

RSNA Screening Mammography Breast Cancer Detection using Vision and Deep Learning

Ahsan Khan 104100196 | Pooja Chandrasekharan 110091306 | Srivatsan Vasudevan 110091377

***Abstract*—Breast cancer is one of the most common cancers and is causing a huge number of deaths in women. The high incidence and mortality of breast cancer are due to its considerably low accuracy of diagnosis. In this paper, we explore machine learning models that can be applied to help increase the accuracy of the diagnosis of breast cancer. The main problem of the project is to detect if the patient has breast cancer or not from Mammographs.**

I. INTRODUCTION

Breast cancer is one of the most common cancer in women and the second leading cause of women's cancer death[1]. Despite the lack of effective treatment, the low accuracy of diagnosis is also a major cause of the high incidence and mortality of breast cancer. Mammography is a traditional method used for diagnosing breast cancer. According to UCHHealth's report, only 78% of breast cancer can be accurately diagnosed by mammography [2]. Many cases such as doctors' negligence or incompetence in addition to a mammography error may also result in a late diagnosis or misdiagnosis, which can be considered a cause of breast cancer death [3]. In the long term, the early-stage diagnosis could significantly increase the survival rate of breast cancer [4], therefore, it is important to improve the accuracy of a breast cancer diagnosis. Machine learning has been applied in medical diagnosis in a large number of papers [5]. In order to increase the accuracy of breast cancer diagnosis, we aim to

use machine learning models and choose the model/hyperparameters with higher performance.

RSNA Screening Mammography Breast Cancer Dataset is a widely used dataset provided by Kaggle. In this paper, we will train our models using this dataset.

Link:

<https://www.kaggle.com/competitions/rsna-breast-cancer-detection>

The dataset for this challenge contains radiographic breast images of female subjects. The goal of this competition is to identify cases of breast cancer in mammograms from screening exams. It is important to identify cases of cancer for obvious reasons, but false positives also have downsides for patients. As millions of women get mammograms each year, a useful machine-learning tool could help a great many people.

Mammography:

Mammography is the standard screening examination for breast cancer. Breast density is a measure used to describe the proportion of fibroglandular tissue in a woman's breast depicted on a digital mammogram. Breast density can be measured qualitatively or quantitatively. Qualitative methods include the original Wolfe criteria, the Tabar classification, and the widely used Breast Imaging and Reporting Data System (BI-RADS) criteria. The BI-RADS mammographic breast density criteria include four qualitative categories: (a) almost entirely fatty, (b) scattered areas of fibroglandular density (or "scattered density" for

short), (c) heterogeneously dense, (d) or extremely dense. Quantitative methods include Cumulus software that interactively determines the skin line and sets a threshold for segmenting dense tissue; the area of dense tissue is expressed in one of six-category percentages: 0, <10%, 10–25%, 26–50%, 51–75%, and >75%. Automated computerized methods include the LIBRA program that is publicly available to estimate an area-based percent density as well as the volume-based commercial software, such as Quantra and Volpara. The volume-based methods function only on the raw (“FOR PROCESSING”) digital mammogram images, which are not routinely stored in most medical centers.

II. DATASET AND FEATURES

The mammograms are in dicom format. You can expect roughly 8,000 patients in the hidden test set. There are usually but not always 4 images per patient. Note that many of the images use the jpeg 2000 format which may you may need special libraries to load.

site_id - ID code for the source hospital.

patient_id - ID code for the patient.

image_id - ID code for the image.

laterality - Whether the image is of the left or right breast.

view - The orientation of the image. The default for a screening exam is to capture two views per breast.

age - The patient's age in years. implant - Whether or not the patient had breast implants. Site 1 only provides breast implant information at the patient level, not at the breast level.

density - A rating for how dense the breast tissue is, with A being the least dense and D being the most dense. Extremely dense tissue can make diagnosis more difficult. Only provided for train.

machine_id - An ID code for the imaging device.

cancer - Whether or not the breast was positive for cancer. The target value. Only provided for train.

biopsy - Whether or not a follow-up biopsy was performed on the breast. Only provided for train.

invasive - If the breast is positive for cancer, whether or not the cancer proved to be invasive. Only provided for train.

BIRADS - 0 if the breast required follow-up, 1 if the breast was rated as negative for cancer, and 2 if the breast was rated as normal. Only provided for train.

prediction_id - The ID for the matching submission row. Multiple images will share the same prediction ID. Test only.

difficult_negative_case - True if the case was unusually difficult. Only provided for train.

