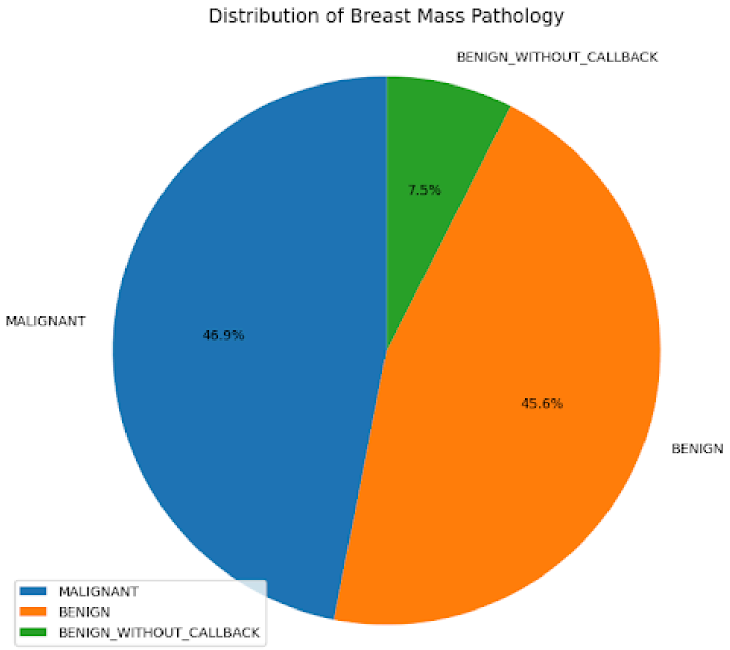
Introduction

Literature Review

Breast cancer is the most common cancer in both genders, affecting mainly women [1]. In 2021, the CDC reported 272,454 new cases of female breast cancer in the U.S. and 42,310 deaths [2]. The CDC categorizes breast cancer into three stages: localized, regional, and distant. About 25% of cases are diagnosed at the regional stage and 6% at the distant stage [2]. The 5-year survival rate is 86.3% for localized cancer and 32.4% for distant cancer [2]. Early diagnosis and intervention are crucial, prompting many clinicians to adopt computer-aided diagnosis, with machine learning effectively addressing challenges of traditional methods. By analyzing mammograms, clinical data, and genetic information, ML models can identify patterns and anomalies indicative of breast cancer [3]. Many studies compare multiple ML models to evaluate which perform best under different conditions [3], [4]. While research on ML for cancer diagnosis has expanded, improvements are still needed. Incorporating multiple modalities could enhance accuracy [4]. Additionally, challenges include the need for large, diverse datasets to reduce bias, transparent models for physicians, and ethical concerns regarding patient privacy [4].

Dataset Description

The CBIS-DDSM (Curated Breast Imaging Subset of DDSM) is an updated, standardized version of the Digital Database for Screening Mammography, featuring decompressed images, mammographer annotations, and detailed metadata. It contains 2,620 scanned mammography studies, including normal, benign, and malignant cases, all verified by pathology. The dataset includes 10,239 images from 1,566 participants, some with multiple patient IDs for different scans.



