**BONE CANCER DETECTION USING**

**IMAGE SEGMENTATION**

**By**

Charan Reddy Jakkireddy

Madhu Kiran Bharatha

Sai Ketan Gande

Department of Data Science, University of Maryland, Baltimore County

DATA 606: Capstone in Data Science

Dr. Muhammad Ali Yousuf

May 20,2022

**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, MobileNetV2, 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.

**Evolution of Classification Algorithms for Image Data**

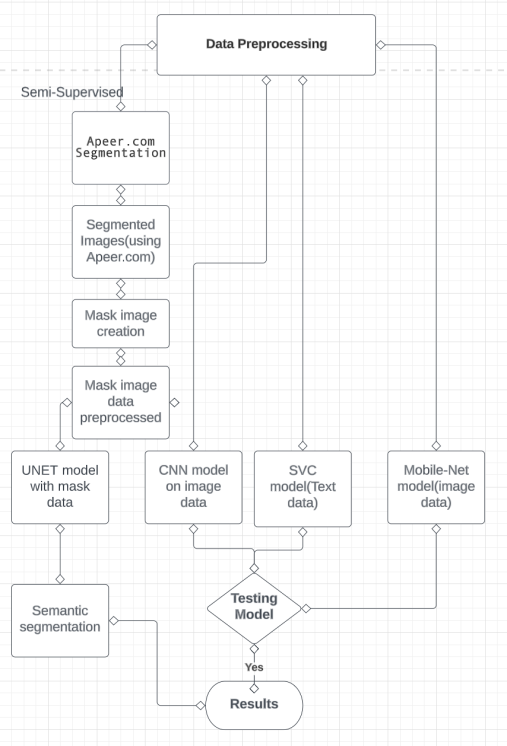
Classification algorithms are fundamental tools in machine learning for categorizing data into predefined classes or categories (Bishop, 2006). Support Vector Machines (SVM) are popular supervised learning algorithms that find an optimal hyperplane to separate different classes within the data (Cortes & Vapnik, 1995). Convolutional Neural Networks (CNNs) have revolutionized image classification by effectively capturing intricate patterns in raw image data (Krizhevsky et al., 2012). CNNs excel at learning hierarchical representations from images, leveraging convolutional layers and pooling operations (Goodfellow et al., 2016). Image data presents unique challenges for traditional classification algorithms due to its high dimensionality and complex structures (Bengio et al., 2013). However, CNNs have demonstrated state-of-the-art performance in image classification tasks by effectively capturing rich visual information in images (LeCun et al., 2015).

**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.

**Methodology**



*Figure 1: Flowchart illustrating the project's steps*

The project involved data preprocessing and manipulation of medical images related to tumors. The first step was to segregate the images into two folders: one for tumor images and another for non-tumor images. This categorization was essential for the subsequent analysis. The images were then processed using Apeer.io, a platform for image analysis, to segment them and extract relevant features. The segmented images were stored in the local disk for further use.

To build a classification model, a Convolutional Neural Network (CNN) was implemented using the raw data obtained from official cancer image data archives. This model aimed to learn patterns and features from the images to distinguish between tumor and non-tumor samples.

Next, a Support Vector Machine (SVM) model was trained using the image data. The images were organized in folders, with each folder representing a specific label. The SVM model utilized these folder labels to classify the images based on their characteristics.

To enhance the analysis, mask images were created using Apeer.com. The mask images provided additional information about the regions of interest in the original images. A CNN model was then implemented, dividing the data into two classes using the image data and the corresponding mask images as inputs. This approach aimed to leverage the combined information from both the original and mask images to improve the classification performance.

Lastly, a UNET model was implemented using the image data and mask image data as inputs. The UNET architecture is specifically designed for image segmentation tasks and has shown promising results in medical image analysis. The UNET model aimed to identify and segment specific regions of interest in the images based on the information provided by both the original images and the corresponding mask images.

In summary, the project involved preprocessing and organizing the medical image data, applying CNN and SVM models for classification, creating mask images for enhanced analysis, and implementing CNN and UNET models for image segmentation. These steps aimed to improve the understanding and detection of tumors in medical images for potential medical applications.

**Data Preprocessing**

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 is done using Apeer.com tool.

**Apeer.com**

* 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.

**Methods**

**Support Vector Classifier (SVC Implementation):**

Support Vector Classifier (SVC) implementation architecture adheres to the concepts of Support Vector Machines (SVM) for classification tasks. SVC seeks an ideal hyperplane in the data space that separates various classes.

