Contents

Abstract

Breast cancer is now commonly recognised as the second-leading cause of death among women. Computer Aided Detection systems provide significant solutions in the early detection and diagnosis of this cancer. The main objective of this study is to develop a system using deep learning algorithms to classify breast lesions into malignant, benign, and normal. This study contains three phases, image preprocessing phase, applying deep learning models, and evaluating our results base using 10-foldcross validation. Five different scenarios have been designed to evaluate different preprocessing techniques and check them with using two different architectures of Convolutional Neural Networks. At first, images have been read. Afterwards, class weight technique and rotation augmentation have been applied in order to make our classes balanced. In the last two scenarios image, contrast enhancement with rotation augmentation and class weight technique have been combined.

The proposed system offers good classification rates. These techniques were applied on the MIAS dataset. Combining image contrast enhancement with class weight technique applied on a 4-layer convolutional neural network showed great performance and promising outcomes using 1o-fold cross validation. For the best method, the accuracy, f-measure, recall, and precision reached 80.99%, 81.07%, 80.99%, and 80.99% respectively. According to the findings of this study, CNNs offer a lot of potential in the field of intelligent medical image diagnosis.

Introduction

Breast cancer is a complex disease which is found as the second cause of cancer-associated death among women. In Canada, breast cancer accounts for around 25% of new cancer diagnoses and 13% of all cancer deaths in Canadian women. The diagnosis of tumour metastases is critical in the treatment of breast cancer patients. Breast screening is evaluated using a three-step approach as follows: 1_suitable imaging, 2_ clinical assessment, 3_if necessary, needle biopsy (i.e. H&E 2 stained histology). Manually detecting breast cancer takes a long time and is difficult for the physician to classify. As a result, early detection of cancer via various automated diagnostic procedures is critical. Support Vector Machine, Naive Bayes, KNN, and Convolution Neural Network are some of the methods and algorithms available for identifying breast cancer. Convolution Neural Network is the most recent deep learning technique that is also used for image classification.

In this project, we build two Convolution Neural Network architecture to classify the images based on whether they are benign and malignant. The Mammographic image analysis society (MIAS) is used for training and testing the model.

In the following part, first a background about convolution Neural Network is presented, then the methodology behind the project is stated. In the next step, the architectures that have been used are described. Finally, the results and evaluation of the model is discussed.

Background

Algorithm:

Since the Convolutional Neural Network (CNN) won the image classification competition 202 (ILSVRC12), deep layer CNN research has gotten a lot of interest. Convolutional neural networks are widely used to solve a wide range of learning problems. They're highly good at solving image classification. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning system that can take an input image, assign relevance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. When compared to other classification methods, the amount of pre-processing required by a ConvNet is significantly less. The higher performance of convolutional neural networks with picture, speech, or audio signal inputs sets them apart from other neural networks. They are divided into three main layers:

- Convolutional layer
- Pooling layer
- Fully connected (FC) layer

A convolutional neural network's first layer is the convolutional layer. While further convolutional layers or pooling layers can be added after convolutional layers, the fully connected layer is the last layer. The CNN becomes more complicated with each layer, detecting larger areas of the image. Earlier layers concentrate on basic elements like colours and borders. As the visual data travels through the CNN layers, it begins to distinguish larger elements or features of the item, eventually identifying the target object.

