Classification of breast cancer abnormalities using mixed machine learning and deep learning methods

L. Guillermo Rodriguez-Lopez[a], J. Angel Gonzalez-Fraga[b], and Vitaly Kober[a,c]

[a]Department of Computer Science, CICESE, Ensenada, B.C. 22860, México.
[b]Facultad de Ciencias. Universidad Autónoma de Baja California, Ensenada, B.C. 22860, México.
[c]Department of Mathematics, Chelyabinsk State University, Russian Federation.

ABSTRACT

Mammography is a standardized imaging technique crucial for the early detection of breast cancer, primarily aimed at identifying abnormalities, or 'findings' in the breasts that cannot be detected through palpation. This study proposes different models for classifying breast abnormalities, integrating machine learning and deep learning methods to improve classification rates. The proposed methodology involves several key steps: preprocessing of image datasets, training of base classification models, and construction of a meta-classifier. By enhancing the performance of individual classifiers, the model is benchmarked against various machine learning models. The evaluation of this method is conducted using the CBIS-DDSM mammography dataset, demonstrating its effectiveness in improving classification accuracy and reliability. The hybrid approach leverages convolutional neural networks (CNNs) such as VGG16, VGG19, and DenseNet121 for feature extraction, followed by machine learning algorithms for final classification. The VGG16 network, combined with machine learning techniques, aimed to surpass the results obtained by VGG19. Ensemble methods, particularly Voting and Stacking Classifiers, showed that combining VGG16 and DenseNet121 yielded the highest accuracy of 91.66%. These findings underscore the potential of hybrid models in breast cancer classification, offering significant improvements over single classifiers and providing valuable insights for future research in medical image analysis.

Keywords: breast cancer, mammography, convolutional neural networks, machine learning, feature extraction, stacking

1. INTRODUCTION

Cancer is a term used to designate a wide range of diseases capable of affecting any part of the human body. This disease involves the rapid multiplication of abnormal cells, which can extend beyond their place of origin, invading adjacent parts of the body or other organs. This process is called metastasis and is the main cause of death worldwide, with breast, lung, colon, rectum, and prostate cancers being the most common.[1] According to the World Health Organization, cancer accounted for nearly 10 million deaths in 2020, equivalent to one in six deaths globally. Breast cancer, also referred to as breast carcinoma, is the most frequent cancer in women, with high incidence and mortality rates. In 2020, breast cancer alone was responsible for 2.3 million new cases and 685,000 deaths.

Mammography is an imaging diagnostic technique obtained through X-rays. Currently, digital mammography has been established as the only approved method for conducting this study,[2] as its use has been proven to reduce mortality, thereby becoming the 'gold standard'[3] for imaging examination in the early detection of breast cancer.

The main factors related to breast cancer, as found in mammograms, are masses and calcifications (abnormalities), which are the focus of this research. Early detection and accurate classification of these abnormalities are crucial for effective treatment and improved survival rates. Conventional diagnostic methods often suffer from human visual limitations, dense breast tissue, and image noise, leading to false positives and negatives. To

Further author information:
J.A. G-F. e-mail: angel_fraga@uabc.edu.mx
V.K. e-mail: vkober@cicese.mx

Applications of Digital Image Processing XLVII, edited by Andrew G. Tescher, Touradj Ebrahimi,
Proc. of SPIE Vol. 13137, 131371H · © 2024 SPIE · 0277-786X · doi: 10.1117/12.3028365

address these challenges, various deep learning models, especially Convolutional Neural Networks (CNNs), have been developed for automatic detection and classification of breast cancer.

In this study, we use the Curated Breast Imaging Subset of the Digital Database for Screening Mammography (CBIS-DDSM) dataset, which contains verified pathology information and is divided into normal, benign, and malignant cases.

2. METHODS

Our approach involves the use of CNNs such as VGG16, VGG19, and DenseNet121 to classify abnormalities into masses and calcifications. These networks are pretrained on large image datasets like ImageNet and are fine-tuned for the specific task of breast cancer classification. The pretrained network VGG16 act as feature extractors, and the extracted features are subsequently used by machine learning algorithms to achieve the final classification.

By leveraging the strengths of CNNs and transfer learning, this study aims to improve the accuracy and reliability of breast cancer classification, ultimately aiding radiologists in making more precise diagnoses and reducing the risk of misdiagnosis.

2.1 Dataset

The Curated Breast Imaging Subset of the Digital Database for Screening Mammography[4] (CBIS-DDSM) was used in these experiments. This dataset is one of the largest publicly accessible resources for breast imaging and is an updated and standardized version of the original DDSM. CBIS-DDSM provides segmentations of regions of interest (ROIs) or patches of abnormalities (masses and calcifications), along with masks for these ROIs and the full mammograms in DICOM format with accompanying data dictionaries in CSV format. Despite some limitations noted in Ref. 5, the primary advantage of this dataset is that the pathologies ('Benign', 'Malignant', 'Benign Without Callback') associated with the abnormalities are verified through biopsies. 'Benign Without Callback' indicates that the radiologist did not find sufficient evidence to warrant a biopsy based on the mammogram but recommended regular monitoring.

