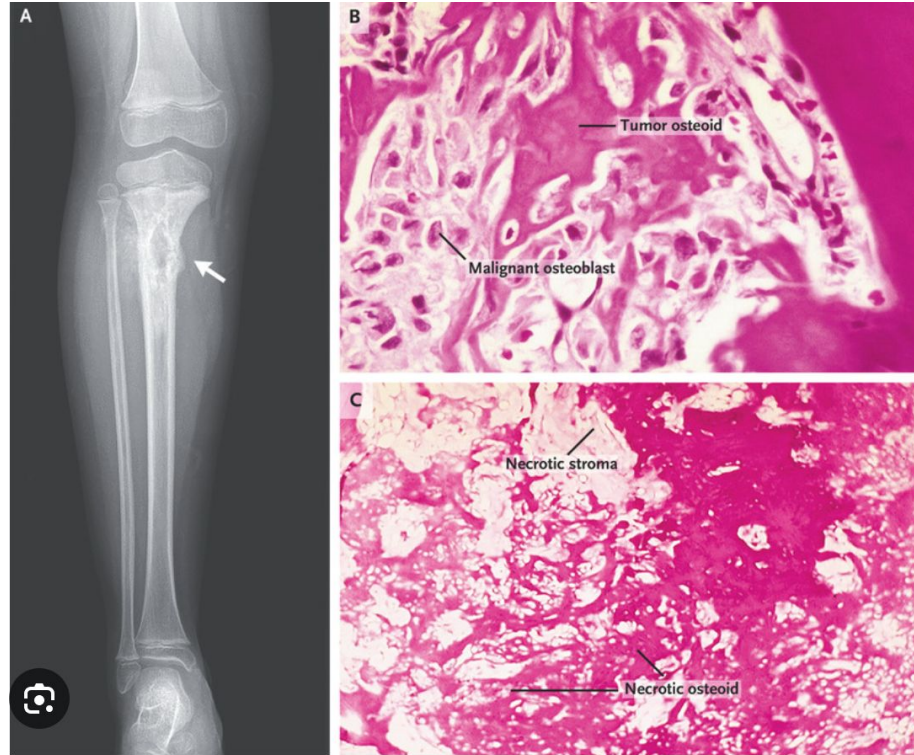


Bone Cancer Detection (Osteosarcoma) using Image Segmentation

- Charan Reddy Jakkireddy
- Madhu Kiran Bharatha
- Sai Ketan Gande



Abstract

- Bone cancer, particularly osteosarcoma, is a serious health concern requiring early and accurate detection.
- Deep learning models offer promising results in medical image analysis tasks.
- This project compares the effectiveness of various models for bone cancer detection: SVC, CNN, Mobile Net, and U-Net.
- Raw image data obtained from various medical imaging techniques is utilized.
- Preprocessing techniques, including resizing, normalization, and augmentation, enhance the dataset.
- The dataset is divided into training and testing sets for evaluation.

Introduction

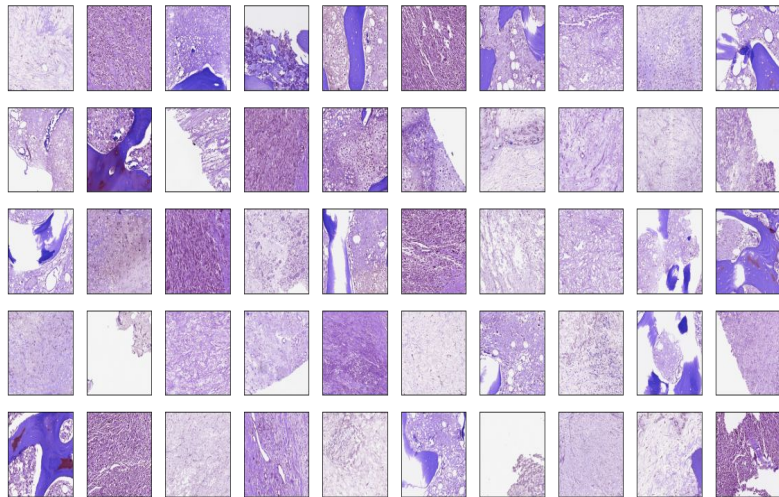
- Osteosarcoma is the most common form of bone cancer, mainly affecting children and young adults. It is a malignant tumor that develops in the cells responsible for bone formation, known as osteoblasts.
- Early detection is vital for improving the prognosis and survival rates of individuals with osteosarcoma. Medical imaging techniques, such as X-rays, CT scans, and MRI, are commonly employed for diagnosis and monitoring the progression of the disease.
- Image segmentation is a computer vision technique that involves dividing an image into distinct regions or objects to extract meaningful information. In the case of osteosarcoma, image segmentation can aid in identifying and outlining tumor regions within medical images, facilitating accurate diagnosis and treatment planning.
- The utilization of image segmentation in bone cancer detection offers several advantages. It enables precise measurement of tumor size, assessment of tumor characteristics, and monitoring changes over time. Additionally, it provides valuable insights into the tumor's location and its relationship with surrounding tissues.