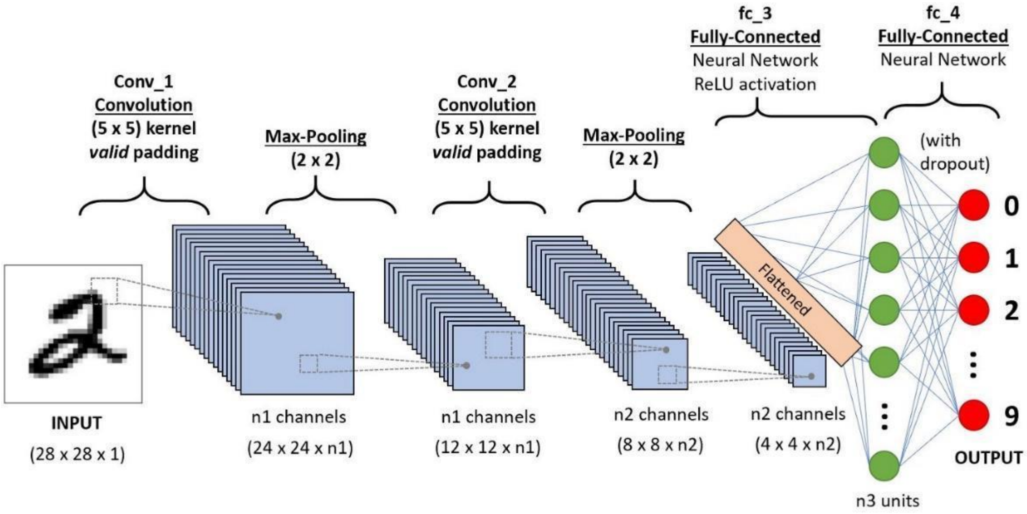


Figure 1: *CNN sequence to classify MNIST digit*

To measure the accuracy of the algorithm, we used four criteria of measurement, which include:

- Accuracy
- Precision
- Recall
- F-measure

which we will discuss in further details.

Accuracy:

One parameter for evaluating classification models is accuracy. Informally, accuracy refers to the percentage of correct predictions made by our model. The following is the formal definition of accuracy:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

The following formula can be used to calculate accuracy in terms of positives and negatives for binary classification[1]:

$$Accuracy = \frac{True\ Negatives + True\ Positive}{True\ Positive + False\ Positive + True\ Negative + False\ Negative}$$

- True Positive: Times that the model correctly classifies a Positive sample as Positive.
- False Negative: Times that the model incorrectly classifies a Positive sample as Negative.
- False Positive: Times that the model incorrectly classifies a Negative sample as Positive.
- True Negative: Times that the model correctly classifies a Negative sample as Negative.

[1] https://developers.google.com/machine-learning/crash-course/classification/accuracy

Precision:

The precision is measured as the ratio of the number of correctly identified Positive samples to the total number of Positive samples (either correctly or incorrectly). The precision of the model in categorising a sample as positive is measured.

$$Precision = \frac{True_{Positive}}{True_{Positive} + False_{Positive}}$$

When the model makes many inaccurate Positive classifications or a small number of correct Positive classifications, the denominator rises and the precision falls. On the other hand, when the model makes many correct Positive classifications (maximise True Positive) and a small number of incorrect Positive classifications, the precision is high (minimize False Positive).

Recall:

The recall is determined by dividing the total number of Positive samples by the number of Positive samples accurately categorised as Positive. The model's ability to recognise Positive samples is measured by the recall. The higher the recall, the greater the number of positive samples found[2].

$$Recall = \frac{True_{Positive}}{True_{Positive} + False_{negative}}$$

F-measure or F-score (F1):

Precision and recall are both considered while calculating the F1 Score. It's the precision and recall harmonic mean (average). The F1 Score is optimal when the system has a good balance of precision (p) and recall (r). On the other hand, if one measure is enhanced at the price of another, the F1 Score isn't as high.

[2] https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/

$$F1 = 2.\frac{Precision.recall}{Precision + recall}$$

Methodology

The suggested system has been explained in this section. First, the mammography datasets that were used to develop the model were described. Following that, the entire development process was outlined, including the "preprocessing" phase and the "Convolution Neural Network model " phase. Five scenarios have been proposed to train the CNN with two different architectures. In the initial strategy, original data was used to feed CNN. In the second approach, the data was balanced with class weight technique. In the third approach, rotation augmentation has been done. In the last two scenarios, image contrast enhancement technique combined with class weight and rotation augmentation technique.

The goal of these experiments is to see if the suggested CNN model, which was trained using the MIAS dataset, can correctly identify lesions in mammography pictures as normal, malignant, or benign.

Dataset:

The MAMMOGRAPHIC IMAGE ANALYSIS SOCIETY(MIAS) dataset has images and labels. This dataset consists of 322 mammography scans (64 benign samples, 51 malignant samples and 207 normal). The labels have seven columns which contains:

1. MIAS database reference number
2. Character of background tissue (Fatty, Fatty-glandular, Dense-glandular)
3. Class of abnormality (Calcification, Well-defined, Spiculated masses, Ill-defined masses, Architectural distortion, Asymmetry and Normal)
4. Severity of abnormality (Benign, Malignant)

5,6. X, Y image-coordinates of the centre of abnormality,

7. approximate radius (in pixels) of a circle enclosing the abnormality.

