Original article

Journal of High School Science

Image segmentation and classification in breast cancer ultrasound images using a Convolutional Neural Network based attention model

Narin A[1], Dulundu A[2]

Submitted: February 12, 2024, Revised: version 1, July 18, 2024
Accepted: July 24, 2024

Abstract
Cancer is one of humanity's biggest challenges. Although there are many ways to prevent cancer, some types of cancer are still largely incurable. One of the most common types of cancer is breast cancer and the most important action toward a favorable outcome is early diagnosis. Correct diagnosis is one of the most critical processes in breast cancer treatment. There are many studies in the literature to predict the type of breast tumors. In this study, Artificial Intelligence and machine learning were used to increase the incidences of correct diagnosis. A convolutional neural network (CNN) was used in the models to classify and segment breast ultrasound images. Two separate models were created; one for image classification, and another for image segmentation, using TensorFlow. The models were based on current Convolutional Neural Network models published previously. The image segmentation model was based on a customized U-Net architecture. Attention gates were added to this customized U-Net model. The image classification model was constructed using a 50-layer ResNet model. The resultant CNN yielded an accuracy of 99.35% for image segmentation. For image classification, the separately constructed CNN model yielded an accuracy of 97.43%. Hence, it was possible to obtain high accuracies to classify and segment breast cancer from ultrasound images using our CNN models.

Keywords
Deep learning, Convolutional Neural Network, Image segmentation, Object detection, Attention U-NET, ResNet, Breast cancer, Ultrasound, Image classification, Mammography

[1]Ali Kaan Narin, Robert College, Kuruçesme Cad. No.8 Arnavutköy - Beşiktaş 34345, Istanbul, Turkey. narinalikaan@gmail.com
[2]Corresponding Author: Alp Dulundu, Robert College, Kuruçeşme Cad. No.8 Arnavutköy - Beşiktaş 34345, Istanbul, Turkey. alpdulundu2029@gmail.com

1 Introduction

Malignant tumors involve the uncontrolled proliferation and spread of cells in tissues (1). Breast cancer is the most common cancer, accounting for 11.7% of all cancer cases for both sexes (2). Although there exist various methods for detecting breast cancer, these methods are not universally optimal. Given that breast cancer is the most common type of cancer in humans, other methods need to be evaluated to improve detection. Currently, approximately 48% of breast cancer cases go undiagnosed when using only one imaging modality (3). In addition, other studies focusing on Triple Negative Breast Cancer, a subtype of breast cancer, have found that in 1 in 3 patients who are imaged, at least 1 of these imaging modalities misdiagnoses the cancer type (4).

A factor contributing to the misdiagnosis of various cancers is asymmetric cancer growths and microcalcification masses that can be missed using imaging techniques. These imperfections can lead to mistakes even when images are examined manually. Artificial intelligence has played an important role in improving the detection rate in cancer imaging by helping experts classify hard-to-see features of images (5). It has been shown that the use of artificial intelligence for breast cancer can increase detection rates by 5% in low-density breast images and 12% in high-density ones (6).

Artificial intelligence models that can help in the medical field have been reported in the literature for several years. However, existing work on the subject focuses more on the classification of masses rather than the on the removal and segmentation of noise in the image. This leads to models that can classify *manually* segmented images with high accuracy but cannot segment the masses in the images on their own. Software is available to help radiologists label and annotate images. However, again, these systems require *manual* annotation, which poses some challenges in terms of processing large datasets and accuracy. As noted by Campanella et al., the accuracy and specificity of such software are important features, and their improvement requires continuous study and research (7). Such software can facilitate and speed up the work of radiologists, but it is difficult to produce reliable results because of the need for manual intervention. Moreover, existing models specialized in mass classification mostly work on microscopy images, however, microscopy is not the main tool used to obtain images of cancerous masses in real life (8). In addition, the accuracy of these systems still needs to be improved. Although in some cases these systems have reached 100% accuracy for cancer detection, it is not easy to transfer the methodologies used in these systems to other cases such as breast cancer (9). Most studies using AI detection for breast cancer focus on whole slide imaging (WSI) or mammography images (10). Few studies have used ultrasound, which we used in our study. Compared to ultrasound images, WSI and mammography images have higher image quality. Therefore, it is technically easier to build AI-assisted systems to classify mammography images and WSI because they require less denoising and detection of microcalcifications. Our work combines aspects of mass segmentation and image classification to minimize human error. We specifically worked on a model applicable to ultrasound images, because ultrasound imaging is more accessible and affordable (11).

1.1 Visual detection methods for breast cancer

Various imaging techniques play an important role in breast cancer detection, allowing radiologists to effectively diagnose the disease and monitor its progress. Mammography is a widely used screening method that uses low-dose X-rays to capture detailed images of breast tissue and detect abnormalities such as tumors or calcifications. Ultrasound uses sound waves to produce real-time images and helps to evaluate suspicious areas that have been previously identified by mammography. Whole slide imaging (WSI), also known as digital pathology, is the scanning and digitization of pathology slides. Each of these techniques has unique strengths and considerations that contribute to a comprehensive approach to detecting and diagnosing breast cancer (12).

1.1.1 *Mammography*

Mammography is a widely adopted breast cancer screening method. Its ability to detect early-stage tumors and microcalcifications makes it effective for early detection. Mammograms are relatively inexpensive and readily available. However, the technique has limitations such as exposure to low-dose ionizing radiation, discomfort during compression, and low sensitivity leading to false negatives in dense breast tissue (13).

1.1.2 *Ultrasound*

Ultrasound imaging is a valuable tool for breast cancer detection. It is safe for frequent use as it does not expose the body to ionizing radiation. Ultrasound provides real-time images that help classify lesions and guide biopsy procedures. However, its effectiveness depends on the operator and its interpretation can be subjective, hence ultrasound images lead to low reproducibility (14). It is particularly useful in the evaluation of dense breast tissue and benign cysts where mammography may be less sensitive (15). Despite its benefits, ultrasound can produce false positives and is often not as effective in detecting microcalcifications due to over-reliance on the radiologist's interpretation.

1.2 Artificial Intelligence

The integration of Artificial Intelligence (AI) into healthcare has revolutionized medical research, diagnosis, and treatment, uncovering significant potential. The impact of AI on diagnostic processes, especially in analyzing medical images and detecting abnormalities, has been highlighted in recent research. This is in line with findings from studies highlighting the role of AI in pathology, where AI improves diagnostic accuracy and eases the workload on healthcare professionals (16). The application of AI extends to personalized medicine, highlighting its role in tailoring treatments based on individual patient characteristics. Moreover, different types of AI such as natural language processing, machine learning, rule-based expert systems, physical robots and robotic process automation are extremely valuable for the healthcare sector (17). Cumulative evidence shows that AI holds promise not only in improving diagnostic accuracy but also in advancing personalized and more effective treatment strategies. As AI continues to evolve, its integration into health systems has the potential to improve patient outcomes, reduce costs and revolutionize care delivery.