Dataset Link

https://www.kaggle.com/datasets/awsaf49/cbis-ddsm-breast-cancer-image-dataset

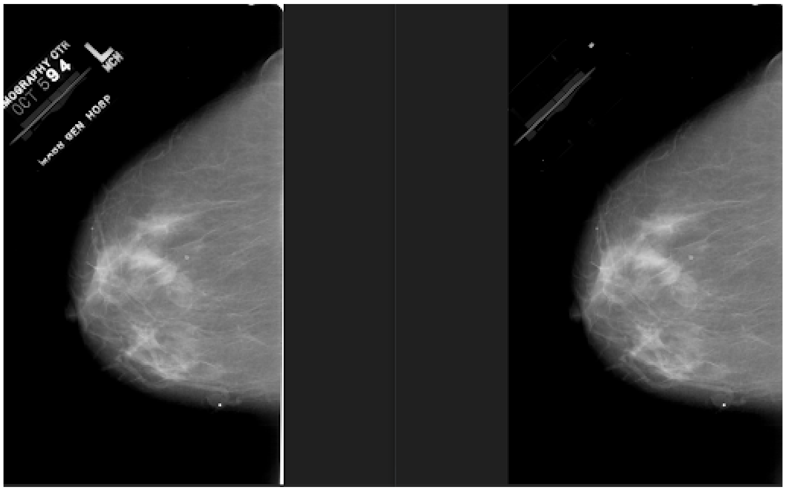
Problem Definition

Traditional diagnostic methods often result in false positives and overdiagnosis. Mammograms miss about 1 in 8 breast cancer cases, and overdiagnosis is a major concern [6]. This leads to unnecessary or missed treatments. Our goal is to improve breast cancer diagnosis accuracy using machine learning, reducing both overdiagnosis and underdiagnosis.

Methods

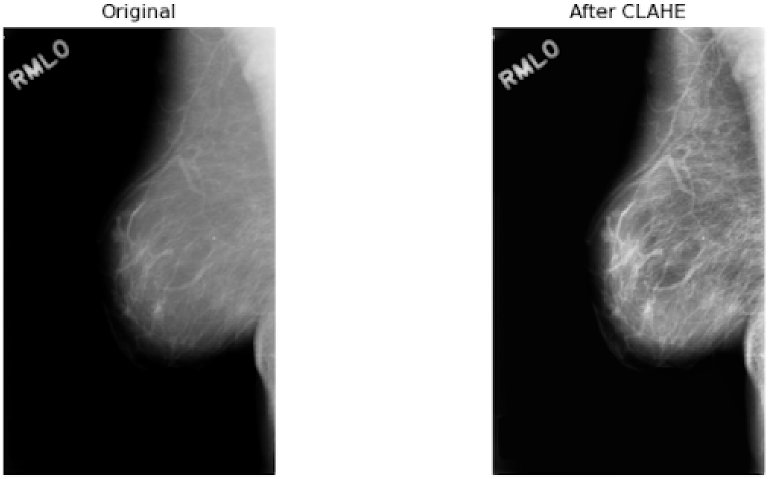
Data Processing

Our dataset consists of 10,239 images from 1,566 participants, accompanied by CSV files with additional information. To prepare the data, we used a python script to separate the images into an 80-20 train-test split, with separate folders for benign and malignant cases. Each image was resized to 240 x 240 using torchvision.transforms, which is the input size required for EfficientNet B1.



To further refine the dataset, we focused on addressing the text present in some images. Our goal was to remove as much noisy and irrelevant data as possible from the model's input. Because some images contain text while others do not, achieving uniformity is essential for reliable data assessment. This allowed us to enhance our predictive accuracy. Our preprocessing approach began with the use of keras_ocr, a model specifically designed for text identification and localization. This model allowed us to efficiently pinpoint and isolate unnecessary information, such as textual regions in the images. Once these regions were identified, we used OpenCV to mask them. By combining keras_ocr and OpenCV, we successfully removed irrelevant and noisy data, providing greater consistency for our model.

Another processing method we explored was the implementation of Contrast Limited Adaptive Histogram Equalization(CLAHE). CLAHE is crucial when using in mammograms due to the slight details that can be missed due to low contrast. This can cause difficulty for radiologists and models to differentiate between malignant and benign tumors. By using CLAHE, we focus on increasing contrast in important regions. This is achieved by dividing the image into smaller components and adjusting these components one by one, then combining them. For our implementation, we used an 8x8 grid to balance contrast effectively without risking the overamplification of noise.



This image displays the contrast difference once CLAHE is applied to the image. The image provided in "After CLAHE" displays enhanced features that were not as visible. These features can allow us to observe potential abnormal features and come to a diagnosis.

ViT Pre-Processing: When preparing the mammogram dataset specifically for our ViT model, we used a series of preprocessing techniques and transformations to enhance the image quality and ensure model robustness. First, to filter out all the irrelevant black and white masks and zoomed-in images of mammograms we parsed the files to filter out everything but the full mammogram images. Then, to improve the visibility of important features in our grayscale mammogram images we used histogram equalization and contrast adjustment, this helped make the images more interpretable for our model. At the same time we used other techniques like normalization per image and sharpening to emphasize the edges of the mammograms, which stabilized and sped up the training process. Along with these techniques we incorporated several transformations in our augmentation pipeline, such as resizing, random horizontal and vertical flips, rotations, color jitter, affine transformations, and cropped resizing. We added these transformations because they introduced a lot more variability to our images, enabling our model to generalize better to unseen data. This helped a lot with overfitting. This preprocessing pipeline for our ViT model improved model performance by creating a higher quality and diverse dataset.