III. RELATED WORK

There have been many studies applying different machine learning techniques on medical analysis. In terms of traditional machine learning methods, Chaurasia et al.[6] used Simple Logistic to reduce the dimension of feature space and applied RepTree and RBF Network to evaluate the performance. Dubey et al. used K-means algorithm to evaluate the impact of clustering using centroid initialization and achieved 92% average positive prediction accuracy [7]. Classification and regression trees (CART) classifier with feature selection and bagging technique was implemented to predict breast cancer in [8]. Wang et al. [9] compared four classifiers: Naive Bayes, Decision Tree, Support Vector Machine and k-nearest neighbor for classification of cancer using gene expression data. These traditional machine learning models has the advantage of low design complexity, but it is not capable of dealing with complex data. In the project, we are going to apply models including these previous mentioned algorithms on the Breast Cancer Wisconsin dataset and compare the performances.

Many deep learning models have also been developed in this objective. In [10], problems in multiple datasets were discussed and Partial Likelihood Artificial Neural Network is applied for prediction of cancer survival. Purwar et al. [11] proposed a hybrid model using a combination of K-means clustering with Multilayer Perceptron with promising results for various medical dataset. Another new classification algorithm for detection of breast abnormalities in digital mammograms using Particle Swarm Optimized Wavelet Neural Network (PSOWNN) is investigated in [12]. These deep learning models are capable of modeling complex and high dimensional data. However, the computational complexity is higher and training time may be long.

IV. THE PROBLEM – MOTIVATION

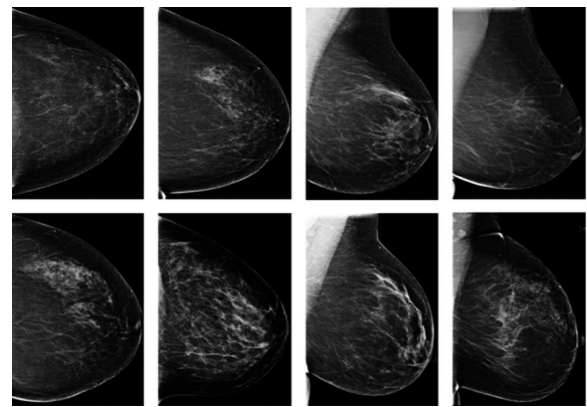
The goal of this paper is to identify breast cancer. We train the model with screening mammograms obtained from regular screening. The work improving the automation of detection in screening mammography may enable radiologists to be more accurate and efficient, improving the quality and safety of patient care. It could also help reduce costs and unnecessary medical procedures.

CONTEXT

Breast cancer is the type of cancer that strikes women most frequently worldwide, according to the WHO. There were 2.3 million new breast cancer diagnoses and 685,000 fatalities in 2020 alone. However, since routine mammography screening was introduced by health authorities in age groups deemed at risk in the 1980s, the mortality rate of breast cancer in high-income countries has decreased by 40%. Your machine learning abilities may be able to speed up the process radiologists use to evaluate screening

mammograms because early identification and treatment are essential to lowering cancer fatality rates.

Currently, screening mammography systems are expensive to operate since early diagnosis of breast cancer necessitates the expertise of highly trained human observers. This issue will probably get worse due to a radiologists shortage that is coming to various nations. A high prevalence of false positive results is another side effect of mammography screening. This may lead to unwarranted concern, inconvenienced follow-up care, more imaging tests, and occasionally the requirement for tissue sampling (often a needle biopsy).



The Radiological Society of North America (RSNA) is a nonprofit organization that covers 145 nations and 31 radiologic subspecialties. Through research, teaching, and technology innovation, RSNA supports excellence in patient care and healthcare delivery.

V. MACHINE LEARNING IN THE MEDICAL FIELD

Machine learning methods are often used in medical imaging to improve the accuracy and efficiency of image analysis. For example, machine learning algorithms can be used to automatically identify abnormalities in medical

images, such as tumors or other abnormalities. These algorithms can also be used to classify different types of medical images, such as X-rays, CT scans, and MRI scans. By using machine learning, radiologists can quickly and accurately interpret medical images, leading to better patient care.