1.3 Object detection

Object recognition, a subset of computer vision, is involved in identifying masses seen in scans as being benign, cancerous or of no malignant consequence (18). It is adept at precisely localizing and outlining abnormalities in medical images, such as tumors or lesions. This technology plays a vital role in early detection of cancer, offering the ability to identify subtle signs in their early stages and enabling timely intervention for improved outcomes. Object recognition algorithms can work seamlessly across various imaging modalities such as ultrasound, mammography and MRI, among others, providing a versatile and comprehensive approach to cancer detection. These algorithms increase efficiency by automating the analysis process, reduce the workload on healthcare professionals, and contribute to faster diagnosis by accelerating the review of medical images (19).

1.4 Image segmentation

Image segmentation is of great importance in image processing and pattern recognition (20). The technique examines an image by segmenting it into different areas. The technique is categorized into 3 subclasses: thresholding or clustering, image edge detection and area extraction. The use of this technique in the medical field is not easy. Different medical image acquisition methods pose different problems. The ultrasound images analyzed in our project contain speckle noise (21). Segmentation of medical images is difficult due to the lack or diffusion of tissue boundaries resulting from low image contrast and noise (22). Therefore, images are segmented manually, which is labor-intensive and time-consuming. This method is both inefficient and prone to human error (23). Recent studies have attempted to automate image segmentation. For this purpose, supervised and unsupervised learning methods are used. New image segmentation methods using deep learning models have emerged and have been reported in the literature. Some of them are convolutional pixel labeling networks, encoder and decoder systems, multiscale and pyramid-based approaches, recurrent networks, visual attention models, and generative models in contrasting environments (24). In our project, we used a visual attention model for segmentation, which we will explain in more detail later.

1.5 Deep learning

Deep learning is the integration of several layers of representation learning techniques into a model that enables a machine to automatically detect what is needed for detection or classification when fed with raw data (25). Deep learning models require less human support than models that use machine learning (26). Therefore, current research prioritizes deep learning models (27,28). Deep learning techniques have made a lot of progress, especially due to emerging multilayer and nonlinear models (29). These developments have enabled, deep learning models to be used in the medical field for image flattening (30), image fusion (31), image annotation (32,33), image segmentation (34), prognosis (35), microscopic image analysis (36) and lesion detection (37).

1.5.1 *Convolutional Neural Networks*

Convolutional neural networks (CNNs) are similar to artificial neural networks (ANNs) in that they are composed of neurons that optimize themselves through learning. Compared to ANNs, CNNs perform more pattern recognition in visuals (38). They can work on larger data sets than ANNs. To work on large data sets, the nerves in CNNs are arranged in 3D. ANNs, on the other hand, are difficult to use on large-scale or greater resolution images such as (for example) those that use 64×64 or more pixels. For ANNs to function on such data sets, a very large network needs to be built. This effort is very laborious and requires computational performance that most systems do not have access to (39). In this respect, CNNs are preferred over ANNs for image segmentation and classification tasks (40). CNNs usually consist of 3 main layers. These are the convolutional layer, the pooling layer, and the fully connected layer (41).

The biggest obstacle to the use of CNNs in the medical field is that labeled datasets are less accessible in the medical field compared to those in other fields (42). As a solution to this problem, algorithms that process images without supervision have emerged. For example, in "A deep convolutional neural network architecture for interstitial lung disease pattern classification" (43), an unsupervised learning model is used to achieve better performance than most currently available supervised systems. Compared to ANNs in general, CNNs are very effective in executing many tasks such as image segmentation, image classification, image line detection, especially in areas where large data needs to be analyzed, such as the medical field, and have consequently led to very rapid developments in these fields (44).

1.6 U-Net

U-Net is an algorithm developed by Olaf Ronneberger, Philipp Fischer, and Thomas Brox to reduce the data requirements and segmentation error of Convolutional Neural Networks (45). In the U-Net algorithm, there is an upsampling layer in addition to the pooling layer found in conventional Convolutional Neural Networks. Upsampling seeks to improve the quality of the image by increasing the sampling rate of the image introduced into the system (46). In this way, another convolutional layer that receives the output of the upsampling layer as input can access a much more accurate image that has better resolution. As can be seen in Figure 1, this process can be repeated several times at different sizes, resulting in a much smaller feature map compared to conventional methods.

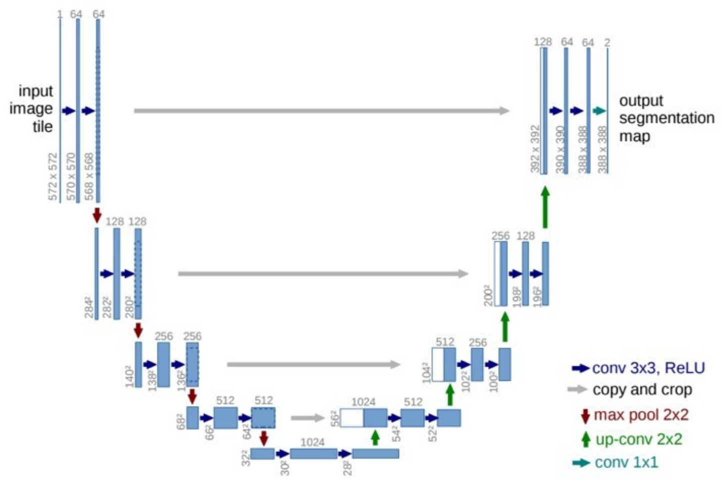

Figure 1. U-Net Architecture Map (45).

This procedure reduces the amount of noise in the algorithm and hence reduces the margin of error (47). U-Net algorithms can thus achieve similar or better results than manual expert analysis. While the basic U-Net algorithm is highly successful, there are several algorithms that aim to improve the encoders and decoders of U-Net (48). Although these algorithms improve the efficiency of U-Net in some cases, in most operations they produce similar results to the basic U-Net (49). Although U-Net algorithms are widely used in the biomedical field, our focus on breast cancer ultrasound images is one of their least common uses in the literature. Figure 2, shows that breast cancer is one of the least researched cancers using the U- Net model among the cancers focused on in the papers written involving U-Net. Figure 2 also shows that the ultrasound images we are analyzing is one of the least studied image methods, regardless of the type of cancer. Accordingly, we believe that the topic of our project presents a knowledge gap and necessitates further research.

