance of the system.There are four classes of bone cancer they are non-tumor, non-viable-tumor, viable tumor and viable_non-viable tumor.From the observation of analysis it was found that the other classes except non-tumor was undersampled.The performance of the model depends on the data used for training.Minimizing the maximum data and maximizing the minimum data may lead to loss of important features.

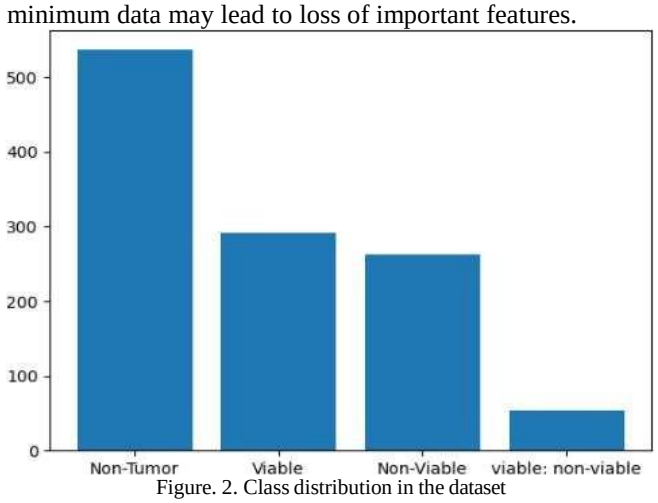


Figure. 2. Class distribution in the dataset

Data Augmentation: A range of data augmentation procedures were utilized to decrease overfitting by increasing the training dataset's effective size and improve our model's generalizability. Since medical imaging data is not readily available for the diagnosis of bone cancer, augmentation is essential in recreating real-world deviations commonly found in medical scans. Among the augmentation techniques were random rotations to mimic different scan orientations, flipping (horizontally and vertically) to account for mirrored or inverted scans, and scaling to introduce variation in the size of regions of interest.

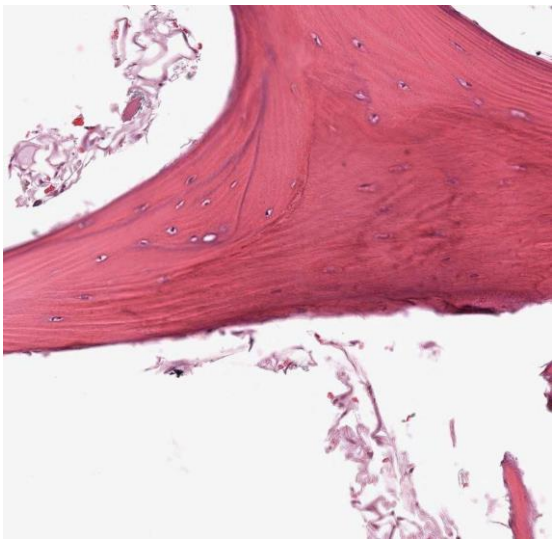


Figure. 3. Before augmentation

Model Architecture:

U-Net: An encoder-decoder structure is used which will not only generate the class labels but generates masks that are of same dimensions as input images, by eliminating need for a fully connected network making it useful for the identification of what is in the image.The encoder framework consists of four encoder blocks,each including an activation function of ReLU after two layers of convolution with a valid padding and 3x3 kernel size.The result is sent into a 2x2 kernel-sized max pooling layer, which splits learnt spatial dimensions into half and lowers the training cost of the model. The bottom layer, known as the bottleneck layer, is composed of two convolutional layers, subsequent to ReLU, and is situated between the encoding and decoding networks. The final aspect of feature map illustration is created by this layer. U-Net's skip connections along with decoding frameworks are its key strengths in segmenting images.

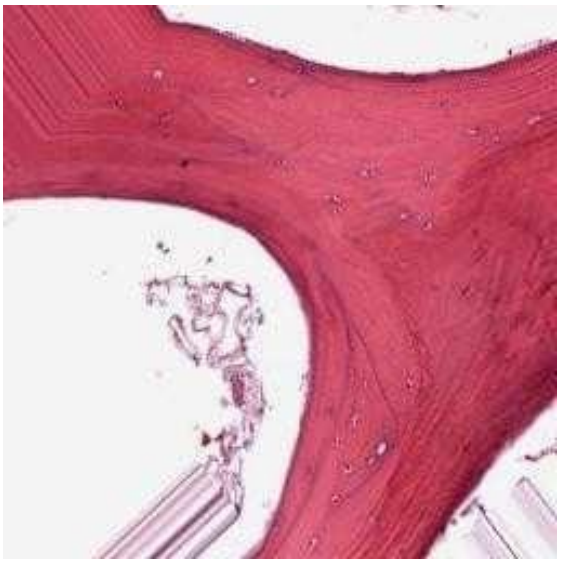


Figure. 4. After augmentation

Decoder network up samples the provided feature maps to imitate input image’s size. This network uses the Feature Map associated with bottleneck layer to produce segmentation masks utilizing skip connections. It assists in determining where the object is located in the image. Each of the four decoder blocks in it has a transpose convolution layer with a 2x2 kernel size. Following this output's concatenation with the encoder block's associated skip layer connection, the network applies a Relu activation function and a pair of convolutional layers with kernel sizes of 3 by 3.

By using skip connections, we can more effectively create our segmentation map using the contextual feature data that was gathered in the encoder blocks. The idea is to project the bottleneck layer's output, or feature map using high-resolution features that have been learnt from the encoder blocks (via skip connections).

The last decoder block is immediately followed by a 1x1 convolution with sigmoid activation, resulting in the output of a segmentation mask with pixel-wise classification. Thus, by employing a U-Net, it is possible to obtain both the feature information and the localization.

DenseNet121 is a densely connected CNN. It has a dense block ,transition layer which reduces the number of channels without losing the feature. In this model every layer gets input

from the preceding layer.Each layer shares the knowledge gained which reduces the duplication of the calculation and makes it easy for the model to learn. The reuse of features makes it pass information efficiently through the dense layers. Residual Network is a deep learning model that was aimed to help train deeper neural networks. It does so by exploiting some pathways called “skip connections” which enable information to leap across some layers.One of the issues, which can occur in deep networks, can be fixed, for instance, one can lose information as it passes through layers.

