



**ÇUKUROVA UNIVERSITY
ENGINEERING AND ARCHITECTURE FACULTY
DEPARTMENT OF COMPUTER ENGINEERING**

GRADUATION THESIS

Cancer Detection from Radiological Images

By

2019555044 – BORAN KAYA

Advisor

Assistant Professor – MEHMET SARIGÜL

Month-Year

May 2024

ADANA

ÖZET

Bu tez çalışmasında, MR ve PET görüntülerine özel olarak geliştirilmiş bir tümör tespiti sistemi CNN modeli kullanılarak geliştirilmiştir. Sistem, hastaların göğüs bölgesinden alınan MR veya PET görüntülerini inceleyerek, sağlık personeline herhangi bir pozitif vakayı bildirmeyi veya olumsuz koşullardan dolayı sağlık çalışanının yanlış teşhis koyduğu durumlarda ikinci bir inceleme önermeyi amaçlamaktadır.

ABSTRACT

In this thesis study, an auxiliary tumor detection system specialized on MR and PET images of the chest area was developed using the CNN model, one of the artificial intelligence models. This system aims to examine MR or PET images taken from patients' chest regions and notify the healthcare personnel of any positively identified cases or suggest a second review of the images if the healthcare worker made a misdiagnosis due to any adverse conditions.

CONTENTS

Page Number

ÖZET	II
ABSTRACT	III
SYMBOLS AND ABBREVIATIONS USED	VI
LIST OF TABLES	VII
LIST OF FIGURES	VIII
1-INTRODUCTION	1
2- LITERATURE REVIEW	2
2.1 Basic Principles of MR and PET Images	2
2.2 Evolution of CNN Models for Medical Image Analysis	3
3- THEORETICAL BACKGROUND	5
3.1 Convolutional Layer	5
3.2 Activation Layer	7
3.3 Pooling Layer	8
3.3.1 Max Pooling	8
3.3.2 Average Pooling	9
3.4 Flattening Process	9
3.5 Fully Connected Layer	10
3.6 Output Layer	11
3.6.1 Output Layer for Classification	11
3.6.2 Output Layer for Regression	11
3.7 Batch Normalization Layer	12

CONTENTS

3.8 Dropout Layer	12
4- DATA COLLECTION and PREPROCESSING	13
5- MODEL DESIGN and APPLICATION	14
5.1 CNN Architecture Used and Reasons for Selection	14
5.2 Training Process and Parameter Settings	15
5.3 Training Process and Parameter Settings	15
5.4 Implementation Details of the Model	16
6- RESULTS and ANALYSIS	17
6.1 Introduction of Data Sets	17
6.2 Evaluation Metrics Definition and Calculations	17
6.3 Results	19
6.3.1 MR and PET Images	19
6.3.2 PET Images Only	25
6.3.3 MR Images Only	31
7- GENERAL EVALUATION	37
8- REFERENCES	38

SYMBOLS AND ABBREVIATIONS USED

MR (Magnetic Resonance Imaging): Magnetic Resonance Imaging (MRI) is a medical imaging technique used to visualize detailed internal structures. MRI provides high-resolution images of organs, tissues, and other internal body structures without using ionizing radiation.

PET (Positron Emission Tomography): Positron Emission Tomography (PET) is a nuclear medicine functional imaging technique that is used to observe metabolic processes in the body. PET scans are commonly used to detect cancer, monitor heart conditions, and evaluate brain disorders.

TP (True Positive): True Positive refers to the instances where the model correctly predicts the positive class. It is the number of positive instances that are correctly identified as positive by the model.

TN (True Negative): True Negative refers to the instances where the model correctly predicts the negative class. It is the number of negative instances that are correctly identified as negative by the model.

FP (False Positive): False Positive refers to the instances where the model incorrectly predicts the positive class. It is the number of negative instances that are incorrectly identified as positive by the model.

FN (False Negative): False Negative refers to the instances where the model incorrectly predicts the negative class. It is the number of positive instances that are incorrectly identified as negative by the model.

CNN (Convolutional Neural Network): Convolutional Neural Network (CNN) is a type of deep learning algorithm specifically designed for processing structured grid data such as images.

LIST OF TABLES

Figure 5.1 ResNet 34 Model in closed format	15
Figure 6.3.1.1 Train and Test Loss Values	19
Figure 6.3.1.2 Train and Test Accuracy Values	20
Figure 6.3.1.3 Confusion Matrix	21
Figure 6.3.1.4 Normalized Confusion Matrix	22
Figure 6.3.1.5 Performance Metrics	23
Figure 6.3.1.6 Roc-Auc Curve	24
Figure 6.3.2.1 Train and Test Loss Values	25
Figure 6.3.2.2 Train and Test Accuracy Values	26
Figure 6.3.2.3 Confusion Matrix	27
Figure 6.3.2.4 Normalized Confusion Matrix	28
Figure 6.3.2.5 Performance Metrics	29
Figure 6.3.2.6 Roc-Auc Curve	30
Figure 6.3.3.1 Train and Test Loss Values	31
Figure 6.3.3.2 Train and Test Accuracy Values	32
Figure 6.3.3.3 Confusion Matrix	33
Figure 6.3.3.4 Normalized Confusion Matrix	34
Figure 6.3.3.5 Performance Metrics	35
Figure 6.3.3.6 Roc-Auc Curve	36

LIST OF FIGURES

Figure 3.1 Basic Fully Connected CNN architecture	5
Figure 3.2 Convolution operator with kernel size 3 and stride 2	6
Figure 3.3 Stride Visualization	7
Figure 3.4 Padded Image	7
Figure 3.5 Activation Functions	8
Figure 3.6 Max Pooling Process Visualization	9
Figure 3.7 Average Pooling Process Visualization	9
Figure 3.8 Example of Flattening	10
Figure 3.9 Fully Connected Layer	10
Figure 3.10 Image Classification Using CNN	11

1- INTRODUCTION

With the rapid increase in the global population, the demand for the healthcare sector is growing more and more every day. Unfortunately, the current healthcare sector is increasingly unable to meet this demand adequately. The scarcity of doctors and personnel, the high number of patients, and the fatigue of healthcare workers due to continuous night shifts have led to insufficient dedication to each patient.

A prime example of this is the increased likelihood of missing diseases in MR and PET images due to the heavy and challenging working conditions of healthcare workers, leading to incorrect diagnoses that healthy individuals are sick. Consequently, the technologies used today come into play to assist these healthcare workers.

Artificial intelligence models used in image processing, such as CNN, have been one of the most important developments in medical imaging in recent years. They have led to groundbreaking advances in medicine thanks to their ability to automatically learn and extract complex patterns and features from medical images such as computed tomography (CT), magnetic resonance imaging (MRI), and x-rays. In addition, increased diagnostic accuracy, decreased error rates and faster processing capacity have enabled doctors to diagnose patients faster.

In this thesis study, an auxiliary tumor detection system specialized on MR and PET images of the chest area was developed using the CNN model, one of the artificial intelligence models. This system aims to examine MR or PET images taken from patients' chest regions and notify the healthcare personnel of any positively identified cases or suggest a second review of the images if the healthcare worker made a misdiagnosis due to any adverse conditions.

2- LITERATURE REVIEW

2.1 Basic Principles of MR and PET Images

Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET) are two important medical imaging methods used to create detailed images of the human body.

Magnetic Resonance Imaging (MRI) is a test that uses powerful magnets, radio waves and a computer to create detailed pictures of the body. MRI is often used to diagnose diseases or track response to treatment. Unlike x-rays and computed tomography (CT) scans, MRI does not use ionizing radiation that is harmful to health. MRI is a technique that uses a magnetic field and radio waves to create detailed images of organs and tissues in the body. Most MRI machines are large, tube-shaped magnets. When you go inside an MRI machine, the magnetic field temporarily rearranges the hydrogen atoms in your body. Radio waves cause these aligned atoms to produce very weak signals. These signals are used to create cross-sectional MRI images of the organ of interest, like slicing a loaf of bread.

Positron Emission Tomography (PET) is a medical imaging method used to study the functioning and metabolism of the human body. It provides information about tissue function by monitoring the distribution of molecules labeled with radioactive substances in the body. PET is used in the diagnosis and treatment of many diseases such as brain, heart, cancer and infection. Pet scan is one type of nuclear medicine imaging. Nuclear medicine imaging systems aim to detect changes in the body by using small amounts of radioactive material. Thanks to nuclear medicine, early diagnosis of the disease can be achieved by detecting changes in the cellular dimension.

2.2 Evolution of CNN Models for Medical Image Analysis

CNN models, used in medical imaging and many other fields, have evolved greatly over time and have begun to play an important role in many different medical application areas.

Head of CNN Applications

- In the late 1990s and early 2000s, CNNs began to be used in medical imaging, primarily for simple tasks. During this period, the models mostly worked on low-resolution images and contained a limited number of layers. For example, in 1998, Yann LeCun developed a successful model trained with the MNIST dataset capable of reading numbers on postal codes and bank checks.

Advanced Architecture and Deepening:

- With the success of models like AlexNet and VGGNet, the architecture of CNNs began to deepen and become more complex. Similar deep structures started to be used in medical imaging, yielding more accurate results.

Transfer Learning and Pre-Trained Models:

- Since the availability of large amounts of training data required for medical image analysis is not always possible, adapting models trained for general image recognition to medical images became a common method. Pre-trained models can quickly adapt to new tasks and perform better with less data.

Segmentation and Classification:

- CNNs began to be used to classify and segment specific structures in medical images. For example, they achieved high success rates in tasks such as cancerous tissue detection and organ segmentation.

3D Imaging and Multilayer Analysis:

- With the advancement of technologies in the field of healthcare, 3D images such as MRI and CT can be taken from patients in addition to 2D images. In addition, CNN architectures have been adjusted to work with 3D images.

Artificial Intelligence Supported Diagnostic Systems:

- CNNs form the basis of artificial intelligence-supported diagnostic systems and are used to support doctors' decision-making processes. These systems can speed up the diagnostic process and help provide more accurate results.

3- THEORETICAL BACKGROUND

The working principle of CNN essentially consists of a certain number of layers that perform certain operations. Images are passed through these layers respectively and the model is enabled to extract features during the training process. The basic layers used by CNN during its working principle are as follows:

- Convolutional Layer
- Activation Layer
- Pooling Layer
- Fully Connected Layer
- Output Layer

In addition to these basic layers, layers such as Batch Normalization and Dropout can be used according to need and scenario during training of the model.

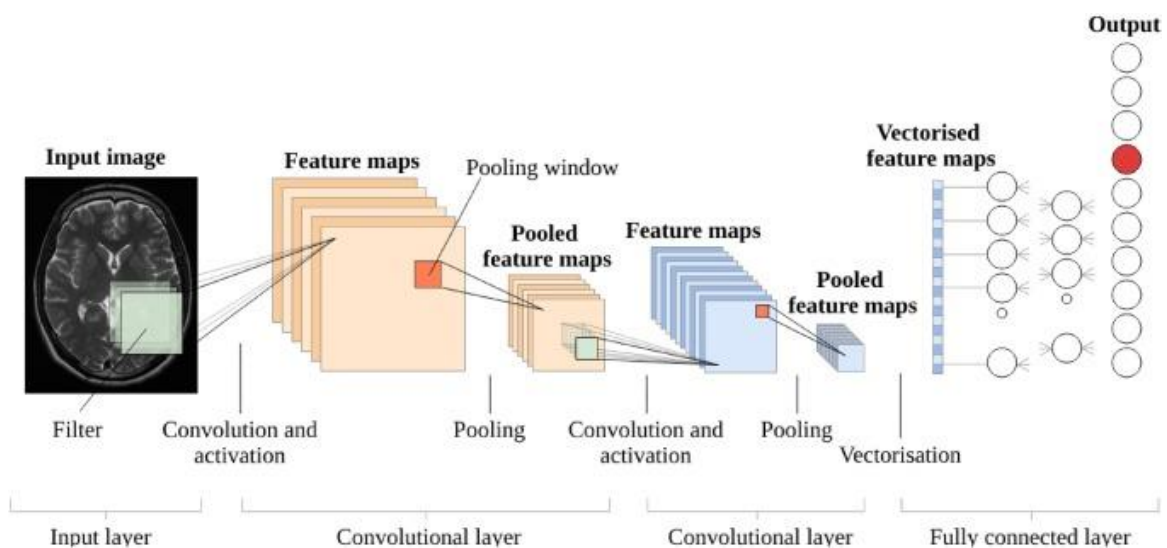


Figure 3.1 Basic Fully Connected CNN architecture

3.1 Convolutional Layer

Feature extraction is done on the images coming to this layer. This inference is achieved through a series of filters.

