Detection and Diagnosis of Breast Tumors using Deep Convolutional Neural Networks

J. D. Gallego-Posada,[†] D. A. Montoya-Zapata,[§] and O. L. Quintero-Montoya[‡]
Research Group on Mathematical Modeling
School of Mathematical Sciences
Universidad EAFIT
Medellín, Colombia
{[†]*jgalle29*, [§]*dmonto39*, [‡]*oquinte1*} *@eafit.edu.co*

***Abstract*—We present an application of deep Convolutional Neural Networks (CNN) for the detection and diagnosis of breast tumors. The images used in this study have been extracted from the mini-MIAS database of mammograms. The proposed system has been implemented in three stages: (a) crop, rotation and resize of the original mammogram; (b) feature extraction using a pretrained CNN model (AlexNet and VGG); (c) training of a Support Vector Machine (SVM) at the classification task using the previously extracted features. In this research, the goal of the system is to distinguish between three classes of patients: those with benign, malign or without tumor. Experiments show that feature extraction using pretrained models provides satisfactory results, achieving a 64.52% test accuracy. This outcome could be improved via fine-tuning of the final layers or training the whole network parameters. The results of additional experiments using a sample of the Caltech-101 database, for which a 99.38% test accuracy was obtained, exhibit the relevance of the similarity between the data used to train the model and the particular application intended. Additionally, it is worth noting the impact of the data augmentation process and the balance of the number of examples per class on the performance of the system.**

***Keywords*—Breast tumor, classification, mammogram, convolutional neural network, support vector machine**

I. Introduction

Breast cancer is the most common cancer in women and is commonly thought to be a disease of the developed world but nearly 50% of breast cancer cases and 58% of deaths occur in less developed countries. It is estimated that around the world over 508.000 women died in 2011 due to this condition. According to the World Health Organization, detection of breast cancer in its early stages dramatically increases the chances of establishing a successful treatment plan [1].

As part of the current efforts to control this condition, the development of computer-aided diagnosis systems which can assist medical personnel with the early detection of tumors pose a crucial alternative. In such systems a high reliability in the accuracy of the classifier is a top priority.

In this study, the diagnosis was performed employing a SVM trained with features extracted using AlexNet and VGG pretrained models fed with preprocessed mammograms. Our data source is the database of the Mammographic Image Analysis Society (MIAS) [2].

The paper is structured as follows: in Section II we provide a review of the application of Deep Learning techniques to the image classification problem. Section III presents an outline of previous studies of breast cancer detection and classification using Deep Learning and Artificial Intelligence-based approaches. In Section IV, the employed methodologies are described. In Section V, the results of the application of the proposed methodology using an extract of the Caltech-101 database are shown. Next, in Section VI the specifications of the implemented system are presented. Finally, Section VII contains the main conclusions of this work and some possibilities for future improvements on this research.

II. Related Work

The study of computer-aided breast cancer diagnosis has been addressed from several perspectives. The aim of this section is to briefly illustrate the state of the art in this field using artificial intelligence and additionally using strictly Deep Learning-related techniques.

A. Analysis of Breast Cancer using Artificial Intelligence Techniques

Alolfe *et al.* in 2009 used a SVM and linear discriminant analysis to distinguish between benign and malign tumors on the MIAS database [3]. Using this approach, they classified 90% and 87.5% of benign and malignant images correctly, respectively. A region of interest (ROI) of 32×32 pixels was selected from the images and 224 features were extracted. These features were divided into five groups: wavelet, first order statistics, second order statistics, shape and fractal dimension data. Finally, 13 features were selected with the forward stepwise linear regression method.

Wang *et al.* in 2013 used the mammographies from 482 patients to compare the accuracies from an extreme learning machine (ELM) and a SVM to classify between images with and without tumors [4]. In the preprocessing stage a median filter was used to reduce the noise and the wavelet transformation of local modulus maxima in conjunction with the region growing algorithm were used as edge segmentation method. Finally, five textural features and five morphological features were extracted from the resulting image and these were used at the classification task. The ELM classifier exhibited better performance than the SVM classifier.