Deep learning is a type of machine learning that uses neural networks to learn complex patterns from data. In the context of breast cancer detection, deep learning algorithms can be trained on large amounts of medical images of both healthy and cancerous breast tissue. The algorithms learn to recognize the characteristic patterns of cancerous tissue, such as the size, shape, and texture of tumors. Once trained, the deep learning algorithm can be used to automatically analyze new medical images and identify potential cases of breast cancer. This can help radiologists and other medical professionals to more quickly and accurately diagnose breast cancer and develop a treatment plan for their patients.

VI. PROPOSED METHOD

Starting with the baseline model, with default parameters of EfficientNet; we try to finetune EfficientNet architecture for breast cancer detection and to get the best accuracy and loss to help the medical field.

VII. SOFTWARE AND PACKAGES

For this project, we used Python 3.6 as the programming language. Python is open-source software and is a cross-platform language. The following table lists the package and its applications:

PACKAGE	APPLICATION
<i>NumPy</i>	<i>Data processing/ wrangling, data calculations.</i>
<i>Pytorch</i>	<i>Open-source deep learning</i>
<i>Timm EfficientNet</i>	<i>Open-source deep learning</i>
<i>wandb</i>	<i>To stream training data</i>
<i>Torch.dataset</i>	<i>To arrange dataset to batch sizes</i>
<i>Sckit-learn</i>	<i>To train and validate splits</i>
<i>OpenCV</i>	<i>To resize mammograms or training data</i>
<i>Pandas</i>	<i>To read the dataset</i>

VIII. METHODS

In this study, we constructed a convolutional neural network (CNN)-based model coupled with a large (i.e., 2,400 images) digital mammogram imaging dataset to evaluate the accuracy and loss to determine if cancer is present or not. In order to measure the classification performance, the CNN classifier was tested on the mammogram images dataset.

IX. EFFICIENT NET

EfficientNet is a convolutional neural network (CNN) architecture that was developed by Google Research. It is designed to improve upon the performance of existing CNNs by scaling up their capacity and improving their efficiency.

EfficientNet uses a novel scaling method to automatically scale up the capacity and size of the network, while also using a combination of carefully chosen convolutional layers, batch normalization, and activations to improve the network's efficiency. This allows EfficientNet to achieve state-of-the-art performance on a wide range of tasks, including image classification, object detection, and semantic segmentation.

One key aspect of EfficientNet is its use of a compound scaling method, which scales up the network in a more structured and efficient manner than previous methods. This method

scales up the network's depth, width, and resolution simultaneously, allowing the network to learn more efficiently and improve its performance on larger and more complex tasks.

Additionally, EfficientNet uses a carefully designed set of convolutional layers, batch normalization, and activations, which are chosen to optimize the network's efficiency and performance. This allows the network to achieve high accuracy with a relatively small number of parameters, which makes it faster and more efficient than other networks.

Overall, EfficientNet is a powerful and efficient CNN architecture that has been shown to outperform existing architectures on a wide range of tasks. It is an important advancement in the field of deep learning and has the potential to improve the performance of many applications that use CNNs.

EfficientNet Architecture

The EfficientNet architecture consists of several key components:

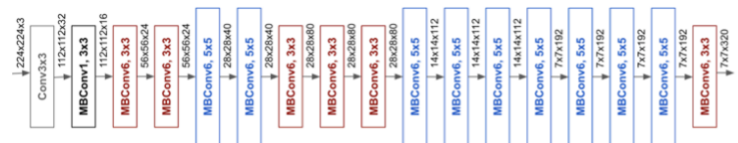
The stem: This is the initial part of the network, which typically consists of a series of convolutional and pooling layers. The stem is used to reduce the input resolution and extract initial features from the input image.

The blocks: The main part of the network is made up of repeated blocks of layers. These blocks typically consist of a combination of convolutional, pooling, and normalization layers. The number and type of layers in each block can vary, depending on the specific EfficientNet model.

The head: This is the final part of the network, which typically consists of a series of fully connected (dense) layers. The head is used to

make predictions based on the features extracted by the blocks.

The EfficientNet architecture also uses advanced techniques such as squeeze-and-excitation blocks and stochastic depth to further improve performance. These techniques are designed to help the network learn more discriminative features and make more accurate predictions.



X. FALSE NEGATIVES

False negatives in healthcare machine learning refer to instances where a model incorrectly predicts that a patient does not have a certain condition, even though the patient does have the condition. This type of error can be particularly problematic in healthcare, as it can lead to patients not receiving the treatment they need. For example, if a machine learning model is used to predict the likelihood of a patient having a certain disease, and the model falsely predicts that the patient does not have the disease, the patient may not receive the necessary treatment. This can have serious consequences for the patient's health.