The CBIS-DDSM dataset contains 2,620 scanned film mammography studies, categorized into normal, benign, and malignant cases. The scans are of large size, approximately 5000×300 pixels. The dataset used in this study includes a total of 3,568 images, with 2,968 images designated for training and 600 independent images for testing. The distribution of the images is illustrated in Figure 1. To evaluate the classifiers, the test set was kept separate. Out of the 2,968 training images, 682 images labeled as 'Benign Without Callback' were excluded, resulting in a total of 2,286 images used for training. A stratified validation set comprising 20% of the training set was created and used initially to monitor the training of the neural networks. All image patches were resized to 224×224 pixels. Figure 2 illustrates examples of masses and calcifications from the CBIS-DDSM dataset.

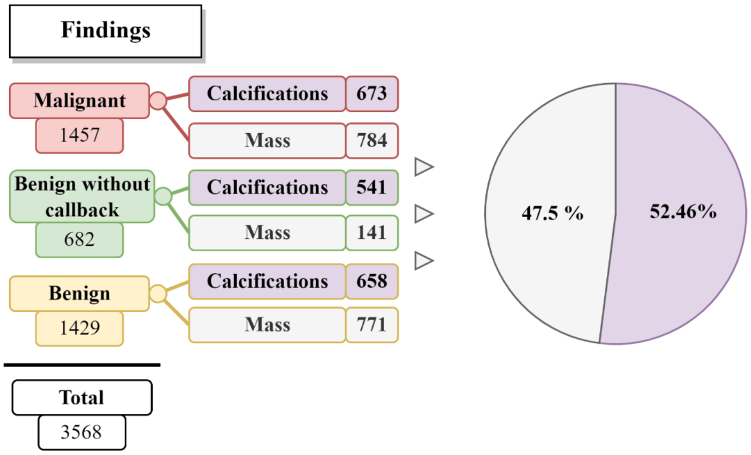

Figure 1: Statistics CBIS-DDSM. Image modified from Ref. 5.

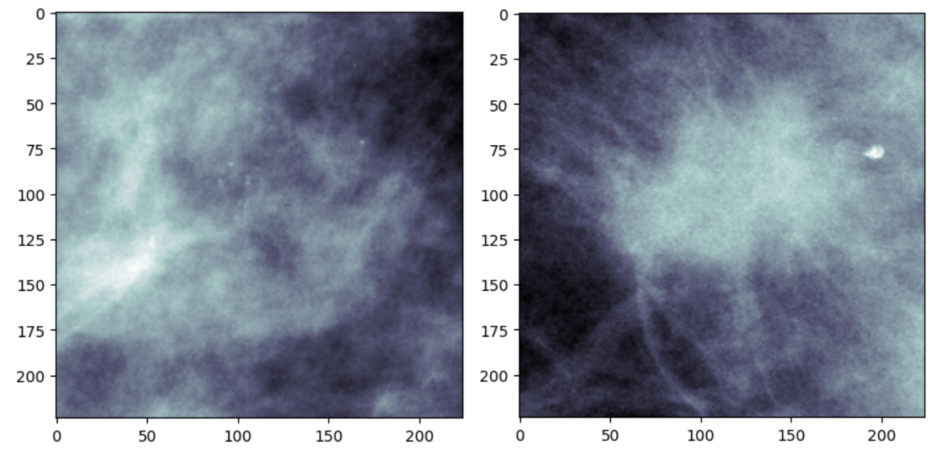

Figure 2: Calcifications and masses used in the experiments.

2.2 Machine Learning Classifiers

As mentioned above, our approach involves the use of VGG16, VGG19, and DenseNet121 models to classify mass and calcification abnormalities in the CBIS-DDSM dataset. These models are pretrained on large image datasets such as ImageNet and tuned for the specific task of breast cancer classification. The pretrained networks act as feature extractors, and the extracted features are then used by machine learning algorithms to achieve the final classification. Data augmentation techniques using TensorFlow's ImageDataGenerator, including horizontal and vertical flips, rotations, zooms, and intensity changes, are used to improve the training process and model performance.

Gaussian Naive Bayes. Naive Bayes classifiers utilize Bayes' theorem with the 'naive' assumption of conditional independence between every pair of features given the class variable. Gaussian Naive Bayes assumes that the probability distribution of the features is Normal.

Logistic Regression. Logistic Regression is used in classification tasks to estimate the probability that an instance belongs to a class. It is also known as *logit regression*.

Stochastic Gradient Descent (SGD). Gradient Descent is an optimization algorithm used to minimize the cost function. In classification problems, various options are available, but in these binary classification experiments, the 'hinge' function was used.

Ridge Classifier. This classifier applies L2 regularization to the linear regression algorithm for classification tasks. Essentially, the Ridge Classifier is a variant of Logistic Regression, but instead of optimizing the logistic loss function, it optimizes a quadratic loss function (as in linear regression). This regularization helps improve the model's generalization by preventing overfitting to the training data.