VGG 16 is a deep learning model, used for image classification and it consists of a systematic design structure with multiple convolution and pooling layers consecutively. It accepts only images of a fixed size but then extracts hierarchical features by utilizing small convolutional filters and using pooling layers that keep reducing the image spatial size. Following the multiple convolutional blocks the feature maps are flattened and fed to fully connected layers, the ones which do the reasoning and classification.

MobileNetV2 is a deep neural network architecture suitable for use in low power platforms Depthwise separable convolutions are applied which decrease load while increasing accuracy. Its main idea is an inverted residual building block that allows carefully preserving the features of the data during the transformation.

In this research, we present the result of a comparative analysis of four state-of-art deep learning architectures namely DenseNet121, MobileNetV2, VGG16, and ResNet50 on a dataset. Each model uses transfer learning where their first layers are set to be non-trainable and then gradually trained on the dataset. The DenseNet121 architecture allowed efficient parameterization, facilitated by dense connections improving feature reuse, and a high accuracy was reached. Another effective model is MobileNetV2 which is narrow and deep both speeding up and delivering results at a needed performance level for mobile use. Although VGG16 is larger and slower, depth makes it a good feature extractor. Last but not the least ResNet50 comes with residual connections which seems to have proved effective in mitigating vanishing gradient problems which makes training to be stable and fairly high accurate especially after fine-tuning. All models have different advantages and disadvantages concerning accuracy, computational time, and inference time where DenseNet121 was the most efficient model with respect to Accuracy, Precision, Recall and F1-score. MobileNetV2 capacitated all parameters very closely while VGG16 and ResNet50 stood lower

Implementation:

TensorFlow and Keras were used for data augmentation after EDA was used to check for disparities in classes in the data cleaning.

IV. Results & Discussions

In this work, we tested four architectures: Among the pre-trained models MobileNetV2, DenseNet121, VGG16, and ResNet50 were included in the framework for bone cancer detection and segmentation. The analysis of the results showed a higher accuracy of DenseNet121 and MobileNetV2, where DenseNet121 had a slightly better accuracy of 79.88% compared to the accuracy achieved by MobileNetV2 of x75.93 %. To assess the performance of models, Precision, Recall, and F1-score were chosen as benchmark parameters based on prior studies; models have been found to outperform the benchmark, and DenseNet121 has been proven to be the most promising model in the current experiment. In the same vein, VGG16 and ResNet50 failed to perform exceptionally as VGG16 had an accuracy of 73.96 while ResNet50 a low rating of 57.50% accuracy.

The best performers in this investigation, DenseNet121 and MobileNetV2, showed their ability to run time for applying in clinical practice. DenseNet121 provided precise and accurate feature learning from the histopathology images with maximum phase. Image segmentation was done through the U-Net model from the TensorFlow and Keras library and classification through models such as ResNet, VGG16, MobileNet V2, DenseNet121 built using TensorFlow. In order to have quick training and cut down the time taken during learning, a GPU was employed which helps process the large dataset.

Recall values along with high accuracy whereas MobileNetV2 has less computational overhead compared to others and hence provides fast completion of procedure. These models used in the study achieved image processing speeds of less than 0.5 seconds and can therefore be useful in diagnostic applications. Although DenseNet121 performed well in feature extraction, MobileNetV2 had an efficient architecture with appropriate performance and speed.

In order to improve bone cancer diagnosis, MobileNetV2's efficiency would provide the ideal balance, while DenseNet121 appeared to perform better than other models on the dense feature extraction front. By highlighting these aspects, these models can improve diagnosis speed and accuracy.

The model's real-time processing capability suggests its potential for clinical applications, enhancing efficiency in diagnostic workflows. These enhanced detection methods discussed in this paper could greatly improve the clinical practice by offering quicker and more reliable diagnosis of bone cancer. With increased accuracy and precision, the model can aid medical professionals in identifying early-stage tumors, which could lead to better patient outcomes and fewer diagnostic errors.

However, the study faced limitations, including the dataset size, as the availability of labeled histopathology images for bone cancer remains limited. Although augmentation techniques were employed to expand the effective training size, this constraint poses a challenge. Additionally, the model required substantial computational resources, which can be a drawback in implementing the model in conditions of limited computational resources.

Table.1. Proposed Model Metrics Comparison

Model / Metrics	Accuracy(%)	Precision (%)	Recall (%)	F1-Score (%)
Mobile NetV2	75.93	76.10	75.93	75.96
Dense Net121	80.01	79.96	79.88	79.85
VGG16	73.96	73.79	73.96	73.44