The following major components are often included in the SVC architecture:

SVC starts by extracting useful features from the input data, which might be numerical or obtained from other approaches like image processing. SVC employs a kernel function to transform the input data into a higher-dimensional feature space in which the classes can be efficiently separated by a hyperplane. SVC identifies a subset of training data points known as support vectors, which are important for determining the decision boundary or hyperplane.

SVC seeks to maximize the margin, which indicates the distance between the decision boundary and the support vectors of various classes. In general, a greater margin leads to higher generalization and classification performance.

Classification: Once the best hyperplane has been found, SVC categorizes fresh data points based on their position relative to the decision boundary.

**Convolutional Neural Networks (CNN):**

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.



*Figure 2: Architecture of Convolution Neural Networks (CNN).*

The model architecture is designed to effectively extract features from input images using a combination of convolutional and pooling layers. These layers progressively analyze the images, capturing relevant patterns and spatial relationships. The extracted features are then passed through fully connected layers, which learn complex relationships between the features. This hierarchical feature extraction and learning process enables the model to make accurate classifications of images into one of the three predefined classes.

To facilitate the efficient handling of the image data, data generators were created. These generators enable batch-wise loading and preprocessing of the image data during training. This approach not only ensures that the model receives a constant flow of training data but also optimizes memory usage.

During training, the model's parameters, including weights and biases, are iteratively updated to minimize the categorical cross-entropy loss. The Adam optimizer, a popular optimization algorithm, is utilized to efficiently adjust these parameters based on the gradients computed during the backpropagation process.

Throughout the training process, the model's performance is monitored and evaluated using training metrics such as loss and accuracy. These metrics are captured in the history variable, which provides insights into the model's learning progress and can be further analyzed to assess the model's performance and make informed decisions.

**MobileNetV2**

MobileNetV2 is an advanced convolutional neural network (CNN) architecture developed specifically for efficient and accurate image classification tasks (Sandler et al., 2018). It addresses the need for lightweight models suitable for resource-constrained environments, such as mobile and embedded devices.

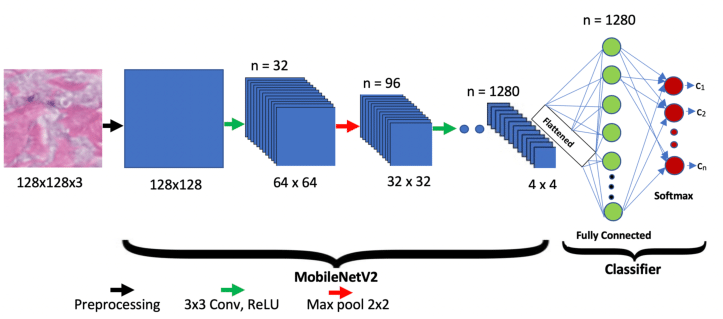
The MobileNetV2 architecture utilizes inverted residual blocks with linear bottleneck layers, which enable efficient feature extraction while reducing computational complexity. This design incorporates a lightweight depth-wise convolution followed by a pointwise convolution to expand the network's capacity.

One key innovation in MobileNetV2 is the introduction of "linear bottlenecks." These bottlenecks, implemented as 1x1 convolutions, act as a bottleneck to reduce the number of parameters without compromising the network's representational power. This approach strikes a balance between computational efficiency and expressive capacity.

Additionally, MobileNetV2 employs the concept of "inverted residuals" to improve information flow between layers. By using shortcut connections within the inverted residual blocks, the model mitigates the vanishing gradient problem and enables the learning of deeper representations.

To further optimize efficiency, MobileNetV2 incorporates depth-wise separable convolutions. This technique splits the standard convolution into depth-wise and pointwise convolutions, reducing parameter count and computation while preserving important spatial and channel-wise correlations.

MobileNetV2 achieves a favorable balance between model size, computational efficiency, and accuracy. It has demonstrated superior performance compared to previous lightweight architectures on various image classification benchmarks. As a result, MobileNetV2 has gained popularity for real-world applications with limited computational resources.



*Figure 3. MobileNetV2 Architecture*

**U-Net**

U-Net is a popular convolutional neural network architecture designed for image segmentation tasks (Ronneberger et al., 2015). It is widely used for various medical image segmentation applications, such as tumor detection, organ segmentation, and cell segmentation.