It is important for healthcare professionals to be aware of the potential for false negatives in machine learning models, and to use these models carefully and cautiously in order to avoid negative outcomes.

In order to fix this issue, we balanced the data in all classes by taking the same number of non-cancerous images of patients: imbalance class.

XI. IMBALANCE CLASS

One common challenge in healthcare machine learning is dealing with imbalanced classes, where some classes have significantly more examples than others. This can cause the machine learning model to be biased towards the majority class, leading to poor performance on the minority classes.

To address this issue, there are several techniques that can be used to balance the classes in a dataset. These techniques can be applied either before or during the training of the machine learning model.

One common technique is oversampling, which involves increasing the number of examples in the minority classes by duplicating existing examples or generating new synthetic examples. This can help the model learn more from the minority classes and improve its performance on those classes.

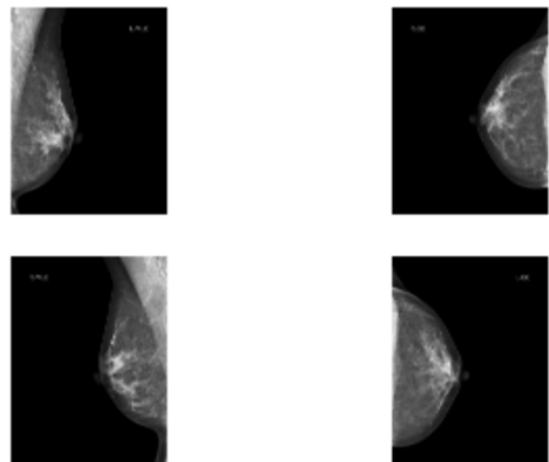
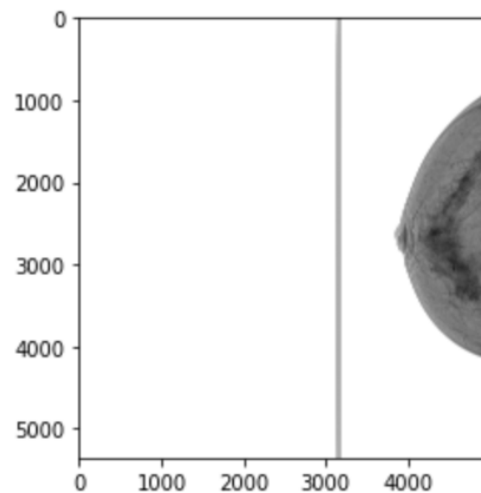
Another technique is undersampling, which involves reducing the number of examples in the majority classes by randomly removing some examples. This can help balance the dataset and improve the model's ability to generalize to the minority classes.

Another approach is to use a weighted loss function, which assigns higher weights to the minority classes during training. This can help the model pay more attention to the minority classes and improve its performance on those classes.

Overall, balancing classes in healthcare machine learning is an important step that can help improve the performance of the machine learning model and ensure that it is able to generalize to all classes in the dataset.

XII. READING THE DATA

pydicom is a Python package for working with DICOM files, which are used for storing and transmitting medical imaging data. DICOM stands for Digital Imaging and Communications in Medicine, and is a widely-used standard in the medical field.



pydicom allows developers to read, modify, and write DICOM files using Python. It also provides tools for working with DICOM metadata, such as patient information, and for

converting DICOM files to and from other image formats.

pydicom is widely used in the medical field, and is a popular choice for developers working on applications related to medical imaging and radiology. It is open-source and available on PyPI, the Python Package Index.

```
def load_dicom(path):
    img=dicom.dcmread(path)
    data=img.pixel_array
    return data

def listdirs(folder):
    return [d for d in os.listdir(folder) if os.path.isdir(os.path.join(folder, d))]

train_dir = '/kaggle/input/rsna-breast-cancer-detection/train_images'
test_dir = '/kaggle/input/rsna-breast-cancer-detection/test_images'
patients = sorted(os.listdir(train_dir))
```

XIII. DATA PREPARATION

From the data exploration, we found out that the data is highly imbalanced because there are more non-cancerous patients than cancerous patients.

	site_id	patient_id	image_id	laterality	view	age	cancer	biopsy	invasive	BIRADS
0	2	10006	482822612	L	CC	61.000000	0	0	0	nan
1	2	10006	1489541791	L	MLC	61.000000	0	0	0	nan
2	2	10006	1864590858	R	MLC	61.000000	0	0	0	nan
3	2	10006	1874946579	R	CC	61.000000	0	0	0	nan
4	2	10011	220375232	L	CC	55.000000	0	0	0	0.000000

We also found out that the dataset contains missing values, there were 37 in count.