Support Vector Classifier (SVC). This model can perform classification on both linear and non-linear data. It uses support vectors to separate the data by utilizing various instances of the training set. The strength of this model lies in its ability to classify non-linear data, supported by 'kernels' to separate the data effectively.

Decision Trees. Decision Trees are algorithms capable of performing well on complex datasets. Like the Support Vector Machine, these algorithms excel in non-linear datasets. Several versions of this algorithm exist, such as 'ID3' and 'CART'.

Ensembles. A group or set of classifiers is called an *ensemble*, and the technique is known as *ensemble learning*. An *ensemble method* combines two or more algorithms to perform regression or classification tasks. Two main recommendations for better results are: combining linear and non-linear algorithms, as used in these experiments, including convolutional neural networks; and training these classifiers on different subsets of the dataset.

Voting Classifiers. The idea is simple. Given different classifiers, the final prediction is obtained by averaging all predictions. There are two ways to do this. Since labels are integer representations, the final prediction is obtained by averaging these representations (hard voting). The other method is averaging the probabilities associated with the labels (soft voting). It is recommended to train the classifiers on different subsets of the dataset, which requires a large dataset, not always possible.

Bagging Classifier. This algorithm enhances the predictions of a single base classifier by sampling the training set and training the base classifier on these subsets. When sampling is done with replacement, it is called *bagging*; without replacement, it is called *pasting.* The final prediction is made by averaging the predictions of several base classifiers trained on these subsets. The **Random Forest Classifier** is a bagging model using Decision Tree classifiers as the base classifier.

AdaBoost Classifier. This classifier starts with a base classifier. First, the base classifier trains and generates some errors. Then, the same classifier focuses on better classifying the misclassified instances. This process repeats, gradually improving the ensemble. **Gradient Boosting** is an AdaBoost variant with a Decision Tree as the base classifier.

Stacking Classifier. This approach is convenient when classifiers can estimate probabilities. Instead of using a Voting Classifier, probabilistic predictions are concatenated by columns, and the labels are copied, creating a new dataset called a *'blending set'.* Then, a classifier, called a *meta-learner* or *meta-classifier*, aggregates the predictions of all predictors in an ensemble.

For more detailed information on the parameters and algorithms described in this subsection, it is recommended to consult the official documentation of **scikit-learn** and the referenced source [6].

2.3 Evaluation metrics

In this subsection, the most common evaluation metrics used for breast cancer classification are explained: Confusion Matrix, Accuracy, Precision, Recall, Matthews Correlation Coefficient and F1-Score.

Confusion Matrix: A binary classifier can classify an instance into either of the two classes it was trained on, but this does not mean it does so with 100% accuracy. There is often some ' 'confusion' in assigning instances to the correct class. Hence, the confusion matrix exists as a visual and matrix representation (2 × 2, four quadrants) of all the classifications of instances. The true labels are positioned on the left side of the matrix, and the predicted labels are at the base. The top-left quadrant is defined as true negative (TN), the bottom-left as false negative (FN), the top-right as false positive (FP), and the bottom-right as true positive (TP). From these definitions, various metrics arise.

Accuracy (Acc). This is the most commonly used metric and measures the number of correctly classified instances. It is the sum of the diagonal (TN + TP) divided by the total number of classified instances.

$$Acc = \frac{TN + TP}{TN + FN + FP + TP} \tag{1}$$

Precision. This is defined as the accuracy of positive predictions.

$$Precison = \frac{TP}{TP + FP} \tag{2}$$

Recall. Also known as Sensitivity or True Positive Rate (TPR), this metric measures the proportion of positive instances correctly detected by the classifier.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Matthews Correlation Coefficient (MCC). This correlation coefficient ranges from -1 to 1, where 1 indicates the best agreement between actual data and predictions. It takes all four quadrants into account and is particularly useful for imbalanced datasets.

$$MCC = \frac{TN \times TP - FN \times FP}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{4}$$

F1-Score (F1). This metric combines Precision and Sensitivity into a single measure. The best possible value is 1, while the worst is 0.

$$F1 = \frac{2}{\frac{1}{Precison} + \frac{1}{Recall}} \tag{5}$$

2.4 Deep feature extraction

A Convolutional Neural Network (CNN) is a specialized type of multilayer deep neural network composed primarily of convolutional layers, pooling layers, and classification layers. The convolutional layers perform convolution operations and non-linearity operations using Rectified Linear Units (ReLUs), which add non-linearity to the model. The pooling layers reduce the size of the output from the convolutional layers through operations like max pooling. The network parameters are updated using the backpropagation algorithm, which adjusts weights and biases .[7]

When performing deep learning feature extraction, we treat the pre-trained network as an arbitrary feature extractor. This involves allowing the input image to propagate forward through the network, stopping at a pre-specified layer, and taking the outputs of that layer as our features. This method allows us to utilize the robust, discriminative features learned by the CNN for recognizing classes the network was not originally trained on.