SVM Pre-Processing: First, we scaled the image down to reduce computation cost of the model. We tested dimensions of 64, 128 and 256. We found that 64 reduced the accuracy of the model while 128 and 256 performed relatively the same. Then we scaled all the features to a range between 0 and 1 using min-max scaler. This also reduces computation of the model while keeping the data importance of each feature the same as well as removing super large numbers. Then we use feature selection to remove columns that have a variance under 0.005. Finally we reduce the number of features and retain 95% of the variance. No further image augmentation was needed or used due to the nature of SVMs.

CNN Pre-Processing: The direction of the breast in the image was determined by splitting the image into two halves down the middle and taking the object on the side with the most ‘white’ pixels; this allows for all images to be flipped uniformly and increased the versatility of the dataset as horizontal flips could be introduced in the data augmentation stage to introduce directionality. While the dataset did include a CSV with data on whether the image was of a right or left breast, it seemed that there wasn’t a consistent direction they faced; however this was the best and simplest solution to uniformly have all the mammograms face the same direction. We isolated the region of interest, the breast, by identifying the largest object in the image and creating a mask around said object and every pixel outside of the mask was set to a value of 0. This effectively got rid of any noise outside of the main area of interest including the name tags that are usually included in the mammograms. This aided helping the model focus on the important parts of the mammograms. It also allows for a more controlled noise to be added back in with Gaussian Noise during the data augmentation which aided with overfitting. CLAHE was also applied during data preprocessing to increase the contrast of the mammograms. The rolling ball algorithm from sci-kit was applied in order to do a final reduction noise on the actual area of interest in the mammograms. It is also to note we also resize the images during this stage to be 2 times the input size of the CNN model and a square aspect ratio during this stage. We found that it drastically reduced the time it took to train the model even if we did resizing during data augmentation due to how large the images were (more than 4000x4000 for most of the images in the dataset). This drastically cut down the training time per epoch by more than half of the original time and allowed for more rapid testing and parameter tuning.

Convolutional Neural Network (CNN)

Our goal is to identify patterns and extract details from mammograms for indications of breast cancer. Given that mammograms are complex images where the difference between normal and malignant can be subtle, we wanted to work on an advanced model. Through our research we found that CNNs mimics human visual systems to recognize patterns in images and videos. Furthermore, CNNs previously had success classifying various medical images.

Support Vector Machine (SVM)

SVM is a good choice for our model as it is known to be suitable for smaller datasets with a large number of features. Given that we had a smaller dataset of mammograms we wanted to try a model that doesn’t incorporate data augmentation to create more data. Our 2 other models required data augmentation due to our small dataset and longer training times. By leveraging our small dataset and fast training time of SVMs we were able to focus on tuning the model with different kernels and hyperparameters. SVM can handle linear and nonlinear data through the use of kernels. In our model we experimented with 3 kernels: linear for simple decision boundaries, polynomial for complex relations andRBF to handle nonlinear patterns.

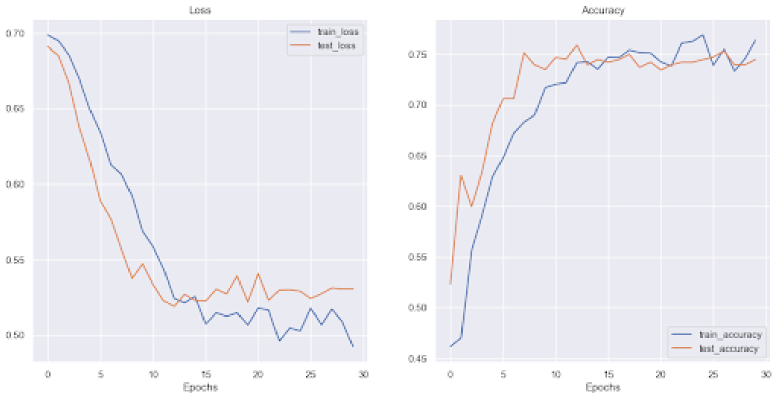
Vision Transformers (ViT)