ResNet50	57.50	60.59	57.50	58.25

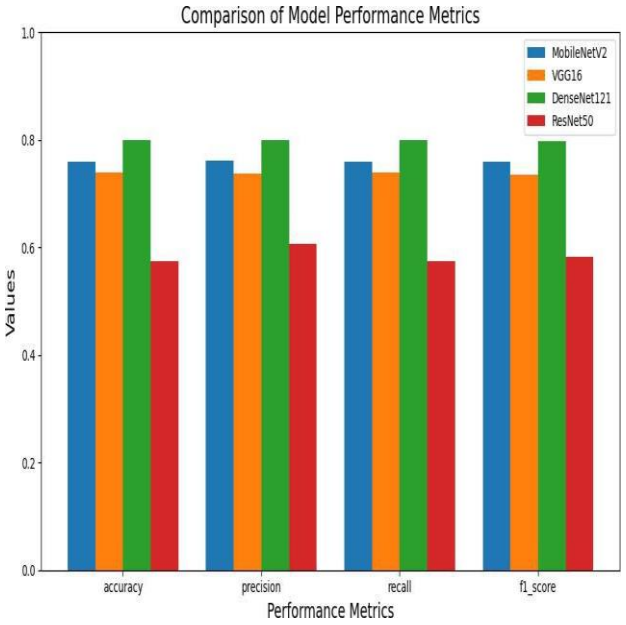


Figure. 5. Performance Metrics

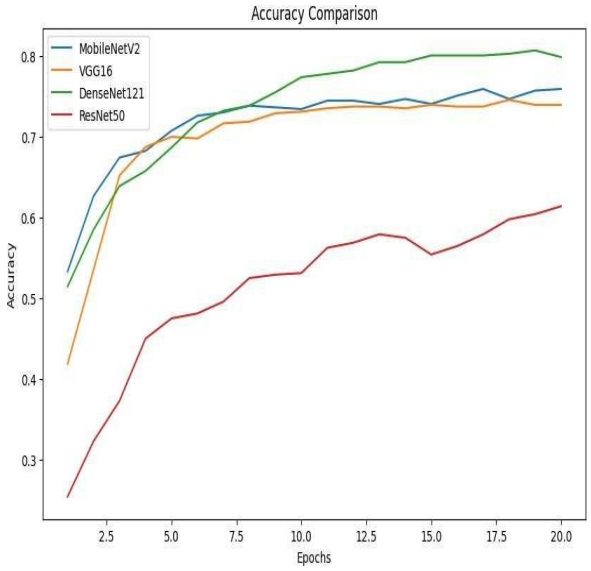


Figure. 6. Accuracy Comparison

V. Conclusion

When performing EDA in this research, we realized that we have a huge class imbalance problem in the provided dataset. To this end, we used some data augmentation strategies as rotation, flipping, and zooming in order to augment the data set and reduce overfitting of the model. First, the histopathology images were selected and segmented using the U-Net to obtain features of interest and classify them using other models including ResNet50, VGG16, MobileNetV2, and DenseNet121. By comparing the results obtained by distinct models, it was possible to measure the accuracy, precision, recall, and F1 score of each one while examining this dataset of medical images in detail. To increase the performance even more, we also looked into transformers and the use of techniques such as fine-tuning, with attention mechanisms to select certain segments within the images. To this end, through the utilization of these sophisticated methods and employing learning through transfer, we were able to compare the improved precision, recall mean F1 score, subsequently strengthening the accuracy of classifications in histopathology images analysis.

References

[1] K. T, B. S. Jeevena, J. A, S. P. S and R. C. V, "Bone Age Assessment (BAA) using Convolutional Transformer and MultiLayer Perceptron," 2024 ICOSEC, Trichy, India, 2024, pp. 1772-1777, doi: 10.1109/ICOSEC61587.2024.10722729.

[2] S. P. S, S. J, K. T and K. P, "Enhanced Classification of Oral Cancer Using Deep Learning Techniques," 2024 ICAIT, Chikkamagaluru, Karnataka, India, 2024, pp. 1-5, doi: 10.1109/ICAIT61638.2024.10690583.

[3] Kumar, P., Senthilselvi, A., Manju, I. et al. HUMRC-PS: Revolutionizing plant phenotyping through Regional Convolutional Neural Networks and Pelican Search Optimization. Evolving Systems 15, 2211–2230 (2024). https://doi.org/10.1007/s12530-024-09612-6

[4] Nagarani, N., et al. "Self-attention based progressive generative adversarial network optimized with momentum search optimization algorithm for classification of brain tumor on MRI image." Biomedical Signal Processing and Control 88 (2024): 105597.

[5] Rajagopal RK, Karthick R, Meenalochini P, Kalaichelvi T. Deep Convolutional Spiking Neural Network optimized with Arithmetic optimization algorithm for lung disease detection using chest X-ray images. Biomedical Signal Processing and Control. 2023 Jan 1;79:104197.

[6] Jasper Gnana Chandran, J., et al. "Dual-channel capsule generative adversarial network optimized with golden eagle optimization for pediatric bone age assessment from hand X-ray image." International Journal of Pattern Recognition and Artificial Intelligence 37.02 (2023): 2354001.

[7] M. A. A. Walid and P. C. Shill, "A Transfer-Learning Based Unbiased Voting Bone Cancer Detection Framework from Histological Osteosarcoma Images," 2023 ICCCNT, Delhi, India, 2023, pp. 1-7.

[8] J. Wu, T. Yuan, J. Zeng and F. Gou, "A Medically Assisted Model for Precise Segmentation of Osteosarcoma Nuclei on Pathological Images," in IEEE Journal of Biomedical and Health Informatics, vol. 27, no. 8, pp. 3982-3993, Aug. 2023.

[9] C. Loraksa, S. Mongkolsomlit, N. Nimsuk, M. Uscharapong and P. Kiatisevi, "Development of the Osteosarcoma Lung Nodules Detection Model Based on SSD-VGG16 and Competency Comparing With Traditional Method," in IEEE Access, vol. 10, pp. 65496-65506, 2022.

[10] A. Saber, M. Sakr, O. M. Abo-Seida, A. Keshk and H. Chen, "A Novel Deep-Learning Model for Automatic Detection and Classification of Breast Cancer Using the Transfer-Learning Technique," in IEEE Access, vol. 9, pp. 71194-71209, 2021.

[11] C. Sun, A. Xu, D. Liu, Z. Xiong, F. Zhao and W. Ding, "Deep Learning-Based Classification of Liver Cancer Histopathology Images Using Only Global Labels," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 6, pp. 1643-1651, June 2020.