Data Set Description

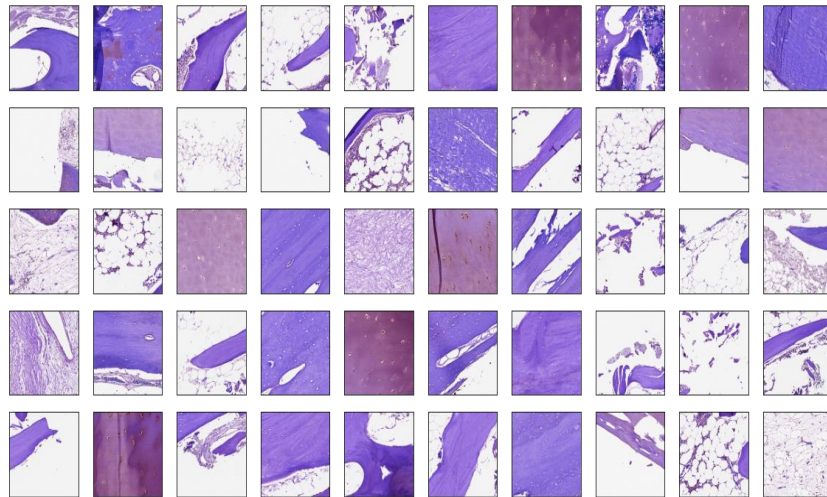
- Data Collection: The data was collected by a team of clinical scientists at the University of Texas Southwestern Medical Center, Dallas.
- Patient Selection: Archival samples from 50 patients treated at Children's Medical Center, Dallas, between 1995 and 2015, were used to create this dataset.
- Labeling: The images are labeled as Non-Tumor, Viable Tumor, and non-viable.
- Dataset Size: The dataset consists of 1144 images of size 1024 X 1024 at 10X resolution.
- Class Distribution: The dataset comprises 536 (47%) non-tumor images, 263 (23%) non-viable images, and 345 (30%) viable tumor images.