This information was determined by the radiologist. It should be noted that the size of all images is 1024 pixels x 1024 pixels. The images have been centred in the matrix. Furthermore, the list is arranged in pairs of films, where each pair represents the left (even filename numbers) and right (odd filename numbers) mammograms of a single patient.

Here the techniques for balancing the dataset is explained.

Class weight Technique (Copy, paste)

This method has been applied to handle imbalanced classes in order to avoid a bias model where each class is represented by the same number of other classes. Balancing can be performed by exploiting one of the following techniques: oversampling. under sampling. class weight.

Figure 1 shows the distribution of the data samples. As we can see, our data samples are imbalanced, so Class weight technique has been used to generate synthetic samples. This method increases the number of samples of the smallest class up to the size of the biggest class. In other words, we use the size of the biggest class, subtract the size of each smaller class from all the size of the biggest class, then we generate the same number of images. In this way, each class will be represented by the same number of other classes.

In this research, the number of normal samples are 207 which is the biggest class, so the size of benign and malignant classes increased to 207. As a result, the number of samples in each class of malignant, benign, and normal are the same.

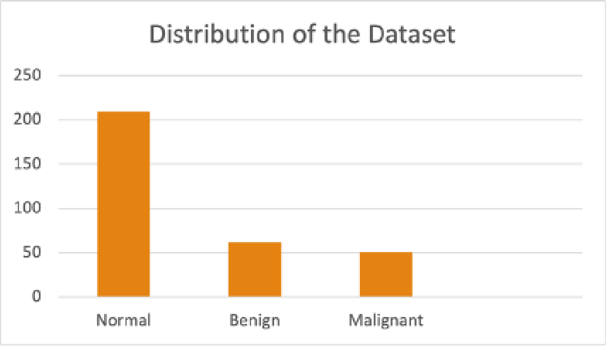


Figure 1- Distribution of the data samples

Rotation Augmentation

One common data augmentation technique is random rotation. A source image is randomly rotated clockwise or counter clockwise by some number of degrees, changing the position of the object in frame. Notably, for object detection problems, the bounding box must also be updated to encompass the resulting object. Random rotation can improve your model without you having to collect and label more data.

In this dataset, due to our two unbalanced classes (malignant, benign) which we want to make the size of our two classes (64 and 51) to be the same as the normal class (207) to be balanced. We choose three degrees (90, 180, 270) to rotate our two unbalanced classes (malignant, benign) to make them equal to our normal 207 class. The reason behind choosing those degrees is that by using 90, 180 and 270 we make sure that our 62 benign and 51 malignant samples are preserved and not to be confused with normal samples because when we rotate the images, the mass we want to identify might be cropped from the edges and considered as a normal image. In other words, if the rotation degree is not suitable for the image, we may lose some part of our image and the data will no longer be preserved post transformation. Finally, we sum the rotated datasets to the two classes and at the end we get 207 malignant, 207 benign and 207 normal data. Figure 2 shows a rotated image with an angle of 90 degrees.

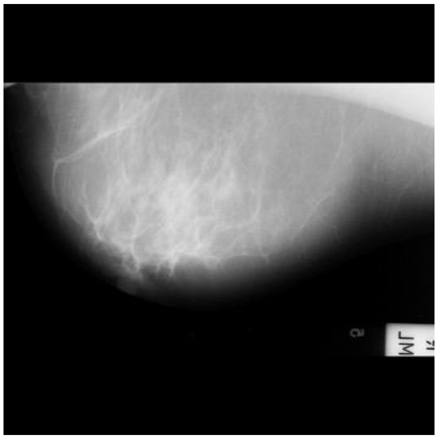


Figure 2 : Rotated image

Image Contrast Enhancement Technique

In different applications and sectors, image processing is a technique for increasing the quality of an image by deleting extraneous image data. In their original format of scanned images, medical photographs contain a lot of unnecessary and undesired portions. In the realm of various medical diagnostics applications, digital image processing is frequently used. In medicine, imaging technology has enabled doctors to observe the inner workings of human anatomical organs for easier diagnosis.

In this part, image enhancement contrast has been applied to adjust the contrast of an image. Regarding our images, we want to increase the contrast to make the light areas lighter and the dark areas darker. we take our classes and change the contrast between the target object and background. We have an enhancement factor that controls the effect. For example, Factors less than one decrease the brightness range, pulling all the colours towards a middle grey. A factor of zero results in a completely grey image. As a result, we applied various factors greater than one and then we chose 1.25. In this process, we have to be careful not to make the contrast too high because when the contrast between object and background is high, the target that we want to use in classification might fade and not be recognizable anymore. In other words, if we choose a very high contrast value, only the basic shapes of the images are visible.