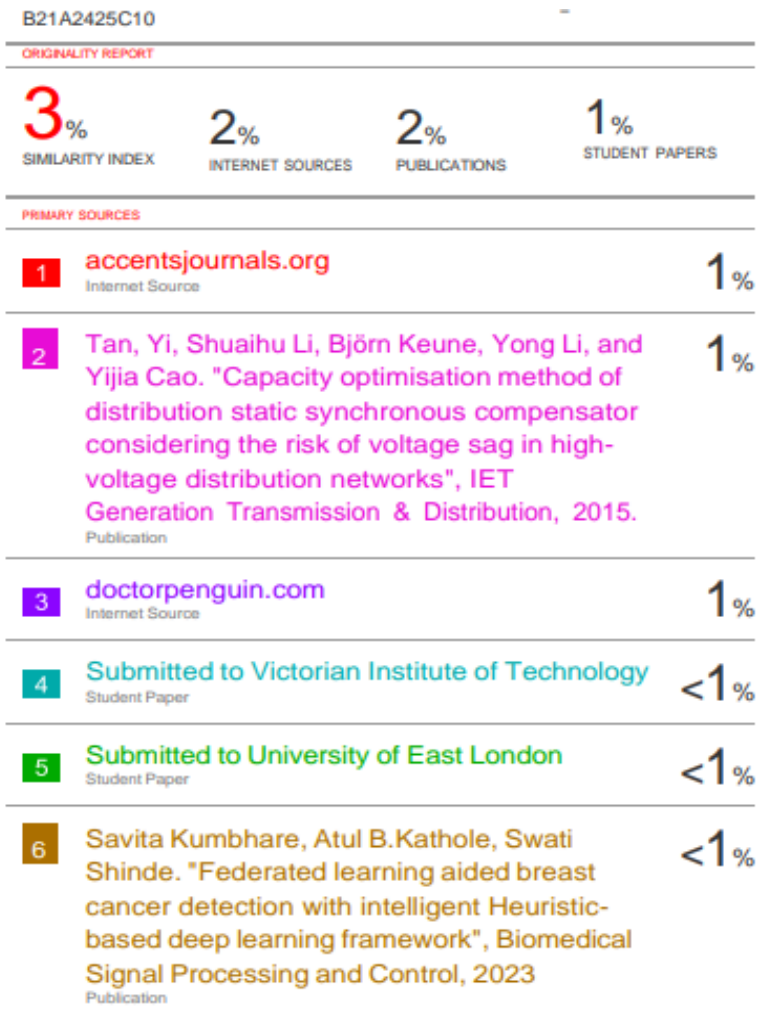
[12] Y. Weng, T. Zhou, Y. Li and X. Qiu, "NAS-Unet: Neural Architecture Search for Medical Image Segmentation," in IEEE Access, vol. 7, pp. 44247-44257, 2019.

[13] Z. Zeng, W. Xie, Y. Zhang and Y. Lu, "RIC-Unet: An Improved Neural Network Based on Unet for Nuclei Segmentation in Histology Images," in IEEE Access, vol. 7, pp. 21420-21428, 2019.

[14] X. Fu, L. Bi, A. Kumar, M. Fulham and J. Kim, "Multimodal Spatial Attention Module for Targeting Multimodal PET-CT Lung Tumor Segmentation," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 9, pp. 3507-3516, Sept. 2021.

[15] V. A. Ashwath, O. K. Sikha and R. Benitez, "TS-CNN: A Three-Tier Self-Interpretable CNN for Multi-Region Medical Image Classification," in IEEE Access, vol. 11, pp. 78402-78418, 2023.

Plagiarism report of the Conference Paper:



B21A2425C10

ORIGINALITY REPORT

3%	2%	2%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	accentsjournals.org Internet Source	1%
2	Tan, Yi, Shuaihu Li, Björn Keune, Yong Li, and Yijia Cao. "Capacity optimisation method of distribution static synchronous compensator considering the risk of voltage sag in high-voltage distribution networks", IET Generation Transmission & Distribution, 2015. Publication	1%
3	doctorpenguin.com Internet Source	1%
4	Submitted to Victorian Institute of Technology Student Paper	<1%
5	Submitted to University of East London Student Paper	<1%
6	Savita Kumbhare, Atul B.Kathole, Swati Shinde. "Federated learning aided breast cancer detection with intelligent Heuristic-based deep learning framework", Biomedical Signal Processing and Control, 2023 Publication	<1%

Plagiarism report of this Report:

B21A2425C10.pdf

ORIGINALITY REPORT

10%	7%	3%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Illinois at Urbana-Champaign Student Paper	5%
2	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in Information, Communication and Computing Technology", CRC Press, 2024 Publication	1%
3	Submitted to University of Canberra Student Paper	<1%
4	Submitted to Victorian Institute of Technology Student Paper	<1%
5	123dok.com Internet Source	<1%
6	assets.researchsquare.com Internet Source	<1%
7	doctorpenguin.com Internet Source	<1%
8	www.hindawi.com Internet Source	<1%

APPENDIX-II

```
# U-Net model
  def unet_model(input_size=(256, 256, 3)):
      inputs = Input(input_size)
      c1 = Conv2D(64, (3, 3), activation='relu', padding='same')(inputs)
      c1 = Conv2D(64, (3, 3), activation='relu', padding='same')(c1)
      p1 = MaxPooling2D((2, 2))(c1)

      c2 = Conv2D(128, (3, 3), activation='relu', padding='same')(p1)
      c2 = Conv2D(128, (3, 3), activation='relu', padding='same')(c2)
      p2 = MaxPooling2D((2, 2))(c2)

      c3 = Conv2D(256, (3, 3), activation='relu', padding='same')(p2)
      c3 = Conv2D(256, (3, 3), activation='relu', padding='same')(c3)
      p3 = MaxPooling2D((2, 2))(c3)

      c4 = Conv2D(512, (3, 3), activation='relu', padding='same')(p3)
      c4 = Conv2D(512, (3, 3), activation='relu', padding='same')(c4)
      p4 = MaxPooling2D((2, 2))(c4)

      c5 = Conv2D(1024, (3, 3), activation='relu', padding='same')(p4)
      c5 = Conv2D(1024, (3, 3), activation='relu', padding='same')(c5)

      u6 = UpSampling2D((2, 2))(c5)
      u6 = concatenate([u6, c4])
      c6 = Conv2D(512, (3, 3), activation='relu', padding='same')(u6)
      c6 = Conv2D(512, (3, 3), activation='relu', padding='same')(c6)

      u7 = UpSampling2D((2, 2))(c6)
      u7 = concatenate([u7, c3])
      c7 = Conv2D(256, (3, 3), activation='relu', padding='same')(u7)
      c7 = Conv2D(256, (3, 3), activation='relu', padding='same')(c7)

      u8 = UpSampling2D((2, 2))(c7)
      u8 = concatenate([u8, c2])
      c8 = Conv2D(128, (3, 3), activation='relu', padding='same')(u8)
      c8 = Conv2D(128, (3, 3), activation='relu', padding='same')(c8)

      u9 = UpSampling2D((2, 2))(c8)
      u9 = concatenate([u9, c1])
      c9 = Conv2D(64, (3, 3), activation='relu', padding='same')(u9)
      c9 = Conv2D(64, (3, 3), activation='relu', padding='same')(c9)
```

```
    outputs = Conv2D(1, (1, 1), activation='sigmoid')(c9)

    model = Model(inputs=[inputs], outputs=[outputs])
    return model

def load_images_and_masks(original_path, mask_path, img_size=(256, 256)):
    images = []
    masks = []

    for img_name in os.listdir(original_path):
        img = cv2.imread(os.path.join(original_path, img_name))
        if img is None:
            print(f"Warning: Unable to load image {img_name}")
            continue

        img = cv2.resize(img, img_size)
        images.append(img)

mask = cv2.imread(os.path.join(mask_path, img_name), cv2.IMREAD_GRAYSCALE)
        if mask is None:
            print(f"Warning: Unable to load mask for {img_name}")
            continue

        mask = cv2.resize(mask, img_size)

        masks.append(mask)

    return np.array(images), np.array(masks)

original_train_path = '/content/drive/MyDrive/Bone Cancer Detection Project/original_train/Non-Tumor'
mask_train_path = '/content/drive/MyDrive/Bone Cancer Detection Project/train_mask2/Non-Tumor'
segmented_output_dir = '/content/drive/MyDrive/Bone Cancer Detection Project/segmented_images_unet/Non-Tumor'

X, y = load_images_and_masks(original_train_path, mask_train_path)

X = X.astype('float32') / 255.0
y = y.astype('float32') / 255.0

model = unet_model(input_size=(256, 256, 3))
model.compile(optimizer=Adam(), loss='binary_crossentropy', metrics=['accuracy'])

model.fit(X, y, batch_size=16, epochs=10, validation_split=0.1)

model.save('/content/drive/MyDrive/Bone Cancer Detection Project/unet_model.h5')
```

```
predicted_masks = model.predict(X)
predicted_masks = (predicted_masks > 0.5).astype(np.uint8)

from tensorflow.keras.preprocessing.image import array_to_img

def apply_and_save_masks(original_images, predicted_masks, save_dir):
    for i in range(len(original_images)):
        mask = predicted_masks[i]

        if mask.ndim == 2:
            mask = np.expand_dims(mask, axis=-1)

        segmented_image = original_images[i] * mask

        segmented_image = np.clip(segmented_image, 0, 1)

        img = array_to_img(segmented_image)
        img.save(os.path.join(save_dir,  f'segmented_{i}.jpg'))

        img.save(os.path.join(save_dir, f'segmented_{i}.jpg'))

apply_and_save_masks(X, predicted_masks, segmented_output_dir)

print("Segmented images have been saved in:", segmented_output_dir)
```