Sample Image data

Tumor: No

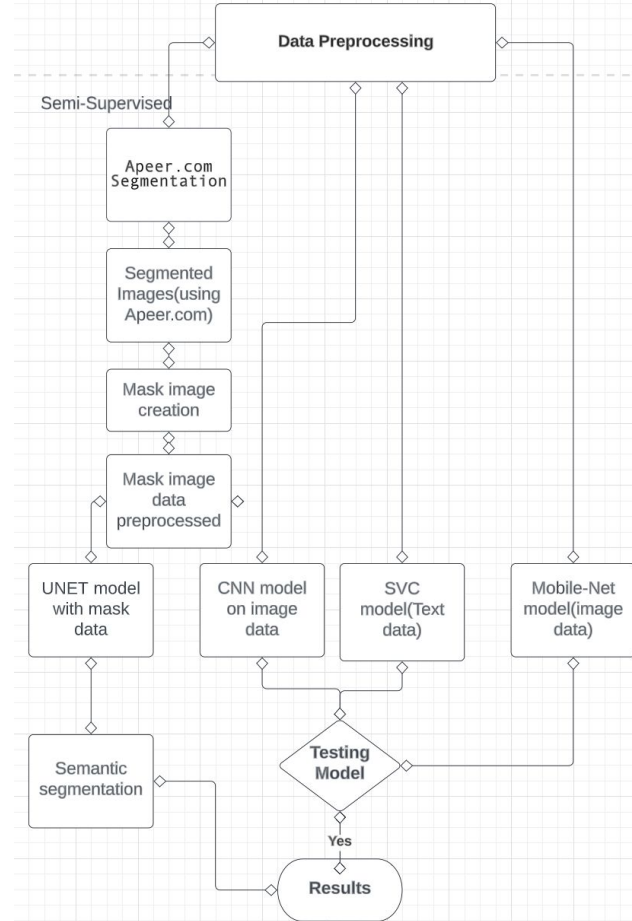


Tumor: Yes



Methodology

1. Data Preprocessing involving image manipulation segregation of images according to the images having tumor, non-tumor as folders and moving images .
2. Segment images using Apeer.io and storing them in local disk
3. Performed basic CNN model on Raw data gathered from the official cancer image data archives.
4. Performed SVM model on image data where the images are present in folder with folder names as labels.
5. Using Apeer.com created mask images
6. Performed CNN with mask image data also as input.
7. Performed RCNN with image data and mask image data as inputs .



Data Pre-processing

- The original image data was divided into training set-1 and training set-2 which had 10 sets of folders with image data and a corresponding CSV file where image data labels are provided in the CSV file.
- We have segregated images into train and test folders and converted them into grey scale , reduced size and normalized them and also made a data frame of all the csv files from the folders which helped to segregate the image data into folders according to their type(viable , non-viable etc.)
- Image segmentation using Apeer.io tool.



Data Pre-processing contd.

- Apeer.com is a platform that facilitates image segmentation, a crucial step in cancer detection.
- Image Segmentation: Image segmentation is the process of dividing an image into distinct regions or objects. In the context of cancer detection, it involves identifying and delineating tumor regions within medical images.
 - Semantic segmentation is detecting objects of one category.
 - Instance segmentation is detecting objects of multiple categories.
- Apeer.com: Apeer.com is an online platform that offers tools and functionalities for image analysis and segmentation. It provides a user-friendly interface and a range of pre-built algorithms specifically designed for medical image processing.
- Segmentation Algorithms: Apeer provides a collection of state-of-the-art segmentation algorithms that can be applied to images. Apeer uses Unet for segmentation.

Evolution of Classification Algorithms for Image Data

- Classification algorithms are fundamental tools in machine learning for categorizing data into predefined classes or categories.
- SVM is a supervised learning algorithm that analyzes data and finds an optimal hyperplane to separate different classes.
- SVC is a variant of SVM designed specifically for classification tasks.
- But, Image data presents unique challenges for traditional classification algorithms due to its high dimensionality and complex structures.
- Images contain rich visual information such as textures, colors, and spatial relationships that traditional algorithms may struggle to capture effectively.
- Deep learning algorithms, particularly Convolutional Neural Networks (CNNs), revolutionized image classification.
- CNNs excel at learning hierarchical representations from raw image data, capturing intricate patterns and achieving state-of-the-art performance.

Support Vector Classifier(SVC) Implementation

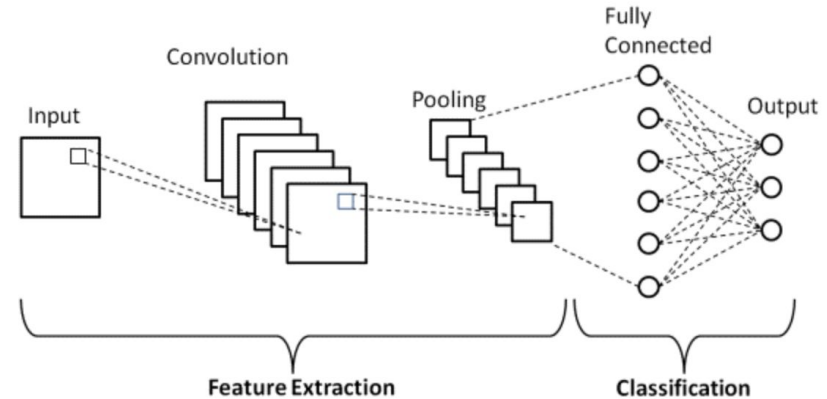
- **Objective:** To explore the classification performance of image data using a traditional classification algorithm.
- **Dataset Overview:** The dataset consisted of CSV files containing image names, features, and labels.
- **Approach:** Implemented a Support Vector Classifier (SVC) algorithm using the image data and the corresponding dataframe.
- **Performance Evaluation:** The model achieved an accuracy of 42% in classifying the images.
- **Purpose:** The implementation of SVC aimed to understand the behavior of image data with conventional classification algorithms.