Figure 3 shows an original image and figure 4 shows the effect of applying the enhancement factor.

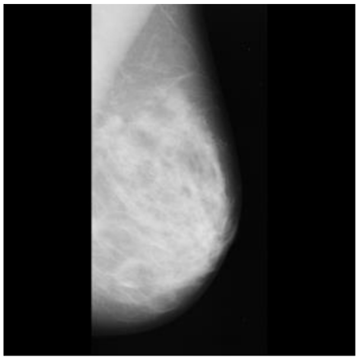


Figure 3: Original image

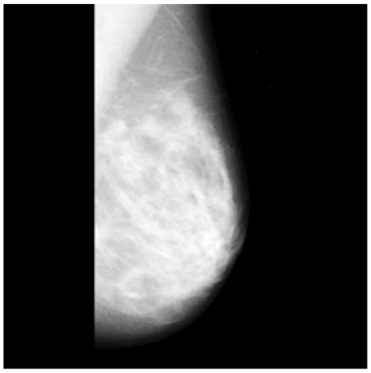


Figure 4: Enhanced image

Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) are comparable to "traditional" neural networks in a way that they contain hidden layers of neurons with "learnable" parameters. These neurons receive inputs, do a dot product, then perform a non-linearity function after that. CNN's success is due to its superior multi-scale high-level image representations.

The data enters the CNN via the input layer and then goes through several hidden layers before reaching the output layer. The network's prediction is represented via the output layer. In terms of loss, the network's output is compared to the actual labels. The partial derivatives of this loss with respect to the trainable weights are calculated and the weights are changed for the network to learn. The network's hidden layers serve as a basic building component for data transformation (input layer or the output of the previously hidden layer). What distinguishes CNN from a simple multilayer perceptron (MLP) network is the use of convolutional layers, pooling and non-linearities such as tanh, sigmoid, and Relu.

Convolutional layers are applied to input feature maps in 2D (or 3D). Weights that could be trained are the 2D (or 3D) filter/kernels that move across the input feature map and, producing dot products with the input feature map’s overlapping region. Three hyperparameters used to define the convolutional layer are as follows:

Kernel Size: The sliding kernel or filter's size.

Stride Length: This specifies how far the kernel is slid before the dot product is used to generate the output pixel.

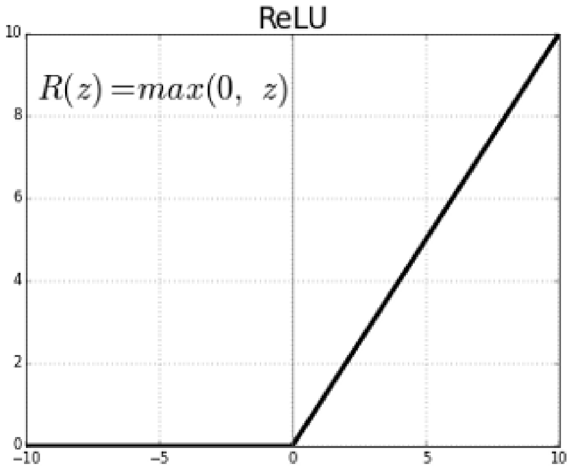
Padding: The number of zeros added around the input feature map in a frame.

Each convolution layer is followed by a pooling layer. The pooling layer decreases the size of input images using the convolution filter's outputs. As a result, the number of parameters in the model is reduced, which is referred to as down-sampling. The two most commonly used pooling layers are Average pooling and Maximum pooling(Max pooling). For this project, Max pooling is used. Max Pooling is a feature map pooling operation that determines the maximum value for patches.

Finally, to introduce non-linearities into the algorithm, an activation function is used such as Relu, Sigmoid or tanh. It was found that in comparison to other two functions, Relu significantly accelerates the convergence of stochastic gradient descent.

In addition, ReLU is obtained by simply thresholding matrix values at zero, as opposed to the considerable computation required by tanh and sigmoid Equitation.

$$f\left(h_{\theta}(x)\right) = h_{\theta}(x)^{+} = max(0, h_{\theta}(x))$$



The output from the last Pooling layer, were flattened and then fed into the fully connected layer. Then there is a 3 layer fully connected layer. After passing through the fully connected layers, the final layer uses the softmax activation function to determine the likelihood that the input belongs to which specific class.