Dheeba *et al.* in 2014 obtained an accuracy of 93.67% classifying between normal and abnormal tissues with an optimized neural network using Particle Swarm Optimization [5]. The experiment was carried out with their private database of mammograms and the classification was done with the Laws Texture Energy Measures extracted from a ROI of dimension 15×15 pixels.

Peng, Mayorga and Hussein in 2015 obtained an accuracy of 96% using an artificial neural network to classify the mammograms from MIAS database [6]. They defined three different categories to carry out the experiment: normal, with presence of a benign tumor and with presence of a malign tumor. A median filter and the seeded region growing algorithm were used to remove the noise of the original images. Then, they extracted 16 features related to the texture properties of the images and five of them were selected. The feature selection algorithm, which is based on the rough-set theory, was developed by the authors.

Mahersia, Boulehmi and Hamrouni in 2015 achieved recognition rates of 97.08% and 95.42% on the MIAS database using a neural network with a Bayesian backpropagation algorithm and an ANFIS system as classifiers, respectively [7]. The breasts were classified into two categories: normal and cancerous. The mammograms from this database were first enhanced, removing the noise and details that may interfere with the recognition of the tumors. Then a generalized Gaussian density model for wavelet coefficients was used as feature extractor.

B. Analysis of Breast Cancer using Deep Learning

Ertosun and Rubin in 2015 used three different architectures of CNNs to locate masses in mammography images [8]. They selected 2420 images from the DDSM dataset and divided these images into training, validation and test sets, containing 80%, 10% and 10% of the images, respectively. They also used cropping, translation, rotation, flipping and scaling techniques to get an augmented training set, in order to improve the generalization ability of the system. The experiment was divided into two stages: the first consisted in the classification of a mammography as containing or not masses and the second in the localization of masses in the images.

Arevalo *et al.* in 2015 obtained 86% of area under the Receiver Operating Characteristic (ROC) curve by classifying mammography mass lesions using a CNN as feature extractor and a SVM as classifier [9]. The data to carry out the experiment was the BCDR-F03 dataset, which is part of the BCDR database. This data was composed by 736 images, 426 containing benign mass lesions and the rest containing malignant lesions. The data augmentation was achieved by flipping and rotating the images. In addition, the mammography images were normalized by the use of global and contrast normalization. The CNN was trained using both dropout and max-norm regularization techniques.

Jiao *et al.* in 2015 obtained an accuracy of 96.7% classifying the breast masses between benign and malign from the DDSM database using a CNN as feature extractor and a SVM as classifier [10]. The images were previously normalized and whitened. On the other hand, the CNN was trained with a subset of ImageNet [11] and the features to perform the classification were extracted from two different layers of the CNN.

Abdel-Zaher and Eldeib in 2015 developed a classifier using the weights of a previously trained deep belief network as the initial parameters for a neural network with Liebenberg

Marquardt learning function [12]. This model was tested on the Wisconsin Breast Cancer Dataset, obtaining an accuracy of 99.68%.

III. Deep Learning for Image Classification

This section is based on the works from Guo *et al.* [13] and LeCun *et al.* [14].

Around 2006, the results obtained by a group of researchers working together in parallel projects in the Canadian Institute for Advanced Research renovated the interest of the community for the deep neural networks. The main four works [15–18], introduced unsupervised learning procedures to pure supervised learning procedures. The objective of each layer in the neural network was to learn the inputs of the previous layer [14]. This approach performed well in comparison with the existent artificial intelligence techniques in tasks such as recognizing handwritten digits, specially when the amount of labeled data was limited [19].

Since the rise of Deep Learning, the CNN model outperformed the fully connected neural networks in tasks related to natural image classification. However, this approach was not seriously used at classification problems until 2012. During six years in which CNNs were laid aside, the methods based on the *Bag of Visual Words* model, that were the state of the art techniques for image classification, were improved by the incorporation of spatial geometry, through the use of spatial pyramids [20].