Convolution Neural Networks(CNNs)

- CNNs (Convolutional Neural Networks) are deep learning models designed for visual data, particularly images.
- They excel in computer vision tasks:
 - Image Classification: Assigning labels to images.
 - Object Detection: Identifying and localizing objects.
 - Image Segmentation: Labeling each pixel in an image.
- CNNs automatically learn hierarchical features from raw pixel data.
- They don't rely on handcrafted features, learning relevant features during training.
- CNNs capture spatial dependencies and patterns using local receptive fields and weight sharing.
- Through convolutions, pooling, and non-linear activations, CNNs learn abstract representations.
- This enables recognition of complex visual patterns, textures, edges, and shapes.
- CNNs have transformed computer vision in applications like autonomous driving, medical imaging, face recognition, and augmented reality.
- They are indispensable for processing and understanding visual data.

CNN Architecture

- Convolutional Layers:
- Pooling Layers:
- Flatten Layer
- Fully Connected Layers:
- Activation Functions:



CNN Implementation

- The Image Data Generator class from Keras is used to generate batches of image data with real-time data augmentation.
- Data generators are beneficial when working with large datasets, as they allow us to load and preprocess images in batches instead of loading all the data into memory at once.

```
In [10]: from keras.preprocessing.image import ImageDataGenerator

train_dir="/Users/bharathamadhukiran/Desktop/capstoneimages/edu/train"
test_dir="/Users/bharathamadhukiran/Desktop/capstoneimages/edu/test"

train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical')

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical')
```

Found 533 images belonging to 3 classes.

Found 556 images belonging to 3 classes.

CNN implementation contd.

