**Breast Cancer Segmentation from Ultrasound Using Deep Learning**

**Abstract:**

Breast cancer is a significant health concern affecting women worldwide. Early and accurate detection of breast cancer is crucial for effective treatment and improved patient outcomes. Image segmentation techniques play a vital role in analyzing medical images, enabling the identification and delineation of cancerous regions. This report presents a study on Breast Cancer Image Segmentation using deep learning techniques. The objective of this study is to develop an accurate and efficient model for segmenting breast cancer lesions from medical images. The proposed methodology utilizes a convolutional neural network (CNN) architecture and is evaluated on a comprehensive dataset. The results demonstrate the potential of deep learning-based image segmentation for breast cancer diagnosis and treatment.

**Introduction and Related Work:**

Breast cancer is the most common type of cancer among women globally, accounting for a significant number of cancer-related deaths. Early detection and precise delineation of cancerous regions in breast images are crucial for effective diagnosis, treatment planning, and monitoring the disease progression. Traditional manual segmentation methods can be time-consuming, subjective, and prone to inter-observer variability. Consequently, automated breast cancer image segmentation techniques have gained considerable attention in recent years.

In recent studies, deep learning-based approaches have shown promising results in medical image segmentation tasks, including breast cancer. Convolutional neural networks (CNN) have emerged as a powerful tool for automatically learning and extracting relevant features from images. These networks can capture complex patterns and spatial relationships, enabling accurate and efficient segmentation of breast cancer lesions.

While significant progress has been made in breast cancer image segmentation using deep learning, there are still challenges that need to be addressed. These challenges include handling class imbalance, limited annotated data, robustness to variations in image quality, and interpretability of the segmentation models.

In this project, we aim to develop a deep learning-based model on the field of breast cancer image segmentation. We utilize a comprehensive dataset consisting of benign and malignant breast images, encompassing diverse variations in image characteristics. The proposed model employs a CNN architecture tailored for breast cancer image segmentation, incorporating advanced techniques for improved performance. Through rigorous experimentation and evaluation, we analyze the effectiveness and limitations of our approach.

**Dataset and Features:**

The dataset used in this study is the Breast Ultrasound Images with Ground Truth (BUSI) dataset from Kaggle. The dataset comprises three main classes: benign, malignant, and normal. The benign class refers to a condition that is not harmful, while the malignant class refers to a state that is indicative of cancer. Malignant breast abnormalities may include invasive ductal carcinoma, invasive lobular carcinoma, or other types of breast cancer. The normal class consists of ultrasound images that do not exhibit any significant benign or malignant conditions. These normal images serve as a reference or baseline for comparison.

In this dataset, every case includes an image captured before segmentation and an image captured after segmentation(more than one image if more than one cancer in the raw data). Here is an example of one case form benign:

A black and white photo of a black animal

Description automatically generated A white object in the dark

Description automatically generated A white oval on a black background

Description automatically generated

The image before segmentation represents the original ultrasound image, while the image after segmentation highlights specific regions of interest, such as tumors or abnormalities. Including both the image before segmentation and the image after segmentation in the same dataset is valuable for training the model.

The image before segmentation provides the model with the raw data and allows it to learn the intrinsic features and patterns present in the breast ultrasound images. This enables the model to understand the overall structure and characteristics of different breast tissues. Then, it gives guidance on segmentation. The image after segmentation provides additional information by highlighting specific regions of interest, such as tumors or abnormalities. By training the model on both the original and segmented images, it can learn to associate these highlighted regions with their corresponding class labels. This can enhance the model's ability to detect and classify specific features in ultrasound images.

This dataset was divided into training and validation subsets, with a validation split of 20% and a random seed of 233 to ensure consistency.

**Method and technique used:**

The CNN model architecture consisted of multiple layers. The input images were first preprocessed using a rescaling layer to normalize the pixel values between 0 and 1. Normalizing images helps stabilize the gradients during backpropagation. Gradients are the values used to update the model's parameters during training. When pixel values are large or have a wide range, the gradients can also become large, resulting in unstable updates and slower convergence. Normalizing the images helps keep the gradients in a more stable range.