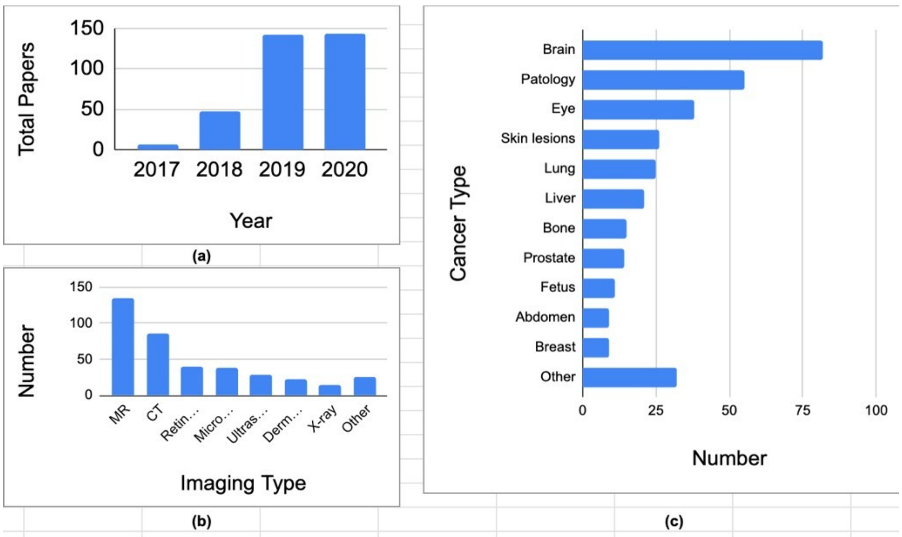

Figure 2. Distribution of Research in the Literature by Image Type and Cancer Type (47)

1.6.3 *Attention U-Net*

Attention U-Net, a U-Net-based algorithm, was constructed based on the principle of attention gates in the article "Attention U-Net: Learning Where to Look for the Pancreas" (50). Apart from Attention U-Net, there are several other U-Net-based systems that have been constructed based on the "attention" principle in machine learning. It has been observed that these systems work more efficiently than traditional CNN models, provide superior results in improving the quality of local features and reduce the false positive rate (51,52). The attention system used by the models consists of several CNNs 'folded' on top of each other. This attention system is divided into stochastic attention and deterministic attention (53). While deterministic models always output the same result for a data set, stochastic models do not always arrive the same result because they include uncertainty and randomness in their algorithm (54). The efficiency of attention-based systems varies according to the attention model they use and the processing power they can access. In addition to increasing the number of CNN layers used, it has been shown that such systems increase the efficiency of the systems by dividing the task into parts and analyzing them by different attention models (55).

1.7 ResNet50

ResNet50 is a 50-layer version of the ResNet CNN model. The ResNet model is a model developed in response to the difficulty of the learning process with the increase in the number of layers seen in deeper and multilayer traditional networks. The model facilitates the learning process by skipping one or more layers using shortcuts. ResNet requires 3.6 billion FLOPs operations (56). This is a much lower requirement compared to the 19.6 billion FLOPs required by the VGG-19 network developed by Karen Simonyan in the same year (57). While the original ResNet had 34 layers, ResNet50 has been expanded to 50 layers for deeper learning tasks. ResNet50 can achieve an ID accuracy of 78.5% when operating on the general ImageNet dataset (58). This is a percentage of correct identification that competes favorably with other models. In addition, accurate identification rates of up to 99.99% have been reported in areas such as building damage analysis using ResNet50 (59).

2 Methods

2.1 Libraries

The libraries used in our research are presented in this section.

2.1.1 *Keras*

Keras is a high-level, advanced, open-source application programming interface (API) of Tensorflow that provides an efficient and accessible interface for solving machine learning (ML) problems (60). Keras covers every step of the ML workflow, from data processing to hyperparameterization and deployment. According to a study by Manaswi , "keras allows developers to focus on the core concepts of deep learning, such as building layers for neural networks, while taking care of the minutiae of tensors, their shapes and mathematical details" (61). In addition, keras can be run on a tensor processing unit (TPU) or large graphics processing unit (GPU) clusters or can be exported to run keras models in a browser or on mobile devices. Keras models can also be served through a web API (62).

2.1.2 *NumPy*

NumPy is the core library for scientific computing in Python, providing features such as various derived objects (masked arrays, matrices, tensors, etc.), including the multidimensional array object. Being open source, NumPy's powerful N-dimensional array object integrates with a wide range of libraries and is the core library for scientific computing, underpinning libraries such as Pandas, Scikit-learn and SciPy. NumPy arrays can execute advanced mathematical operations with large datasets more efficiently and with fewer lines of code than when using Python's built- in arrays. With these advantages, it is an indispensable library for building machine learning models (63).

2.1.3 *Pandas*

Pandas is a fast, powerful, flexible and easy-to-use open-source data analysis and manipulation tool built on top of the Python programming language. It was created in 2008 by Wes McKinney. Since pandas is built on top of the Numpy package, it is open for use with Numpy, but also works with other libraries such as SciPy and scikit-learn

(64). It presents data in a way suitable for data analysis, especially through the "DataFrame" and "Series" classes (65).

2.2 Data set

In this project, the dataset used to build deep learning models for image segmentation and classification includes breast ultrasound images of women between the ages of 25 and 75. The images were collected from a total of 600 different women (66). Images are categorized into 3 classes as benign, malignant, and normal. Apart from the ultrasound images themselves, the cancer mass of each ultrasound image is segmented in black and white as separate images in the dataset. The sample ultrasound and the segmented black and white cancer mass are shown in Figure 3. The original dimensions of the images in the dataset are 500 by 500.

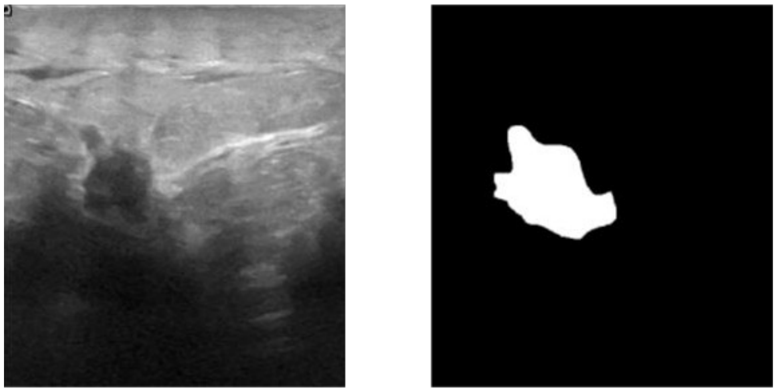

Figure 3. Ultrasound Image (left) and Mask (Right)

2.3 Segmentation algorithm

Attention U-Net was used for the segmentation algorithm. Figure 4 is a representation of the underlying U-Net architecture. Figure 5 is a representation of the architecture of the "Attention Gate" that is shown in Figure 4.