- The model architecture is designed to progressively extract features from the input images using convolutional and pooling layers. The fully connected layers learn complex relationships between features, leading to the final classification of images into one of the three classes.
- The model is trained using the data generators created earlier, allowing for efficient loading and preprocessing of image data in batches.
- The training process aims to optimize the model's parameters (weights and biases) using the Adam optimizer and minimize the categorical cross-entropy loss.
- The history variable captures the training metrics (loss and accuracy) for analysis and evaluation.

```
model = Sequential()

model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(3, activation='softmax'))

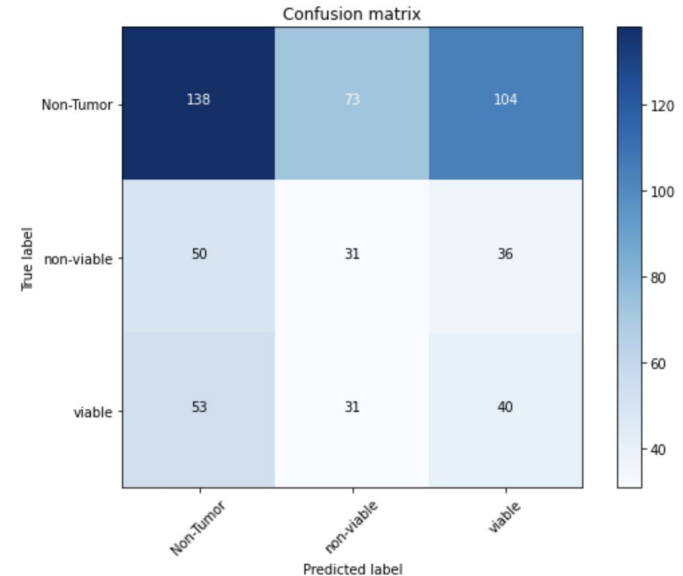
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```


Model performance

- Classification Report

	precision	recall	f1-score	support
Non-Tumor	0.53	0.40	0.46	315
non-viable	0.23	0.26	0.25	117
viable	0.21	0.31	0.25	124
accuracy			0.35	556
macro avg	0.32	0.32	0.32	556
weighted avg	0.39	0.35	0.37	556

- Confusion Matrix



MobileNetV2

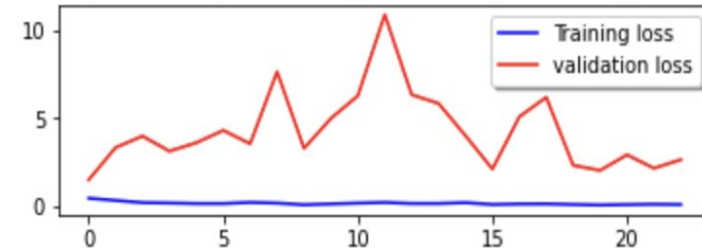
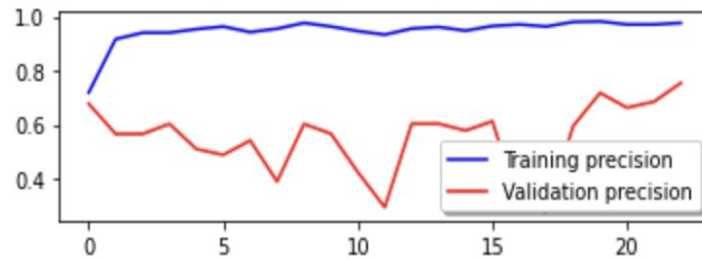
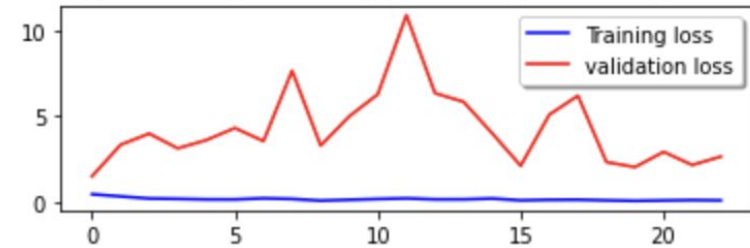
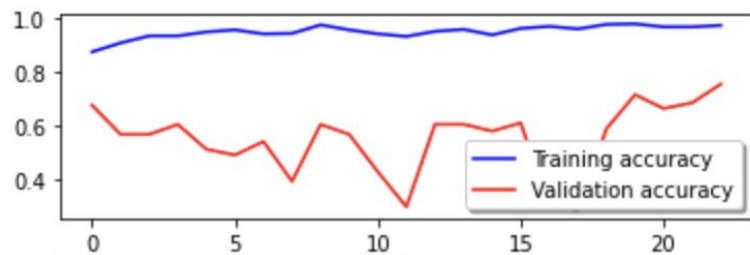
Mobile Net is a convolutional neural network that employs a type of convolution known as depth wise separable convolution. Convolution of this form separates spatial and channel-wise operations, making it more efficient.

The Mobile Net architecture contains 17 bottleneck levels, each with a variable number of output channels and strides.

Each bottleneck layer is made up of three parts: Expansion Block: By increasing the number of input channels, the network is able to capture more diverse information from the input data.

Depth wise Block: A depth wise convolution is used with a specific stride in this block. This convolution operation is primarily concerned with capturing spatial relationships between input data, such as edges and textures.

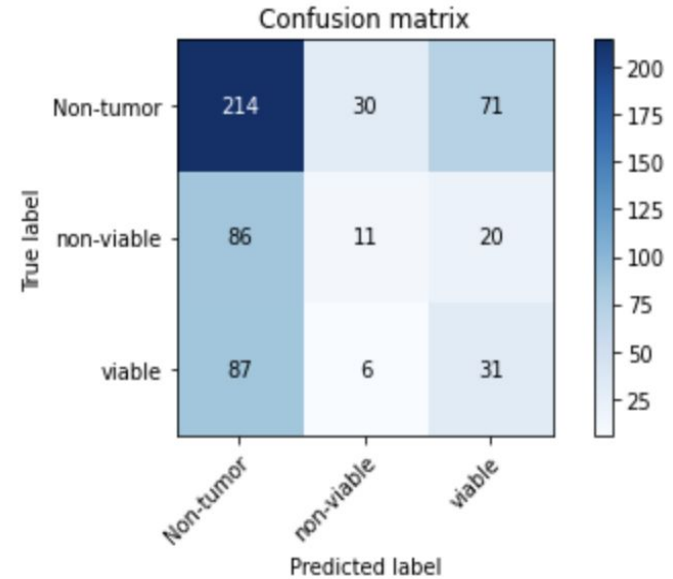
Model performance



Model performance contd.

- Test accuracy for this model is