Model Implementation:

The dataset includes 322 images of the size 1024 * 1024. In the first step, the format of the images was changed to pgm which is a bitmap-based format with a four-line header and data saved as an unsigned char type, with a maximum of 256 grayscale levels or 8-bit data per pixel. Then, the format of each image was changed to npy format in order to make it readable for Python. Consequently, the size of arrays became 1024 * 1024 * 1.

In this section, we used two CNN models to evaluate our dataset which will be explained. The first CNN used for this dataset is called Architecture 1 and the second CNN is called Architecture 2.

Architecture 1:

This architecture has 3 convolutional layers, 3 max pooling and 3 fully connected layers. The number of layers in each channel is as follows:

Architecture 1	
Input Layer	1024 * 1024 *1
Convolution Layer 1	64 Filters
Max pooling	Kernel Size: 10 *10
Convolution Layer 2	32 Filters
Max pooling	Kernel Size: 10 *10
Convolution Layer 3	16 Filters
Max pooling	Kernel Size: 10 *10
Fully Connected Layer 1	128 Neurons
Fully Connected Layer 2	64 Neurons
Fully Connected Layer 3	32 Neurons

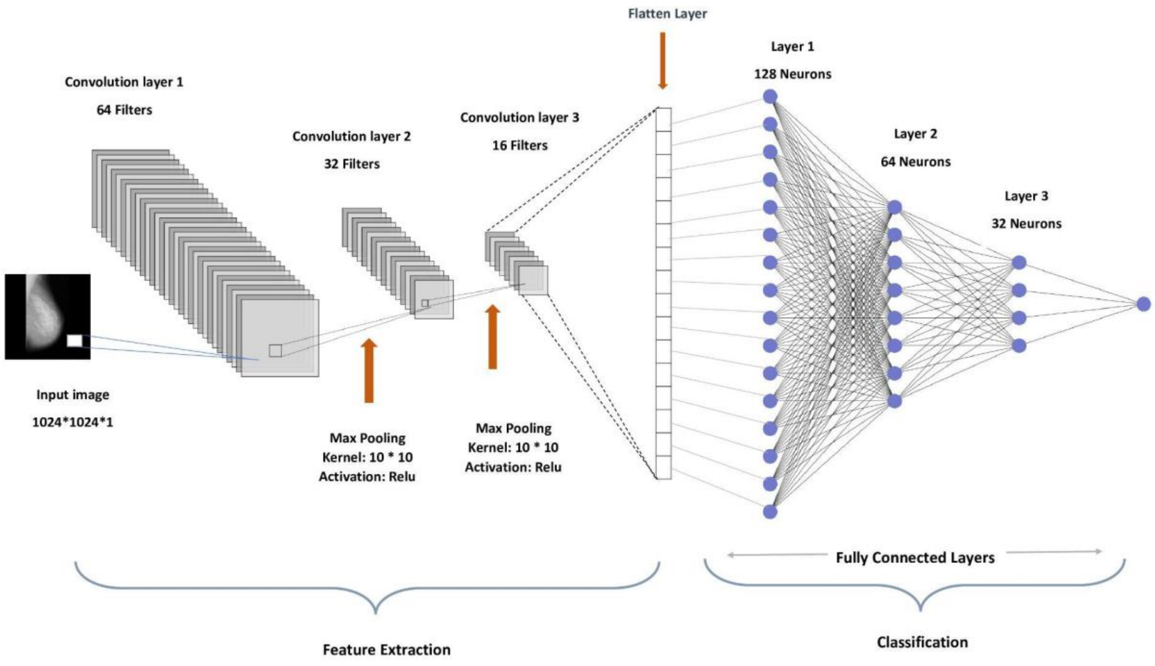


Figure 5: Architecture 1

Architecture 2:

This architecture is more complex than the architecture 1, so it has 4 convolutional layers, 4 max pooling and 3 fully connected layers. Their difference is that the second one has one convolutional and one max pooling layer more than the first one which can make our process more accurate. Each channel has the following number of layers:

Architecture 2	
Input Layer	1024*1024*1
Convolution Layer 1	128 Filters
Max pooling	Kernel Size: 10 *10
Convolution Layer 2	64 Filters
Max pooling	Kernel Size: 10 *10
Convolution Layer 3	32 Filters
Max pooling	Kernel Size: 10 *10
Convolution Layer 4	32 Filters
Max pooling	Kernel Size: 10 *10
Fully Connected Layer 1	256 Neurons
Fully Connected 2	128 Neurons
Fully Connected 3	64 Neurons