Same padding is a technique in CNN where zero-valued rows and columns are added around the input data before convolution. It ensures that the output feature map has the same size as the input, unlike valid padding which reduces the size.

Subsequently, three convolutional layers with ReLU activation were applied. ReLU activation introduces non-linearity to the network, allowing it to model complex relationships between the input and output. ReLU activation sets negative values to zero while preserving positive values, effectively introducing sparsity, and promoting the network's ability to learn more discriminative features.

Each followed by a max-pooling layer for down sampling. This was done to capture important features at different spatial scales. By reducing the spatial dimensions of the feature maps, max-pooling helps to capture the most salient features while providing some degree of translation invariance. It also reduces the computational complexity of the network and helps prevent overfitting by enforcing local spatial invariance.

The model was compiled using the Adam optimizer with a learning rate of the default value. The loss function used was Sparse Categorical Cross entropy, which is suitable for multi-class classification tasks. The model was trained for a total of 10 epochs, with early stopping applied based on the validation loss. Early stopping was used to prevent overfitting and restore the weights of the best epoch.

The output of the convolutional layers was flattened and fed into two fully connected (dense) layers. The first dense layer had 128 units and used ReLU activation, while the final dense layer had 3 units with a SoftMax activation function, representing the three classes for breast cancer segmentation. By using Softmax, the model can make a confident prediction by selecting the class with the highest probability. The class with the highest probability is considered as the predicted class label for the input sample.

During training, the model's performance was monitored by tracking the accuracy and loss on both the training and validation sets. The training process was visualized using line plots, illustrating the changes in accuracy and loss over the epochs. The CNN model implemented in this project provides a framework for breast cancer image segmentation. The use of convolutional layers enables the model to automatically learn and extract relevant features from the images, improving the accuracy of segmentation. The training process with early stopping helps prevent overfitting and ensures the model's generalizability to unseen data.

**Results:**

Finally, visualizations of the training and validation accuracy and loss are plotted using matplotlib. These plots help in visualizing the model's learning progress over the training epochs. Here is the output and we obtained the following results:

79/79 [==============================] - 13s 145ms/step - loss: 0.8829 - accuracy: 0.5938 - val\_loss: 0.6346 - val\_accuracy: 0.7016

Epoch 2/10

79/79 [==============================] - 12s 145ms/step - loss: 0.6110 - accuracy: 0.7379 - val\_loss: 0.4305 - val\_accuracy: 0.8127

Epoch 3/10

79/79 [==============================] - 11s 143ms/step - loss: 0.4755 - accuracy: 0.7973 - val\_loss: 0.4095 - val\_accuracy: 0.8413

Epoch 4/10

79/79 [==============================] - 11s 143ms/step - loss: 0.3846 - accuracy: 0.8472 - val\_loss: 0.2820 - val\_accuracy: 0.9048

Epoch 5/10

79/79 [==============================] - 11s 141ms/step - loss: 0.2414 - accuracy: 0.9105 - val\_loss: 0.2655 - val\_accuracy: 0.9143

Epoch 6/10

79/79 [==============================] - 11s 140ms/step - loss: 0.1697 - accuracy: 0.9438 - val\_loss: 0.2518 - val\_accuracy: 0.9333

Epoch 7/10

79/79 [==============================] - 11s 140ms/step - loss: 0.1193 - accuracy: 0.9620 - val\_loss: 0.2647 - val\_accuracy: 0.9397

Epoch 8/10

79/79 [==============================] - 11s 141ms/step - loss: 0.0746 - accuracy: 0.9739 - val\_loss: 0.2448 - val\_accuracy: 0.9365

Epoch 9/10

79/79 [==============================] - 11s 140ms/step - loss: 0.0846 - accuracy: 0.9707 - val\_loss: 0.2007 - val\_accuracy: 0.9587