```
28/28 [=====]
l1_1: 0.7554
Test loss: 2.6363558769226074
Test accuracy: 0.7553957104682922
```



Binary Classification for Tumor Detection

Problem Background:

- Initial attempt: Multiclass classification using CNNs for tumor detection.
- Challenge: CNNs were not performing well in the multiclass setting.

Solution: Converted the problem into a binary classification task.

- Categorized images into two classes:
 - a) Tumor: Combined non-viable and viable tumor images.
 - b) No Tumor: All remaining images without tumor.

Model Training:

- Trained a new CNN model tailored for binary classification.
- Modified architecture with two output neurons representing Tumor and No Tumor classes.
- Focused on optimizing the model to accurately distinguish tumor and non-tumor images.

CNN for binary classification

Input: Shape of the input image (image_width, image_height, #_channels)

Output: Model object representing the bone detection model

Architecture Overview:

- Input placeholder
- Zero-Padding
- Convolutional Block
- Max Pooling
- Flatten
- Fully Connected Layer

Model Summary:

- Convolutional and pooling layers extract features
- Flattening transforms features into a 1D vector
- Fully connected layer performs binary classification
- Sigmoid activation for probability output

```
# Define the input placeholder as a tensor with shape input_shape.
X_input = Input(input_shape) # shape=(?, 240, 240, 3)

# Zero-Padding: pads the border of X_input with zeroes
X = ZeroPadding2D((2, 2))(X_input) # shape=(?, 244, 244, 3)

# CONV -> BN -> RELU Block applied to X
X = Conv2D(32, (7, 7), strides = (1, 1), name = 'conv0')(X)
X = BatchNormalization(axis = 3, name = 'bn0')(X)
X = Activation('relu')(X) # shape=(?, 238, 238, 32)

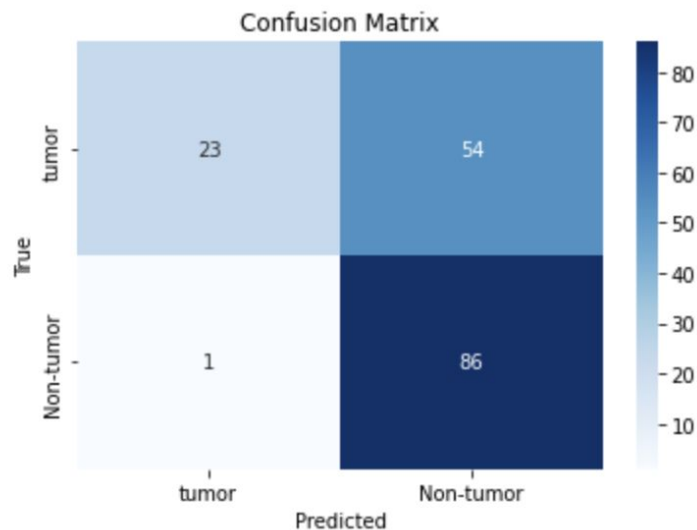
# MAXPOOL
X = MaxPooling2D((4, 4), name='max_pool0')(X) # shape=(?, 59, 59, 32)

# MAXPOOL
X = MaxPooling2D((4, 4), name='max_pool1')(X) # shape=(?, 14, 14, 32)

# FLATTEN X
X = Flatten()(X) # shape=(?, 6272)
# FULLYCONNECTED
X = Dense(1, activation='sigmoid', name='fc')(X) # shape=(?, 1)

# Create model. This creates your Keras model instance, you'll use this instance to train/test the model.
model = Model(inputs = X_input, outputs = X, name='BoneDetectionModel')
```


Model Performance



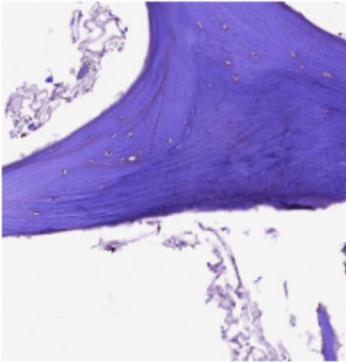
	precision	recall	f1-score	support
0	0.75	0.66	0.70	79
1	0.72	0.80	0.76	85
accuracy			0.73	164
macro avg	0.73	0.73	0.73	164
weighted avg	0.73	0.73	0.73	164

Results