The U-Net architecture consists of an encoder-decoder structure with skip connections. The encoder part gradually down-samples the input image, capturing coarse-level features through convolutional and pooling layers. This encoding process helps extract abstract features from the input image.

The decoder part, on the other hand, gradually up-samples the encoded features to the original image resolution using transposed convolutions. The skip connections, which connect corresponding encoder and decoder layers, help in the fusion of low-level and high-level features. These skip connections enable the U-Net architecture to capture both local and global contextual information, enhancing the segmentation accuracy.

The U-Net architecture's unique aspect is its expansive path during decoding, which enables precise localization of segmented regions. This expansive path helps to recover spatial information and reconstruct detailed segmentation maps.

During training, the U-Net model is optimized using appropriate loss function such as ‘binary cross-entropy’. The model is trained to minimize the discrepancy between predicted segmentation maps and ground truth labels. This process involves backpropagation and gradient descent algorithms to update the network's parameters.

U-Net has demonstrated remarkable performance in various image segmentation tasks, especially in scenarios with limited training data. It has become a popular choice due to its ability to handle complex and intricate segmentation tasks, making it a valuable tool in medical imaging and other fields requiring accurate pixel-level segmentation.



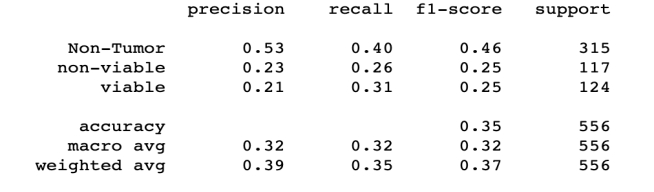
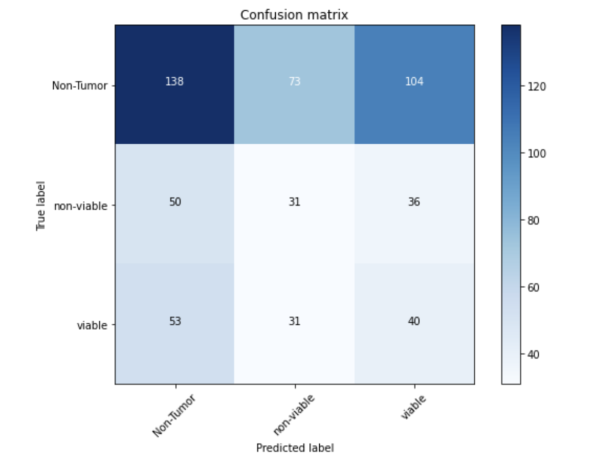
*Figure 4. U-Net Architecture*

**Results and Analysis**

In our project, we preprocessed the image data and stored them in NumPy arrays. Then, we loaded this image data along with the data frame that consists of image filenames, features and labels into a classification algorithm (Support Vector Classifier) and achieved an accuracy of 42%. So, we implemented a customized CNN on the image data using a data generator and achieved an accuracy of 35%. To improve the accuracy, we then used MobileNetV2 with the same image data and this time, the accuracy improved to around 70%. Since, the data is divided into 3 classes and the number of images is small, the models did not achieve best results. So, we divided the dataset into just 2 classes making the problem a binary classification. After converting the dataset into 2 classes, we again implemented a customized CNN which achieved around 86% accuracy. We then moved onto semantic segmentation using U-Net which is again a CNN which is majorly used for segmentation problems. Our dataset did not have annotated images. So, we used an online tool called Apeer to annotate images and generate masks. We then loaded the images and the corresponding masks that are created using Apeer into the U-Net model to create segmentations. The U-Net model achieved an accuracy of around 92% which shows that segmentation works best for medical image analysis because segmentation is practically more useful in the field of medicine.

Below are the results for different models.

1. **CNN Architecture:** The model architecture is designed to extract features progressively from input images using convolutional and pooling layers. These layers analyze the images and capture important patterns and information. Fully connected layers then learn complex relationships between these features, enabling the model to classify the images into one of the three predefined classes. To ensure efficient handling of the image data, data generators are employed, allowing for batch-wise loading and preprocessing. During the training process, the model's parameters, including weights and biases, are optimized using the Adam optimizer to minimize the categorical cross-entropy loss.

****

*Figure 5: Confusion Matrix and Classification report of CNN Model.*

Since the data is divided into three classes and size of dataset is too small for each class the performance of CNN didn’t provide good results. According to the above confusion matrix approximately 50 % non-tumor images were incorrectly classified. More than 75 % of non-viable images are incorrectly classified. Approximately 80 % of viable images were incorrectly classified. Since the accuracy is low we tried Mobile-Net as the next model to perform Multi class classification.