2.5 Proposed approach

This study evaluates a hybrid approach, utilizing Convolutional Neural Networks (CNNs) for feature extraction followed by machine learning algorithms for classification. The convolutional networks VGG16, VGG19, and DenseNet121 were trained to classify abnormalities between masses and calcifications, closely following the parameters provided by Ref. 8. For these models, the transfer learning (TL) technique was employed, initially using weights from models pretrained on the ImageNet dataset, which contains numerous real-world object images. Subsequently, fine-tuning (FT) was performed on the VGG16 and VGG19 networks up to the fourth convolutional block. Our approach in both FT and TL involves removing the dense and output layers, known as fully connected layers (FC), from each of the CNNs used for classification in the original ImageNet dataset and customizing new ones. This allows us to reuse the feature extraction part of the model while adding new layers suited to the new classification task of breast abnormalities. The layers we added were consistent across all CNNs: a dropout layer with 0.5, a dense layer, and a classification layer with sigmoid activation for the binary classification task. The only parameter that varied was the number of neurons in the dense layer of each CNN. Initially, we experimented with 64 neurons, then 128, and continued doubling the number of neurons until reaching 1024. The dense layer with 1024 neurons provided the best results for the VGG16 and DenseNet121 models, while 128 neurons were optimal for VGG19.

After training, we utilized the VGG16 network to extract features in two different ways. First, we removed its fully connected layer, and the feature map from the last pooling layer was converted into a vector. The second way to leverage this CNN as a feature extractor is to take the output of one of the fully connected (FC) layers as a feature vector.

3. RESULTS

The results of the experiments conducted are presented below. First, in Sec. 3.1, the results of the CNNs used as classifiers are shown. In Sec. 3.2, the feature map from the VGG16 network was converted into a vector to serve as input for the machine learning models. Subsequently, in Sec. 3.2.2, a dense layer with 1024 neurons in the VGG16 network, was used as a feature extractor. Finally, in Sec. 3.3.1 and Sec. 3.3.2, the results of combining the outputs of each of the CNNs are presented.

3.1 CNNs as classifiers

The individual results and confusion matrices obtained by each of the CNNs in the classification problem are shown in Tab.1 and Figure 3. It can be observed that the best result was obtained by the VGG19 network misclassifying 58 images. The detailed metrics further highlight the superior performance of VGG19, with the highest accuracy, F1 score, MCC, and precision among the three CNNs evaluated. Specifically, VGG19 achieved an accuracy of 90.33%, a recall of 84.55%, an F1 score of 88.3%, an MCC of 0.8032, and a precision of 92.4%. In comparison, VGG16 and DenseNet121 also performed well but fell short of VGG19's performance.

Throughout this document, results equal to or better than those obtained by the VGG19 network are marked in blue, and the model with the fewest misclassified images is marked in red.

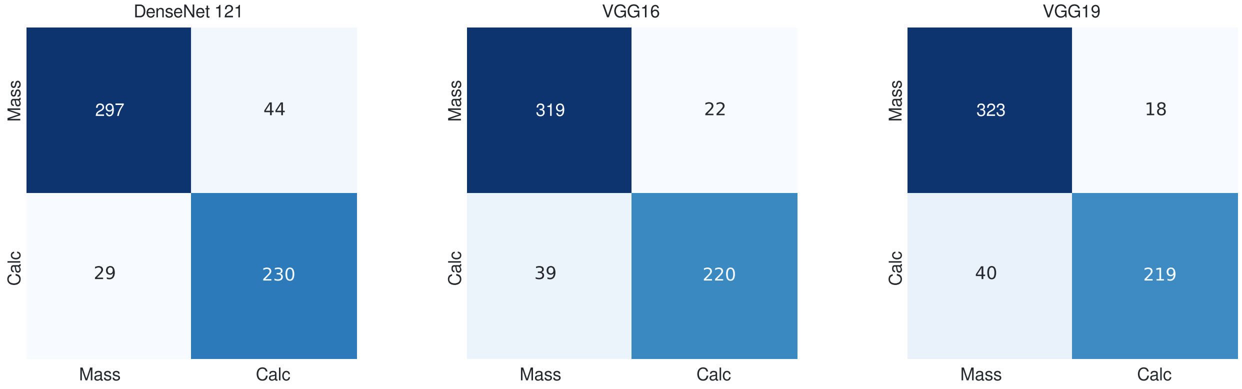

Figure 3: Individual results of each CNN.

Table 1: Individual Results.

Classifier	Accuracy	Recall	F1	MCC	Precision	Misclassified Images
DenseNet 121	0.8783	0.8880	0.8635	0.7547	0.8394	73
VGG16	0.8983	0.8494	0.8782	0.7924	0.9090	61
VGG19	0.9033	0.8455	0.8830	0.8032	0.9240	58

3.2 VGG16 as a Feature Extractor

VGG16 [9] is one of the most widely used deep learning models. The VGG16 architecture has 16 weight layers, including thirteen convolutional layers and three fully connected (FC) layers. Max pooling layers always follow convolutional layers. The input for VGG16 is fixed at 224×224 pixels. In VGG16, the first two FC layers have 4096 channels, and the last one has 1000 channels, representing the number of class labels in the ImageNet dataset. The output layer is a softmax layer, which provides the probability for each class label for the input image.