#DenseNet121

```
import tensorflow as tf
import numpy as np
from tensorflow.keras.preprocessing import image_dataset_from_directory
from tensorflow.keras.applications import DenseNet121
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.model_selection import train_test_split

from sklearn.metrics import precision_score, recall_score, f1_score

# Set directory for the dataset
dataset_dir = '/content/drive/MyDrive/segmented_images_unet'

# Image size (DenseNet121 default is 224x224)
img_size = (224, 224)
batch_size = 32

# Load the entire dataset
dataset = image_dataset_from_directory(dataset_dir, image_size=img_size,
batch_size=batch_size, shuffle=True)
```

```
# Extract images and labels from the dataset
image_list = []
label_list = []

for images, labels in dataset:
    image_list.extend(images.numpy())
    label_list.extend(labels.numpy())

# Convert lists to numpy arrays
image_list = np.array(image_list)
label_list = np.array(label_list)

# Split the dataset into 80% training and 20% testing
X_train, X_test, y_train, y_test = train_test_split(
    image_list, label_list, test_size=0.2, random_state=42, stratify=label_list
)

# Convert the split data back to TensorFlow datasets
train_dataset = tf.data.Dataset.from_tensor_slices((X_train, y_train))
test_dataset = tf.data.Dataset.from_tensor_slices((X_test, y_test))

# Batch and shuffle the datasets
train_dataset = train_dataset.shuffle(buffer_size=1000).batch(batch_size)
test_dataset = test_dataset.batch(batch_size)

# Normalize pixel values
normalization_layer = tf.keras.layers.Rescaling(1./255)
train_dataset = train_dataset.map(lambda x, y: (normalization_layer(x), y))
test_dataset = test_dataset.map(lambda x, y: (normalization_layer(x), y))

# Load DenseNet121 with pre-trained weights (ImageNet)

base_model = DenseNet121(input_shape=(224, 224, 3), include_top=False, weights='imagenet')

# Freeze the base model layers to avoid training them
base_model.trainable = False

# Add custom classification head
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)  # Add dense layer
predictions = Dense(4, activation='softmax')(x)  # 4 classes

# Build the model
model = Model(inputs=base_model.input, outputs=predictions)

# Compile the model
model.compile(optimizer=Adam(learning_rate=0.00001),
```

```
loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Set early stopping to avoid overfitting
early_stopping = EarlyStopping(monitor='val_loss', patience=5)

# Train the model
history = model.fit(train_dataset, validation_data=test_dataset, epochs=20, callbacks=[early_stopping])

# Unfreeze some layers of the base model for fine-tuning
base_model.trainable = True

# Recompile with a lower learning rate for fine-tuning
model.compile(optimizer=Adam(learning_rate=0.000001),
loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Fine-tune the model (optional)
# history_fine = model.fit(train_dataset, validation_data=test_dataset, epochs=10, callbacks=[early_stopping])

# Evaluate the model
test_loss, test_acc = model.evaluate(test_dataset)
print(f"Test accuracy: {test_acc}")

# Step 1: Get predictions from the model on the test set
y_pred_probs = model.predict(test_dataset)
y_pred = np.argmax(y_pred_probs, axis=1)

# Step 2: Extract true labels from the test dataset
y_true = np.concatenate([y.numpy() for x, y in test_dataset], axis=0)

# Step 3: Calculate Precision, Recall, and F1 Score using scikit-learn
precision = precision_score(y_true, y_pred, average='weighted')
recall = recall_score(y_true, y_pred, average='weighted')
f1 = f1_score(y_true, y_pred, average='weighted')

print(f'Precision: {precision:.4f}')
print(f'Recall: {recall:.4f}')
print(f'F1 Score: {f1:.4f}')
```