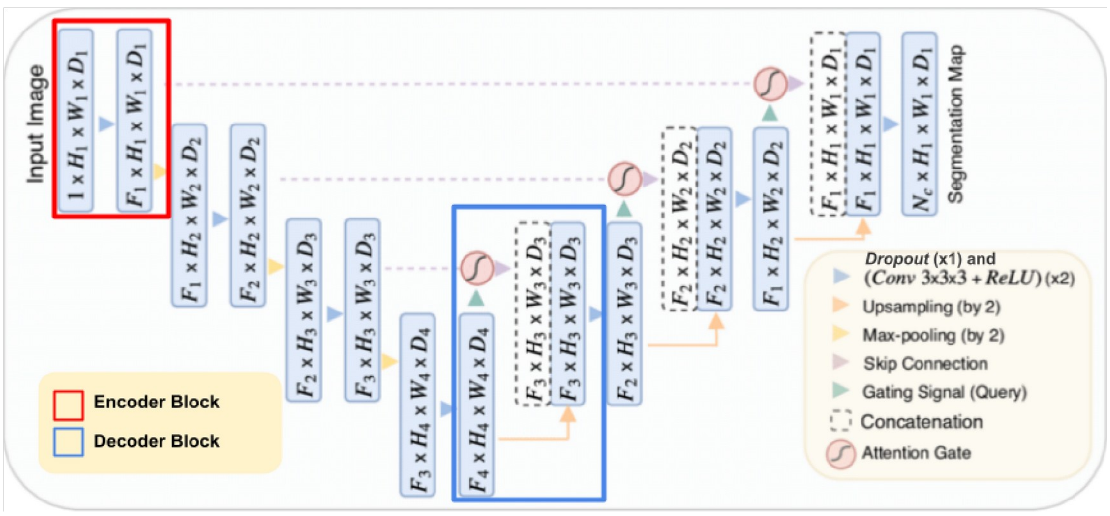

Figure 4. U-Net Attention Gate Model Map (50)

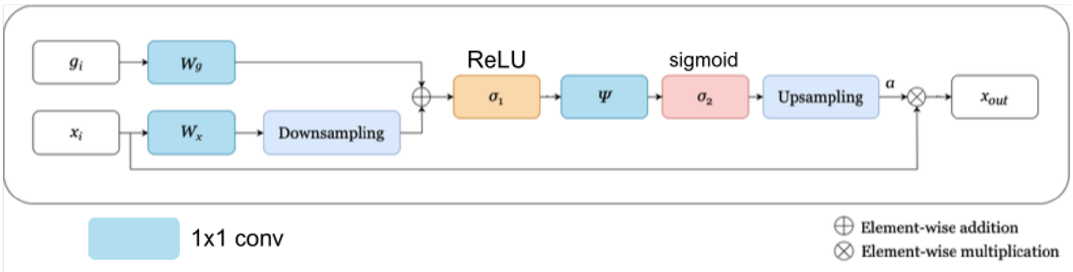

Figure 5. Attention Gate Sketch (67)

2.3.1 *Data preprocessing*

Before training the Attention U-Net model, the size of each image needed to be adjusted so that the data was of consistent size and converted into an array for the deep learning model. As can be seen in Figure 6, the images were first converted to a NumPy array. The images were then resized to a size equal to the value of the SIZE variable. According to prior studies, working with 256 by 256 images provided better results with U-Net (68). In addition, working with a smaller image compared to the original image speeded up the training process of the model. Therefore, in this project, the images were fixed to 256 by 256. Finally, the image values were divided by 255, simplified to the range 0 to 1, and the values were rounded to four decimal places using NumPy's np.round function, again, with the objective to speed up the training process.

```
def load_image(image, SIZE):
    return np.round(tf.image.resize(img_to_array(load_img(image))/255.,(SIZE, SIZE)),4)
```

Figure 6. Data Preprocessing Code Chunk

2.3.2 *Encoder*

The encoder block shown in Figure 7 contained the Conv2D, Dropout and MaxPool2D functions from the Keras library. It contained two of the Conv2D functions with a kernel size of 3, which were repeated multiple times throughout the architecture, and a Dropout function to reduce overfitting.

2.3.3 *Decoder*

The decoder block shown in Figure 4 is shown in Figure 8 by combining functions from the Keras libraries. It contained the Upsampling2D function from the Keras library and the function that combined the data processed in the attention gate of the block opposite to the block processed in the U-Net architecture.

2.3.4 *AttentionGate*

The attention gate shown in Figure 5 is shown as code in Figure 9 written with the functions in the Keras library.

```python
class EncoderBlock(Layer):

    def __init__(self, num_filters, dropout_rate, use_pooling=True, **kwargs):
        super(EncoderBlock, self).__init__(**kwargs)

        self.num_filters = num_filters
        self.dropout_rate = dropout_rate
        self.use_pooling = use_pooling

        self.conv1 = Conv2D(num_filters, kernel_size=3, strides=1, padding='same', activation='relu', kernel_initializer='he_normal')
        self.dropout = Dropout(dropout_rate)
        self.conv2 = Conv2D(num_filters, kernel_size=3, strides=1, padding='same', activation='relu', kernel_initializer='he_normal')
        self.pooling_layer = MaxPool2D()

    def call(self, inputs):
        x = self.conv1(inputs)
        x = self.dropout(x)
        x = self.conv2(x)
        if self.use_pooling:
            pooled_output = self.pooling_layer(x)
            return pooled_output, x
        else:
            return x
```

Figure 7. Encoder Block Code Chunk

```python
class DecoderBlock(Layer):

    def __init__(self, num_filters, dropout_rate, **kwargs):
        super(DecoderBlock, self).__init__(**kwargs)

        self.num_filters = num_filters
        self.dropout_rate = dropout_rate

        self.upsample = UpSampling2D()
        self.encoder_block = EncoderBlock(num_filters, dropout_rate, use_pooling=False)

    def call(self, inputs):
        x, skip_x = inputs
        upsampled_x = self.upsample(x)
        concatenated_x = concatenate([upsampled_x, skip_x])
        decoded_x = self.encoder_block(concatenated_x)
        return decoded_x
```

Figure 8. Decoder Block Code Chunk

```python
class AttentionGate(Layer):
    def __init__(self, filters, bn, **kwargs):
        super(AttentionGate, self).__init__(**kwargs)

        self.filters = filters
        self.bn = bn

        self.normal = Conv2D(filters, kernel_size=1, padding='same', activation='relu', kernel_initializer='he_normal')
        self.down = Conv2D(filters, kernel_size=1, strides=2, padding='same', activation='relu', kernel_initializer='he_normal')
        self.learn = Conv2D(1, kernel_size=1, padding='same', activation='sigmoid')
        self.resample = UpSampling2D()
        self.BN = BatchNormalization()

    def call(self, inputs):

        X, skip_X = inputs

        normal_branch = self.normal(X)
        downsampled_skip = self.down(skip_X)
        combined = Add()([normal_branch, downsampled_skip])
        attention_map = self.learn(combined)
        upsampled_attention = self.resample(attention_map)
        attention_applied = Multiply()([upsampled_attention, skip_X])

        if self.bn:
            return self.BN(attention_applied)
        else:
            return attention_applied
```