Incoming images come in matrix format so that they can be processed by the model. The values in the matrices include the pixel values of the image. Let's

examine an example of convolution.

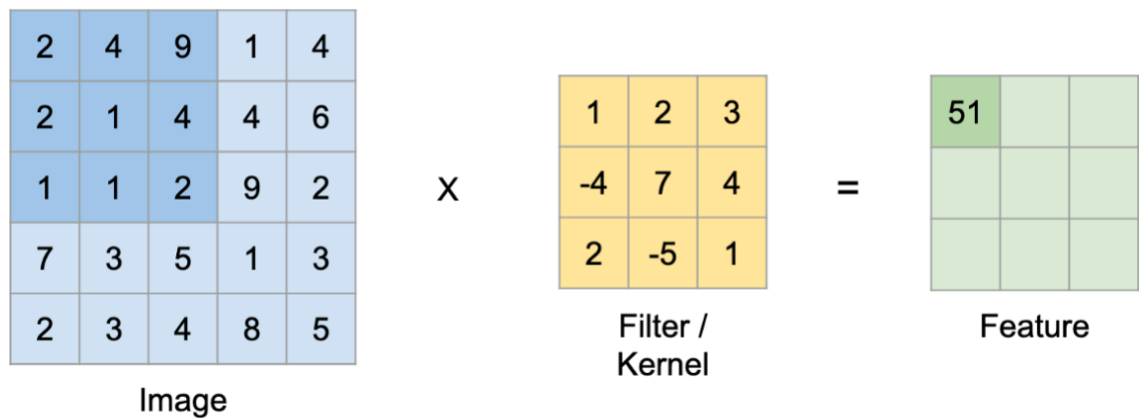


Figure 3.2 Convolution operator with kernel size 3 and stride 2

There is a 5x5 image matrix and a 3x3 filter matrix. In this case, the size of the new output matrix will be 3x3. The filter matrix is placed in the upper left corner of the image matrix, and the values in the intersecting indices of these two matrices are multiplied and added to obtain the first index of the output matrix. This process is repeated by shifting the filter matrix one step to the right on the image matrix. This process ends after the filter matrix has completely traveled through the image matrix. The same process is repeated by shifting the filter matrix one step to the right. When the filter matrix is moved completely over the image, all values in the output matrix are calculated. Thanks to this process, it ensures that the necessary features are determined in the convolution layer. Additionally, multiple convolution layers can be used in convolutional neural networks for multiple features.

Stride and Padding:

- **Stride:** Determines how far the filter is moved over the input image. With a larger stride value, less detail is extracted from the image, but a more computationally efficient activation map is created. With a smaller stride value, more detailed feature extraction is achieved. In the example below, the stride value is taken as 2.

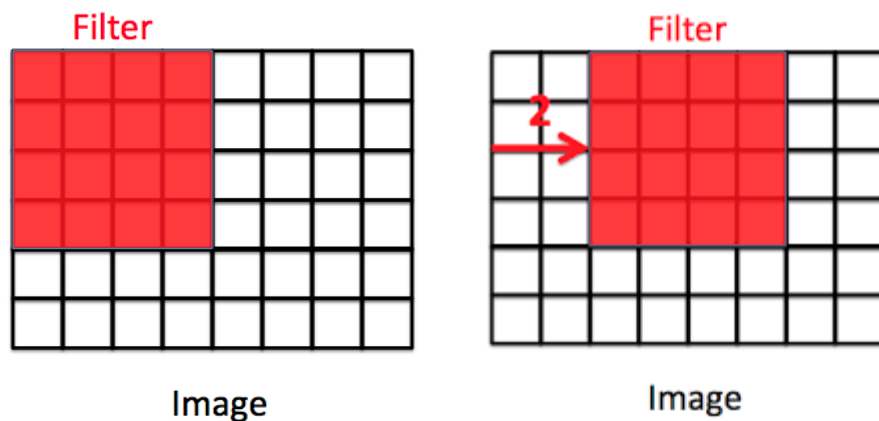


Figure 3.3 Stride Visualization

- **Padding:** By adding zeros around the input matrix or taking pixel values from neighboring indices, padding ensures that the filters fit perfectly at the edges of the input matrix. This is done to preserve the input dimensions after convolution.

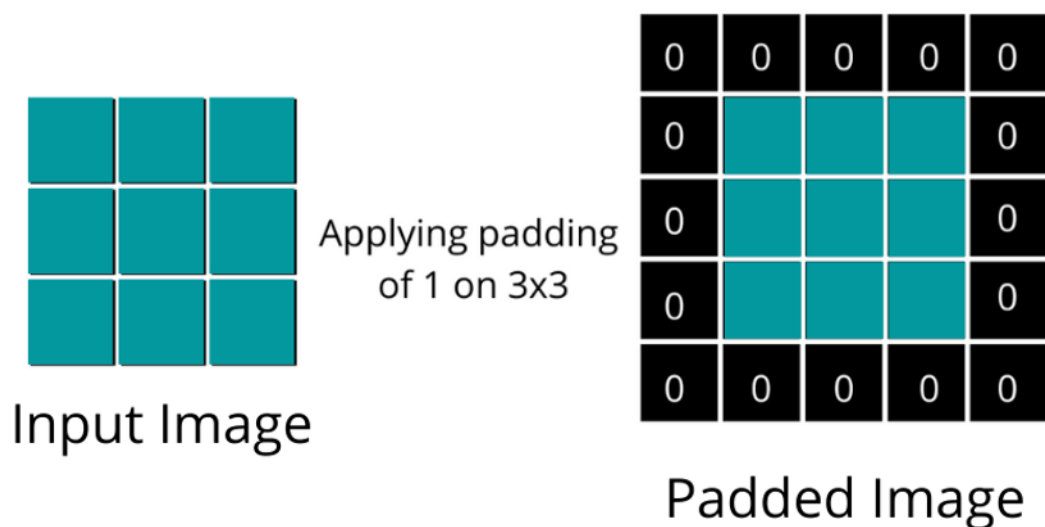


Figure 3.4 Padded Image

3.2 Activation Layer

Activation functions have an important role in artificial neural networks. Through these functions, it takes the total input signal calculated by each neuron in each layer of a neural network and applies a function to convert this signal into an output signal.

The main purpose of activation functions is to add nonlinear features to the network, thus allowing the network to learn more complex models. Although there are many activation functions in the literature such as ReLU, Sigmoid, Tanh and Softmax, ReLU is the most frequently used activation function in terms of efficiency factor.

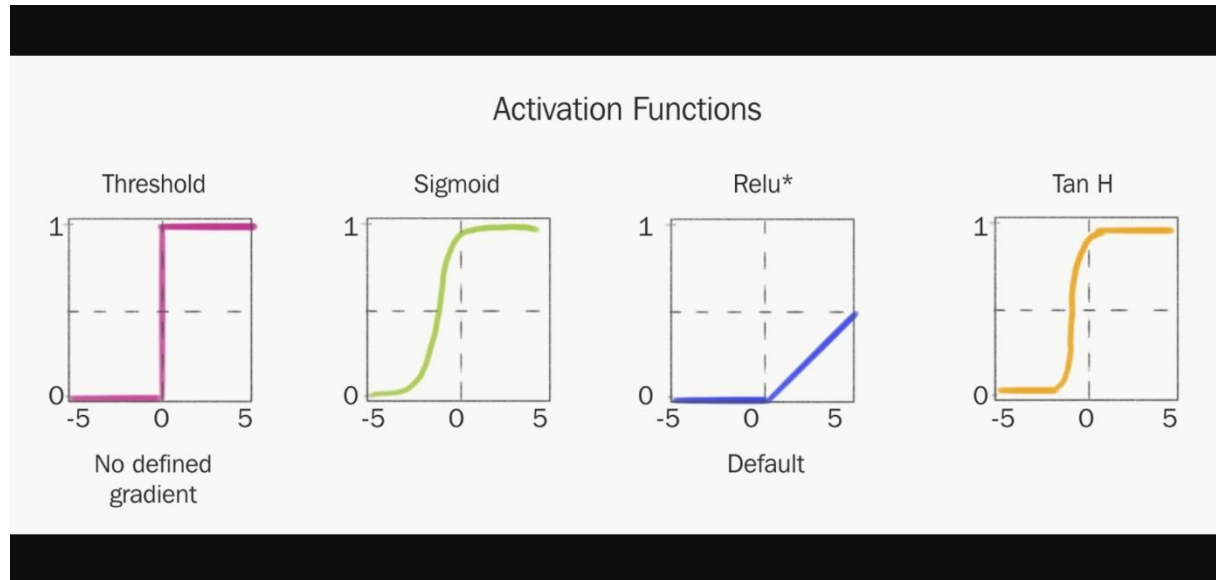


Figure 3.5 Activation Functions

3.3 Pooling Layer

In this layer, no learning occurs; instead, it is used to reduce the dimensions of the incoming images. The aim here is to reduce the computational burden of the training process and try to prevent overfitting. In this regard, there are two types of pooling.

3.3.1 Max Pooling

In this method, the maximum value among the pixel values at the intersections where the filter moves over the image is selected. These values are then transferred to the new matrix that will be formed. In this way, new images with reduced size are created. At the same time, by selecting the largest pixel value, the most prominent

features in the image are preserved.

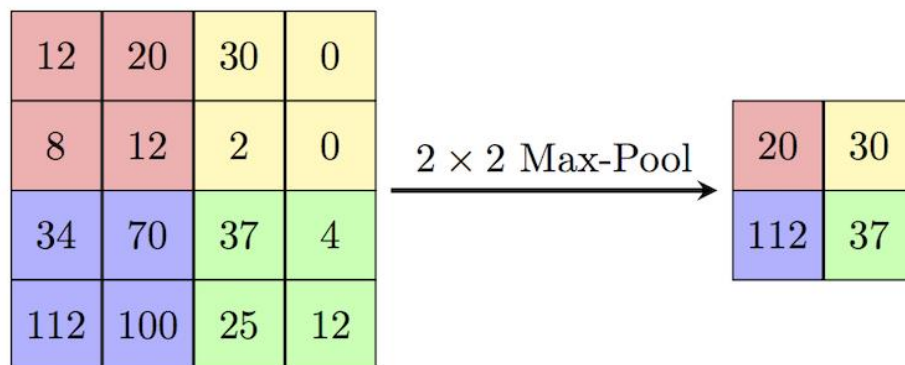


Figure 3.6 Max Pooling Process Visualization

3.3.2 Average Pooling

In this method, the pixel values at the intersections are averaged throughout the time the filter travels in the image. These average values are then transferred to the new matrix to be created. In this way, new images with reduced size are created. Although this method is less common than Max Pooling, it provides smoother feature extraction.