As mentioned in Sec.2.5, we removed our proposed FC layer from the VGG16 network to utilize its feature map. In the case of VGG16, an image is allowed to forward propagate to the final max-pooling layer (prior to the fully connected layers), and the activations at that layer are extracted. The output of the max-pooling layer has a volume shape of $(7, 7, 512)$ (feature map), which is flattened into a feature vector of $25,088$ dimensions $(7 \times 7 \times 512 = 25,088)$. These features can then be used to train a standard machine learning model.

3.2.1 Feature map as Input Vector

After training the VGG16 network, it was used as a feature extractor, resulting in a feature vector of $25,088$ dimensions. These vector was standarized before being input into any machine learning model. In this experiment, none of the machine learning models used achieved an accuracy greater than 70%, with most models biased towards one of the two classes, including SVM, Random Forest, Gradient Boost, etc. Surprisingly, one of the least biased models was the basic Naive Bayes classifier, which achieved an accuracy of 65.16%. This Naive Bayes classifier was then used as the base classifier in an AdaBoost model to enhance its classifications, increasing its classification rate to 81.33%. The individual results and confusion matrices obtained by the Naive Bayes and AdaBoost (Naive Bayes) classifiers are shown in Table 2 and the confusion matrices in Figure 4. The Naive Bayes classifier misclassifying 209 images. The AdaBoost (Naive Bayes) classifier significantly improved the performance, achieving an accuracy of 81.33%, a recall of 78.76%, an F1 score of 78.46%, an MCC of 0.6199, and a precision of 78.16%, reducing the number of misclassified images to 112.

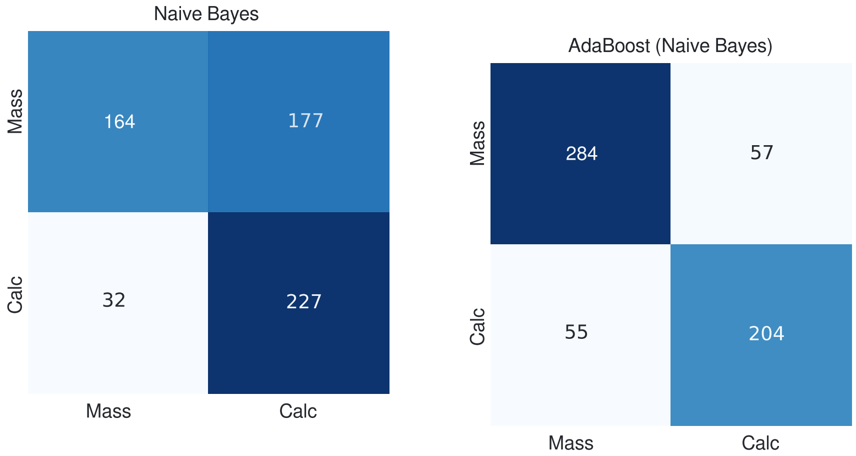

Figure 4: Results of the Naive Bayes model with feature extraction.

Table 2: Naive Bayes Results.

Classifier	Accuracy	Recall	F1	MCC	Precision	Misclassified Images
Naive Bayes	0.6516	0.8764	0.6847	0.3774	0.5618	209
AdaBoost (Naive Bayes)	0.8133	0.7876	0.7846	0.6199	0.7816	112

3.2.2 Dense Layer as Input Vector

As mentioned in Sec.2.5, we utilized the 1024-neurons dense layer from the FC of the VGG16 network as the output vector. This feature vectors was standardized before being input into a machine learning model. The experiments yielded models biased towards each class, as well as a Logistic Regression model, which achieved an accuracy of 90.33%, matching the performance of the VGG19 network and improving upon the accuracy obtained solely by the VGG16 network.

From the confusion matrices shown in Figure 5 and the detailed metrics in Table 3, it is evident that the Logistic Regression model outperformed other models like AdaBoost (SVC) and SGD. The Logistic Regression model misclassified 58 images, achieving the highest accuracy of 90.33%, a recall of 84.55%, an F1 score of 88.30%, an MCC of 0.8032, and a precision of 92.40%.

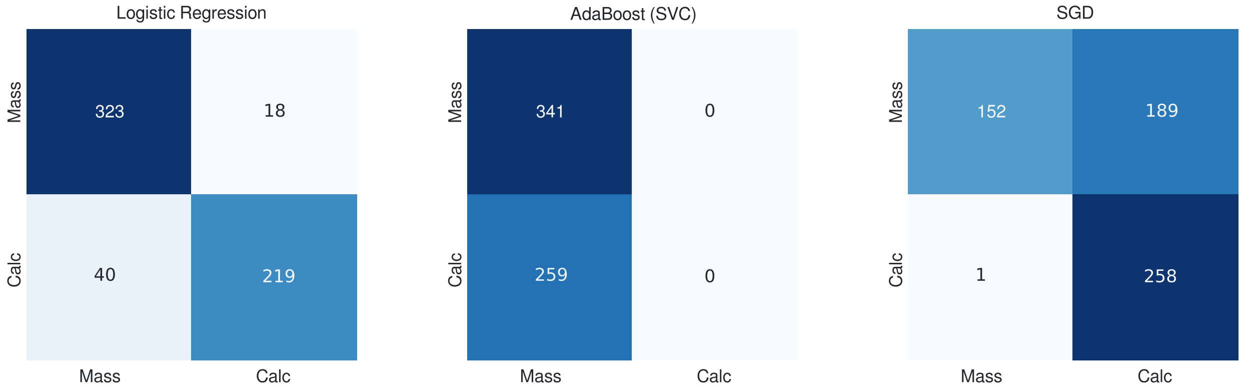

Figure 5: Logistic Regression and class-biased models: AdaBoost (SVC) and SGD.