Figure 9. Attention Gate Code Chunk

2.3.5 *Attention U-Net*

The model created with the defined blocks is shown in Figure 10. The functions defined by considering the U-Net architecture shown in Figure 4 were called.

```
input_layer = Input(shape=images.shape[-3:])


pool1, conv1 = EncoderBlock(neurons, dropout_rate=0.1, name="Encoder1")(input_layer)
pool2, conv2 = EncoderBlock(neurons * 2, dropout_rate=0.1, name="Encoder2")(pool1)
pool3, conv3 = EncoderBlock(neurons * 4, dropout_rate=0.2, name="Encoder3")(pool2)
pool4, conv4 = EncoderBlock(neurons * 8, dropout_rate=0.2, name="Encoder4")(pool3)

bottleneck = EncoderBlock(neurons * 16, dropout_rate=0.3, pooling=False, name="BottleNeck")(pool4)

attention1 = AttentionGate(neurons * 8, use_bn=True, name="Attention1")([bottleneck, conv4])
decoder1 = DecoderBlock(neurons * 8, dropout_rate=0.2, name="Decoder1")([bottleneck, attention1])
attention2 = AttentionGate(neurons * 4, use_bn=True, name="Attention2")([decoder1, conv3])
decoder2 = DecoderBlock(neurons * 4, dropout_rate=0.2, name="Decoder2")([decoder1, attention2])
attention3 = AttentionGate(neurons * 2, use_bn=True, name="Attention3")([decoder2, conv2])
decoder3 = DecoderBlock(neurons * 2, dropout_rate=0.1, name="Decoder3")([decoder2, attention3])
attention4 = AttentionGate(neurons, use_bn=True, name="Attention4")([decoder3, conv1])
decoder4 = DecoderBlock(neurons, dropout_rate=0.1, name="Decoder4")([decoder3, attention4])

output_layer = Conv2D(1, kernel_size=1, activation='sigmoid', padding='same')(decoder4)

model = Model(inputs=[input_layer], outputs=[output_layer])

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

Figure 10. Attention U-Net Code

For this segmentation model, 10% of our data was allocated for validation. The model was trained with the remaining images. The "batch" size was kept at 8 to avoid overfitting due to unequal values of the model's dataset for each cancer type and low graphics processing unit memory. The model was run for a total of 51 epochs (17 + 17 + 17 + 17).

```
BATCH_SIZE = 8
VALID_SPLIT = 0.1
SPE = (int ((1 - VALID_SPLIT) * len(images)))//BATCH_SIZE


results = model.fit(
    images, masks,
    validation_split=VALID_SPLIT,
    epochs=17,
    steps_per_epoch=SPE,
    batch_size=BATCH_SIZE
)
```

Figure 11. Data Splitting Code Chunk

The predictions were given a value between 0 and 1, and the values ≥ 0.5 were set equal to 1 and the remaining values < 0.5 were set to 0. Thus, a mass was obtained as shown in Figure 12 with labels “Original Mask”, “Predicted Mask” and “Processed Mask” respectively.

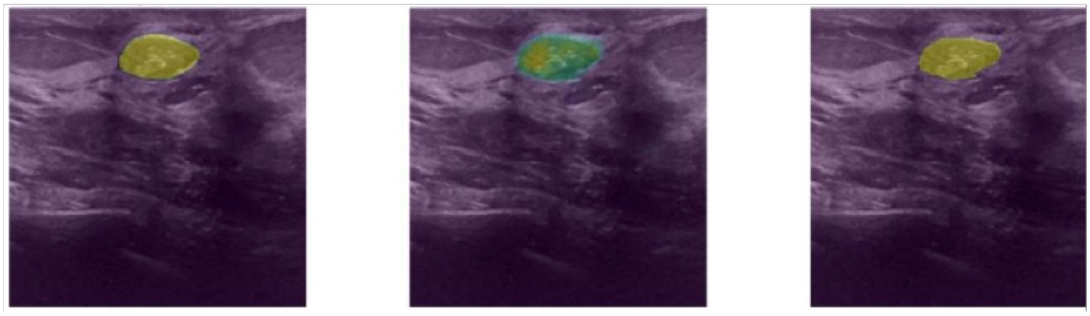

Figure 12. Mask Images, Original (left), Predicted (middle), Processed (right)

2.4 Classification algorithm

For classification, the pre-trained Resnet50 model was trained with superimposed ultrasound and segmented mass images.

2.4.1 *Data preprocessing*

Since the images in the dataset were separated into ‘ultrasound’ and ‘segmented mass’, the images were merged with the code in Figure 13 before training the model. Figure 14 shows one of the merged images.

```
def overlay(image_path, mask_path):

            image = Image.open(image_path)
            mask = Image.open(mask_path)
            mask = mask.convert(image.mode)

            if image.size != mask.size:
                image = image.resize(mask.size)

            overlayed = Image.blend(image, mask, alpha=0.5)

            label = os.path.basename(os.path.dirname(image_path))
            output_path = os.path.join(data_dir, label, os.path.basename(image_path))
            overlayed.save(output_path)

for i in range(len(image_paths)):
    image_path = image_paths[i]
    mask_path = mask_paths[i]
    overlay(image_path, mask_path)
```

Figure 13. Data Preprocessing Code Chunk for Classifcation Model

Figure 14. Merged Image

2.4.2 *Resnet50*

A summary of the Resnet50 model is shown in Figure 15. The model was run for 10 epochs with a batch size of 8. The dataset of 780 images was divided into 3 parts: benign, malignant, and normal, and 20% of the data was allocated for validation. The validation and training dataset extracted with the "image_dataset_from_directory" function of Keras was fitted to the Resnet50 model using the ".fit()" function as shown in Figure 16.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 2048)              23587712

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 512)               1049088

 dense_1 (Dense)             (None, 3)                 1539

=================================================================
Total params: 24,638,339
Trainable params: 1,050,627
Non-trainable params: 23,587,712
_________________________________________________________________
```

Figure 15. Resnet50 Summary

```
EPOCHS = 10
results = resnet_model.fit(
    train_ds,
    validation_data=val_ds,
    epochs = EPOCHS
)
```

Figure 16. Epoch Code Chunk

3 Results

This section presents the results of our study.

3.1 Image segmentation

Figure 17 shows the accuracy and loss rates of the image segmentation deep learning model for the last 17 epochs. This model showed better results on validation data that it had not seen before. There is a significant difference in the accuracy of the training set.

The model was not trained further to avoid overfitting due to the visible increase in training accuracy. The model achieved a prediction accuracy of 99.35%.