 # ResNet50

```
import tensorflow as tf
import numpy as np
from sklearn.utils import class_weight
from tensorflow.keras.preprocessing import image_dataset_from_directory
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, BatchNormalization
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
```

```
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_score, recall_score, f1_score

# Set directory for the dataset
dataset_dir = '/content/drive/MyDrive/Bone Cancer Detection Project/segmented_images_unet'

# Image size (ResNet50 default is 224x224)
img_size = (224, 224)
batch_size = 32

# Load the entire dataset
dataset = image_dataset_from_directory(
    dataset_dir, image_size=img_size, batch_size=batch_size, shuffle=True
)

# Extract images and labels from the dataset
image_list = []
label_list = []

for images, labels in dataset:

    image_list.extend(images.numpy())
    label_list.extend(labels.numpy())

# Convert lists to numpy arrays
image_list = np.array(image_list)
label_list = np.array(label_list)

# Split the dataset into 80% training and 20% testing
X_train, X_test, y_train, y_test = train_test_split(
    image_list, label_list, test_size=0.2, random_state=42, stratify=label_list
)

# Convert the split data back to TensorFlow datasets
train_dataset = tf.data.Dataset.from_tensor_slices((X_train, y_train))
test_dataset = tf.data.Dataset.from_tensor_slices((X_test, y_test))

# Define data augmentation layers
data_augmentation = tf.keras.Sequential([
    tf.keras.layers.RandomFlip("horizontal_and_vertical"),
    tf.keras.layers.RandomRotation(0.3),  # Increased rotation
    tf.keras.layers.RandomZoom(0.3),      # Increased zoom
    tf.keras.layers.RandomContrast(0.3),  # Increased contrast
])

# Batch and shuffle the datasets
train_dataset = train_dataset.shuffle(buffer_size=1000).batch(batch_size)
```

```
test_dataset = test_dataset.batch(batch_size)

# Normalize pixel values and apply data augmentation
normalization_layer = tf.keras.layers.Rescaling(1./255)

def preprocess_data(image, label):
    image = normalization_layer(image)
    image = data_augmentation(image)  # Apply data augmentation
    return image, label

train_dataset = train_dataset.map(preprocess_data)
test_dataset = test_dataset.map(lambda x, y: (normalization_layer(x), y))

# Load ResNet50 with pre-trained weights (ImageNet)
base_model = ResNet50(input_shape=(224, 224, 3), include_top=False, weights='imagenet')

# Freeze the base model layers to avoid training them initially
base_model.trainable = False


# Add custom classification head with batch normalization
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
x = BatchNormalization()(x)  # Add Batch Normalization
predictions = Dense(4, activation='softmax')(x)  # 4 classes

# Build the model
model = Model(inputs=base_model.input, outputs=predictions)

# Compile the model
model.compile(optimizer=Adam(learning_rate=0.00001),
loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Set early stopping and learning rate reduction
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=3, min_lr=1e-7)  # More aggressive reduction


# Ensure y_train is 1D
y_train_flat = y_train.flatten()  # Flatten in case it's multi-dimensional

# Calculate class weights
class_weights = class_weight.compute_class_weight('balanced', classes=np.unique(y_train_flat), y=y_train_flat)
```

```
# Convert class_weights to a dictionary
class_weight_dict = {i: class_weights[i] for i in range(len(class_weights))}

# Train the model with class weights
history = model.fit(train_dataset, validation_data=test_dataset, epochs=20,
                    callbacks=[early_stopping, reduce_lr], class_weight=class_weight_dict)

# Increase learning rate reduction aggressive
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=2, min_lr=1e-7)

# # Unfreeze some layers of the base model for fine-tuning
# base_model.trainable = True

# # Fine-tune by unfreezing more layers
# for layer in base_model.layers[:143]:  # Only freeze the first few layers


#     layer.trainable = False

# # Recompile with a lower learning rate for fine-tuning
#                                 model.compile(optimizer=Adam(learning_rate=0.000001), loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# # Fine-tune the model
# fine_tune_history = model.fit(train_dataset, validation_data=test_dataset, epochs=20, callbacks=[early_stopping, reduce_lr])

# Evaluate the model
test_loss, test_acc = model.evaluate(test_dataset)
print(f"Test accuracy: {test_acc}")

# Step 1: Get predictions from the model on the test set
y_pred_probs = model.predict(test_dataset)
y_pred = np.argmax(y_pred_probs, axis=1) # Convert softmax probabilities to class labels

# Step 2: Extract true labels from the test dataset
y_true = np.concatenate([y.numpy() for x, y in test_dataset], axis=0)

# Step 3: Calculate Precision, Recall, and F1 Score using scikit-learn
precision = precision_score(y_true, y_pred, average='weighted')
recall = recall_score(y_true, y_pred, average='weighted')
f1 = f1_score(y_true, y_pred, average='weighted')

print(f'Precision: {precision:.4f}')
print(f'Recall: {recall:.4f}')
print(f'F1 Score: {f1:.4f}')
```

#VGG16

```
import tensorflow as tf
import numpy as np
from tensorflow.keras.preprocessing import image_dataset_from_directory
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_score, recall_score, f1_score

# Set directory for the dataset
dataset_dir = '/content/drive/MyDrive/Bone Cancer Detection Project/segmented_images_unet'

# Image size (VGG16 default is 224x224)
img_size = (224, 224)
batch_size = 32

# Load the entire dataset
dataset = image_dataset_from_directory(
    dataset_dir, image_size=img_size, batch_size=batch_size, shuffle=True
)

# Extract images and labels from the dataset
image_list = []
label_list = []

for images, labels in dataset:
    image_list.extend(images.numpy())
    label_list.extend(labels.numpy())

# Convert lists to numpy arrays
image_list = np.array(image_list)
label_list = np.array(label_list)

# Split the dataset into 80% training and 20% testing
X_train, X_test, y_train, y_test = train_test_split(
    image_list, label_list, test_size=0.2, random_state=42, stratify=label_list
)

# Convert the split data back to TensorFlow datasets
train_dataset = tf.data.Dataset.from_tensor_slices((X_train, y_train))
test_dataset = tf.data.Dataset.from_tensor_slices((X_test, y_test))

# Batch and shuffle the datasets
train_dataset = train_dataset.shuffle(buffer_size=1000).batch(batch_size)
test_dataset = test_dataset.batch(batch_size)
```

```
# Normalize pixel values
normalization_layer = tf.keras.layers.Rescaling(1./255)
train_dataset = train_dataset.map(lambda x, y: (normalization_layer(x), y))
test_dataset = test_dataset.map(lambda x, y: (normalization_layer(x), y))

# Load VGG16 with pre-trained weights (ImageNet)
base_model = VGG16(input_shape=(224, 224, 3), include_top=False, weights='imagenet')

# Freeze the base model layers to avoid training them
base_model.trainable = False

# Add custom classification head
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)  # Add dense layer
predictions = Dense(4, activation='softmax')(x)  # 4 classes

# Build the model
model = Model(inputs=base_model.input, outputs=predictions)

# Compile the model
model.compile(optimizer=Adam(learning_rate=0.00001),
loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Set early stopping to avoid overfitting
early_stopping = EarlyStopping(monitor='val_loss', patience=5)

# Train the model
history = model.fit(train_dataset, validation_data=test_dataset, epochs=20,
callbacks=[early_stopping])

# Unfreeze some layers of the base model for fine-tuning
base_model.trainable = True

# Recompile with a lower learning rate for fine-tuning
model.compile(optimizer=Adam(learning_rate=0.000001),
loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Fine-tune the model (optional)
# history_fine = model.fit(train_dataset, validation_data=test_dataset, epochs=10,
callbacks=[early_stopping])

# Evaluate the model
test_loss, test_acc = model.evaluate(test_dataset)
print(f"Test accuracy: {test_acc}")

# Step 1: Get predictions from the model on the test set
y_pred_probs = model.predict(test_dataset)
```