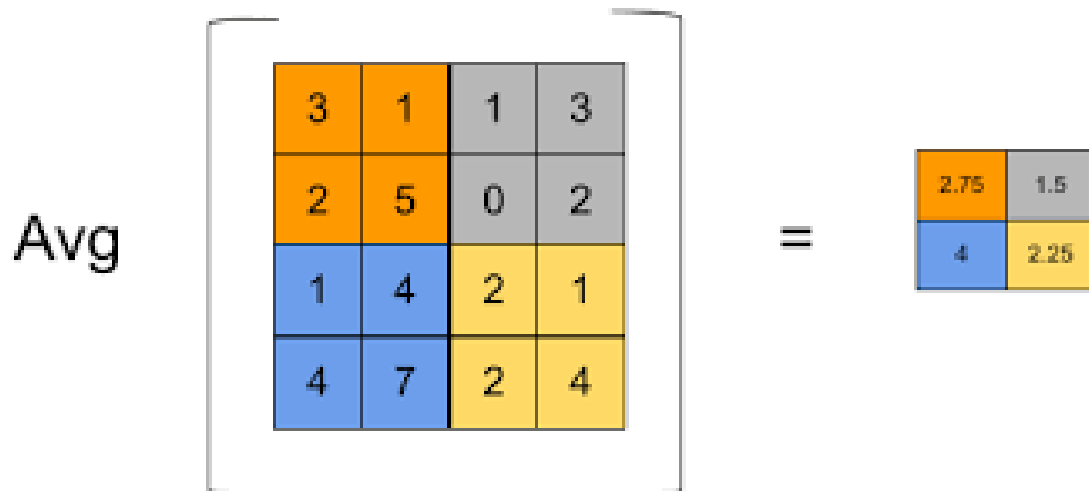


Figure 3.7 Average Pooling Process Visualization

3.4 Flattening Process

Before moving to the final layer of the model, we need to perform a flattening process on the matrix we have. The reason for this is that while all our operations up to this

point have been performed on matrices, we need to reduce our data to a one-dimensional vector to feed it into the model's final layer. This process is called flattening.

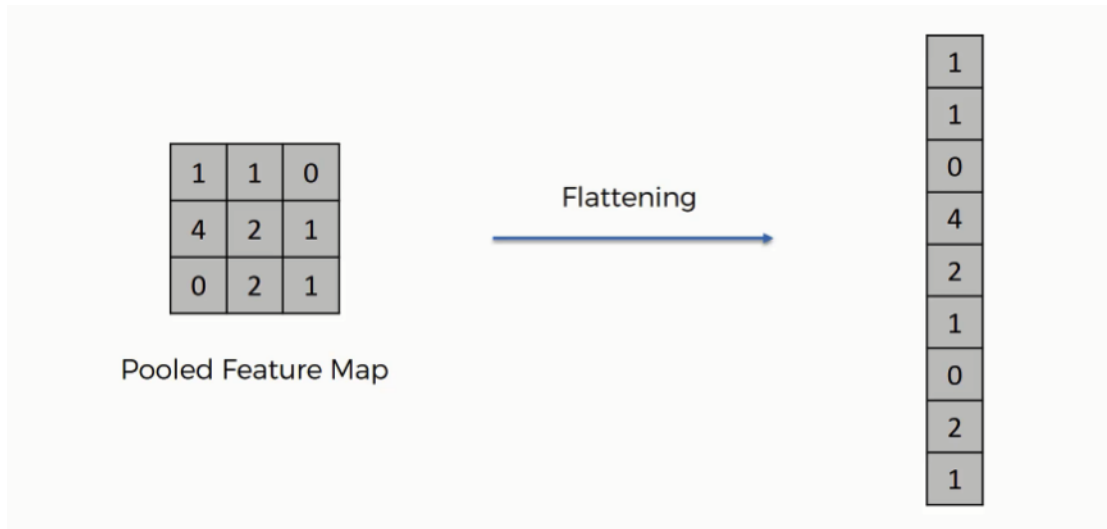


Figure 3.8 Example of Flattening

3.5 Fully Connected Layer

In this layer, the vectors converted to single plane in the flattening layer are taken and given as input to the artificial neural networks. In this way, the process for the relevant learning process begins.

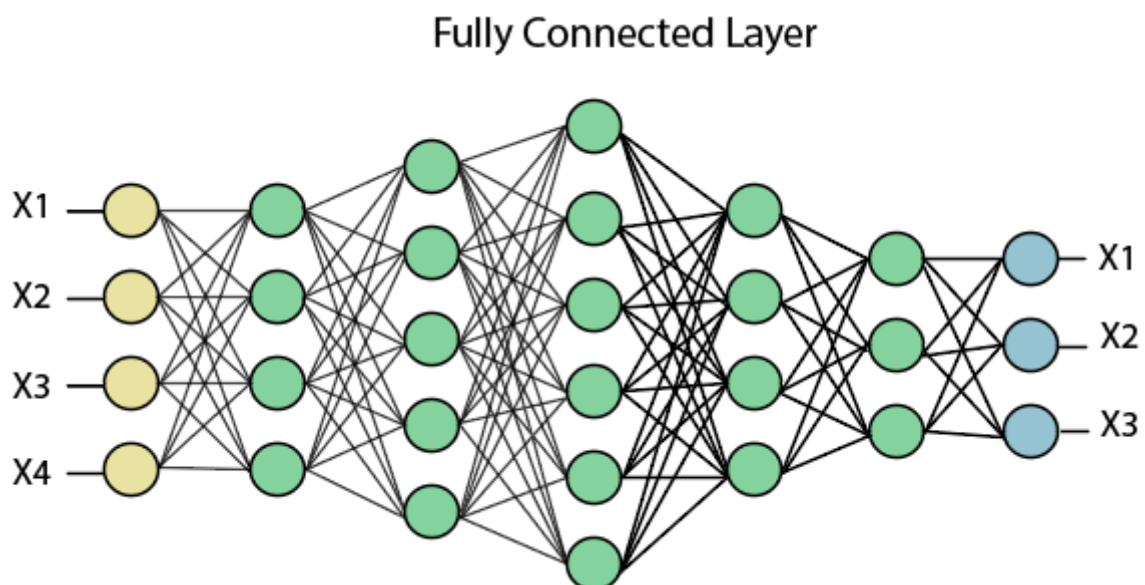


Figure 3.9 Fully Connected Layer

3.6 Output Layer

In this layer, final predictions are made using information from previous layers. There are multiple output layer types and uses.

3.6.1 Output Layer for Classification

Classification layers are mostly used to predict which class an image will belong to. For example, it is used in situations such as problems such as which species the animal in a picture belongs to or whether the patient is healthy according to an MRI image. If we need to make more than 2 classifications in the problem at hand, then multi-class classification is used. Here, the number of output neurons is equal to the number of classes. If we have 2 classes that we need to classify, binary classification is used and a neuron is usually used here.

3.6.2 Output Layer for Regression

In this case, usually one neuron is used, and this neuron produces a continuous value.

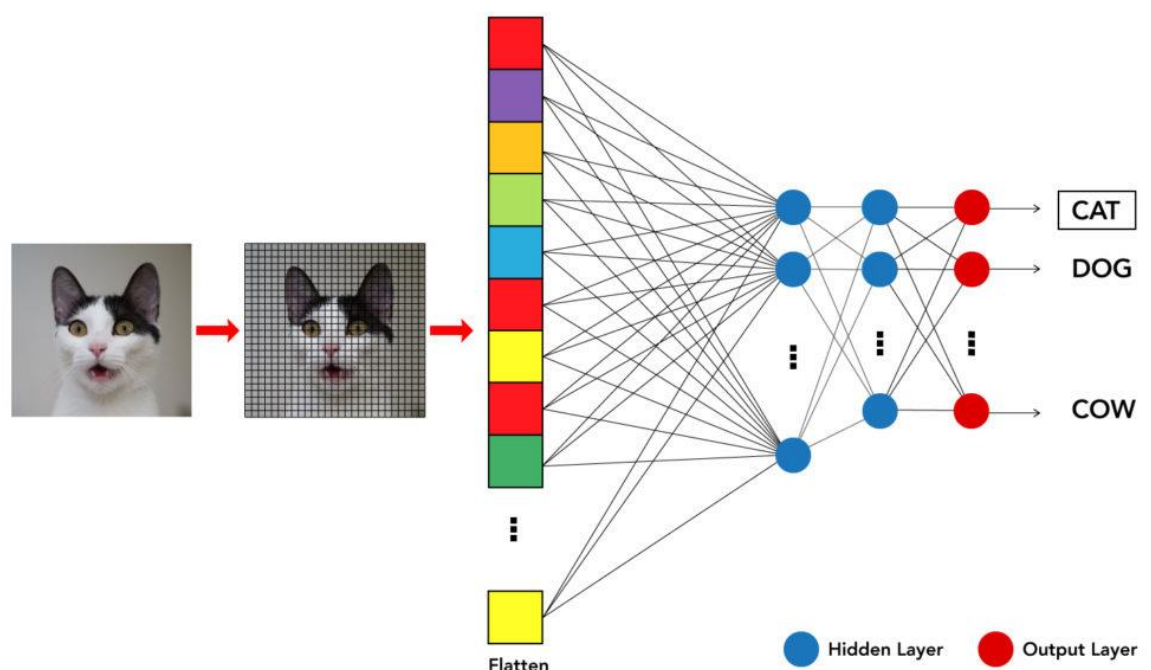


Figure 3.10 Image Classification Using CNN

3.7 Batch Normalization Layer

Normalization is a preprocessing technique used to standardize data. Batch Normalization is a normalization process between the layers of a neural network. Normalization is done in mini groups rather than the entire data. Learning is facilitated by allowing to accelerate training and use higher learning rates. It is usually located between the convolution layer and the activation layer.

3.8 Dropout Layer

Dropout randomly deactivates neurons during the training process, preventing the model from overfitting to the noise in the dataset and enhancing its generalization ability.

4- DATA COLLECTION and PREPROCESSING

We have three different datasets for training, each organized into folders for positive and negative cases. The first dataset consists of MR and PET images. To maintain a balanced distribution, only 60 positive patients were selected and used from all available positive cases, along with all 24 negative patients. The second data set contains only PET images, with all available positive and negative patients included: 112 positive and 47 negatives. Finally, the third dataset consists of only MR images, with 67 positive patients and 24 negative patients. All these data were provided by doctors at Adana City Training and Research Hospital.

Since the original form of the images was in DICOM format, they were converted to 224 x 224 JPEG format so that the data could be given to the model for training and testing purposes. The main reason for choosing this size is that this size is considered standard in many widely used Convolutional Neural Network (CNN) models, as well as the need to minimize computational costs when processing large data sets. The conversion to JPEG format was carried out automatically through a special function written in Python programming language, and the resulting images were stored in folders under the name of the patient number.

Subsequently, the images converted to JPEG format were split into training and test sets. For the datasets using only MR and only PET images, 60% were allocated for training and 40% for testing. For the combined MR and PET images, the split was 55% for training and 45% for testing.

The images obtained are subjected to normalization before the data is given to the model. For this process, pixel averages and standard deviations of the images in the training and test sets are calculated by using a function written in Python. After the calculation, images are resized using the 'transforms.Compose()' function, optionally converted to black-and-white format, and random horizontal and vertical rotation and randomly sized cropping operations are applied for data augmentation. After these operations, since CNNs must expect image pixel values within a certain range, this requirement is met by normalizing the pixel values using the 'transforms.Normalize()' function.

5- MODEL DESIGN and APPLICATION

5.1 CNN Architecture Used and Reasons for Selection

The CNN model used in this study is the ResNet 34 model, which was created using the PyTorch library and has proven itself in many educational competitions. ResNet 34 is a member of the Residual Network (ResNet) family. It is a model developed to solve the vanishing gradient problem, which is one of the frequently encountered problems in deep neural networks. This model is designed for image classification tasks and has a 34-layer convolutional neural network architecture. These layers mainly consist of convolutional layers, batch normalization layers, ReLU activation functions and pooling layers.

The most important feature of ResNet 34 is that it consists of blocks, each of which is called "residual block". The residual block consists of two or three convolution layers. Input values are added to the outputs of these layers via a skip connection. This allows the block's input to be collected before and after the block is processed using skip connections. In this way, it offers a safe solution to the gradient disappearance problem encountered in training deep networks. Layers generally have the following structure:

- The first layer is a convolution layer of size 7x7 with a stride value of 2, followed by a max pooling layer.
- There are 4 main sections (stages) that deepen in four different dimensions, and each contain a different number of remnant blocks. These sections consist of 3, 4, 6 and 3 residual blocks, respectively.
- Within each block, the size of the feature maps is reduced and deepened by performing 3x3 convolutions.
- At the end of the network, global average pooling is performed, and the features are reduced to a single vector. Finally, classification is done with a fully connected layer.