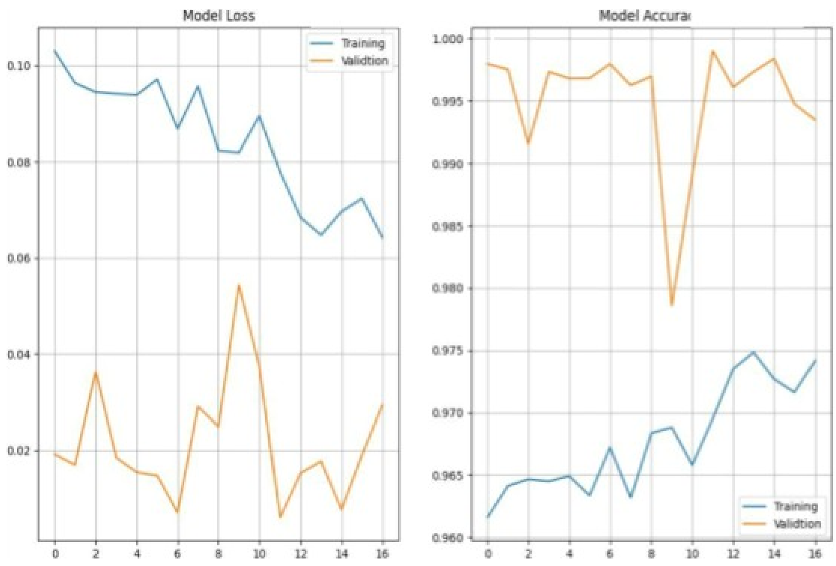

Figure 17. Model Loss and Accuracy Graphs

3.2 Classification model

Figure 18 shows a confusion matrix extracted from the validation dataset of the classification model using the pre-trained Resnet50 model. The model made the most errors in the presence of cancer masses. A cancer-free individual was correctly predicted to be cancer-free every time. Based on the validation data 97.43% of the predictions were correct. An example of a prediction is shown in Figure 12.

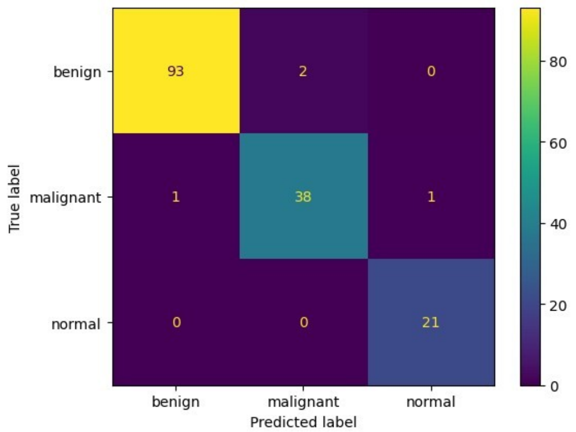

Figure 18. Confusion Matrix

4 Conclusion and discussion

The results show that our model has an accuracy rate of 99.35% for image segmentation and 97.43% for image classification. This is better than current deep learning models compared to the results of Kim et al. where the authors compared the accuracy of radiologists and their own deep learning models.

An important reason for this high accuracy may be the simplicity of the dataset we used. The variance between images in the dataset was low. The images in the dataset were generally simple cyst images with less artifacts than other types of images. For these reasons, the model needs to be trained with larger datasets containing other types of ultrasound images in order to be used with other datasets. The most important advantage of our model over current models that perform both segmentation and classification is that it blended 2 different models. Since other models use a single model, their data processing speeds are slightly faster, but their accuracy rates are usually lower due to the number of layers and model-specific limitations.

Although the segmentation and classification models developed in this project presented with very high accuracy rates, they are not without their limitations. As mentioned in the introduction, breast cancer is a very common disease today and can have serious consequences. Therefore, its diagnosis is a serious matter. While the segmentation model we have created is a tool that can help doctors in mass detection, it is not yet to be able to perform at a level that can provide a definitive diagnosis on its own.

However, with a much larger and more detailed dataset, that goes beyond classifying just benign and malignant cancer masses to including BI-RADS categories of masses, it is likely that this model will be able to detect masses that can even escape manual detection.

5 Recommendations

There is room for improvement in the software and data set of the model. Several integration methods for using the model in real life are described in this section.

5.1 Increasing processor power for the CNN using more layers

The model created in the project was trained and run on a home computer with RTX3060 and 16GB RAM. This process took a large amount of time due to the lack of processing power of the computer. If there was a computer with more processing power, the model could have run with a higher accuracy rate using CNN models that use more layers, such as ResNet101. The processing power can be increased by using a computer with more advanced features or machine learning platforms such as Azure over the cloud.

5.2 Making the software accessible *via* the Turkish National Health System (E-Nabız) for the use of radiologists and specialists

E-Pulse (E-Nabız) is a personal health record system used in Turkey, implemented by the Ministry of Health of the Republic of Turkey, where citizens can view and manage their personal health information individually on the Internet-based service and phone application.

Our model can be integrated into E-Pulse, making it accessible to both radiologists and patients independently. Ultrasound images can be analyzed by our model as soon as they are uploaded to E-Pulse, even before they are available to radiologists. This makes it easier for radiologists to analyze the images and, as mentioned in our article, reduces the margin of error in the diagnosis of the image. Even if the image in front of radiologists still contains a margin of error, it would still serve as a reference point for professionals.

6 References

1. What is Cancer? Diagnosis and Treatment Methods. (2017). Anadolu Sağlık. https://www.anadolusaglik.org/saglik-rehberi/kanser#:~:text=Kanser%2C%20v%C3%BCcudun%20herhangi%20organ%20ya,karaci%C4%9Fer%2C%20kolon%20ve%20meme%20kanseridir

2. All Cancers Fact Sheet [Fact sheet]. (2020). Global Cancer Observatory. https://gco.iarc.fr/today/data/factsheets/cancers/39-All-cancers-fact-sheet.pdf

3. Berg, W. A., Zhang, Z., Lehrer, D., Jong, R. A., Pisano, E. D., Barr, R. G., et al. (2012). Detection of breast cancer with addition of annual screening ultrasound or a single screening MRI to mammography in women with elevated breast cancer risk. Jama, 307(13), 1394-1404. https://doi.org/10.1001/jama.2012.388

4. Elfgen, C., Varga, Z., Reeve, K., Moskovszky, L., Bjelic-Radisic, V., Tausch, C., Güth, U. (2019). The impact of distinct triple-negative breast cancer subtypes on misdiagnosis and diagnostic delay. Breast cancer research and treatment, 177, 67-75. https://doi.org/10.1007/s10549-019-05298-6

5. Kim, H. E., Kim, H. H., Han, B. K., Kim, K. H., Han, K., Nam, et al. (2020). Changes in cancer detection and false-positive recall in mammography using artificial intelligence: A retrospective, multireader study. The Lancet Digital Health, 2(3), e138-e148. https://doi.org/10.1016/s2589-7500(20)30003-0

6. Aliyeva, K., Yilmaz, C. (2023). MEME MERKEZİ GÜNLÜK PRATİĞİNDE MAMOGRAFİ YAPAY ZEKA "3450 OLGUNUN DEĞERLENDİRİLMESİ" [Poster presentation]. 1st International 4th Turkish Breast Surgery Conference, Nevşehir, Türkiye.