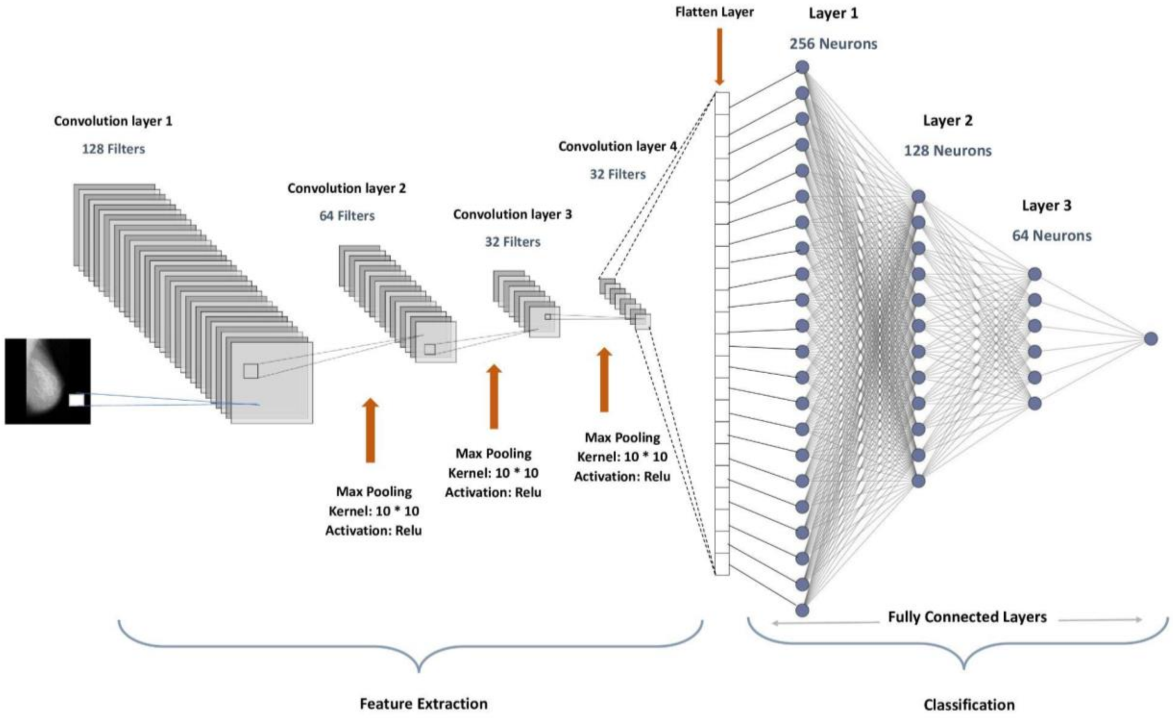


Figure 6: Architecture 2

In CNN 1, first we import related keras, sklearn and other libraries. Then, we use one-hot encoding to categorize the three classes (Benign, Malignant, normal) into 2, 1 and 0. One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. The model of our CNN architectures is **sequential**. The Sequential model API allows us to develop deep learning models by creating an instance of the Sequential class and adding model layers to it. The activation function is **Relu**.

Then, for the model compiler, **Adam** optimizer was used. Adam optimization is a stochastic gradient descent method based on adaptive first- and second-order moment estimation. For the metrics, **accuracy** was calculated. This function computes subset accuracy in multilabel classification: the set of labels predicted for a sample must exactly match the corresponding set of labels in y true.

The dataset was balanced using different techniques including the Class weight technique (Copy, Paste), rotation technique, Image enhancement (contrast). Furthermore, different techniques were combined in order to increase the accuracy of the model. For this project, once we combine the Image enhancement technique (Contrast) with Copy paste technique. And the other time, the contrast technique was combined with rotation. The results of each one of these methods were stated in the next part.

Evaluation/Results

In this section, we review and compare the results. As we talked about, we had 5 preprocessing scenarios, then we evaluated each of them with our two CNN architectures. The tables are shown the confusion matrix for each evaluation. A confusion matrix N x N is used to evaluate the performance of a classification model, where N is the number of target classes. The matrix compares the actual goal values to the machine learning model's predictions. This provides us with a comprehensive picture of how well our classification model is working and the types of errors it makes.