The Vision Transformer (ViT) model is a relatively new model that emerged in 2021. Originally designed for natural language processing (NLP), it has since been adapted for computer vision applications. ViT models achieve impressive results compared to Convolutional Neural Networks (CNN) and offer several advantages. Notably, ViT models deliver comparable performance while requiring less computational power. However, they don’t have the same ability to naturally “understand” image patterns like CNNs. Instead they have to rely more on extra techniques like regularization and data augmentation. Rather than looking at pixel arrays like CNNs do, ViT models split images into patches, small equal-sized pieces. It uses these patches along with its position in the image and sends this data to the transformer model for analysis. With all of this said, a ViT model was an obvious choice for our team since they show promising results in computer vision applications while also outperforming CNNs when it comes to computational efficiency and accuracy.

Results and Discussion

Visualizations and Quantitative Metrics of CNN

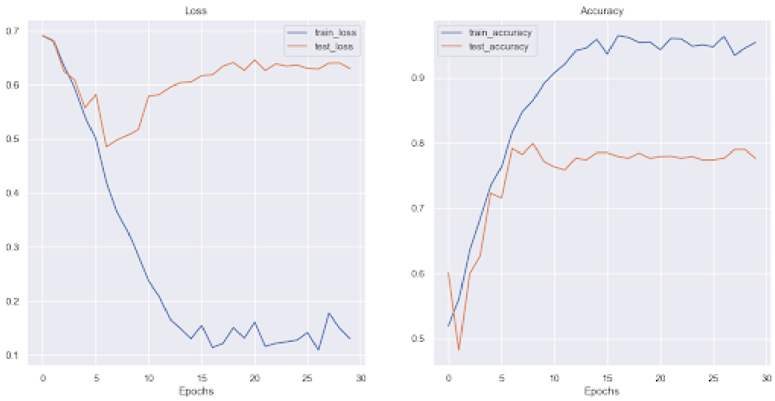
Final Accuracy and Training Loss (CNN)



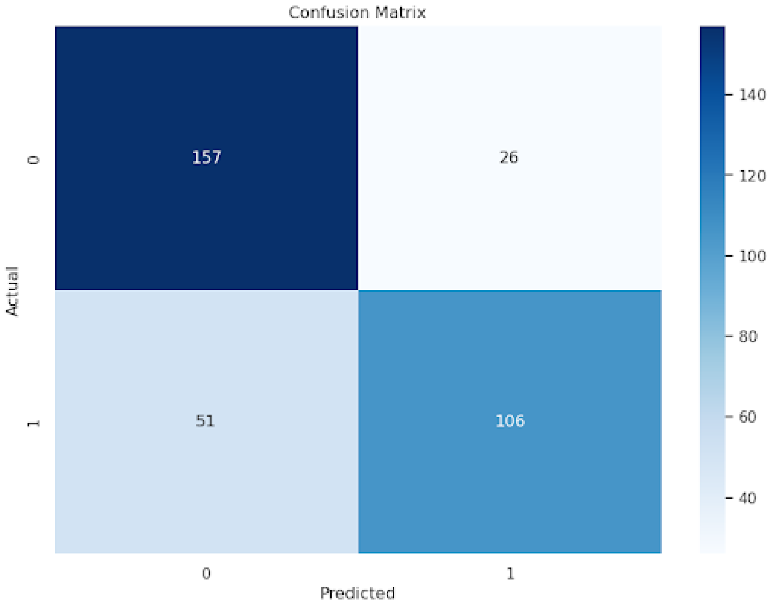
From our training data, we saw that the model’s accuracy increased over each epoch and that the training loss decreased consistently. The test accuracy and loss both follow a similar trend, meaning the model is learning our dataset effectively without major signs of overfitting. Through each epoch, the data shows that the test loss stabilizes as the model progresses and test accuracy lines up with the training accuracy. This implies that the model is adequately generalizing to unseen data, suggesting good robustness.