7. Campanella, G., Hanna, M. G., Geneslaw, L., Miraflor, A., Werneck Krauss Silva, V., Busam, K. J., et al. (2019). Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. Nature medicine, 25(8), 1301-1309. https://doi.org/10.1038/s41591-019-0508-1

8. Haase, R., Fazeli, E., Legland, D., Doube, M., Culley, S., Belevich, I., Jokitalo, E., Schorb, M., Klemm, A., Tischer, C. (2022). A Hitchhiker's guide through the bio-image analysis software universe. FEBS Letters, 596, 2472-2485. https://doi.org/10.1002/1873-3468.14451

9. Chen, C., Zheng, S., Guo, L., Yang, X., Song, Y., Li, Z., et al. (2022). Identification of misdiagnosis by deep neural networks on a histopathologic review of breast cancer lymph node metastases. Scientific reports, 12(1), 13482. https://doi.org/10.1038/s41598-022-17606-

0

10. Wetstein, S.C., de Jong, V.M.T., Stathonikos, N., Opdam, M., Dackus, G. M. H. E., Pluim, J. P. W., et al. (2022). Deep learning-based breast cancer grading and survival analysis on whole-slide histopathology images. Scientific Reports, 12, 15102. https://doi.org/10.1038/s41598-022-19112-9

11. Omidiji, O. A., Campbell, P. C., Irurhe, N. K., Atalabi, O. M., Toyobo, O. O. (2017). Breast cancer screening in a resource poor country: Ultrasound versus mammography. Ghana medical journal, 51(1), 6–12. https://doi.org/10.4314/gmj.v51i1.2

12. Different Techniques for Cancer Detection. (2022). Health Travellers Worldwide. https://www.healthtravellersworldwide.com/cancer-detection-techniques/#:~:text=Some%20of%20the%20tried%20and,medicine%2C%20various%20blood%20tests%20and

13. Heywang-Köbrunner, S. H., Hacker, A., Sedlacek, S. (2011). Advantages and disadvantages of mammography screening. Breast Care, 6(3), 2–2. https://doi.org/10.1159/000329005

14. Sencha, A. N., Evseeva, E. V., Mogutov, M. S., Patrunov, Y. N. (2013). Breast Ultrasound. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-36502-7

15. Sickles, E. A., Filly, R. A., Callen, P. W. (1984). Benign breast lesions: ultrasound detection and diagnosis. Radiology, 151(2), 467-470. https://doi.org/10.1148/radiology.151.2.6709920

16. Davenport, T., Kalakota, R. (2019). The potential for artificial intelligence in healthcare. Future healthcare journal, 6(2), 94-98. https://doi.org/10.7861/futurehosp.6-2-94

17. Velichko, Y. (2021). Types and Applications of AI in Healthcare. PostIndustria. https://postindustria.com/types-and-applications-of-ai-in-healthcare/

18. Amit, Y., Felzenszwalb, P., Girshick, R. (2021). Object detection. In Computer Vision: A Reference Guide (pp. 875-883). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-63416-2_660

19. Li, Z., Dong, M., Wen, S., Hu, X., Zhou, P., Zeng, Z. (2019). CLU-CNNs: Object detection for medical images. Neurocomputing, 350, 53-59. https://doi.org/10.1016/j.neucom.2019.04.028

20. Cheng, H. D., Jiang, X. H., Sun, Y., Wang, J. (2001). Color image segmentation: advances

and prospects. Pattern recognition, 34(12), 2259-2281. https://doi.org/10.1016/S0031-3203(00)00149-7

21. Akar, E., Kara, S., Akdemir, H., Kiris, A. (2015). A MATLAB tool for an easy application and comparison of image denoising methods [Paper presentation]. 2015 Medical Technologies National Conference, Muğla, Türkiye. https://www.researchgate.net/publication/304286915_A_MATLAB_tool_for_an_easy_application_and_comparison_of_image_denoising_methods

22. Ghosh, P., Mitchell, M. (2006). Segmentation of medical images using a genetic algorithm. In Proceedings of the 8th annual conference on Genetic and evolutionary computation (pp. 1171-1178). https://doi.org/10.1145/1143997.1144183

23. Sharma, N., Aggarwal, L. M. (2010). Automated medical image segmentation techniques. Journal of medical physics, 35(1), 3–14. https://doi.org/10.4103/0971-6203.58777

24. Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D. (2022). Image Segmentation Using Deep Learning: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(7), 3523-3542. https://doi.org/10.1109/TPAMI.2021.3059968

25. LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521, 436–444. https://doi.org/10.1038/nature14539

26. Shen, D., Wu, G., Suk, H.-I. (2017). Deep learning in medical image analysis. Annual Review of Biomedical Engineering, 19(1), 221–248. https://doi.org/10.1146/annurev-bioeng-071516-044442

27. Dildar, M., Akram, S., Irfan, M., Khan, H. U., Ramzan, M., Mahmood, A. R., et al. (2021). Skin cancer detection: a review using deep learning techniques. International journal of environmental research and public health, 18(10), 5479. https://doi.org/10.3390/ijerph18105479

28. Mambou, S. J., Maresova, P., Krejcar, O., Selamat, A., Kuca, K. (2018). Breast Cancer Detection Using Infrared Thermal Imaging and a Deep Learning Model. Sensors, 18(9), 2799. https://doi.org/10.3390/s18092799

29. Baghaei, K. T., Payandeh, A., Fayyazsanavi, P., Rahimi, S., Chen, Z., Ramezani, S. B. (2022). Deep representation learning: Fundamentals, perspectives, applications, and open challenges. https://doi.org/10.48550/arXiv.2211.14732

30. Wu, G., Kim, M., Wang, Q., Munsell, B. C., Shen, D. (2016). Scalable High-Performance

Image Registration Framework by Unsupervised Deep Feature Representations Learning. IEEE transactions on bio-medical engineering, 63(7), 1505–1516. https://doi.org/10.1109/TBME.2015.2496253

31. Suk, H. I., Lee, S. W., Shen, D., Alzheimer's Disease Neuroimaging Initiative (2014). Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. NeuroImage, 101, 569–582. https://doi.org/10.1016/j.neuroimage.2014.06.077

32. Shin, H.-C., Roberts, K., Lu, L., Demner-Fushman, D., Yao, J., Summers, R. M. (2016). Learning to Read Chest X-Rays: Recurrent Neural Cascade Model for Automated Image Annotation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2497-2506). https://doi.org/10.1109/CVPR.2016.274

33. Özgür, S. N., Bozkurt Keser, S. (2021). Meme Kanseri Tümörlerinin Derin öğrenme Algoritmaları Ile Sınıflandırılması. Türk Doğa ve Fen Dergisi, 10(2), 212–222. https://doi.org/10.46810/tdfd.957618

34. Işın, A., Direkoğlu, C., Şah, M. (2016). Review of MRI-based brain tumor image segmentation using Deep Learning Methods. Procedia Computer Science, 102, 317–324. https://doi.org/10.1016/j.procs.2016.09.407