Table 3: Results from 1024 FC.

Classifier	**Accuracy**	**Recall**	**F1**	**MCC**	**Precision**	**Misclassified Images**
AdaBoost (SVC)	0.5683	0.000	0.000	0.000	0.000	259
SGD	0.6833	0.9961	0.7308	0.5021	0.5771	190
Logistic Regression	0.9033	0.8455	0.8830	0.8032	0.9240	58

Another model used was the Random Forest Classifier with hyperparameter tuning (Best Params Random Forest), as well as the creation of various Voting Classifiers. A Voting classifier (Voting Cl 1) was created using 'hard voting' from the models shown in Table 3 in an attempt to improve accuracy but achieved the same result. However, when creating another Voting Classifier (Voting Cl 2), combining the classifiers used in Voting Cl 1 with Best Params Random Forest, a model with an accuracy of 90.83% was obtained, surpassing the results achieved by the VGG19 network.

From the confusion matrices shown in Figure 6 and the detailed metrics in Table 4. Voting Cl 2 provides the best results with an accuracy of 90.83%, a recall of 83.78%, an F1 score of 88.75%, an MCC of 0.8146, and a precision of 94.34%, reducing the number of misclassified images to 55. These results highlight the effectiveness of combining multiple classifiers to enhance performance in the classification of breast abnormalities.

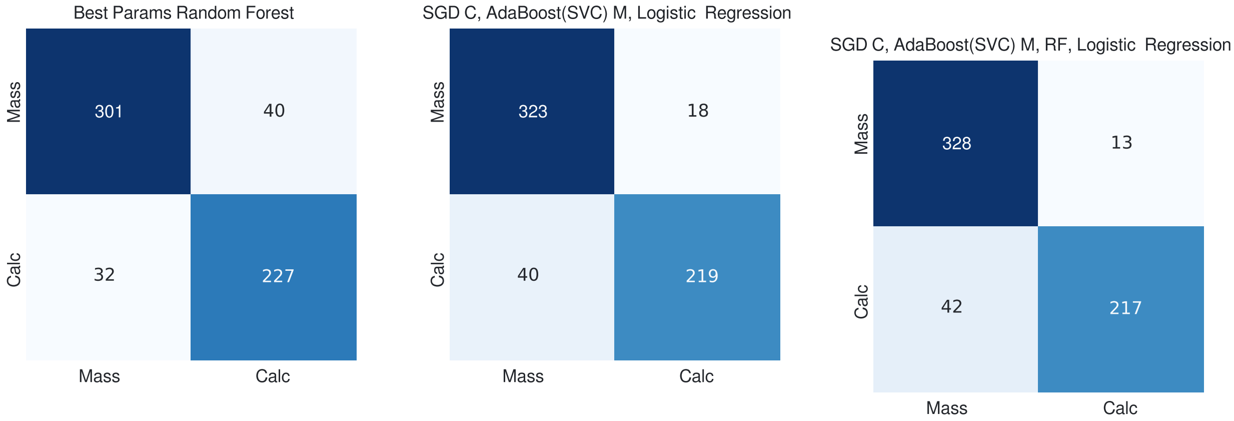

Figure 6: Random Forest Hiperparameter Tuning, Voting Cl 1 and Voting Cl2.

Table 4: Importance results from 1024 FC.

Classifier	Accuracy	Recall	F1	MCC	Precision	Misclassified Images
BP Random Forest	0.8800	0.8764	0.8631	0.7566	0.8500	72
Voting Cl 1	0.9033	0.8455	0.8830	0.8032	0.9240	58
Voting Cl 2	0.9083	0.8378	0.8875	0.8146	0.9434	55

From these experiments, we can conclude that it was possible to improve the classification rate of the VGG16 network. Not only that, but the classification rate also surpassed that of the VGG19 network, which had the highest accuracy among the three selected networks. By utilizing feature extraction with the VGG16 and combining it with various machine learning models, particularly through the creation of effective Voting Classifiers, the overall performance in classifying breast abnormalities was significantly enhanced. The best results were achieved with Voting Cl 2, which demonstrated the highest accuracy and precision, thus validating the effectiveness of combining multiple classifiers to improve the accuracy of breast cancer classification.

3.3 CNNs Voting and Stacking Classifiers

We can construct Voting Classifiers using Convolutional Neural Networks (CNNs). Since CNNs are probabilistic classifiers, ”soft voting” can be employed. Additionally, since the validation set was not used to adjust the parameters of the CNNs, it can be used to generate a “blending set” and thus utilize the Stacking technique, as used by Ref. 10 in their pathology classification problem.