1. **Mobile NetV2:**

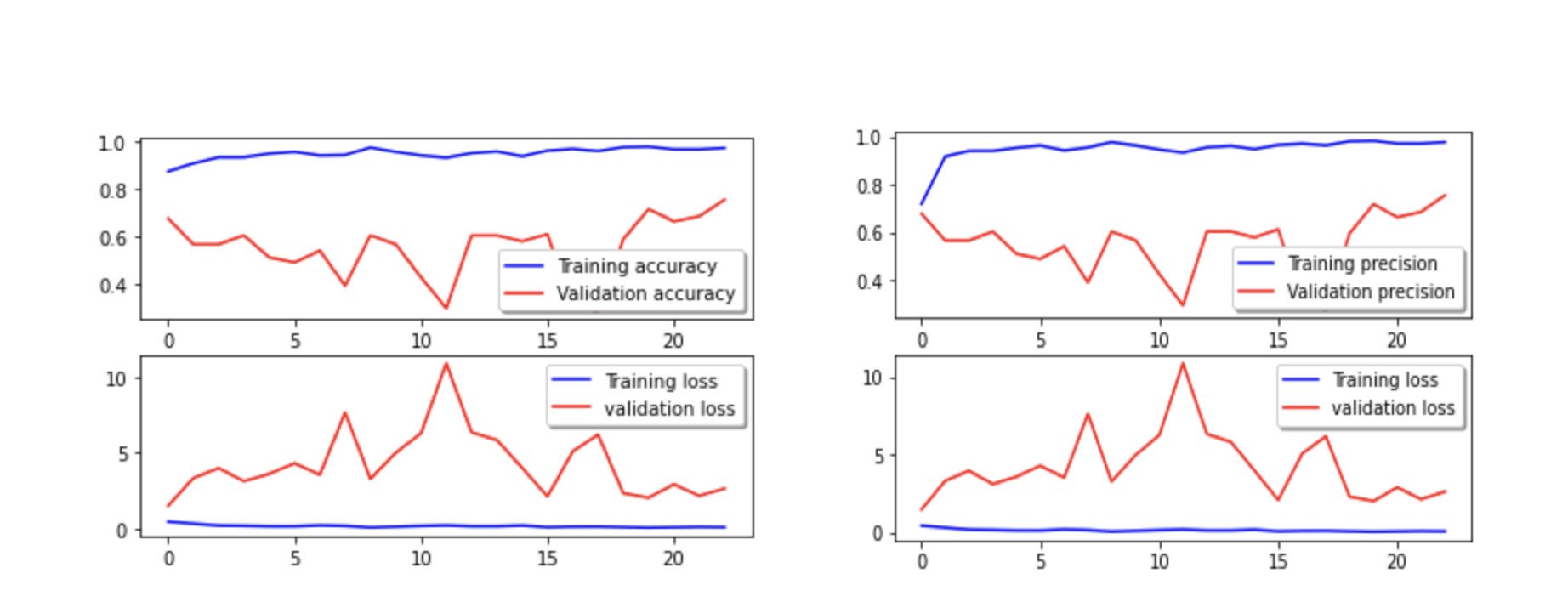
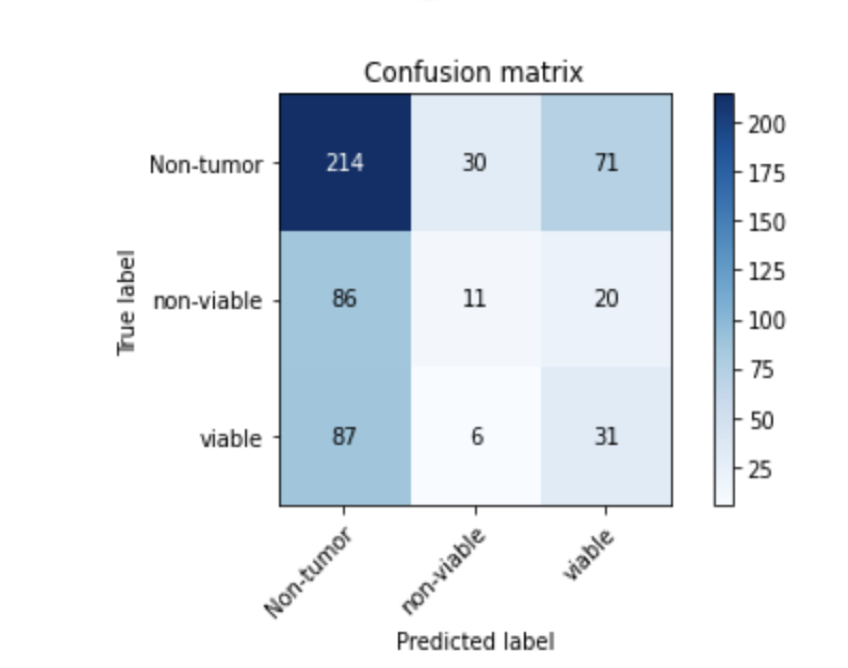