The turning point for image classification was 2012. In this year, AlexNet, a CNN with five convolutional layers and three fully connected layers developed by Krizhevsky *et al.* [21], outperformed the existing methodologies and won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, almost halving the error rate of the model in second place [11]. This success reflected the new developments in graphic hardware and algorithms [14]: the increased chip processing abilities (GPU units), the use of Rectified Linear Unit (ReLU) as neural activation functions, a novel regularization technique called dropout [22] and the developments of algorithms for data augmentation.

Since the success of AlexNet in 2012, several improvements of this model have been performed. In 2013, Zeiler and Fergus established a technique to analyze the responses of intermediate layers, what enabled them to implement Clarifai, winning the ILSVRC [23].

In 2014, deeper architectures were finally used. VGG [24] and GoogLeNet [25] networks obtained the second and first place in ILSVRC, respectively. The VGG network from Simonyan *et al.* [24] had 13-16 convolutional layers, while GoogLeNet, developed by Szegedy *et al.* [25], had 21 convolutional layers.

In 2015, He *et al.* [26] proposed a model that surpassed for the first time human-level performance on the ImageNet 2012 test dataset, with a network with the same architecture of VGG [24]. In addition, He *et al.* also established a new framework to train deeper networks called the residual learning [27]. They developed ResNet, a 152-layers network and won ILSVRC.

Currently, the researchers are focusing in three main aspects to further improve the performance of Deep Learning models [13]: (a) the implementation of larger networks: ResNet, GoogLeNet and VGG models have shown that the networks with a larger number of layers outperform the simpler ones; (b) the use of multiple networks, where every network can execute all the process independently, so the responses of all the networks are combined in order to obtain the final result; and (c) the introduction of external information from other resources and the use of shallow structures. In this aspect, one of the most important developments is *Regions with CNN Features* method [28], in which the features extracted from a CNN feed a SVM.

Other research projects have focused their efforts on getting a further understanding of what deep neural networks learn, addressing the problem from both a theoretical and a empirical perspective. For instance, Li *et al.* [29] have recently studied convergent learning, aiming to analyze cases in which different neural networks learn similar representations. In this work, they propose a method for quantifying the similarity between deep neural networks and showed that there exist basic features which are learned by multiple networks with the same architectures but different random initialization.

IV. Theoretical Background

A. Convolutional Neural Networks

CNNs are a type of biologically-inspired feed-forward networks characterized by a sparse local connectivity and weight sharing among its neurons. A CNN can also be seen as a sequence of convolutional and subsampling layers in which the input is a set of $H \times W \times D$ images, where H is the height, W is the width and D is the number of channels which, in the case of RGB images corresponds to $D = 3$.

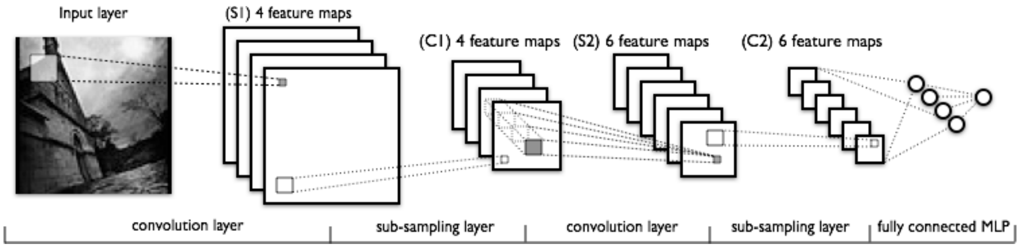


Figure 1: EXAMPLE OF A CNN ARCHITECTURE. TAKEN FROM [30].

A typical convolutional layer (volume) is formed by K filters (kernels) of size $F \times F \times D$, where $F \leqslant H$ and $F \leqslant W$. These filters are usually randomly initialized and are the parameters to be tuned in the training process. Since the size of the filter is generally strictly smaller than the dimensions of the image, this leads to a local connectivity structure among the neurons. Each of this convolutional volumes has an additional hyper-parameter, S, which corresponds with the stride that the filter is going to slide spatially in the image.