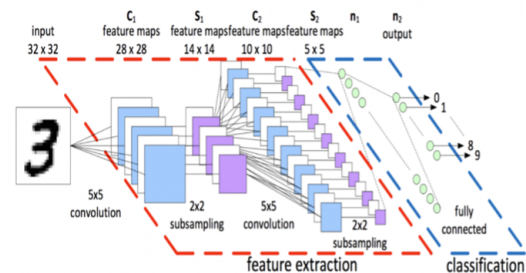
	Null Values Count
age	37
BIRADS	28420
density	25236

XIV. SUPERVISED LEARNING TECHNIQUES

Everyone who is new to computer vision realizes that one of the most powerful supervised deep learning techniques is the Convolutional Neural Networks (abbreviated as “CNN”). The final structure of a CNN is actually very similar to Regular Neural Networks (Regular Nets) where there are neurons with weights and biases. In addition, just like in Regular Nets, we use a loss function (e.g. cross-entropy or softmax) and an optimizer (e.g. adam optimizer) in CNNs.

CONVOLUTIONAL NEURAL NETWORK (CNN)

A typical CNN design begins with feature extraction and finishes with classification. Feature extraction is performed to convolution layers with sub-sampling. Classification is performed with dense layers followed by a final SoftMax layer. For image classification, this architecture performs better than an entirely fully connected feed-forward neural network.



THE 5-STEP MODEL LIFE-CYCLE

A model has a life-cycle, and this very simple knowledge provides the backbone for both modeling a dataset:

There are 5 steps for this model:

1. Creating an EfficientNet Model.
2. Compile the model with adam optimizer and cross-entropy loss function.
3. Fit the model.
4. Evaluate the model.

1. Create an EfficientNet Model.

```
import timm
def train_model(config=None):
    model = timm.create_model('efficientnet_b4', pretrained=True, in_chans=1)
    # for i in range(2):
    #     if i:
    #         else:
    #             model = timm.create_model('resnet34', pretrained=True, in_chans=1)
    criterion = nn.CrossEntropyLoss()
    best_acc = 0.0
```

Using timm library, we created an EfficientNet architectures head and added our final layers.

We finetune this model to predict whether the patient has cancer or not. This is how we create an EfficientNet Model.

2. Compile the model with adam optimizer and cross-entropy loss function.

To train the model, we used cross-entropy as the loss function and adam as the optimizing function.

3. Fit the Model

```
for epoch in range(config.epochs):
    model.cuda()

    for phase in ['train', 'val']:
        if phase == 'train':
            model.train()
        else:
            model.eval()

        epoch_loss = 0.0
        epoch_acc = 0

        dataloader = dataloaders_dict[phase]
        for item in tqdm(dataloader, leave=False):
            images = item[0].cuda().float()
            classes = item[1].cuda().long()

            optimizer.zero_grad()
```

We arranged the data batches of 32, split them into 2 sets: training and validation sets then fitted the model on the training set and validated its performance in the validation set.

4. Evaluate the Model

```
with torch.set_grad_enabled(phase == 'train'):
    output = model(images)
    loss = criterion(output, classes)
    _, preds = torch.max(output, 1)

    if phase == 'train':
        loss.backward()
        optimizer.step()

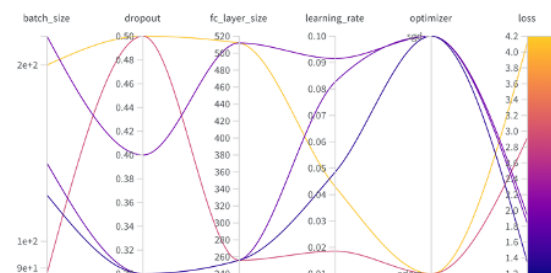
    epoch_loss += loss.item() * len(output)
    epoch_acc += torch.sum(preds == classes.data)
```

We submitted the predictions from the model on the provided test dataset.