Figure 6. Accuracy loss and precision graphs of Mobile Net.

From the graph, we can observe that the training accuracy , loss and precision performed really well which show signs of overfitting as the validation accuracy, loss and precision are not that good.



*Figure 7: Confusion Matrix of Mobile Net Model*

According to our results we have got about 45% of Non-tumor images were incorrectly classified. 75% of non-viable images were incorrectly classified.75 % of viable images were incorrectly classified.

Here, we identified that the dataset is also a problem. The number of images in the dataset is too small to perform multiclass classification. So, we decided to make the dataset into two classes – tumor and non-tumor. Now, we are left with 553 tumor images and 536 non-tumor images making it a binary classification. So, we implemented a customized CNN again on this binary class data.

1. **Binary Classification using CNN:**

Converted the problem into a binary classification task.

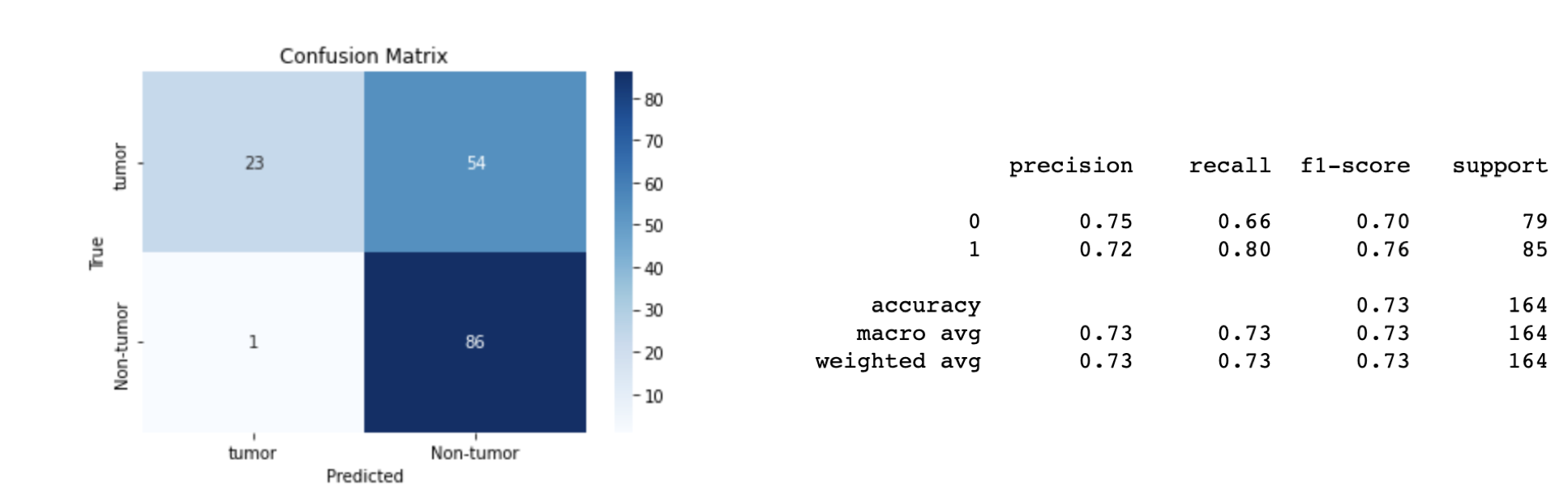
Categorized images into two classes:

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.