```
In [240]: # Assuming 'image' is your input image and 'best_model' is
path = '/Users/bharathamadhukiran/Downloads/binary/images',
prediction_prob, prediction_class = predict_image(path)
```

1/1 [=====] - 0s 34ms/step

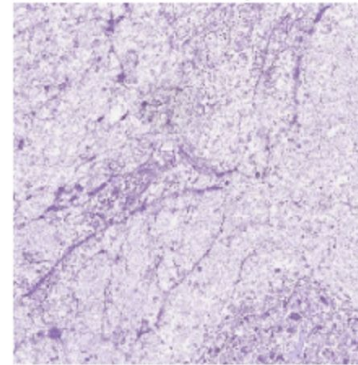
Predicted Class: Non-Tumor, Probability of No Tumor: 0.9857



```
In [244]: # Assuming 'image' is your input image and 'best_model'
path = '/Users/bharathamadhukiran/Downloads/binary/image',
prediction_prob, prediction_class = predict_image(path)
```

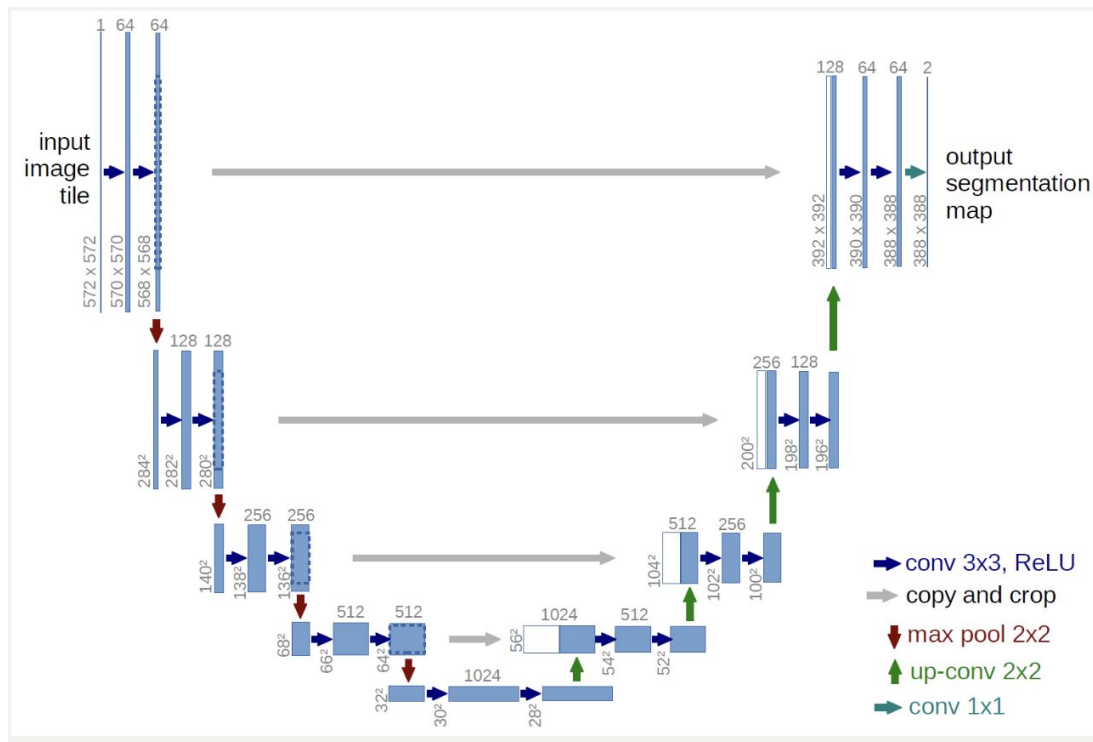
1/1 [=====] - 0s 31ms/step

Predicted Class: Tumor, Probability of No Tumor: 0.3861



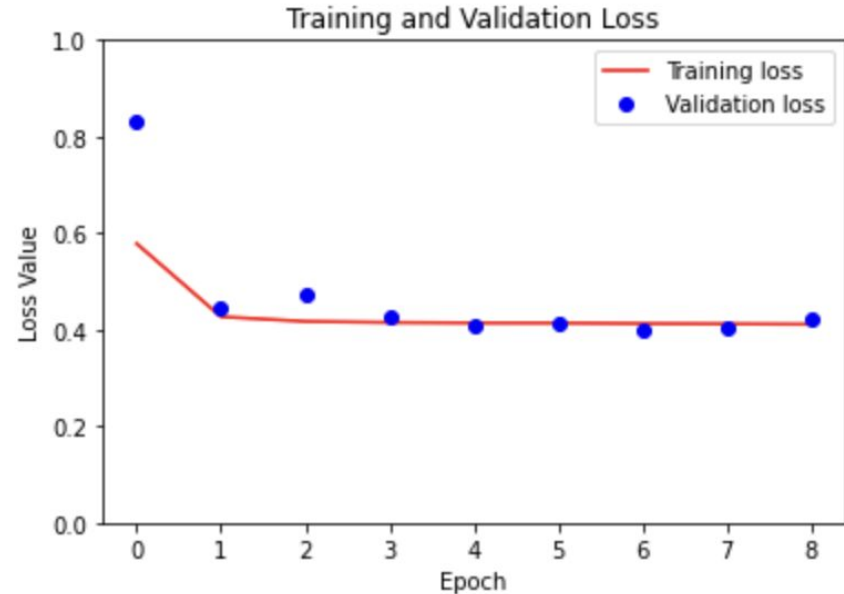
U-Net Model

- U-Net is a popular architecture for image segmentation tasks, particularly in biomedical image analysis.
- It consists of an encoder-decoder structure with skip connections. The encoder part captures contextual information and extracts features, while the decoder part generates a high-resolution segmentation map.
- Skip connections enable the fusion of low-level and high-level features, aiding in precise segmentation.
- U-Net has been widely used and achieved state-of-the-art performance in various image segmentation challenges.



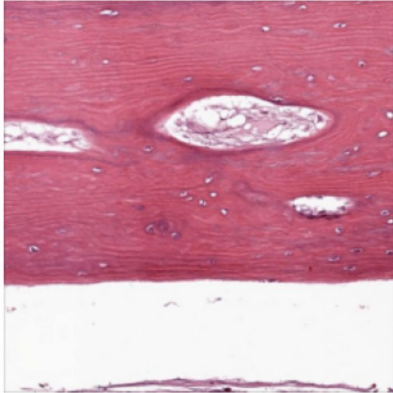
Model Performance

- During the training process, the model's performance improved as the loss decreased and accuracy increased.
- The validation loss and accuracy also showed positive trends. The model achieved a training accuracy of around 93-94% and a validation accuracy above 90%. The validation loss consistently decreased, indicating the model's ability to generalize well.
- Overall, the training and validation metrics demonstrated the model's learning progress and its capability to make accurate predictions.



Segmented image

Input image



True mask



Predicted mask



CONCLUSION

- In conclusion, CNN as expected has performed well in the binary classification. For multi-class classification the results would have been better with more data.