Let's denote a particular training example as $X_{H \times W \times D}$ and a convolution filter $W_{F \times F \times D}$. As it is familiar from the usual Multi-Layer Perceptron, it is customary to add a bias term b to each of the linear combinations formed. Finally, a (commonly non-linear) activation function, for example ReLU, is applied to the convolution between the input image and the kernels, which yields an activation map A of the dimensions $1 + \frac{N-F}{S} \times 1 + \frac{N-F}{S} \times 1$:

$$A = f(X \circledast W + b)$$

where $\circledast$ represents the *valid* convolution between the operands and f is the activation function.

Appending the activation maps found by applying K diferent kernels to the input example, an activation volume of dimensions $1 + \frac{N-F}{S} \times 1 + \frac{N-F}{S} \times K$ is obtained. Note that depending on the dimensions of the image, the filter and the size of the stride, the resulting activation volume may reduce its spatial dimensions very quickly. An alternative to control this situation in advance is the use of *padding* techniques to the original image [31].

Finally, in order to perform dimensionality reduction directly on the data, pooling layers are applied to an activation volume or even the input image itself. These layers subsample its inputs, typically with mean or max pooling, over contiguous regions of size $P \times P$.

Figure 1 shows an example of a typical architecture for a CNN in which two convolutional and two pooling layers are applied to the original image. In this case, the extracted features obtained as are fed into a fully connected layer to perform the classification task. Note that it is possible to change the classifier set up at the end of the network with, for example, a SVM or a softmax classifier.

B. Back-propagation Algorithm

The summary presented in this section is heavily based on the Unsupervised Feature Learning and Deep Learning Tutorial [31]. For simplicity, we will illustrate the algorithm assuming that we have a CNN with the input layer followed by a convolutional volume, a pooling layer and finally a fully connected layer.

Let's denote by $\delta^{(l+1)}$ the error term in the $(l+1)$-th layer in the network with labeled training data (x, y), parameters (W, b) and cost function $J(W, b; x, y)$. If the l-th layer is densely connected to the former, the error for this layer can be computed by:

$$\delta^{(l)} = \left((W^{(l)})^t \delta^{(l+1)} \right) \bullet f'(z^{(l)})$$

where $\bullet$ represents element-wise multiplication and f is the activation function.

The gradients are:

$$\nabla_{W^{(l)}} J(W, b; x, y) = \delta^{(l+1)} (a^{(l)})^t$$

$$\nabla_{b^{(l)}} J(W, b; x, y) = \delta^{(l+1)}$$

If the l-th layer is a convolutional and subsampling layer, then the error is propagated through as:

$$\delta_k^{(l)} = upsample \left((W_k^{(l)})^t \delta_k^{(l+1)} \right) \bullet f'(z_k^{(l)})$$

where k indexes the filter number and the $upsample$ function propagates the error through the pooling layer by calculating the error related to each input unit.

Finally, the gradient for each filter map can be found by:

$$\nabla_{W_k^{(l)}} J(W,b;x,y) = \sum_{i=1}^{m} (a_i^{(l)}) \ast rot90(\delta_k^{(l+1)}, 2)$$

$$\nabla_{b_k^{(l)}} J(W,b;x,y) = \sum_{a,b} \left(\delta_k^{(l+1)}\right)_{a,b}$$

where $a^{(l)}$ is the input to the l-th layer and $rot90(A,k)$ rotates the input array A counterclockwise by $k*90$ degrees.

C. Linear Support Vector Machines

Suppose we are given a training data set of size n examples of the form:

$$\{(X_1, y_1), (X_1, y_1), ..., (X_n, y_n)\}$$

where each y_i is either 1 or -1 and each X_i is a p-dimensional vector. Thus, assuming that the data is linearly separable, we want to find the hyperplane that separates the group of $\{X_i\}$ for which $y_i = 1$ from those for which $y_i = -1$ so that the distance between the hyperplane and the nearest point from either group is maximized. For that reason, it is also called a *maximum-margin classifier*. This can be formally expressed as:

$$\min_{w \neq 0, b} \frac{1}{2} ||w||^2$$

$$\text{s.t.} \quad y_i(w^t X_i + b) \geqslant 1 \quad (i = 1, 2, ..., n)$$

. Recall that $\frac{b}{||w||}$ represents the separation of the hyperplane from the origin along the normal vector w when the hyperplane is expressed as $wX - b = 0$.