Epoch 10/10

79/79 [==============================] - 11s 142ms/step - loss: 0.0703 - accuracy: 0.9762 - val\_loss: 0.2035 - val\_accuracy: 0.9587

20/20 [==============================] - 1s 28ms/step - loss: 0.2035 - accuracy: 0.9587

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

rescaling (Rescaling) (None, 224, 224, 3) 0

conv2d (Conv2D) (None, 224, 224, 16) 448

max\_pooling2d (MaxPooling2 (None, 112, 112, 16) 0

D)

conv2d\_1 (Conv2D) (None, 112, 112, 32) 4640

max\_pooling2d\_1 (MaxPoolin (None, 56, 56, 32) 0

g2D)

conv2d\_2 (Conv2D) (None, 56, 56, 64) 18496

max\_pooling2d\_2 (MaxPoolin (None, 28, 28, 64) 0

g2D)

dropout (Dropout) (None, 28, 28, 64) 0

flatten (Flatten) (None, 50176) 0

dense (Dense) (None, 128) 6422656

dense\_1 (Dense) (None, 3) 387

=================================================================

Total params: 6446627 (24.59 MB)

Trainable params: 6446627 (24.59 MB)

Non-trainable params: 0 (0.00 Byte)

A graph of loss and validation

Description automatically generated

Loss: The loss values represent the value of the loss function during training and evaluation. Lower loss values indicate that the model's predictions are closer to the true labels.

Training Loss: The training loss decreases gradually from 0.8829 in the first epoch to 0.0703 in the tenth epoch. This decreasing trend indicates that the model is effectively learning from the training data and improving its predictions over time.

Validation Loss: The validation loss also decreases from 0.6346 in the first epoch to 0.2035 in the tenth epoch. This reduction in the validation loss suggests that the model is generalizing well and not overfitting to the training data.

Accuracy: The accuracy values represent the proportion of correctly classified samples during training and evaluation. Higher accuracy values indicate that the model's predictions align well with the true labels.

Training Accuracy: The training accuracy increases from 0.5938 in the first epoch to 0.9762 in the tenth epoch. This upward trend demonstrates that the model is learning to classify the training samples more accurately as the training progresses.

Validation Accuracy: The validation accuracy improves from 0.7016 in the first epoch to 0.9587 in the tenth epoch. This improvement suggests that the model is performing well on unseen data and generalizing effectively.

In this seed, the model doesn't exhibit significant signs of overfitting during training. So early stopping is not used.

Overall, the decreasing loss values and increasing accuracy values for both the training and validation sets indicate that the model is learning and making progress over the course of training. The relatively high accuracy and low loss on the validation set in the final epoch suggest that the model has achieved a good level of performance for the breast cancer image segmentation task.

**Future Work:**

While the achieved results are promising, there are several areas for future exploration and improvement:

Data Augmentation: Applying various data augmentation techniques, such as rotation, scaling, or flipping, can help increase the diversity of the training data and improve the model's ability to generalize to unseen examples.

Hyperparameter Tuning: Experimenting with different hyperparameter settings, such as learning rate, batch size, and dropout rate, may lead to further improvements in the model's performance. Techniques like grid search or random search can be employed to find optimal hyperparameter configurations.

More class: Three cases in this dataset may not accurately represent the complexity of the real-world problem. To address this limitation, it would be beneficial to expand the dataset with additional classes and labels representing different types of cancer, enabling the model to generalize better and be more useful for real-world applications.

Parameter Tuning: Explore different regularization strengths for L1 and L2 regularization. The regularization strength determines the amount of penalty applied to the model's parameters during training. By tuning the regularization strength, you can find the optimal balance between reducing overfitting and preserving important model features. Conduct experiments with various regularization strengths and evaluate their impact on the model's segmentation performance.