Additionally, although train and test losses have shown improvement, they remain relatively high. We had experimented with switching over to pretrained models with larger parameter counts to improve accuracy. However, this adjustment had resulted in apparent overfitting due to overlearning the training data without any gain in test accuracy.

Loss and Accuracy of Previous Attempts (CNN)

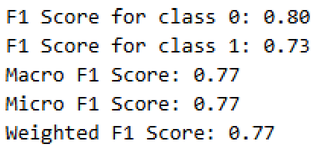


Confusion Matrix (CNN)



The confusion matrix shows the true positive and false positives of the data. True positives (106) and True Negatives (157) indicate correctly identified cases for both classes. False Positives (26) and False Negatives (51) indicate times where the model misclassified the image.

F1 Scores (CNN)



```
F1 Score for class 0: 0.80
F1 Score for class 1: 0.73
Macro F1 Score: 0.77
Micro F1 Score: 0.77
Weighted F1 Score: 0.77
```

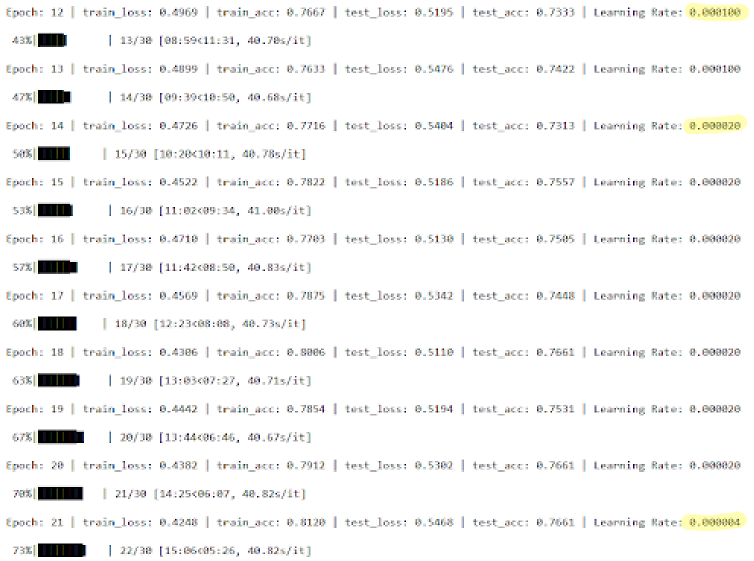
The F1 scores offer a balanced measure of precision and recall: F1 score for class 0 (0.80): suggests relatively high precision and recall for the Negative class, meaning the model is quite effective at recognizing benign tumors. F1 score for class 1 (0.73): suggests slightly lower performance in detecting malignant cases, likely due to subtle variations and issues with the images that make malignant signs harder to recognize. The other three scores, macro, micro, and weighted F1 scores all sit at 0.77 indicating an overall performance across both classes.

Analysis of CNN

While working with the CNN model there were many experimentations, including adjustments with the learning rate to control how quickly our model adapted to new data, higher rates would cause the model to overshoot optimal solutions while lower rates could lead to slower convergence. To balance this, we tried different learning rates but improvement was limited due to the complex dataset.

In terms of loss functions, we experimented with binary cross entropy (binary classification) and cross entropy (multi class classification) for multi-class attempts. We found that using binary cross entropy with logits loss for classification, as it utilizes sigmoid activation directly into the loss function, allowing to efficient distinction between benign and malignant tumors. Another issue was image acquisition differences. The mammograms were taken with varying orientations and angles, sometimes the left of the image was the front of the breast while other images have the front of the breast towards the right of the image. This inconsistency led to issues with our model identifying classifying correctly.