D. Confusion Matrix

Consider a classification problem with only two classes: positive (P) and negative (N). For every training example, there are only four possible outcomes. If the training example is positive and the prediction is positive, we call it a *true positive*; and if the prediction is negative, it is called a *false negative*. On the other hand, if the training example is negative and it is classified as negative, it is called a *true negative*; otherwise, it is a *false positive*. Table I displays an example of a confusion matrix for a two-class problem.

Table I: CONFUSION MATRIX

		Predicted Class	
		P	**N**
Actual Class	**P**	True Positives	False Negatives
	N	False Positives	True Negatives

A confusion matrix is a tool that allows to visualize the performance of a classifier in a supervised learning problem. By means of this matrix it is possible to asses whether the system is commonly confusing pairs of classes. In the aforementioned problem, the confusion matrix summarizes the four possible outcomes from the classifier [32].

V. PRELIMINARY EXPERIMENTS

Prior to the final implementation of our diagnosis system, several experiments were carried out. The purpose of this experiment was to evaluate the performance of a system with a CNN as feature extractor and a SVM as classifier, previous to the implementation of this strategy in our core problem.

A. Data and Data Augmentation

The images to carry out this experiment were retrieved from the Caltech-101 database [33]. This database contains pictures from 101 different categories. The intensity of every pixel is between 0 and 255, in which 0 represents black and 255 represents white. For our purpose, only the images from four categories were used: airplanes, faces, motorbikes and watches. After the selection of these four categories, every image belonging to these sets were flopped, so the length of the dataset of interest was duplicated.

B. Experiment Description

From the dataset generated for the four categories of interest, 100 images of each category were randomly extracted and were divided into training (60 images) and test (40 images) sets. The extraction of the characteristics to carry out the classification stage was done with two pretrained CNNs on the ImageNet database [11]: AlexNet [21] and VGG-F [34]. The features chosen were the activations of the last convolutional layer. In addition, we selected a SVM as the classifier, which was trained using the features obtained for the pictures of the training set.

Finally, the classification accuracy of the two trained SVMs was evaluated with the 160 images (40 for each category) corresponding to the test set.

C. Results

In Tables II and III the confusion matrices obtained using pretrained feature extractors are presented. These confusion matrices correspond to the results given by the SVM classifier on the test set. The accuracies achieved using VGG-F and AlexNet as feature extractors were 98.75% and 99.38%, respectively, which correspond to 158 and 159 well classified examples, taking into account that the total number of examples was 160. Results show that this approach of using a SVM in conjunction with a CNN obtains a good performs at classifying natural images.

Table II: CONFUSION MATRIX FOR CALTECH101 TEST SET PREDICTIONS AND FEATURE EXTRACTION USING VGG.

		Target				
		Airplanes	**Faces**	**Motorbikes**	**Watches**	**Total**
Output	**Airplanes**	**97.5**	0	0	2.5	**97.5**
	Faces	0	**97.5**	0	2.5	**97.5**
	Motorbikes	0	0	**100**	0	**100**
	Watches	0	0	0	**100**	**100**
	Total	**97.5**	**100**	**100**	**95.24**	**98.75**

Table III: CONFUSION MATRIX FOR CALTECH101 TEST SET PREDICTIONS AND FEATURE EXTRACTION USING ALEXNET.