In the confusion matrix, class 1 represents the normal class. Class 2 represents the malignant class, and class 3 represents the class benign. For each one of the algorithms accuracy, f-measure, recall and precision are calculated and can be seen in the following tables.

Scenario 1: Applying Architectures on Original Images

The first result to mention is our original imbalanced data without any preprocessing.

For the original data set, Architecture 1 could classify 150 samples as True Positive. 34 samples of malignant class considered as True Positive. 19 samples of benign were considered false positive, and 32 samples as True Positive.

Architecture 1				
Confusion Matrix				
		Predicted Output		
	Class	1	2	3
Real Output	1	150	36	21
	2	18	34	12
	3	6	13	32

Criteria	
Accuracy	0.670807
precision	0.588005
recall	0.62778
f-measure	0.607242

Architecture 2				
Confusion Matrix				
		Predicted Output		
	Class	1	2	3
Real Output	1	153	33	21
	2	16	39	9
	3	4	14	33

Criteria	
Accuracy	0.698758
precision	0.620564
recall	0.665188
f-measure	0.642101

From those tables we can conclude that architecture 2 outperformed architecture 1 with accuracy 0.69 which had more complexity than the first one.

Scenario 2 : Applying Architectures on balanced dataset with Class Weight Technique (Copy, Paste)

Now, we will apply the CNN 1 and CNN 2 to the first preprocess scenario which is balanced data (copy/ paste scenario)

Architecture 1				
Confusion Matrix				
		Predicted Output		
	Class	1	2	3
Real Output	1	165	18	24
	2	16	151	40
	3	32	38	137

Criteria	
Accuracy	0.729469
precision	0.72857
recall	0.729469
f-measure	0.729019

Architecture 2				
Confusion Matrix				
		Predicted Output		
	Class	1	2	3
Real Output	1	168	18	21
	2	16	157	34
	3	13	51	143

Criteria	
Accuracy	0.753623
precision	0.756568
recall	0.753623
f-measure	0.755093

As the previous comparison, in these two tables we can see the accuracy criteria for architecture 2 was better with a 0.03 difference.

Scenario 3 : Applying Architectures on balanced dataset with rotation augmentation

Architecture 1				
Confusion Matrix				
		Predicted Output		
	Class	1	2	3
Real Output	1	168	24	15
	2	24	128	55
	3	27	39	141

Criteria	
Accuracy	0.703704
precision	0.701842
recall	0.703704
f-measure	0.702772

Architecture 2				
Confusion Matrix				
		Predicted Output		
	Class	1	2	3
Real Output	1	165	20	22
	2	24	143	40
	3	12	56	139

Criteria	
Accuracy	0.719807
precision	0.721802
recall	0.719807
f-measure	0.720803

As we can see about the accuracy, this method of balancing data compared to the previous method (copy/ paste) has poor performance.

Scenario 4: Applying Architectures on a balanced dataset with Image contrast enhancement technique and Copy/Paste method.

In this scenario, we combine two techniques. First, we enhance the image contrast and then balance the dataset with the Copy/Paste method.

Architecture 1				
Confusion Matrix				
		Predicted Output		
	Class	1	2	3
Real Output	1	172	18	17
	2	12	161	34
	3	15	39	153

Criteria	
Accuracy	0.782609
precision	0.784285
recall	0.782609
f-measure	0.783446

Architecture 2				
Confusion Matrix				
		Predicted Output		
	Class	1	2	3
Real Output	1	172	18	17
	2	12	161	34
	3	15	39	153

Criteria	
Accuracy	0.809984
precision	0.811599
recall	0.809984
f-measure	0.81079

In this scenario, we get really good results compared to the three other processes with accuracy of 0.78 and 0.80. It can also be shown that with architecture 2, our results get better.

Scenario 5: Applying Architectures on a balanced dataset with Image contrast enhancement technique and Rotation technique.