- The Unet model has demonstrated good performance for the segmentation task based on the masks provided to it.
- The Unet model's advanced architecture, incorporating the concept of skip connections and encoder-decoder structures, has allowed it to excel in tasks such as image segmentation and medical imaging. Its ability to capture fine-grained details and accurately identify boundaries and features has contributed to its good performance.

Future Scope

- Continuously refining and optimizing the image classification model for Osteosarcoma can lead to improved accuracy and reliability. Exploring different architectures, such as deep convolutional neural networks (CNNs) or advanced models like ResNet or Inception, can help capture more intricate patterns and features from medical images, resulting in better prediction outcomes.
- Exploring the use of image classification models for early detection and prognosis of Osteosarcoma can have significant clinical implications. Developing models that can predict disease progression, treatment response, or overall patient survival can assist in personalized treatment planning and monitoring of the disease.

References

- New Horizons in the Treatment of Osteosarcoma (Paul S. Meltzer,2021).
- (Khobragade, 2022). Revaluating Pretraining in Small Size Training Sample Regime.
- https://www.academia.edu/88103650/Revaluating_Pretraining_in_Small_Size_Training_Sample_Regime
- Bone Cancer Detection from X-Ray and MRI Images through Image Segmentation Techniques
- <https://www.ijrte.org/wp-content/uploads/papers/v8i6/F7159038620.pdf>
- Bone Cancer Detection Using Feature Extraction Based Machine Learning Model
- <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8712164/>
- Deep Learning for Classification of Bone Lesions on Routine MRI
- <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8190437/>
- AX-U-net: A Deep Learning Framework for Image Segmentation to Assist Pancreatic Tumor Diagnosis
- <https://www.frontiersin.org/articles/10.3389/fonc.2022.894970/full>