Another important reason for choosing the ResNet 34 model as the training model is that the number of data sets, we have is medium-sized. Thanks to this model, the

model will prevent problems such as overfitting or underfitting during training due to the amount of data we have.

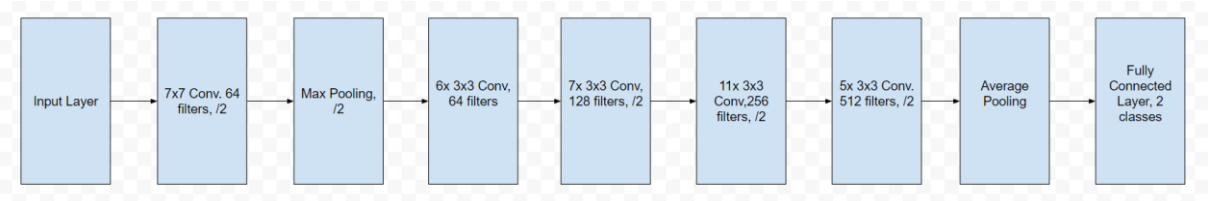


Figure 5.1 ResNet 34 Model in closed format

5.2 Training Process and Parameter Settings

In order to enable more efficient learning during the training of the model, we created a special data loader function that will load the images sequentially according to patient number, using the folder structure classified by patient number, instead of giving the patients' images collectively and randomly. This function organizes data paths and labels for different classes and makes them compatible with PyTorch's DataLoader.

5.3 Training Process and Parameter Settings

Since the model will make a class prediction as "Positive" or "Negative" as a result of training, the error function we will use during the training of the model is 'CrossEntropyLoss', which is used in classification problems.

Among the optimization algorithms, "Adam (Adaptive Moment Estimation)" was preferred. Adam uses adaptive learning rates for each parameter. At the core of the algorithm are both first moment (mean) and second moment (variance) estimates, allowing the learning rate of each parameter to be dynamically adjusted based on historical knowledge of the gradient.

Adam is a successful optimization algorithm that can be easily used in modern deep learning libraries in the literature. Sometimes when using this algorithm, it may become unstable due to excessive learning rate adjustments, especially when approaching the minimum.

Due to this situation, the model training process was optimized by using a dynamic learning rate and the overlearning problem was tried to be avoided.

5.4 Implementation Details of the Model

Before training the model, empty arrays were created to hold the loss values and accuracy values of the train and test data.

After determining how many epochs the model would run, the model was put into training mode with the “`model.train()`” code for each epoch cycle. At this stage, images and tags of images are moved to the appropriate device. If there is "CUDA" in the environment where the training will take place, the video card is used for training, if not, "CPU" is used. In the next step, the optimizer gradients are reset. Outputs are taken from the model and loss values are calculated. These calculated values are propagated back and the optimizer parameters are updated. The total loss and correct predictions are then recorded.

After the training process is completed, the '`model.eval()`' command is used to move on to the evaluation phase. At this stage, gradient calculations are stopped and using the test data loader created at the beginning of the model, images and labels are processed for each mini-batch we have determined, and loss and prediction values are recorded. Total test loss and correct predictions are recorded. The final test loss and accuracy are calculated and added to the relevant lists. At this stage, empty directories were created to calculate the confusion matrices that will be used to evaluate the success of the model in the later stages of the training.

In the last stage, values such as train loss, train accuracy, test loss, test accuracy are printed on the screen at the end of each epoch.

At the end of each cycle, the dynamic learning rate timer (“`scheduler`”) is updated according to the test loss. In this way, it is tried to prevent the model from overlearning.

6- RESULTS and ANALYSIS

6.1 Introduction of Data Sets

We will discuss our analysis and results separately for 3 different data groups. These are respectively:

- MR and PET Images
- PET Images Only
- MR Images Only

6.2 Evaluation Metrics Definition and Calculations

Definitions of metrics used to evaluate the success of models during model reviews and information on how they are calculated:

Accuracy

Definition: The proportion of correctly predicted instances out of the total instances.

Calculation: $Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)}$

Precision

Definition: The proportion of true positive instances out of the instances predicted as positive.

Calculation: $Precision = \frac{TP}{(TP+FP)}$

Recall (Sensitivity)

Definition: The proportion of true positive instances out of the actual positive instances.

Calculation: $Recall = \frac{TP}{(TP+FN)}$

F1- Score

Definition: The harmonic means of precision and recall.

Calculation: $F1\ Score = 2 * \frac{Precision * Recall}{(Precision + Recall)}$

Specificity

Definition: The proportion of true negative instances out of the actual negative instances.

Calculation: $Specificity = \frac{TN}{(TN + FP)}$

Positive Predictive Value (PPV)

Definition: The proportion of true positive instances out of the instances predicted as positive (same as precision).

Calculation: $PPV = \frac{TP}{(TP + FP)}$

False Positive Rate (FPR)

Definition: It is the rate at which true negatives are predicted as false positives.

Calculation: $FPR = \frac{FP}{(FP + TN)}$

ROC-AUC

ROC-AUC metric is an important metric frequently used in classification models. This metric indicates how well the model can distinguish positive and negative classes. ROC (Receiver Operating Characteristic) curve and AUC (Area Under Curve) visually represent the relationship between the model's true positive rate (TPR), also known as Recall, and false positive rate (FPR) and summarize the model's performance in a single value.

The Roc curve is essentially a graph obtained by showing TPR and FPR values at different threshold values in the classification model. AUC represents the area under the ROC curve and thus summarizes the overall performance of the model with a single value. Although the AUC value varies between 1 and 0; It correctly classifies all positive and negative examples as it approaches one. As it approaches zero, it indicates that the model performs poorly and identifies positive examples as negative and negative examples as positive.

6.3 Results

6.3.1 MR and PET Images

In this evaluation, the MR and PET images we had were combined and used. While we had 67 patients with positive labels in the original data folder; The number of patients with a negative label is 27. Due to this situation, we used 35 patients with positive labels and 15 patients with negative labels in the training cycle of the model to bring the data to a balanced distribution. We used 25 positive patients and 9 negative patients for the testing cycle.

Train and Test Loss Values

While it was observed that the loss values in the train cycle decreased at the end of each epoch, when it came to the test cycle, an increase in the test loss values began to be observed after the 2nd cycle. This tells us that we may have encountered overfitting after the second cycle.

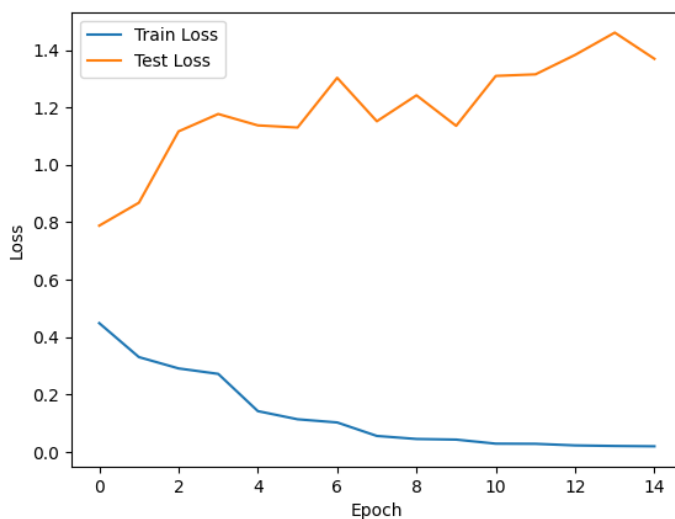


Figure 6.3.1.1 Train and Test Loss Values

Train and Test Accuracy Values

In the train cycle, accuracy values increased regularly at the end of each epoch. The same situation was observed in the test cycle up to a certain point. However, test

accuracy remained constant at a certain point and was observed at these points in the remaining training cycles.

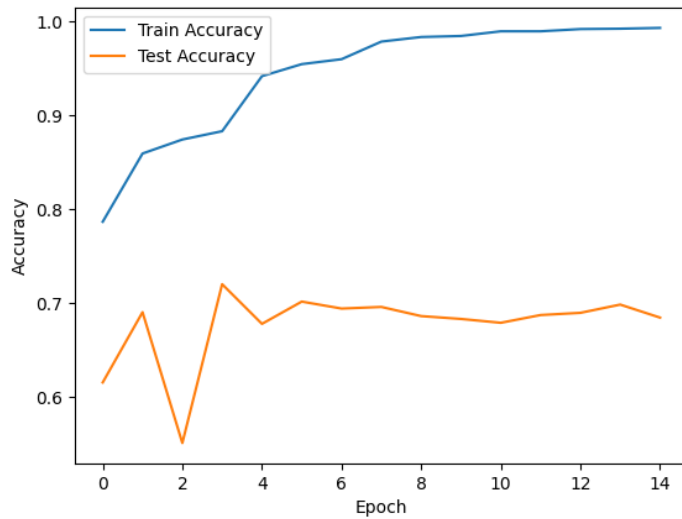


Figure 6.3.1.2 Train and Test Accuracy Values

Confusion Matrix

Confusion matrix results of the model:

True Positive (TP): 10,183

True Negative (TN): 1,807

False Positive (FP): 3,129

False Negative (FN): 2,398

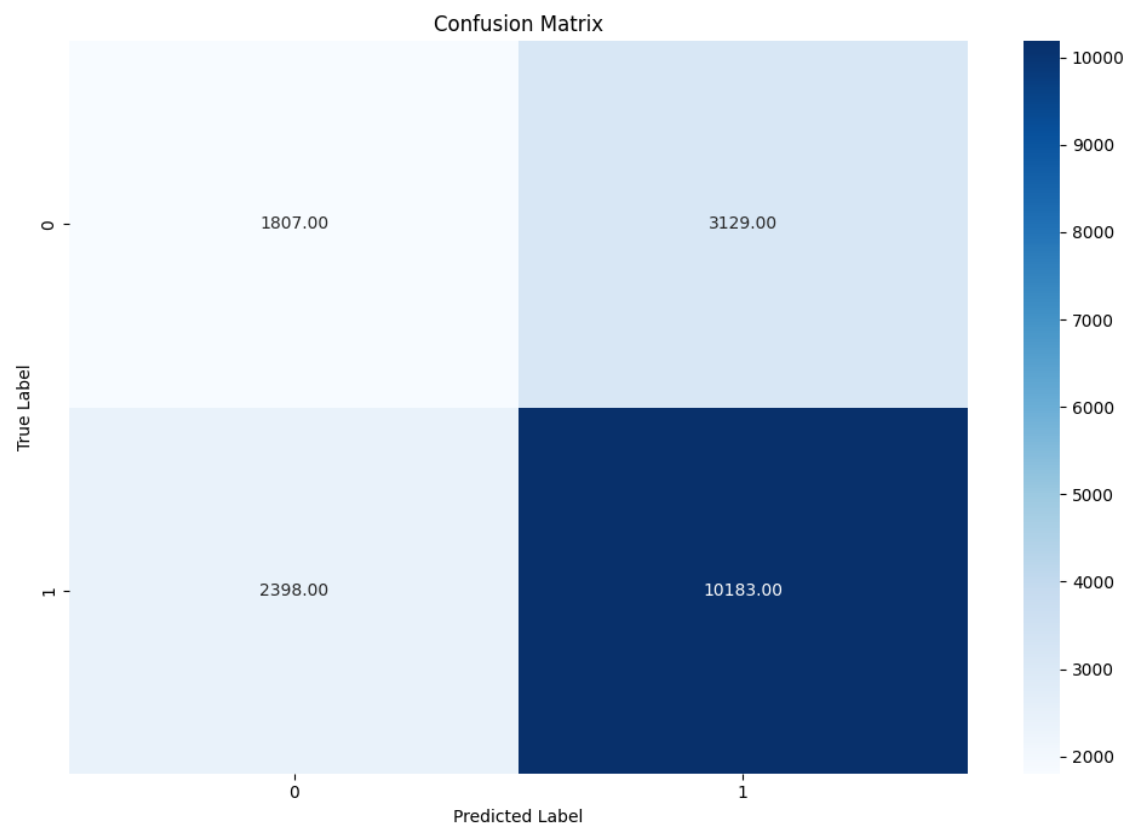


Figure 6.3.1.3 Confusion Matrix

Confusion matrix ratios of the model:

True Positive (TP): 0.81

True Negative (TN): 0.37

False Positive (FP): 0,63

False Negative (FN): 0,19

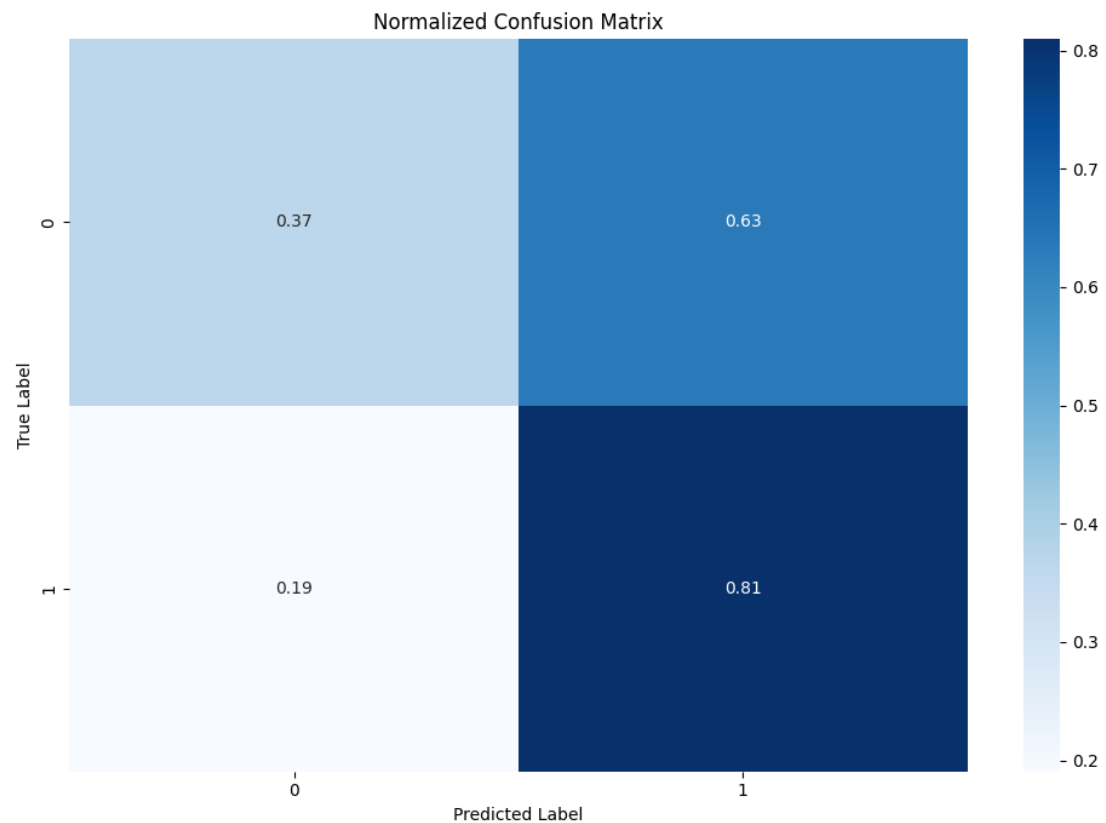


Figure 6.3.1.4 Normalized Confusion Matrix

Performance Metrics

If we examine the performance metrics of the model.

- Accuracy value is 68%, indicating that only 68% of the samples in the data set are classified correctly.
- The precision value is 76%, which indicates that 76% of the samples predicted by the model to be positive are positive.
- Recall value is 81%, indicating that the model correctly detected 81% of true positive samples
- The F1 score value is 79%, indicating that both false positives and false negatives are minimized.
- The specificity value is 37%, indicating that only 37% of the examples in the model that should be negative are correctly classified as negative. The biggest reason for

this is that the number of patients and images belonging to the negative class is low in the train and test sets we have.

- The Positive Predict Value (PPV) value is 76%, indicating that 76% of the samples predicted as positive were positive.

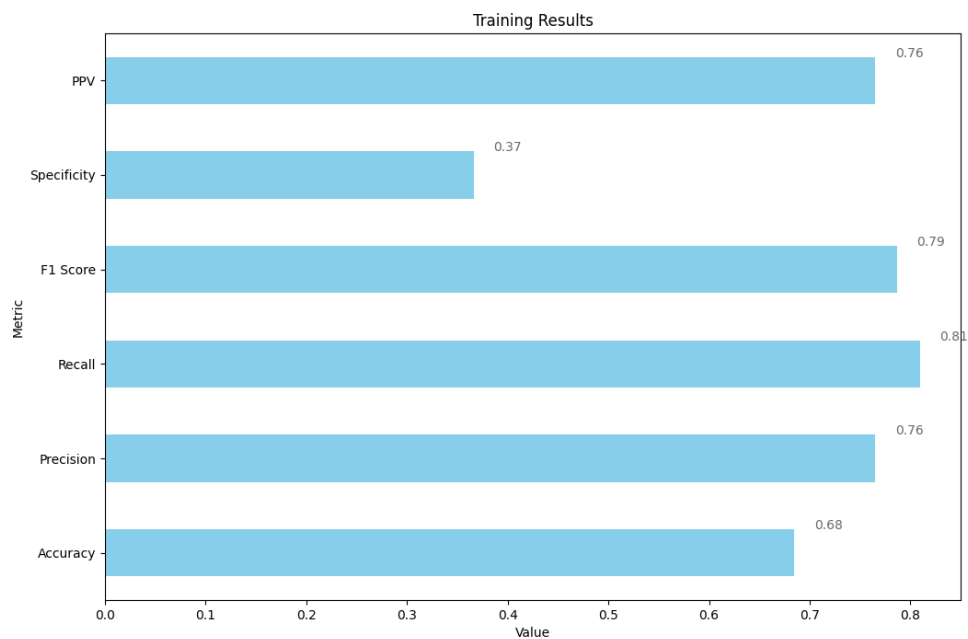


Figure 6.3.1.5 Performance Metrics

Roc-Auc Value

An AUC Value of 0.65 indicates that the model performs reasonably well, but not enough to be considered a high-performance model.

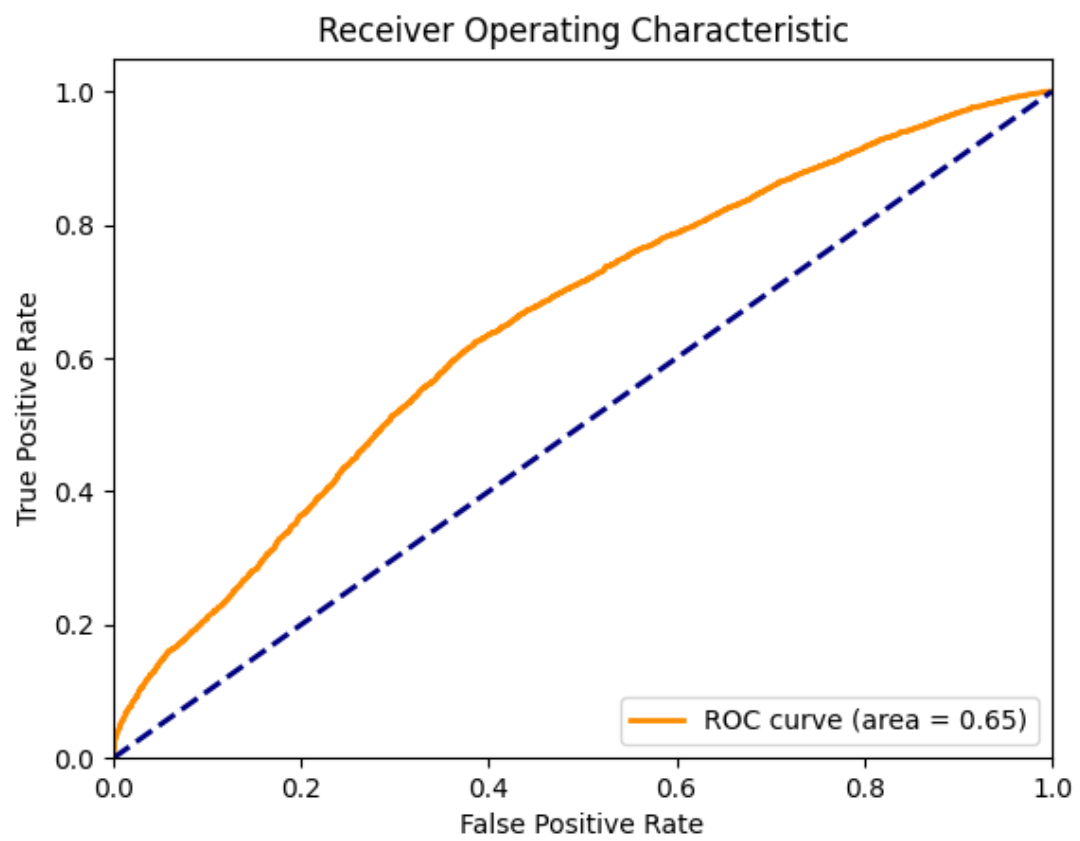


Figure 6.3.1.6 Roc-Auc Curve

6.3.2 PET Images Only

In this evaluation, the PET images we had were used. While we had 112 patients with positive labels in the original data folder; The number of patients with a negative label is 47.

Train and Test Loss Values

While it was observed that the loss values in the train cycle decreased at the end of each epoch, when it came to the test cycle, an increase in the test loss values began to be observed after the 3rd cycle. This tells us that we may have encountered overlearning after the 3rd cycle.

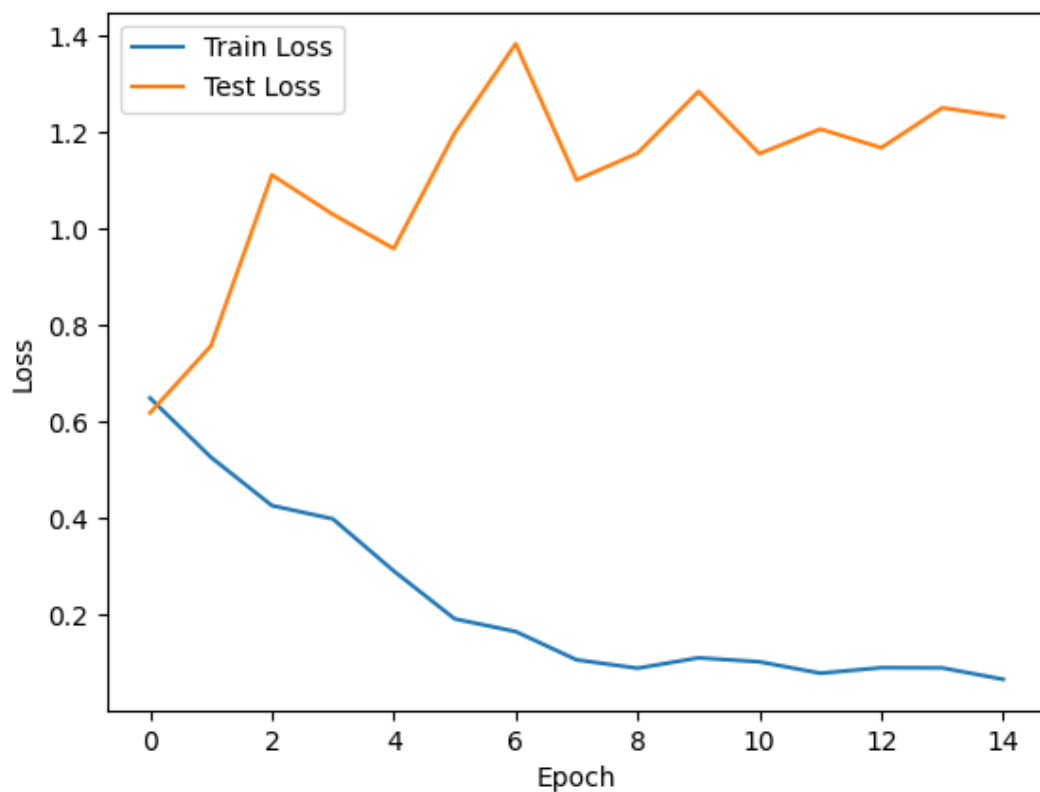


Figure 6.3.2.1 Train and Test Loss Values

Train and Test Accuracy Values

In the train cycle, accuracy values increased regularly at the end of each epoch. The same situation was observed in the Test cycle up to a certain point. However, test accuracy remained constant at a certain point and was observed at these points in the remaining training cycles.