3.3.1 Voting Classifiers

In general, the number of distinct Voting Classifiers that can be constructed if there are n classifiers is $2^n-(n+1)$. Since we have three CNNs, we can construct $2^3-(3+1)=8-(4)=4$, which are shown in Figure 7, and the obtained results are shown in Tab.5.

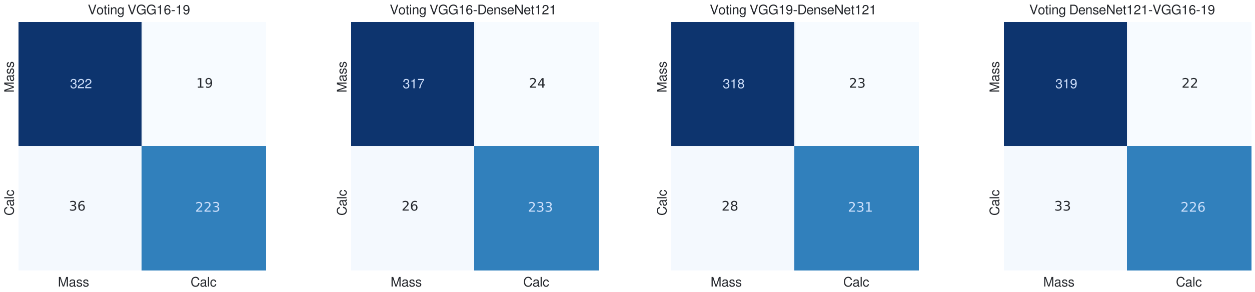

Figure 7: Combinations of VGG16, VGG19, and DenseNet121 used to create Voting Classifiers.

Table 5: Voting Classifiers created by CNNs.

Voting Classifier	Accuracy	Recall	F1	MCC	Precision	Misclassified Images
VGG16-19	0.9083	0.8610	0.8902	0.8130	0.9214	55
VGG16-Dens	0.9166	0.8996	0.9029	0.8301	0.9066	50
VGG19-Dens	0.915	0.8918	0.9005	0.826	0.9094	51
All CNNs	0.9083	0.8725	0.8915	0.8127	0.9112	55

From the confusion matrices shown in Figure 7 and the detailed metrics in Table 5, it is evident that various combinations of Voting Classifiers provide significant improvements in accuracy and performance metrics. Specifically, the combination of VGG16 and DenseNet121 (VGG16-Dens) achieved the best results, with an accuracy of 91.66%, a recall of 89.96%, an F1 score of 90.29%, an MCC of 0.8301, and a precision of 90.66%, misclassifying only 50 images.

Other combinations, such as VGG16-19 and VGG19-Dens, also performed well, achieving accuracies of 90.83% and 91.50%, respectively, with misclassification rates of 55 and 51 images.

When all CNNs were combined into a single Voting Classifier (All CNNs), the model achieved an accuracy of 90.83%, a recall of 87.25%, an F1 score of 89.15%, an MCC of 0.8127, and a precision of 91.12%, with 55 misclassified images.

3.3.2 Stacking

Similar to constructing Voting Classifiers, we can create the same number of blending sets with n classifiers (in this case, four) formed by various concatenations of the predictions on the validation set from each CNN. In this experiment, disregarding the previous votes, we began achieving better results from the start.

Meta-classifiers were sought for each of the four blending sets. For the blending set formed by the predictions of VGG16 and VGG19, the Naive Bayes model restored the accuracy of the VGG19 network, misclassifying 58 images. The SGD model misclassified 57 images, the same as Logistic Regression. However, using a Bagging model with Logistic Regression as the base classifier (Figure 8 (a)), 56 images were misclassified, and after tuning the hyperparameters, 53 images were misclassified.

For the blending set formed by the predictions of DenseNet121 and VGG16, tuning the hyperparameters of an SVC meta-classifier resulted in 52 misclassified images. However, the Ridge Classifier meta-classifier (Figure 8 (b)) achieved an accuracy of 91.66% in this experiment, misclassifying only 50 images, matching the voting accuracy of these two networks.

For the blending set formed by the predictions of the VGG19 and DenseNet121 networks, the SVC and Bagging models with SVC as the base classifier misclassified 55 images. The best model was Logistic Regression, which misclassified 52 images (Figure 8 (c)).

Finally, in the blending set formed by the predictions of all three CNNs, the SVC meta-classifier quickly misclassified 57 images, followed by the KNN model misclassifying 54 images. The model that correctly classified the most images was a Bagging meta-classifier with SVC as the base classifier (Figure 8 (d)), misclassifying 53 images.