Architecture 1				
Confusion Matrix				
		Predicted Output		
	Class	1	2	3
Real Output	1	164	19	24
	2	12	157	38
	3	19	37	151

Criteria	
Accuracy	0.760064
precision	0.762345
recall	0.760064
f-measure	0.761203

Architecture 2				
Confusion Matrix				
		Predicted Output		
	Class	1	2	3
Real Output	1	171	14	22
	2	10	157	40
	3	21	27	159

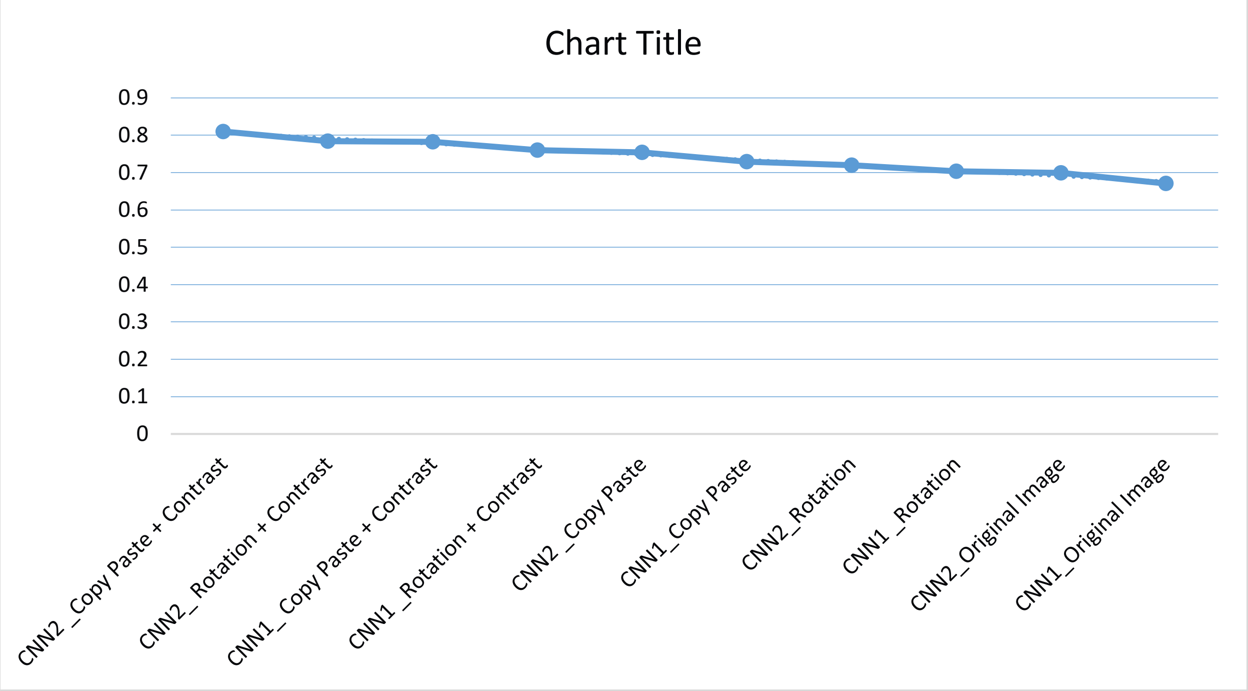
Criteria	
Accuracy	0.784219
precision	0.786307
recall	0.784219
f-measure	0.785262

The accuracy of this scenario was quite good but compared to the contrast + copy/paste had poor performance.

Overall, the more complex our real model, the more accurate the data will be. In all 5 scenarios that were discussed, architecture 2 outperformed architecture one, and contrast + copy/paste had better accuracy criteria than the others.

Conclusion

As could be concluded from the results, when the methods are combined, better accuracy is achieved. The best results belong to architecture 2 when the image contrast enhancement and then copy/paste method were applied. The second accuracy was also achieved by the technique when image contrast enhancement and Copy/Paste methods were combined. The Rotation technique resulted in the lowest accuracy among both architectures.



References

1. https://developers.google.com/machine-learning/crash-course/classification/accuracy
2. https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/
3. Shorten, C., Khoshgoftaar, (2019),T.M. A survey on Image Data Augmentation for Deep Learning. J Big Data 6, 60 (2019). https://doi.org/10.1186/s40537-019-0197-0
4. Yu X., Zhou Q., Wang S., Zhang Y.-D.(2021), A systematic survey of deep learning in breast cancer, International Journal of Intelligent Systems.
5. Oyelade O.N., Ezugwu A.E. (2021), Characterization of abnormalities in breast cancer images using nature-inspired metaheuristic optimized convolutional neural networks model Concurrency and Computation: Practice and Experience.
6. Perumal, S. & Thambusamy, Velmurugan. (2018). Preprocessing by contrast enhancement techniques for medical images. International Journal of Pure and Applied Mathematics. 118. 3681-3688.