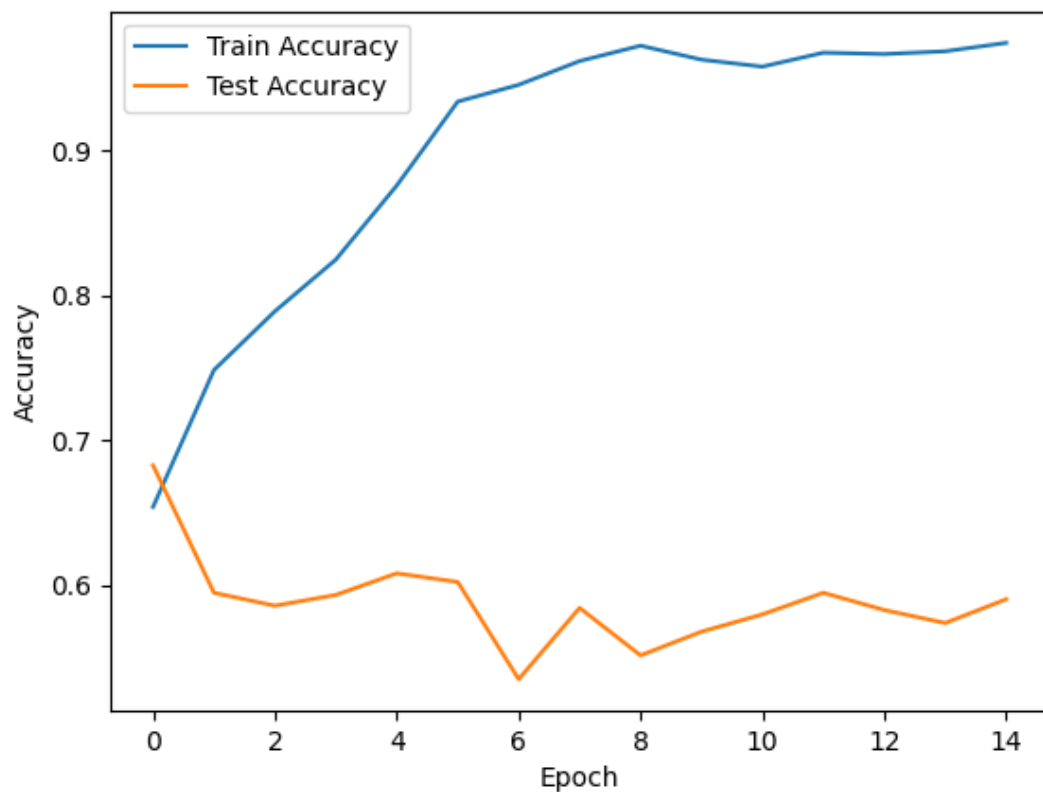


Figure 6.3.2.2 Train and Test Accuracy Values

Confusion Matrix

Confusion matrix results of the model:

True Positive (TP): 311

True Negative (TN): 85

False Positive (FP): 133

False Negative (FN): 142

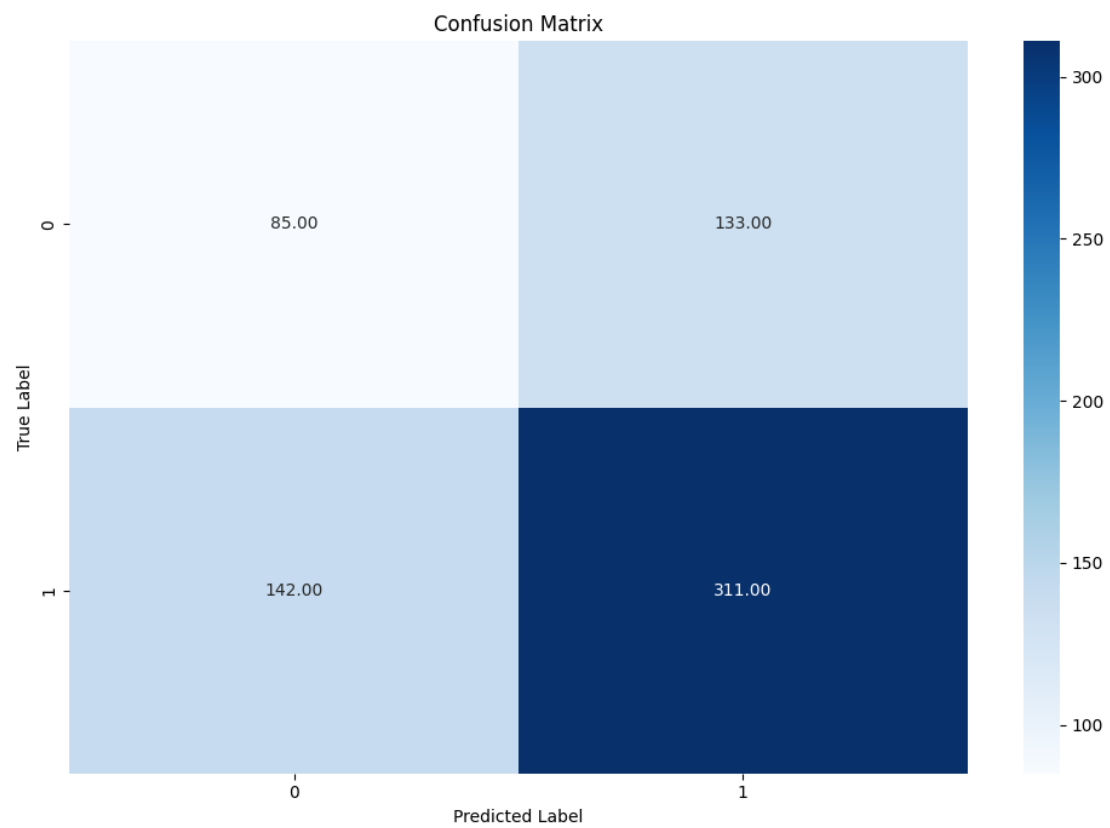


Figure 6.3.2.3 Confusion Matrix

Confusion matrix ratios of the model:

True Positive (TP): 0.69

True Negative (TN): 0.39

False Positive (FP): 0,61

False Negative (FN): 0,31

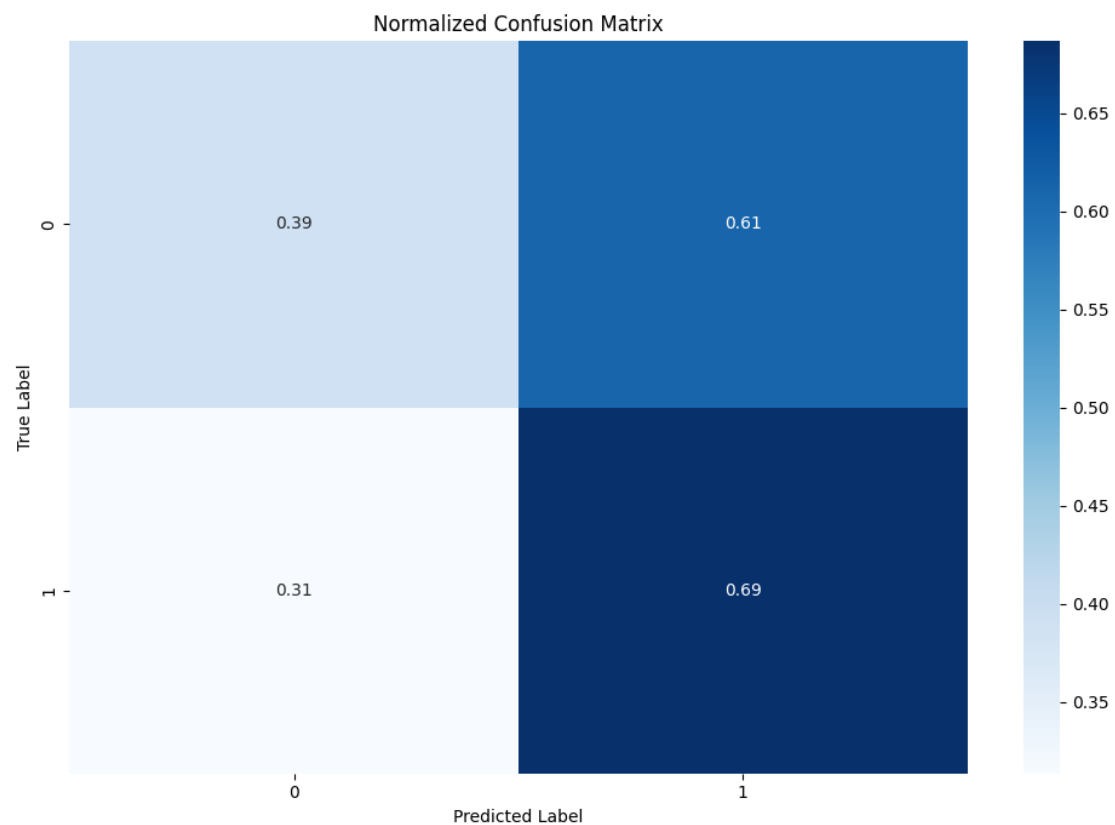


Figure 6.3.2.4 Normalized Confusion Matrix

Performance Metrics

If we examine the performance metrics of the model;

- Accuracy value is 59%, indicating that only 59% of the samples in the data set are classified correctly.
- The precision value is 70%, which indicates that 70% of the samples predicted as positive by the model are actually positive.
- Recall value is 69%, indicating that the model correctly detected 69% of true positive samples
- The F1 score value is 69%, indicating that both false positives and false negatives are minimized.

- The specificity value is 39%, indicating that only 39% of the examples in the model that should be negative are correctly classified as negative. The biggest reason for this is that the number of patients and images belonging to the negative class is low in the train and test sets we have.
- The Positive Predict Value (PPV) value is 70%, indicating that 70% of the samples predicted as positive are actually positive.

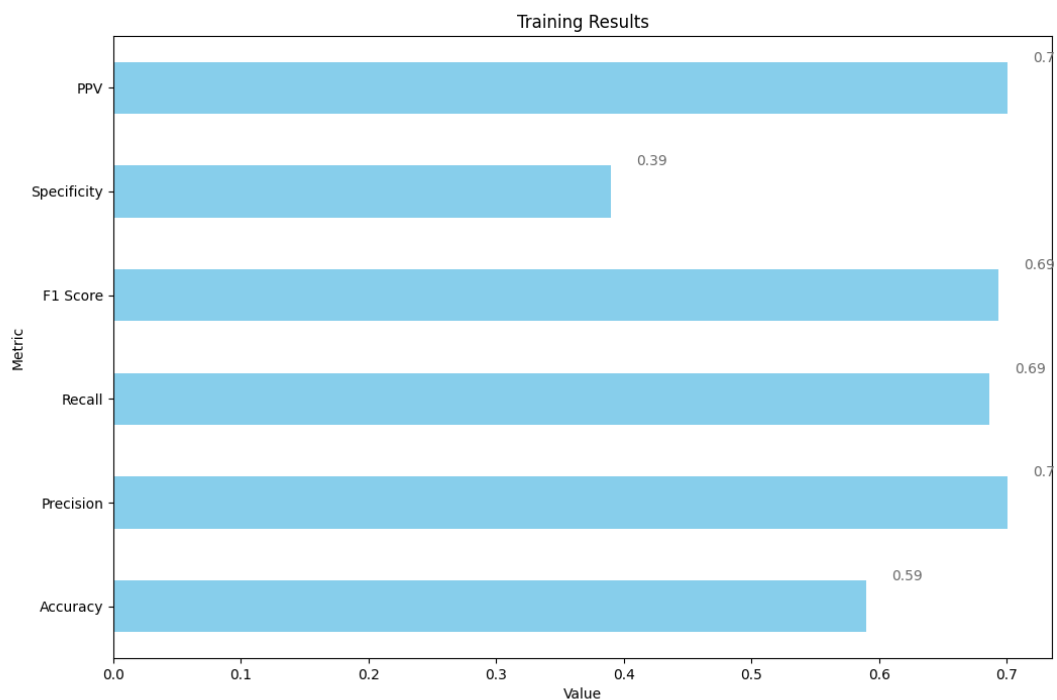


Figure 6.3.2.5 Performance Metrics

ROC-AUC Value

The AUC value is 0.57, indicating that the model is slightly better at distinguishing between positive and negative classes. The value of 0.57 indicates that the model performs better than random guessing in some cases, but it still does not make a reliable classification in general.