Once we build the baseline model, we used random searchCV to find the best hyperparameters for the baseline model. For this we used: weights and bias sweeps.

XV. RESULT

1. Hyper Parameter Tuning:



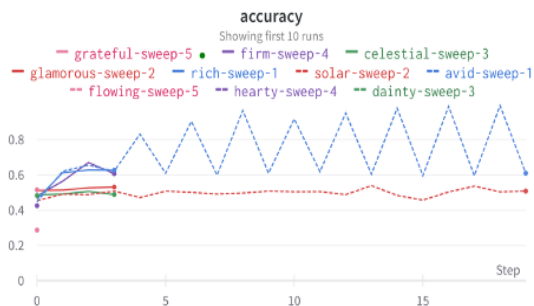
Using the Weights and Bias platform we observed the accuracy and loss of our models by tuning the hyperparameters mentioned below:

1. Batch Size: 32 to 128
2. Dropout: 0.3, 0.4, 0.5

3. Fully connected layer size: 128, 256, 512
4. Learning Rate: 0 to 0.1
5. Optimizer: SGD and adam

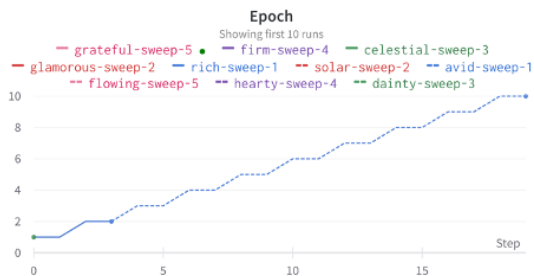
From these experiments, we found the best hyperparameters is a batch size of 120, a dropout of 0.3, a fully-connected layer of 256, a learning rate of 0.04, and lastly the optimizer being SGD. We obtained a loss of 0.6 and an accuracy of 60%.

2. Epochs/Training Time:

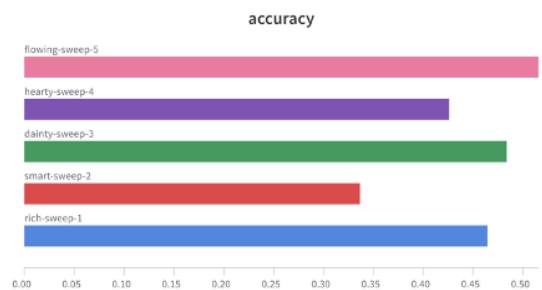
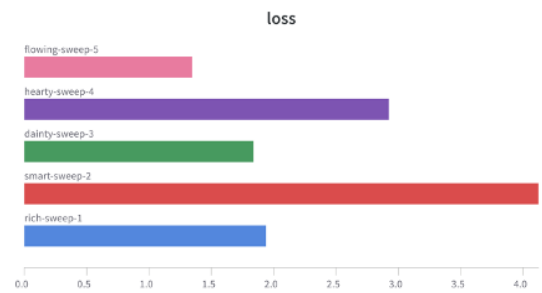
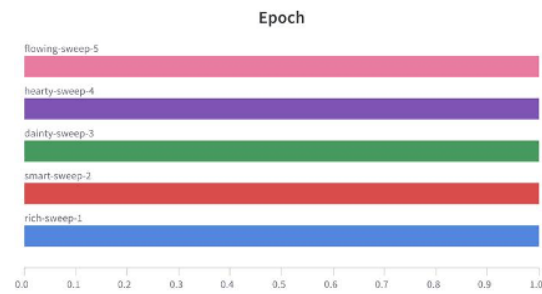


After finding the best hyperparameters, we trained the model for 10 epochs and got the best accuracy of 61% and a loss of 0.5.

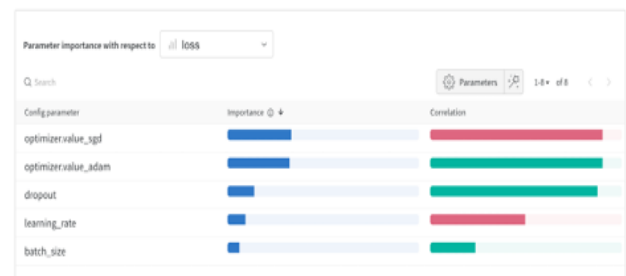
Since we are constrained on the hardware (lack of GPU), we were not able to train with a deeper model and longer epochs.