		Target				
		Airplanes	**Faces**	**Motorbikes**	**Watches**	**Total**
Output	**Airplanes**	**97.5**	0	0	2.5	**97.5**
	Faces	0	**100**	0	0	**100**
	Motorbikes	0	0	**100**	0	**100**
	Watches	0	0	0	**100**	**100**
	Total	**97.5**	**100**	**100**	**97.56**	**99.38**

VI. MAMMOGRAMS CLASSIFICATION

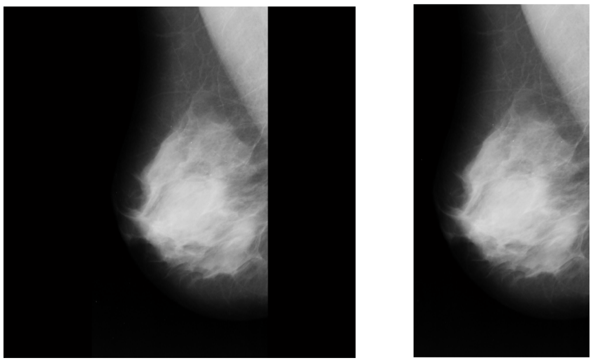
A. Data

The mammograms used for the commitment of this work were retrieved from the database of the MIAS [2], which is known as mini-MIAS since the images of the original MIAS database has been reduced to 200 micron pixel edge and the dimension of the mammograms has been fixed to 1024×1024 pixels. This database contains 322 mammograms and the intensity of every pixel is between 0 and 255. This database also includes information about the class and the severity of abnormalities that may be present in the mammograms, as well as the coordinates of the center of these abnormalities.

It must be mentioned that we only used the mammogram images and the required information to divide the mammograms into three categories: patients with benign, malign or without tumor.

B. Data Preprocessing

1) Mammogram Cropping: Mammograms contain black zones in the borders which may difficult the classification task. For this reason, we designed an algorithm to eliminate these black zones based on the sum of the pixels over the column. The algorithm finds the first column, say C_l, on the left of the mammogram in which the sum of the pixels exceeds a given threshold P. Now, from this point, the algorithm finds the first column, C_r, in which the sum of the pixels is not greater than P. Then, the new image is the one enclosed between C_l and C_r. This algorithm was applied to every mammogram of the 322 retrieved from the aforementioned database, taking $P = 500$. An example of the images obtained at this stage is illustrated in Figure 2, in which Figure 2a is an original mammogram of the mini-MIAS database and Figure 2b is the resultant image after the application of the cropping algorithm.



(a) Original.

(b) Cropped.

Figure 2: MAMMOGRAMS OBTAINED AFTER THE CROPPING STAGE.

2) Data Augmentation: Due to the lack of mammograms corresponding to malign tumors (51 out of 322), it was necessary to perform a data augmentation operation in order to get a balanced dataset with at least 600 mammograms. For this purpose, after the cropping procedure, every resultant mammogram was rotated -90°, 90° and 180°. The label assigned to the three artificially generated mammograms corresponded with the label of the original image.

C. Feature Extraction

As mentioned in Sections II and III, CNNs are being widely used to carry out image classification tasks because of their outstanding performance in comparison with other classification techniques. For this reason, they have become an emerging alternative in the computer-aided diagnosis field.

In this work, two different experiments were carried out using a CNN previously trained on the ImageNet database as feature extractor. In the first experiment, the CNN used was AlexNet [21] while in the second experiment the CNN used was VGG-F [34]. The features selected to perform the classification of mammograms were the activations of the last convolutional layer of the CNN. Then, in both cases, 4096 features have been extracted for each image.

In order to feed both pretrained CNNs with the cropped images, it was necessary to convert every mammogram into a three channel image by repeating the single channel three times. Then, the resulting image was resized depending on the input dimension of the CNN (227 × 227 pixels for AlexNet and 224 × 224 pixels for VGG-F). Finally, the average image (which is included with the tuned parameters of the pretrained models used) was subtracted from the resized image.

D. Classification

The goal of the system was to distinguish between three classes: patients with benign, malign or without tumor, then based on the works [3, 9, 10], we decided to adopt a SVM as our classifier.

In order to evaluate our methodology, 120 and 80 mammograms of each category were selected from the augmented dataset to define the training and test stages of the SVM, respectively. Hence, our training set was composed by 360 mammograms and our testing set by 240 mammograms.

To carry out the training of the SVM, each of the 360 mammograms selected was given as the input for the CNN and the features obtained at this step became the inputs for the SVM. Then, using the Statistics and Machine Learning Toolbox from MATLAB, the SVM was trained.