Furthermore, we experimented with AdamW optimization to update the weights in each epoch. The algorithm adjusts how much it changes through weight decay. This helps the model prevent overfitting. As seen from our training, the learning rate was adjusted gradually.



```
Epoch: 12 | train_loss: 0.4969 | train_acc: 0.7667 | test_loss: 0.5195 | test_acc: 0.7333 | Learning Rate: 0.000100

 43%|████      | 13/30 [08:59<11:31, 40.70s/it]

Epoch: 13 | train_loss: 0.4899 | train_acc: 0.7633 | test_loss: 0.5476 | test_acc: 0.7422 | Learning Rate: 0.000100

 47%|████▋     | 14/30 [09:39<10:50, 40.68s/it]

Epoch: 14 | train_loss: 0.4726 | train_acc: 0.7716 | test_loss: 0.5404 | test_acc: 0.7313 | Learning Rate: 0.000020

 50%|█████     | 15/30 [10:20<10:11, 40.78s/it]

Epoch: 15 | train_loss: 0.4522 | train_acc: 0.7822 | test_loss: 0.5186 | test_acc: 0.7557 | Learning Rate: 0.000020

 53%|█████▎    | 16/30 [11:02<09:34, 41.00s/it]

Epoch: 16 | train_loss: 0.4710 | train_acc: 0.7703 | test_loss: 0.5130 | test_acc: 0.7585 | Learning Rate: 0.000020

 57%|█████▋    | 17/30 [11:42<08:50, 40.83s/it]

Epoch: 17 | train_loss: 0.4569 | train_acc: 0.7875 | test_loss: 0.5342 | test_acc: 0.7448 | Learning Rate: 0.000020

 60%|██████    | 18/30 [12:23<08:08, 40.73s/it]

Epoch: 18 | train_loss: 0.4306 | train_acc: 0.8006 | test_loss: 0.5110 | test_acc: 0.7661 | Learning Rate: 0.000020

 63%|██████▎   | 19/30 [13:03<07:27, 40.71s/it]

Epoch: 19 | train_loss: 0.4442 | train_acc: 0.7854 | test_loss: 0.5194 | test_acc: 0.7531 | Learning Rate: 0.000020

 67%|██████▋   | 20/30 [13:44<06:46, 40.67s/it]

Epoch: 20 | train_loss: 0.4382 | train_acc: 0.7912 | test_loss: 0.5302 | test_acc: 0.7661 | Learning Rate: 0.000020

 70%|███████   | 21/30 [14:25<06:07, 40.82s/it]

Epoch: 21 | train_loss: 0.4248 | train_acc: 0.8120 | test_loss: 0.5468 | test_acc: 0.7661 | Learning Rate: 0.000004

 73%|███████▎  | 22/30 [15:06<05:26, 40.82s/it]
```

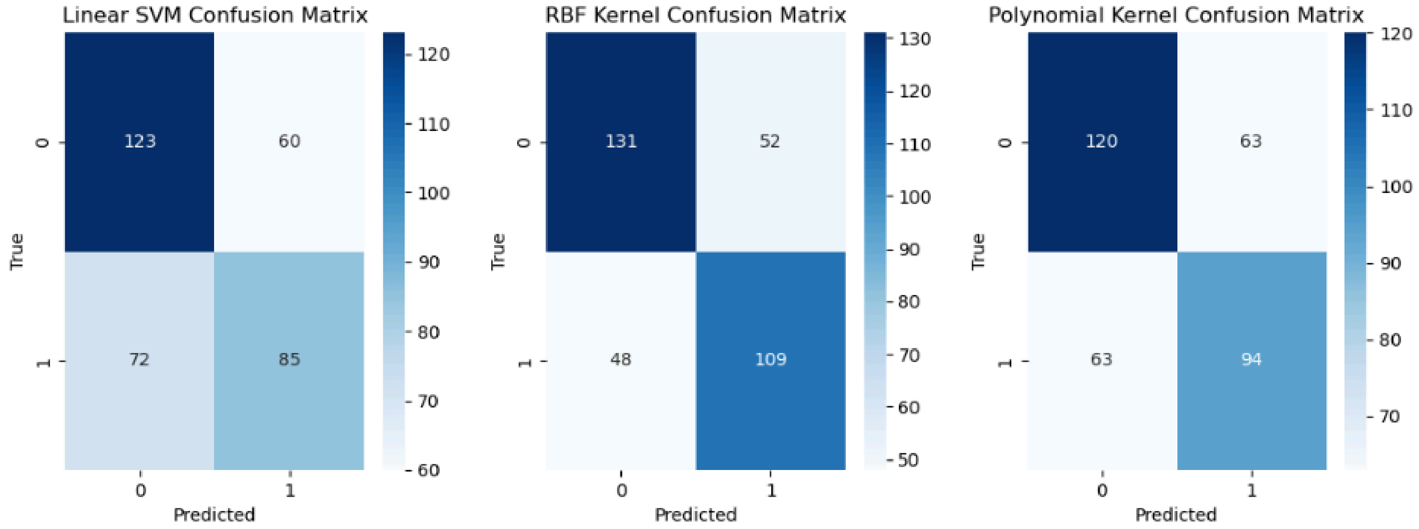
After experimenting with pretrained architectures like ResNet, we found that the advantages offered by using a pretrained model were worth looking into, due to the powerful foundation it provides. We looked at various pretrained models that are trained on larger datasets like ImageNet which could give our model an advantage as it would not have to start from scratch. Although ResNet was difficult to work with, we found success with Efficient Net B1. Starting with a pretrained model allows us to take advantage of its starting weights that encode knowledge and patterns from the original dataset, which gives the model an “idea” of where to begin. In the fine-tuning phase we can adapt the pretrained weights for our specific dataset, which increases the efficiency of our model because it doesn’t have to relearn basic visual features. It recognizes the nuances of the new but can still be fine tuned to deal with a smaller dataset. This accelerated convergence provided better initial accuracy and faster training times. As well as saving computational resources, pre training offers an edge in generalization as well, identifying general patterns despite being used on a broad range of image orientations and angles.