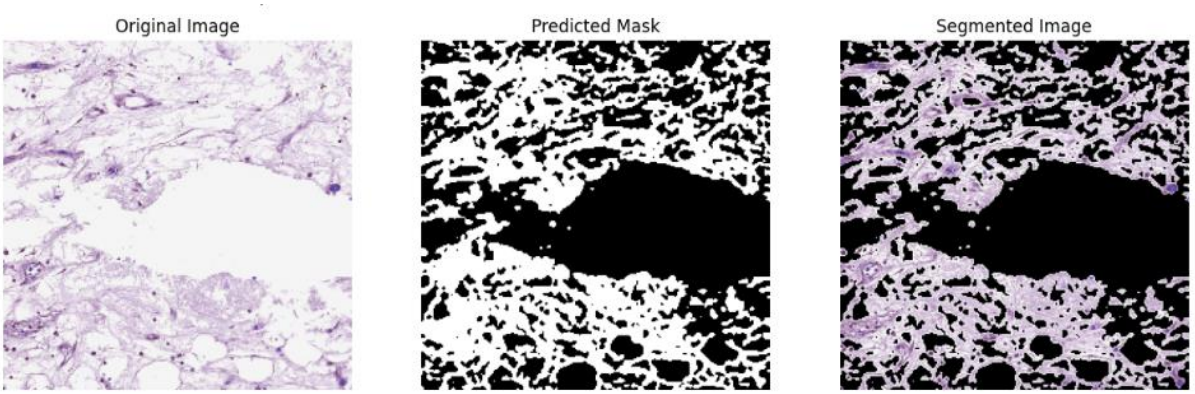
```
y_pred = np.argmax(y_pred_probs, axis=1)  # Convert softmax probabilities to class labels

# Step 2: Extract true labels from the test dataset
y_true = np.concatenate([y.numpy() for x, y in test_dataset], axis=0)
# Step 3: Calculate Precision, Recall, and F1 Score using scikit-learn
precision = precision_score(y_true, y_pred, average='weighted')
recall = recall_score(y_true, y_pred, average='weighted')
f1 = f1_score(y_true, y_pred, average='weighted')

print(f'Precision: {precision:.4f}')
print(f'Recall: {recall:.4f}')
print(f'F1 Score: {f1:.4f}')
```

#MobileNetV2

```
import tensorflow as tf
import numpy as np
from tensorflow.keras.preprocessing import image_dataset_from_directory
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_score, recall_score, f1_score
# Set directory for the dataset
dataset_dir = '/content/drive/MyDrive/segmented_images_unet'
# Image size (MobileNetV2 default is 224x224)
img_size = (224, 224)
batch_size = 32
# Load the entire dataset (without validation split, we will split manually)
dataset = image_dataset_from_directory(
    dataset_dir, image_size=img_size, batch_size=batch_size, shuffle=True
)
# Extract images and labels from the dataset
image_list = []
label_list = []

for images, labels in dataset:
    image_list.extend(images.numpy())
    label_list.extend(labels.numpy())

# Convert lists to numpy arrays
image_list = np.array(image_list)
label_list = np.array(label_list)
```

```
# Split the dataset into 80% training and 20% testing
X_train, X_test, y_train, y_test = train_test_split(
    image_list, label_list, test_size=0.2, random_state=42, stratify=label_list
)

# Convert the split data back to TensorFlow datasets
train_dataset = tf.data.Dataset.from_tensor_slices((X_train, y_train))
test_dataset = tf.data.Dataset.from_tensor_slices((X_test, y_test))

# Batch and shuffle the datasets
train_dataset = train_dataset.shuffle(buffer_size=1000).batch(batch_size)
test_dataset = test_dataset.batch(batch_size)

# Normalize pixel values
normalization_layer = tf.keras.layers.Rescaling(1./255)
train_dataset = train_dataset.map(lambda x, y: (normalization_layer(x), y))
test_dataset = test_dataset.map(lambda x, y: (normalization_layer(x), y))

# Load MobileNetV2 with pre-trained weights
base_model = MobileNetV2(input_shape=(224, 224, 3), include_top=False, weights='imagenet')

# Freeze the base model layers to avoid training them
base_model.trainable = False

# Add custom classification head
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)  # Add dense layer
predictions = Dense(4, activation='softmax')(x)  # 4 classes

# Build the model
model = Model(inputs=base_model.input, outputs=predictions)

# Compile the model
model.compile(optimizer=Adam(learning_rate=0.00001),
loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Set early stopping to avoid overfitting
early_stopping = EarlyStopping(monitor='val_loss', patience=5)

# Train the model
history = model.fit(train_dataset, validation_data=test_dataset, epochs=20, callbacks=[early_stopping])
```

```
# Unfreeze some layers of the base model for fine-tuning
base_model.trainable = True

# Recompile with a lower learning rate for fine-tuning
model.compile(optimizer=Adam(learning_rate=0.000001),
loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Fine-tune the model (optional)
# history_fine = model.fit(train_dataset, validation_data=test_dataset, epochs=10,
callbacks=[early_stopping])

# Evaluate the model
test_loss, test_acc = model.evaluate(test_dataset)
print(f"Test accuracy: {test_acc}")

# Step 1: Get predictions from the model on the test set
y_pred_probs = model.predict(test_dataset)
y_pred = np.argmax(y_pred_probs, axis=1) # Convert softmax probabilities to class
labels

# Step 2: Extract true labels from the test dataset
y_true = np.concatenate([y.numpy() for x, y in test_dataset], axis=0)

# Step 3: Calculate Precision, Recall, and F1 Score using scikit-learn
precision = precision_score(y_true, y_pred, average='weighted')
recall = recall_score(y_true, y_pred, average='weighted')
f1 = f1_score(y_true, y_pred, average='weighted')

print(f'Precision: {precision:.4f}')
print(f'Recall: {recall:.4f}')
print(f'F1 Score: {f1:.4f}')
```