The classification accuracy of the trained SVM was evaluated with the 240 mammograms belonging to the test set, following the same process described above for the extraction of the features for every mammogram.

E. Results

In Table IV the confusion matrix obtained using AlexNet as feature extractor without augmenting the dataset is shown. This experiment was carried out with a training set of 30 mammograms per category and a test set of 20 per category. This confusion matrix is based on the response of the system on the test set and this low accuracy rate of only 35%, which corresponds to 21 well classified mammograms of the 60 that conformed the test set, is an evidence of the necessity of performing a data augmentation operation.

Table IV: CONFUSION MATRIX FOR MIAS TEST SET PREDICTIONS AND FEATURE EXTRACTION USING ALEXNET.

		Target			
		Benign	**Malign**	**Normal**	**Total**
Output	**Benign**	**36.53**	48.12	15.35	**36.53**
	Malign	27.39	**56.12**	16.49	**56.12**
	Normal	31.34	56.29	**12.36**	**12.36**
	Total	**38.35**	**34.96**	**27.97**	**35.01**

In Tables V and VI the confusion matrices corresponding to the response of the system when AlexNet and VGG-F CNNs are used as feature extractors in conjunction with a SVM as classifier are exhibited. Table V shows the accuracy of the system on the test set after performing the data augmentation when the CNN used is AlexNet.

Table V: CONFUSION MATRIX FOR AUGMENTED MIAS TEST SET PREDICTIONS AND FEATURE EXTRACTION USING ALEXNET.

		Target			
		Benign	**Malign**	**Normal**	**Total**
Output	**Benign**	**61.79**	20.33	17.87	**61.79**
	Malign	18.79	**61.75**	19.46	**61.75**
	Normal	22.88	20.67	**56.46**	**56.46**
	Total	**59.73**	**60.10**	**60.20**	**60.01**

On the other side, Table VI shows the response of the system when VGG-F is the feature extractor. It can be noted that the performance of the system has dramatically increased after artificially augmenting the dataset: from 35% to 60.01% and 64.52% using AlexNet and VGG-F, respectively.

Table VI: Confusion matrix for Augmented MIAS test set predictions and feature extraction using VGG.

		Target			
		Benign	**Malign**	**Normal**	**Total**
Output	**Benign**	**63.63**	18.45	17.92	**63.63**
	Malign	17.86	**64.37**	17.77	**64.37**
	Normal	16.91	17.54	**65.55**	**65.55**
	Total	**64.66**	**64.14**	**64.75**	**64.52**

VII. Conclusions

Based on the results obtained in this work, the Deep Learning approach, particularly using pretrained CNNs as feature extractors, is a promising methodology when addressing the problem of diagnosing breast cancer with mammogram images. Since in this context the reliability of the system is highly relevant, it is desirable to increase the achieved 64.52% test accuracy. This outcome could be improved by cropping the image to a specific ROI in which a tumor could be located; via fine-tuning of the final layers or training the whole network parameters. The results of additional experiments using a subset of the Caltech-101 database, for which a 99.38% test accuracy was obtained, exhibit the relevance of the similarity between the data used to train the model and the particular application intended. Additionally, it is worth noting the impact of the data augmentation process and the balance of the number of examples per class on the performance of the system.

Future research could be focused on the evaluation of the following techniques:

- To extract features from multiple layers of the CNN instead of only using the activations obtained from the last convolutional layer.
- To use different pretrained CNNs as feature extractors, such as GoogLeNet [25] or ResNet [27].
- To include a feature selection phase in which the best extracted features from a CNN could be selected to perform the classification of the mammograms.
- To test other classifier structures: neural networks, fuzzy inference systems or clustering techniques.

References

[1] World Health Organization, “Breast cancer: prevention and control,” Jan 2016, [Accessed: 19- May- 2016]. [Online]. Available: http://www.who.int/cancer/detection/breastcancer/en/

[2] J. Suckling, J. Parker, D. Dance, S. Astley, I. Hutt, C. Boggis, I. Ricketts, E. Stamatakis, N. Cerneaz, S. Kok *et al.*, “The mammographic image analysis society digital mammogram database,” in *Exerpta Medica. International Congress Series*, vol. 1069, 1994, pp. 375–378.