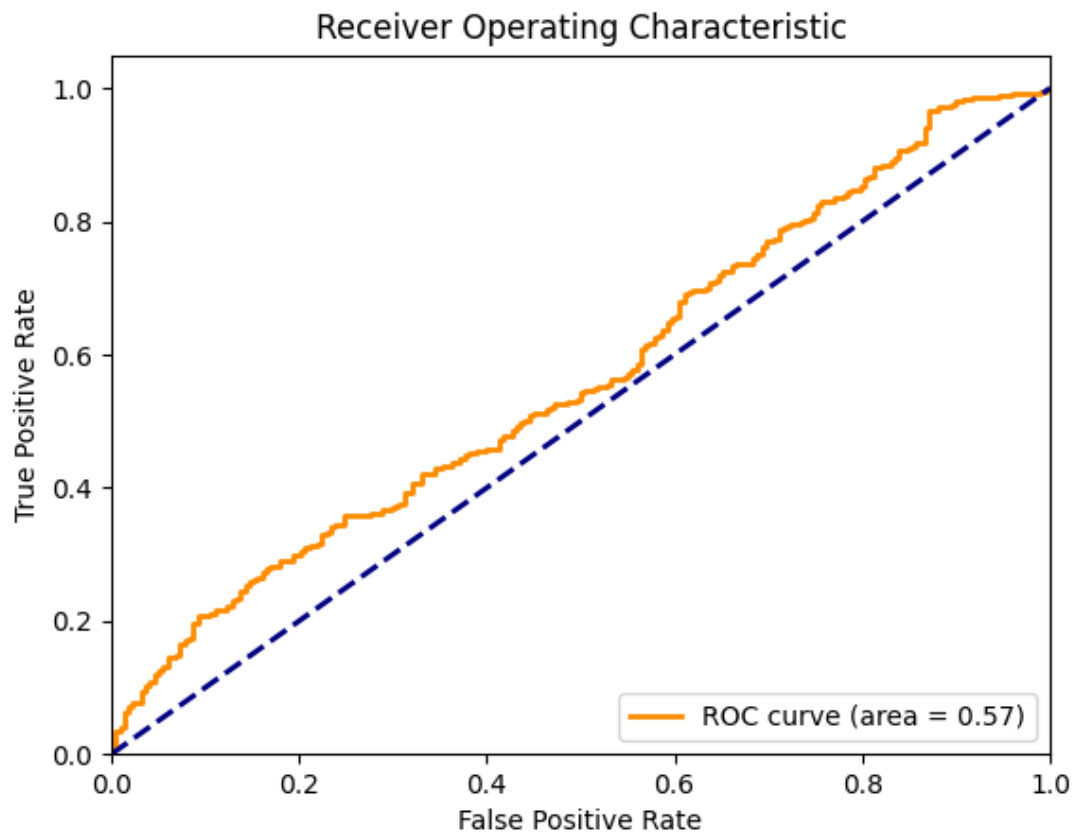


Figure 6.3.2.6 Roc-Auc Curve

6.3.3 MR Images Only

In this evaluation, the PET images we had were used. While we had 67 patients with positive labels in the original data folder; The number of patients with a negative label is 24.

Train and Test Loss Values

While it was observed that the loss values in the train cycle decreased at the end of each epoch, when it came to the test cycle, an increase in the test loss values began to be observed after the 3rd cycle. This tells us that we may have encountered overlearning after the 3rd cycle.

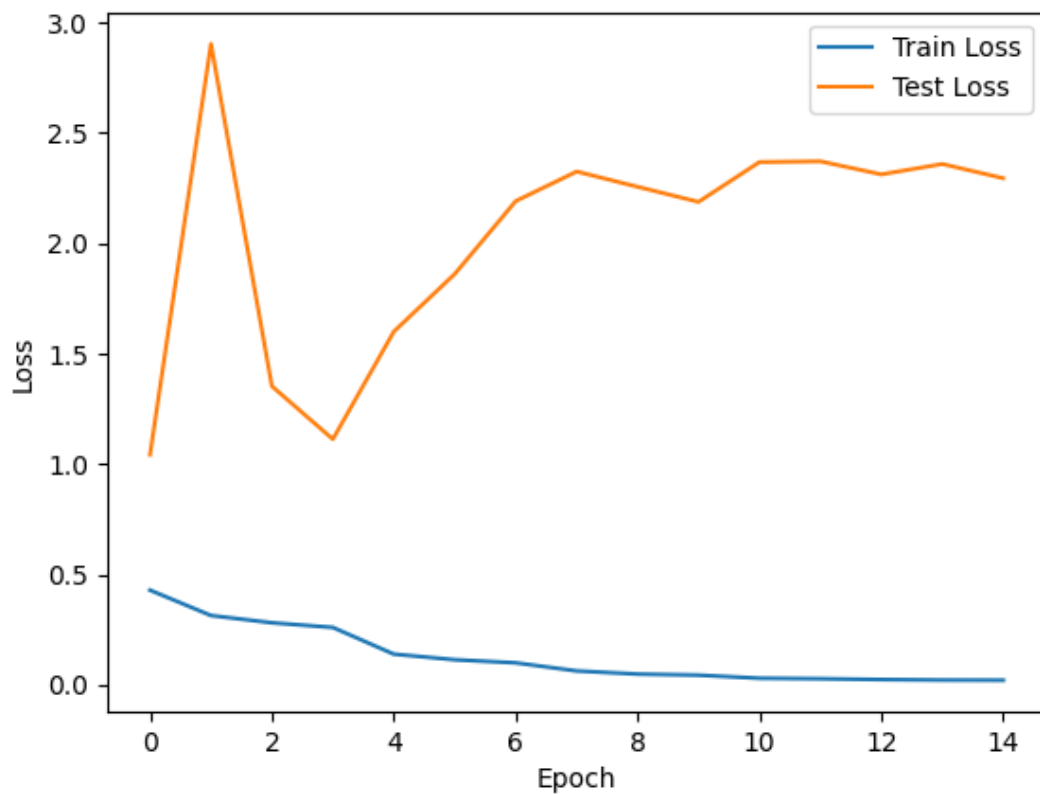


Figure 6.3.3.1 Train and Test Loss Values

Train and Test Accuracy Values

In the train cycle, accuracy values increased regularly at the end of each epoch. The same situation was observed in the Test cycle up to a certain point. However, test accuracy remained constant at a certain point and was observed at these points in the remaining training cycles.

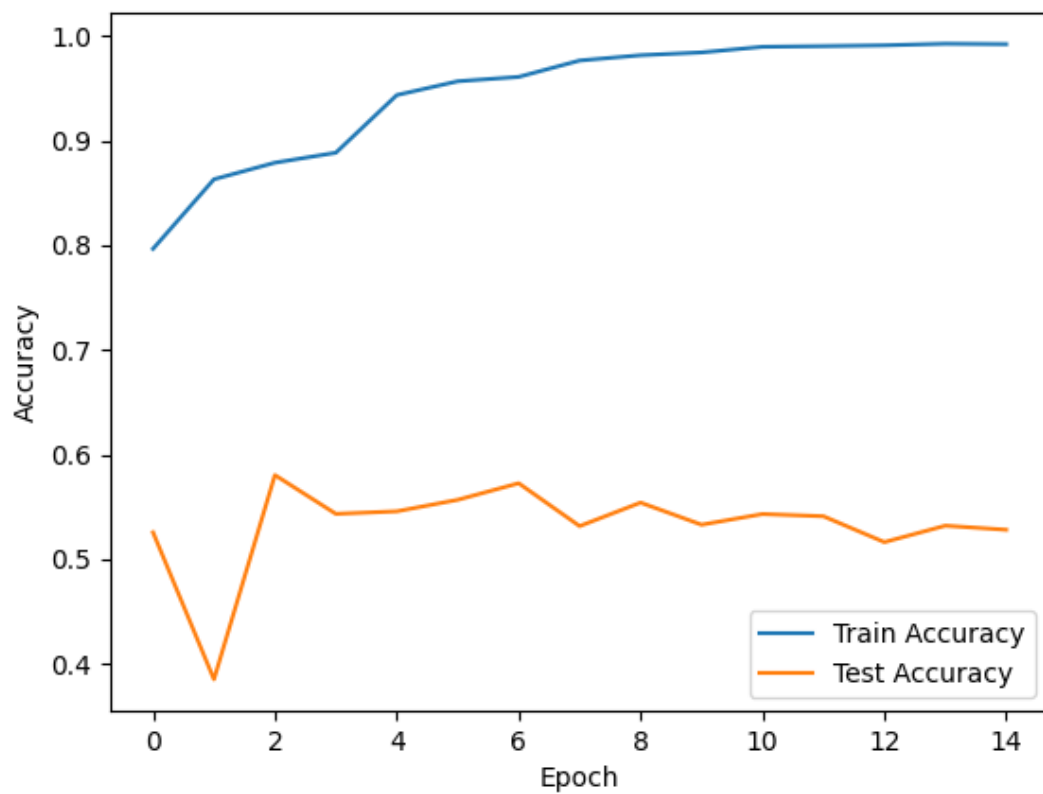


Figure 6.3.3.2 Train and Test Accuracy Values

Confusion Matrix

Confusion matrix results and ratios of the model:

True Positive (TP): 6,907

True Negative (TN): 1,395

False Positive (FP): 4,420

False Negative (FN): 2,999

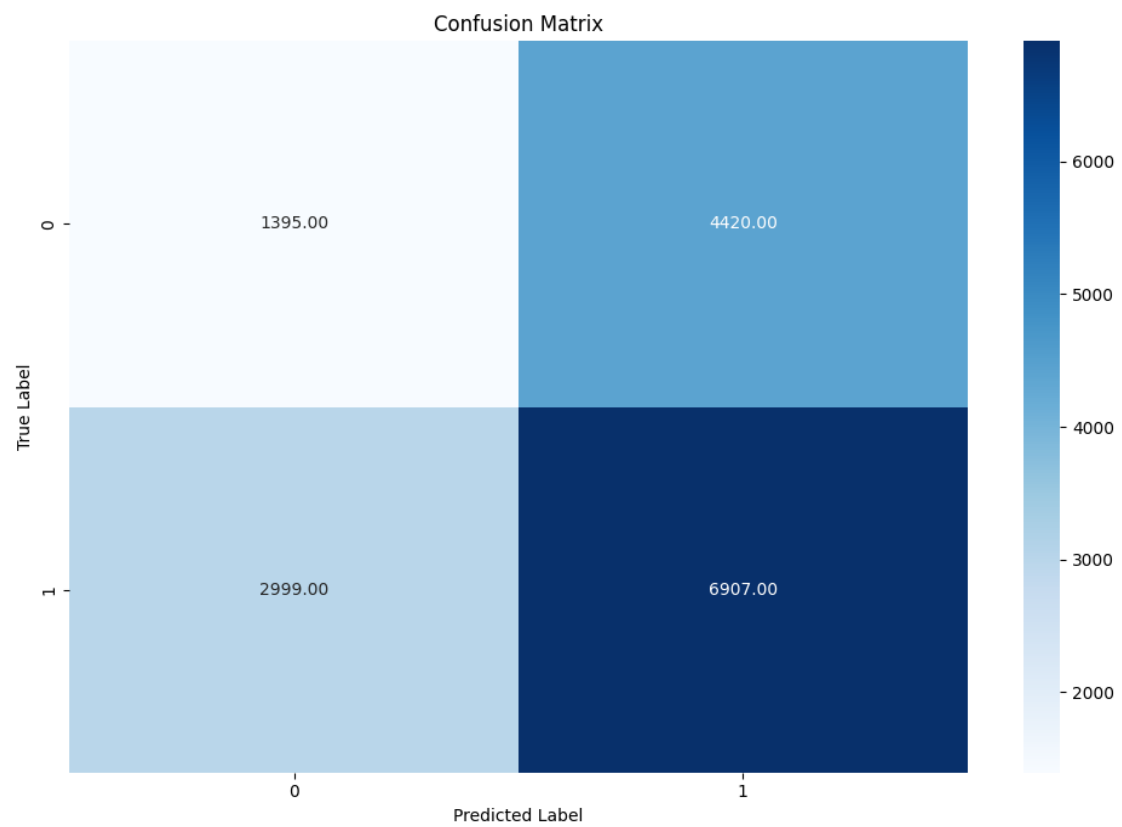


Figure 6.3.3.3 Confusion Matrix

Confusion matrix ratios of the model:

True Positive (TP): 0.70

True Negative (TN): 0.24

False Positive (FP): 0,76

False Negative (FN): 0,30

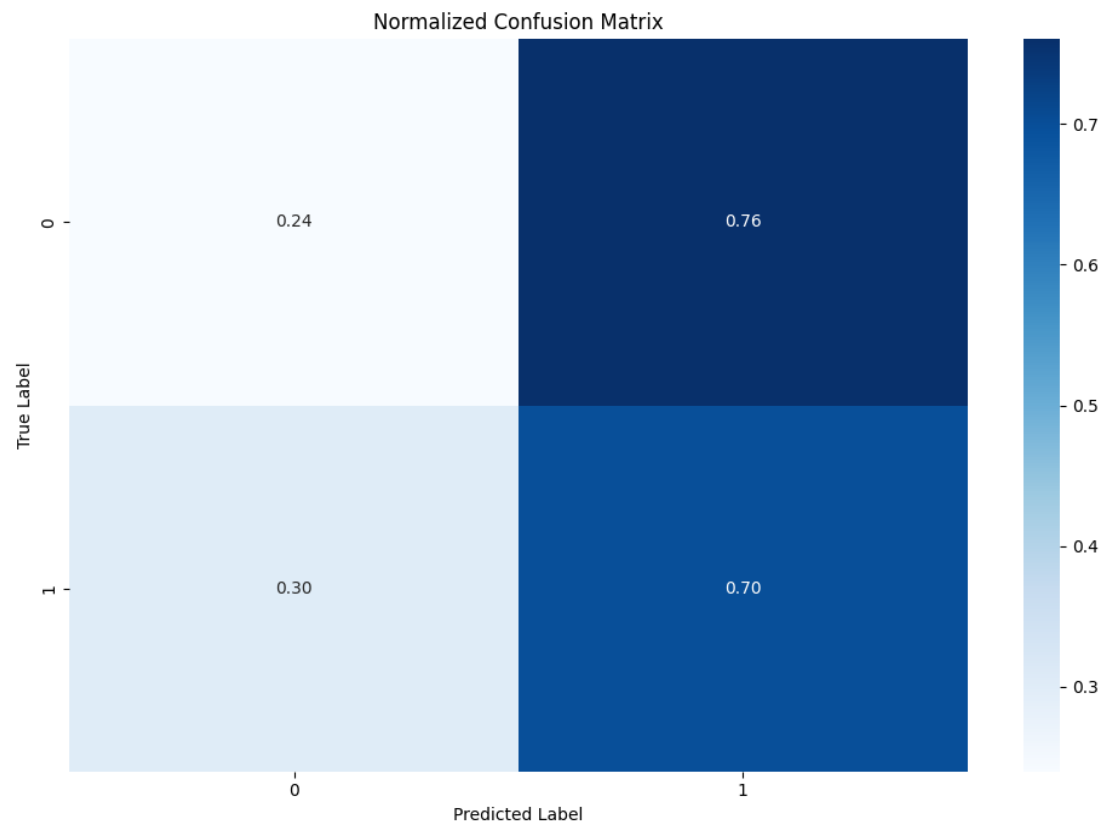


Figure 6.3.3.4 Normalized Confusion Matrix

Performance Metrics

If we examine the performance metrics of the model.

- Accuracy value is 53%, indicating that only 53% of the samples in the data set are classified correctly.
- The precision value is 61%, indicating that 61% of the samples predicted as positive by the model are positive.
- Recall value is 70%, indicating that the model correctly detects 70% of true positive samples
- F1 score value was found to be 65%.
- The specificity value is 24%, indicating that only 24% of the examples in the model that should be negative are correctly classified as negative. The biggest reason for this is that the number of patients and images belonging to the negative class is low in the train and test sets we have.

- The Positive Predict Value (PPV) value is 61%, indicating that 61% of the samples predicted as positive are positive.

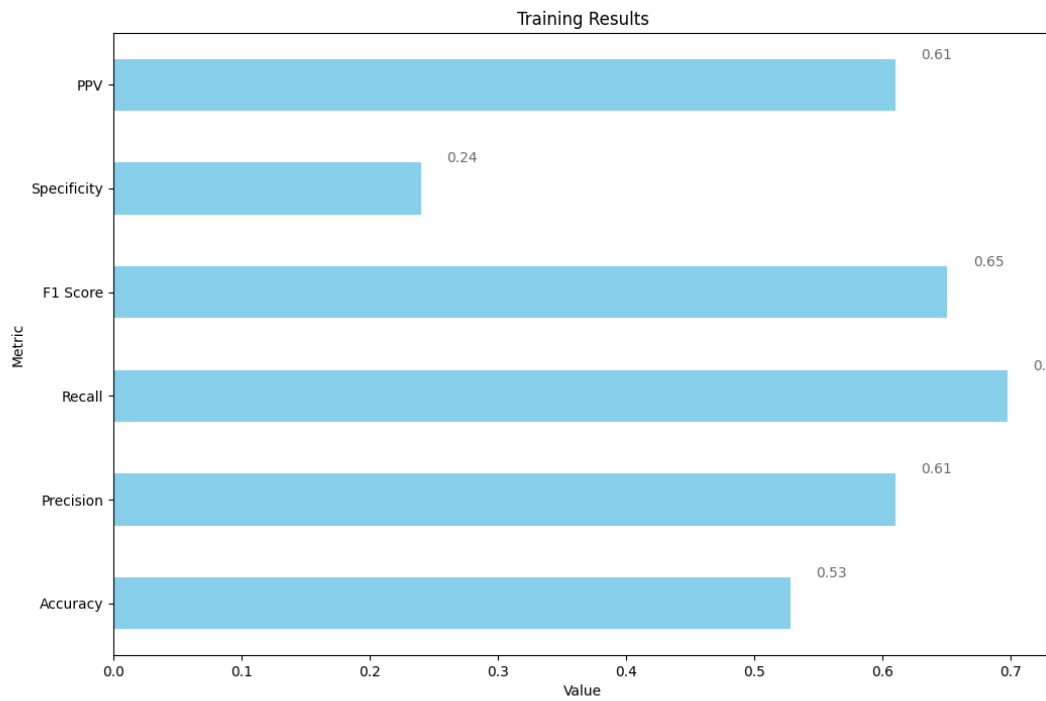


Figure 6.3.3.5 Performance Metrics

ROC-AUC Value

The AUC value is 0.48, indicating that the model's discrimination in positive and negative classes is worse than random guessing.

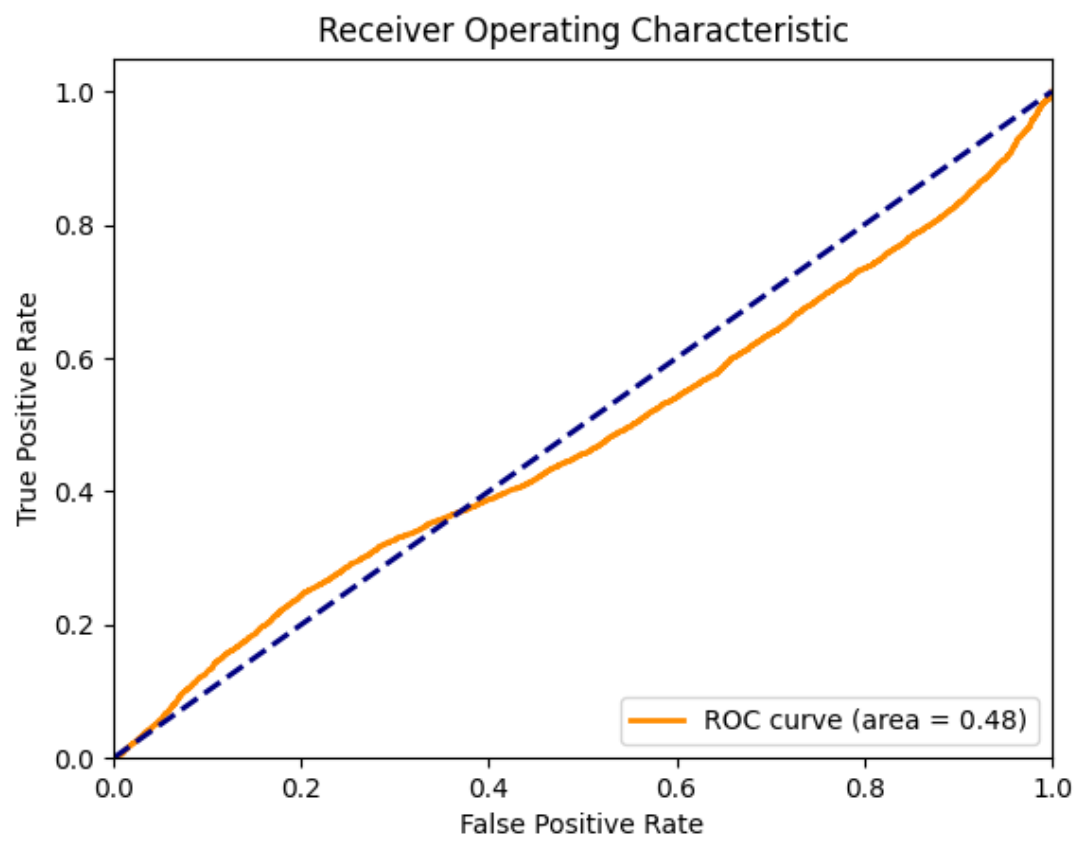


Figure 6.3.3.6 Roc-Auc Curve

7- GENERAL EVALUATION

It has been observed that the model created using MR and PET images has higher performance than the models created using only PET or only MR images. The model consisting of MR and PET images was more successful in correctly detecting positive classes with a higher AUC value (0.65) and F1 score (0.79). However, this model also appears to have extreme learning problems and limited generalization ability.

The model created using PET images showed a reasonable performance but remained poor with a lower AUC value (0.57) and F1 score (0.69) than the model created with MR and PET images. The model created using only MR images showed the lowest performance, with an AUC value of 0.48, which was worse than random guessing.

As a result, the model created from MR and PET images was found to be more effective for tumor detection. In future studies, studies such as optimizing data preprocessing techniques, expanding the data set, and improving model hyperparameters can be done to increase model performance.

REFERENCES

1. BULUT, Dursun (2024). "**Emar (MR) nedir ?**", Medical Park,
<https://www.medicalpark.com.tr/emar/hg-2048>.
2. BENLİ Mari (2024). "**Pet-CT taraması nedir?**", Florence,
<https://www.florence.com.tr/pet-ct>.
3. ŞAKAR, Hilal (2024). "**Evrişimli Sinir Ağları (Convolutional Neural Networks - CNN)**", Medium,
<https://hilalgozutok.medium.com/evrişimli-sinir-ağları-convolutional-neural-networks-cnn-e61470e9bdb1>.
4. SÜHEDA, Akdağ (2024). "**ResNet (Residual Network) nedir?**", Medium,
<https://suhedacilek.medium.com/resnet-residual-network-nedir-49105e642566>.