Additional experimentation with larger models were further tested by freezing all but the classification and last few layers. This was tested on EfficientNetV2 and Inception-ResNet-V2 both of which have around 50 million parameters. However the results were insufficient with accuracy stagnating around just 70%. However, freezing the first layers did lower the number of trainable parameters to under 4 million and prevented any substantial overfitting by the model. From this we chose to advance with the model that worked the best for us, which was the much smaller EfficientNet-b1 model.

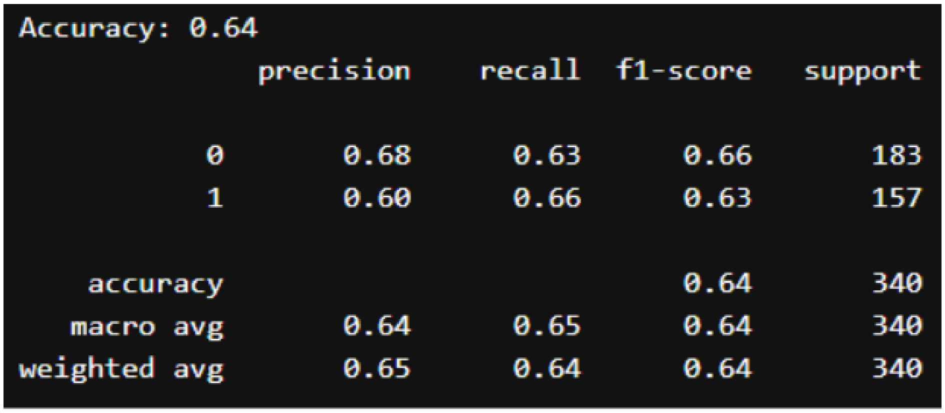
Our CNN model’s current performance would be improved by addressing overfitting issues. Overfitting occurs when the model memorizes details specific to the training data, failing to generalize to new unseen data. To reduce this overfitting we could look at parameter reduction techniques like decreasing the depth of layers or experimenting with dropout regularization. At the same time, it's necessary to look at improving our preprocessing method to better find and delete unnecessary artifacts such as letterings or markings on the mammograms. Currently we use a model that detects letters and deletes them but this can result in certain non-letter markings not getting deleted. Enhancing our preprocessing and attempting to fix the overfitting would lead to better generalization and more accurate results.