[3] M. A. Alolfe, W. A. Mohamed, A. B. M. Youssef, A. S. Mohamed, and Y. M. Kadah, “Computer aided diagnosis in digital mammography using combined support vector machine and linear discriminant analyasis classification,” in *2009 16th IEEE International Conference on Image Processing (ICIP)*, Nov 2009, pp. 2609–2612.

[4] Z. Wang, G. Yu, Y. Kang, Y. Zhao, and Q. Qu, “Breast tumor detection in digital mammography based on extreme learning machine,” *Neurocomputing*, vol. 128, pp. 175 – 184, 2014.

[5] J. Dheeba, N. A. Singh, and S. T. Selvi, “Computer-aided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach ,” *Journal of Biomedical Informatics*, vol. 49, pp. 45 – 52, 2014.

[6] W. Peng, R. Mayorga, and E. Hussein, “An automated confirmatory system for analysis of mammograms,” *Computer Methods and Programs in Biomedicine*, vol. 125, pp. 134 – 144, 2016.

[7] H. Mahersia, H. Boulehmi, and K. Hamrouni, “Development of intelligent systems based on Bayesian regularization network and neuro-fuzzy models for mass detection in mammograms: A comparative analysis ,” *Computer Methods and Programs in Biomedicine*, vol. 126, pp. 46 – 62, 2016.

[8] M. G. Ertosun and D. L. Rubin, “Probabilistic visual search for masses within mammography images using deep learning,” in *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*, Nov 2015, pp. 1310–1315.

[9] J. Arevalo, F. A. González, R. Ramos-Pollán, J. L. Oliveira, and M. A. G. Lopez, “Convolutional neural networks for mammography mass lesion classification,” in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug 2015, pp. 797–800.

[10] Z. Jiao, X. Gao, Y. Wang, and J. Li, “A deep feature based framework for breast masses classification,” *Neurocomputing*, vol. 197, pp. 221 – 231, 2016.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[12] A. M. Abdel-Zaher and A. M. Eldeib, “Breast cancer classification using deep belief networks,” *Expert Systems with Applications*, vol. 46, pp. 139 – 144, 2016.

[13] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, “Deep learning for visual understanding: A review,” *Neurocomputing*, 2015.

[14] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] G. E. Hinton, “What kind of graphical model is the brain?” in *International Joint Conference on Artificial Intelligence*, vol. 5, 2005, pp. 1765–1775.

[16] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm

for deep belief nets." *Neural computation*, vol. 18, no. 7, pp. 1527–54, 2006.

[17] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy Layer-Wise Training of Deep Networks," *Advances in neural information processing systems*, vol. 19, no. 1, p. 153, 2006.

[18] C. P. Marc'Aurelio Ranzato, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model," in *Advances in Neural Information Processing Systems*, vol. 19, 2006, pp. 1137–1144.

[19] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3626–3633.

[20] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2169–2178.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances In Neural Information Processing Systems*, pp. 1–9, 2012.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the ECCV International Workshop on Statistical Learning in Computer Vision*. Springer, 2014, pp. 818–833.

[24] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Proceedings of the ICLR*, pp. 1–14, 2015.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *Proceedings of the ICCV*, pp. 1–11, 2015.

[27] ——, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[29] Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. Hopcroft, "Convergent Learning: Do different neural networks learn the same representations?" in *ICLR*, 2016, pp. 1–21.

[30] LISA Lab, *My LeNet*, Retrieved 2016-5-26. [Online]. Available: http://deeplearning.net/tutorial/_images/mylenet.png

[31] A. Ng, J. Ngiam, C. Y. Foo, Y. Mai, and C. Suen, "UFLDL tutorial," *http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial*, 2010.

[32] T. Fawcett, "An introduction to ROC analysis ," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861 – 874, 2006, ROC Analysis in Pattern Recognition .

[33] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," 2004.

[34] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference*, 2014.