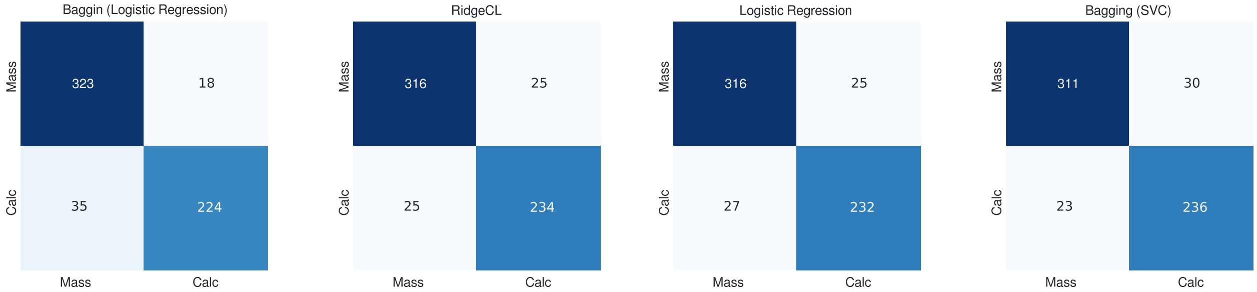

Figure 8: Meta-Learners: (a) Bagging(Logistic Regresion), (b) RidgeCL, (c) Logistic Regression, (d) Bagging(SVC).

The results of this experiment (Tab.6) show that using machine learning methods increases the classification rate of the CNNs used in this problem.

Table 6: Best results from each blending set

Blending Set	Meta-Learner	Accuracy	Recall	F1	MCC	Precision	Misclassified Images
VGG16-19	Bagging(LogReg)	0.9116	0.8648	0.8942	0.8199	0.9256	53
VGG16-Dens	RidgeCL	0.9166	0.9034	0.9034	0.8300	0.9034	50
VGG19-Dens	Logistic Reg.	0.9133	0.8957	0.8992	0.8232	0.9027	52
All CNNs	Bagging(SVC)	0.9116	0.9110	0.8990	0.8207	0.8872	53

4. CONCLUSION

The main objective of this study was to explore the synergy of combining convolutional neural networks (CNNs) as feature extractors with machine learning algorithms as classifiers.

The use of the VGG16 model aimed to achieve better results than those obtained by VGG19 by combining machine learning techniques, as shown in Sec.3.2.2 and throughout Sec.3.3. However, the experimental results in Sec.3.2.1 were not as successful when using the 25,088 features, suggesting that applying a dimensionality reduction technique such as PCA could help achieve better results.

The results in Sec. 3.3.1 seems to indicate that it would be advisable to start experimenting with a Voting Classifier. Additionally, it is interesting to observe that, in the experiments conducted using Voting Sec. 3.3.1 and Stacking Sec. 3.3.2 Classifiers, the best results were obtained by using the VGG16 and DenseNet121 networks together, rather than using the VGG19 network, which initially had the highest classification rate.

Overall, these findings demonstrate that the combination of CNNs with machine learning classifiers can significantly enhance performance in the classification of breast abnormalities. Carefully designed ensemble methods, particularly those involving VGG16 and DenseNet121, outperform individual models, underscoring the potential of such hybrid approaches in medical image analysis.

Acknowledgments

L.G. Rodríguez-López thanks the CONAHCYT scholarship granted, No. CVU:1205950. V. Kober thanks the Russian Science Foundation, grant No. 22-19-20071. J.A. Gonzalez-Fraga thanks the Autonomous University of Baja California, grant No. 246.

REFERENCES

[1] Organización Mundial de la Salud, "Cáncer." https://www.who.int/es/news-room/fact-sheets/detail/cancer (Feb. 2022).

[2] Vinnicombe, S., Pinto Pereira, S. M., McCormack, V. A., Shiel, S., Perry, N., and Dos Santos Silva, I. M., "Full-field digital versus screen-film mammography: Comparison within the uk breast screening program and systematic review of published data," *Radiology* **251**(2), 347–358 (2009).

[3] Drukteinis, J. S., Mooney, B. P., Flowers, C. I., and Gatenby, R. A., "Beyond Mammography: New Frontiers in Breast Cancer Screening," *The American Journal of Medicine* **126**(6), 472–479 (2013).

[4] Lee, R. S., Gimenez, F., Hoogi, A., Miyake, K. K., Gorovoy, M., and Rubin, D. L., "A curated mammography data set for use in computer-aided detection and diagnosis research," *Scientific data* **4**, 170177 (2017).

[5] Mračko, A., Vanovčanová, L., and Cimrák, I., "Mammography Datasets for Neural Networks—Survey," *Journal of Imaging* **9**(5) (2023).

[6] Géron, A., [*Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*], O'Reilly Media, 2 ed. (2019).

[7] K., R. R. and Wilscy, M., "Pretrained convolutional neural networks as feature extractor for image splicing detection," in [*2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*], 1–5 (2018).

[8] Lai, L., "Medicalcnn: Abnormality detection in mammogram images using deep convolutional neural networks," (2021). GitHub repository.

[9] Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," in [*3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*], Bengio, Y. and LeCun, Y., eds. (2015).

[10] Nemade, V., Pathak, S., and Dubey, A. K., "Deep learning-based ensemble model for classification of breast cancer," *Microsystem Technologies* **30**, 513–527 (May 2023).