35. Zhu, W., Xie, L., Han, J., Guo, X. (2020). The Application of Deep Learning in Cancer Prognosis Prediction. Cancers, 12(3), 603. https://doi.org/10.3390/cancers12030603

36. von Chamier, L., Laine, R. F., Jukkala, J., Spahn, C., Krentzel, D., Nehme, E., et al. (2021). Democratising deep learning for microscopy with ZeroCostDL4Mic. Nature communications, 12(1), 2276. https://doi.org/10.1038/s41467-021-22518-0

37. van Tulder, G., de Bruijne, M. (2016). Combining Generative and Discriminative Representation Learning for Lung CT Analysis With Convolutional Restricted Boltzmann Machines. IEEE Transactions on Medical Imaging, 35(5), 1262-1272. https://doi.org/10.1109/TMI.2016.2526687

38. Kose, U., Alzubi, J. (2021). Deep learning for cancer diagnosis. Springer. https://doi.org/10.1007/978-981-15-6321-8

39. O'Shea, K., Nash, R. (2015). An Introduction to Convolutional Neural Networks. https://doi.org/10.48550/arXiv.1511.08458

40. Xi, X., Meng, X., Yang, L., Nie, X., Yang, G., Chen, et al. (2018). Automated segmentation of choroidal neovascularization in optical coherence tomography images using

multi-scale convolutional neural networks with structure prior. Multimedia Systems, 25(2), 95–102. https://doi.org/10.1007/s00530-017-0582-5

41. Huang, S., Lee, F., Miao, R., Si, Q., Lu, C., Chen, Q. (2020). A deep convolutional neural network architecture for interstitial lung disease pattern classification. Medical & biological engineering & computing, 58(4), 725–737. https://doi.org/10.1007/s11517-019-02111-w

42. Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., et al. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE transactions on medical imaging, 35(5), 1285–1298. https://doi.org/10.1109/TMI.2016.2528162

43. Huang, S. C., & Le, T. H. (Eds.). (2021). Principles and Labs for Deep Learning (pp. 27-55). Academic Press. https://doi.org/10.1016/B978-0-323-90198-7.00006-9

44. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., et al. (2018). Recent advances in convolutional neural networks. Pattern Recognition, 77, 354-377. https://doi.org/10.1016/j.patcog.2017.10.013

45. Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science(), vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28

46. Dumitrescu, D., Boiangiu, C. A. (2019). A study of image upsampling and downsampling filters. Computers, 8(2), 30. https://doi.org/10.3390/computers8020030

47. Siddique, N., Paheding, S., Elkin, C. P., Devabhaktuni, V. (2021). U-Net and its variants for medical image segmentation: A review of theory and applications. IEEE Access, 9, 82031-82057. https://doi.org/10.1109/ACCESS.2021.3086020

48. Ghaznavi, A., Saberioon, M., Brom, J., Itzerott, S. (2024). Comparative performance analysis of simple U-Net, residual attention U-Net, and VGG16-U-Net for inventory inland water bodies. Applied Computing and Geosciences, 21, 100150. https://doi.org/10.1016/j.acags.2023.100150

49. Leclerc, S., Smistad, E., Østvik, A., Cervenansky, F., Espinosa, F., Espeland, T., et al. (2020). Lu-net: a multi-task network to improve the robustness of segmentation of left ventriclular structures by deep learning in 2d echocardiography. https://doi.org/10.48550/arXiv.2004.02043

50. Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., et al. (2018). Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999. https://doi.org/10.48550/arXiv.1804.03999

51. Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., et al. (2019). Attention gated networks: Learning to leverage salient regions in medical images. Medical Image Analysis, 53, 197-207. https://doi.org/10.1016/j.media.2019.01.012.

52. Zhang, J., Jiang, Z., Dong, J., Hou, Y., Liu, B. (2020). Attention Gate ResU-Net for Automatic MRI Brain Tumor Segmentation. IEEE Access, 8, 58533-58545. https://doi.org/10.1109/ACCESS.2020.2983075

53. Kim, M., Go, K., Yun, S. Y. (2022). Neural processes with stochastic attention: Paying more attention to the context dataset. https://doi.org/10.48550/arXiv.2204.05449

54. Pinsky, M. A., Karlin, S. (2011). Introduction. In M. A. Pinsky & S. Karlin (Eds.), An Introduction to Stochastic Modeling (4th ed., pp. 1-45). Academic Press. https://doi.org/10.1016/B978-0-12-381416-6.00001-0

55. Yin, W., Ebert, S., Schütze, H. (2016). Attention-based convolutional neural network for machine comprehension. https://doi.org/10.48550/arXiv.1602.04341

56. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778). https://doi.org/10.1109/CVPR.2016.90.

57. Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. https://doi.org/10.48550/arXiv.1409.1556

58. Gao, P., Lu, J., Li, H., Mottaghi, R., Kembhavi, A. (2021). Container: Context aggregation network. https://doi.org/10.48550/arXiv.2106.01401

59. Wen, L., Li, X., Gao, L. (2019). A transfer convolutional neural network for fault diagnosis based on resnet-50. Neural Computing and Applications, 32(10), 6111–6124. https://doi.org/10.1007/s00521-019-04097-w

60. Ketkar, N. (2017). Introduction to keras. Deep learning with python: a hands-on introduction, 97-111. https://doi.org/10.1007/978-1-4842-2766-4

61. Manaswi, N. K. (2018). Understanding and working with Keras. Deep learning with

applications using Python: Chatbots and face, object, and speech recognition with TensorFlow and Keras, 31-43. Apress, Berkeley, CA, USA.

62. Gupta, D. (2023). Image segmentation keras: Implementation of segnet, fcn, unet, pspnet and other models in keras. https://doi.org/10.48550/arXiv.2307.13215

63. Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., et al. (2020). Array programming with NumPy. Nature, 585(7825), 357-362. https://doi.org/10.1038/s41586-020-2649-2

64. McKinney, W. (2012). pandas: powerful Python data analysis toolkit (Release 0.7.3). https://pandas.pydata.org/pandas-docs/version/0.7.3/pandas.pdf

65. Chen, D. Y. (2017). Pandas for everyone: Python data analysis. Pearson Education, Upper Saddle river, NJ, USA.

66. Al-Dhabyani, W., Gomaa, M., Khaled, H., Fahmy, A. (2020). Dataset of breast ultrasound images. Data in brief, 28, 104863. https://doi.org/10.1016/j.dib.2019.104863

67. Tran, T. T., Pham, V. T. (2022). Fully convolutional neural network with attention gate and fuzzy active contour model for skin lesion segmentation. Multimedia Tools and Applications, 81. https://doi.org/10.1007/s11042-022-12413-1

68. Rukundo, O. (2023). Effects of image size on Deep Learning. Electronics, 12(4), 985. https://doi.org/10.3390/electronics12040985