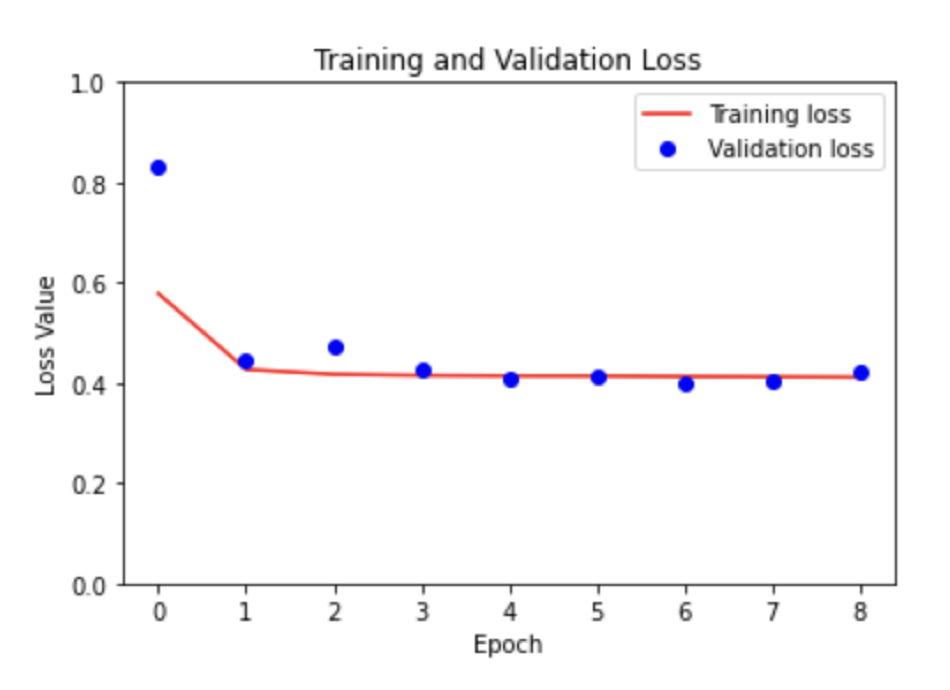


*Figure 8: Confusion Matrix and Classification Report of CNN.*

From the graph and classification report we can say that after converting the data set into binary classification our model has performed better approximately 73% of accuracy is achieved. We have got approximately 62% non-tumor images are accurately classified and 98% of tumor images are accurately classified. Hence we have achieved best accuracy out of all the models we trained till now.

1. **U-Net for Segmentation:**

We used the dataset with 2 classes. Tumor and non-tumor for segmentation. Since, the dataset did not have masked images, we used the masked images generated from Apeer. We loaded the images with their corresponding masks into the U-Net model for segmentation.

****

*Figure 9: Training and Validation Graph of U-Net Model.*

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.



*Figure 10: Semantic Segmentation of Tumor Image*.

Even though segmentation is not very accurate the segmentation depends on the masks provided along with the images. Since the masks used for this project were not from original dataset and are created by us using Apeer there is some inconsistency in predicting the masks as shown in the figure above.

**Comparison of CNNs Performance:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Performance Measures** | Custom CNN  (Multi-class) | MobileNetV2  (Multi-class) | Custom CNN  (Binary) | U-Net  (Semantic segmentation) |
| Image Dimensions | 224\*224 | 128\*128 | 240\*240 | 256\*256 |
| Optimizer | adam | adam | adam | adam |
| Activation  Function | relu | relu | sigmoid | sigmoid |
| Loss | Categorical cross-entropy | Categorical cross-entropy | Binary cross-entropy | Binary cross-entropy |
| **Accuracy** | 72% | 76% | 95% | 85% |

**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 U-Net model has demonstrated good performance for the segmentation task based on the masks provided to it. The U-Net 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.

**GIT HUB Link:** <https://github.com/bharatha-kiran/Capstone-606/>

**REFERENCES**

1. Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence.
2. Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
3. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning.
4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
5. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems.
6. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature.
7. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
8. Akay, M., Du, Y., Cheryl, L. S., Wu, M., Chen, T. Y., Assassi, S., Mohan, C., Akay, Y. M. "Deep Learning Classification of Systemic Sclerosis Skin Using the MobileNetV2 Model."
9. Paul S. Meltzer. (2021). New Horizons in the Treatment of Osteosarcoma
10. Bone Cancer Detection from X-Ray and MRI Images through Image Segmentation Techniques
11. https://www.ijrte.org/wp-content/uploads/papers/v8i6/F7159038620.pdf
12. Bone Cancer Detection Using Feature Extraction Based Machine Learning Model https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8712164/
13. Deep Learning for Classification of Bone Lesions on Routine MRI https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8190437/
14. 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