**Conclusion:**

In conclusion, the development of a deep learning-based model for breast cancer image segmentation. The model's ability to segment breast cancer lesions accurately and efficiently from ultrasound images. By automating the segmentation process, medical professionals can save valuable time and reduce subjectivity, leading to quicker and more precise diagnoses. With further research and refinement, deep learning models for breast cancer image segmentation can become an invaluable tool in the fight against this devastating disease, ultimately improving patient care and outcomes.

**Appendix**

import tensorflow as tf

import numpy as np

import matplotlib.pyplot as plt

import pathlib

from keras.callbacks import EarlyStopping

path = 'D:/Downloads/Study/deep\_learning/dataset/archive/Dataset\_BUSI\_with\_GT/'

data\_dir = pathlib.Path(path)

class\_names = np.array(sorted([item.name for item in data\_dir.glob("\*")]))

class\_names

batch\_size = 16

img\_height = 224

img\_width = 224

from keras.utils import image\_dataset\_from\_directory

train\_data = image\_dataset\_from\_directory(

    data\_dir,

    validation\_split=0.2,

    subset="training",

    seed=233,

    image\_size=(img\_height, img\_width),

    batch\_size=batch\_size

)

val\_data = image\_dataset\_from\_directory(

    data\_dir,

    validation\_split=0.2,

    subset="validation",

    seed=250,

    image\_size=(img\_height, img\_width),

    batch\_size=batch\_size

)

from keras import layers

model = tf.keras.Sequential([

    layers.Rescaling(1./255, input\_shape=(img\_height, img\_width, 3)),

    layers.Conv2D(16, 3, padding='same', activation='relu'),

    layers.MaxPooling2D(),

    layers.Conv2D(32, 3, padding='same', activation='relu'),

    layers.MaxPooling2D(),

    layers.Conv2D(64, 3, padding='same', activation='relu'),

    layers.MaxPooling2D(),

    layers.Dropout(0.5),

    layers.Flatten(),

    layers.Dense(128, activation='relu'),

    layers.Dense(3, activation="softmax")

])

model.compile(

    optimizer="Adam",

    loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True),

    metrics=["accuracy"]

)

epochs = 10

early\_stopping = EarlyStopping(

    monitor='val\_loss',

    patience=3,

    restore\_best\_weights=True

)

history = model.fit(

    train\_data,

    epochs=epochs,

    validation\_data=val\_data,

    batch\_size=batch\_size,

    callbacks=[early\_stopping]

)

history.history.keys()

acc = history.history['accuracy']

val\_acc = history.history['val\_accuracy']

loss = history.history['loss']

val\_loss = history.history['val\_loss']

epochs\_range = range(len(acc))

plt.figure(figsize=(8, 8))

plt.subplot(1, 2, 1)

plt.plot(epochs\_range, acc, label='Accuracy')

plt.plot(epochs\_range, val\_acc, label="Validation Accuracy")

plt.legend()

plt.subplot(1, 2, 2)

plt.plot(epochs\_range, loss, label='Loss')

plt.plot(epochs\_range, val\_loss, label="Validation Loss")

plt.legend()

plt.show()

model.evaluate(val\_data)

model.summary()

**References**

Dataset: Breast Ultrasound Images Dataset

https://www.kaggle.com/datasets/aryashah2k/breast-ultrasound-images-dataset

Author: ARYA SHAH

Code based on: Breast Cancer Image Segmentation | CNN

<https://www.kaggle.com/code/ardawrld/breast-cancer-image-segmentation-cnn#Make-Plotting-of-Random-%C4%B0mages>

Author: ARDA.WRLD

Adam, R., Dell&rsquo;Aquila, K., Hodges, L., Maldjian, T., & Duong, T. Q. (2023, July 24). *Deep learning applications to breast cancer detection by Magnetic Resonance Imaging: A literature review - breast cancer research*. BioMed Central. https://breast-cancer-research.biomedcentral.com/articles/10.1186/s13058-023-01687-4

Breast cancer segmentation methods: Current status and future ... - hindawi. (n.d.). https://downloads.hindawi.com/journals/bmri/2021/9962109.pdf