All the experiments that were conducted for finding the best hyperparameters and their corresponding accuracy and loss are represented by the images below:



We also found out that the following parameters had a high core relation with the prediction:



XVI. CONCLUSION

Using machine learning algorithms to detect breast cancer can be a useful tool for improving the accuracy and efficiency of diagnosis. Machine learning algorithms can be trained on large datasets of medical images, such as mammograms, and can learn to identify patterns and features that are associated with breast cancer. This can help to improve the sensitivity and specificity of diagnosis and can reduce the number of false positives and false negatives.

However, it is important to note that machine learning algorithms should not be used as a replacement for clinical examination and other diagnostic methods. These algorithms should be used as part of a broader diagnostic strategy, and should be carefully validated and tested before being used in clinical settings. Additionally, the ethical implications of using machine learning in healthcare should be carefully considered.

Overall, the use of machine learning for detecting breast cancer shows promise, but further research and development is needed to fully realize its potential.

Future Work:

To obtain better results we can do the following for our experiments:

1. Use a deeper model
2. Use GPUs to decrease the training time
3. Explore other architectures
4. Use a higher number of epochs
5. Train on more data

Project Dashboard:

<https://wandb.ai/srivatsan25/Topics%20in%20AI%20Project%202/table?workspace=user-srivatsan25>

References

- [1] A. C. Society, "How common is breast cancer?." <https://www.cancer.org/cancer/breastcancer/about/how-common-is-breast-cancer.html>, 2011.
- [2] UCHealth, "How accurate are mammograms?." <https://www.uchealth.org/today/how-accurate-are-mammograms/>, 2015.
- [3] "Breast cancer misdiagnosis and mammography errors." <https://hackmd.io/@shaochia/Hk6fwgI1r?type=view1-Introduction>, 2011.
- [4] L. Wang, "Early diagnosis of breast cancer," *Sensors*, vol. 17, no. 7, p. 1572, 2017.
- [5] K. R. Foster, R. Koprowski, and J. D. Skufca, "Machine learning, medical diagnosis, and biomedical engineering research-commentary," *Biomedical engineering online*, vol. 13, no. 1, p. 94, 2014.
- [6] V. Chaurasia and S. Pal, "Data mining techniques: to predict and resolve breast cancer survivability," *International Journal of Computer Science and Mobile Computing IJCSMC*, vol. 3, no. 1, pp. 10–22, 2014.
- [7] A. K. Dubey, U. Gupta, and S. Jain, "Analysis of K-means clustering approach on the breast cancer Wisconsin dataset," *International journal of computer assisted radiology and surgery*, vol. 11, no. 11, pp. 2033–2047, 2016.
- [8] D. Lavanya and K. U. Rani, "Ensemble decision tree classifier for breast cancer data," *International Journal of Information Technology Convergence and Services*, vol. 2, no. 1, p. 17, 2012.
- [9] X. Wang and O. Gotoh, "A robust gene selection method for microarray-based cancer classification," *Cancer*

informatics, vol. 9, pp. CIN–S3794, 2010.

- [10] R. Marshall, “Artificial neural networks in cancer management,”
- [11] A. Purwar and S. K. Singh, “Hybrid prediction model with missing value imputation for medical data,” *Expert Systems with Applications*, vol. 42, no. 13, pp. 5621– 5631, 2015. [
- [12] J. Dheeba, N. A. Singh, and S. T. Selvi, “Computeraided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach,” *Journal of biomedical informatics*, vol. 49, pp. 45–52, 2014.
- [13] D. Team, “Kernel functions-introduction to svm kernel examples.”
<https://data-flair.training/blogs/svm-kernel-functions/>, 2018.
- [14] S. B. Ranjit Panigrahi, “Classification and analysis of facebook metrics dataset using supervised classifiers,” *Social Network Analytics*, 2019.
- [15] A. Chakure, “Random forest classification and its implementation in python.”
<https://towardsdatascience.com/random-forestclassification-and-its-implementation-d5d840d8ead0>, 2019.
- [16] M. B. Kursu, W. R. Rudnicki, et al., “Feature selection with the boruta package,” *J Stat Softw*, vol. 36, no. 11, pp. 1–13, 2010.
- [17] vishabh goel, “Building a simple machine learning model on breast cancer data.”
<https://towardsdatascience.com/building-a-simplemachine-learning-model-on-breast-cancer-dataeca4b3b99fa3>, 2018.