REFERENCES

[1] Anaya-Isaza, L. Mera-Jiménez and A. Fernandez-Quilez, "CrossTransUnet: A New Computationally Inexpensive Tumor Segmentation Model for Brain MRI," in IEEE Access, vol. 11, pp. 27066-27085, 2023.

[2] Saber, M. Sakr, O. M. Abo-Seida, A. Keshk and H. Chen, "A Novel Deep-Learning Model for Automatic Detection and Classification of Breast Cancer Using the Transfer-Learning Technique," in IEEE Access, vol. 9, pp. 71194-71209, 2021.

[3] Tabbakh and S. S. Barpanda, "A Deep Features Extraction Model Based on the Transfer Learning Model and Vision Transformer “TLMViT” for Plant Disease Classification," in IEEE Access, vol. 11, pp. 45377-45392, 2023.

[4] Loraksa, S. Mongkolsomlit, N. Nimsuk, M. Uscharapong and P. Kiatisevi, "Development of the Osteosarcoma Lung Nodules Detection Model Based on SSD-VGG16 and Competency Comparing With Traditional Method," in IEEE Access, vol. 10, pp. 65496-65506, 2022.

[5] Sun, A. Xu, D. Liu, Z. Xiong, F. Zhao and W. Ding, "Deep Learning-Based Classification of Liver Cancer Histopathology Images Using Only Global Labels," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 6, pp. 1643-1651, June 2020.

[6] Kumar et al., "Automatic Detection of White Blood Cancer From Bone Marrow Microscopic Images Using Convolutional Neural Networks," in IEEE Access, vol. 8, pp. 142521-142531, 2020.

[7] Alabdulkreem, M. K. Saeed, S. S. Alotaibi, R. Allafi, A. Mohamed and M. A. Hamza, "Bone Cancer Detection and Classification Using Owl Search Algorithm With Deep Learning on X-Ray Images," in IEEE Access, vol. 11, pp. 109095-109103, 2023.

[8] H. N. Dao, T. Nguyen, C. Mugisha and I. Paik, "A Multimodal Transfer Learning Approach Using PubMedCLIP for Medical Image Classification," in IEEE Access, vol. 12, pp. 75496-75507, 2024.

[9] J. Hu et al., "S-UNet: A Bridge-Style U-Net Framework With a Saliency Mechanism for Retinal Vessel Segmentation," in IEEE Access, vol. 7, pp. 174167-174177, 2019.

[10] J. Wu, T. Yuan, J. Zeng and F. Gou, "A Medically Assisted Model for Precise Segmentation of Osteosarcoma Nuclei on Pathological Images," in IEEE Journal of Biomedical and Health Informatics, vol. 27, no. 8, pp. 3982-3993, Aug. 2023.

[11] K. D. Shah, D. K. Patel, M. P. Thaker, H. A. Patel, M. J. Saikia and B. J. Ranger, "EMED-UNet: An Efficient Multi-Encoder-Decoder Based UNet for Medical Image Segmentation," in IEEE Access, vol. 11, pp. 95253-95266, 2023.
[12] K. Neamah, F. Mohamed, S. R. Waheed, W. H. M. Kurdi, A. Y. Taha and K. A. Kadhim, "Utilizing Deep improved ResNet50 for Brain Tumor Classification Based MRI," in IEEE Open Journal of the Computer Society, vol. 5, pp. 446-456, 2024.
[13] M. A. A. Walid and P. C. Shill, "A Transfer-Learning Based Unbiased Voting Bone Cancer Detection Framework from Histological Osteosarcoma Images," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-7.
[14] N. Siddique, S. Paheding, C. P. Elkin and V. Devabhaktuni, "U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications," in IEEE Access, vol. 9, pp. 82031-82057, 2021.
[15] R. Zaitoon and H. Syed, "RU-Net2+: A Deep Learning Algorithm for Accurate Brain Tumor Segmentation and Survival Rate Prediction," in IEEE Access, vol. 11, pp. 118105-118123, 2023.
[16] S. Alsubai et al., "Group Teaching Optimization With Deep Learning-Driven Osteosarcoma Detection Using Histopathological Images," in IEEE Access, vol. 12, pp. 34089-34098, 2024.
[17] V. A. Ashwath, O. K. Sikha and R. Benitez, "TS-CNN: A Three-Tier Self-Interpretable CNN for Multi-Region Medical Image Classification," in IEEE Access, vol. 11, pp. 78402-78418, 2023.
[18] X. Fu, L. Bi, A. Kumar, M. Fulham and J. Kim, "Multimodal Spatial Attention Module for Targeting Multimodal PET-CT Lung Tumor Segmentation," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 9, pp. 3507-3516, Sept. 2021.
[19] Y. Weng, T. Zhou, Y. Li and X. Qiu, "NAS-Unet: Neural Architecture Search for Medical Image Segmentation," in IEEE Access, vol. 7, pp. 44247-44257, 2019.
[20] Z. Zeng, W. Xie, Y. Zhang and Y. Lu, "RIC-Unet: An Improved Neural Network Based on Unet for Nuclei Segmentation in Histology Images," in IEEE Access, vol. 7, pp. 21420-21428, 2019.