Visualizations and Quantitative Metrics of SVM

Confusion Matrix (SVM)



Linear Kernel Scores (SVM)

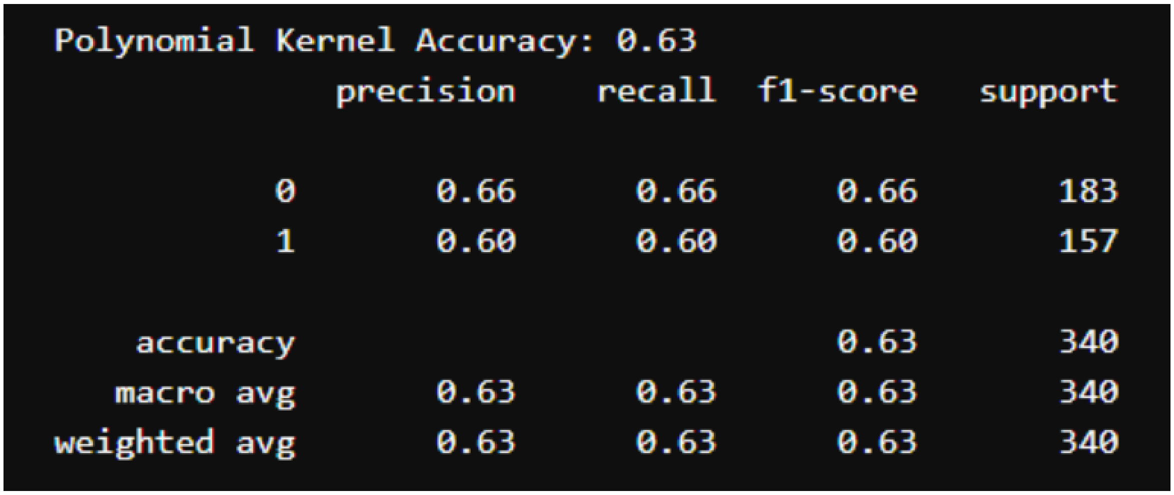


```
Accuracy: 0.64
              precision    recall  f1-score   support

           0       0.68      0.63      0.66       183
           1       0.60      0.66      0.63       157

    accuracy                           0.64       340
   macro avg       0.64      0.65      0.64       340
weighted avg       0.65      0.64      0.64       340
```

Polynomial Kernel with Degree 4 Scores (SVM)

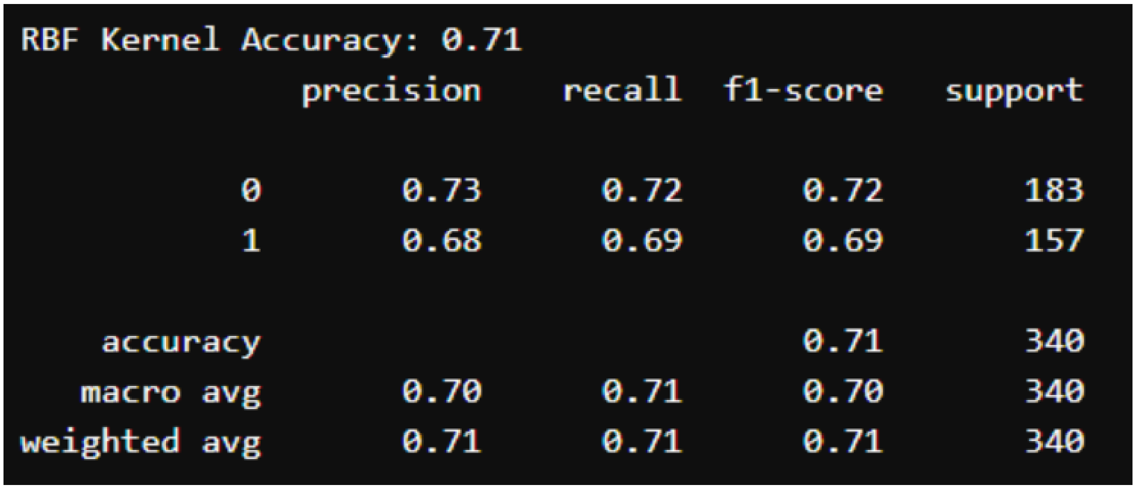


```
Polynomial Kernel Accuracy: 0.63
              precision    recall  f1-score   support

           0       0.66      0.66      0.66       183
           1       0.60      0.60      0.60       157

    accuracy                           0.63       340
   macro avg       0.63      0.63      0.63       340
weighted avg       0.63      0.63      0.63       340
```

RBF Kernel Scores (SVM)

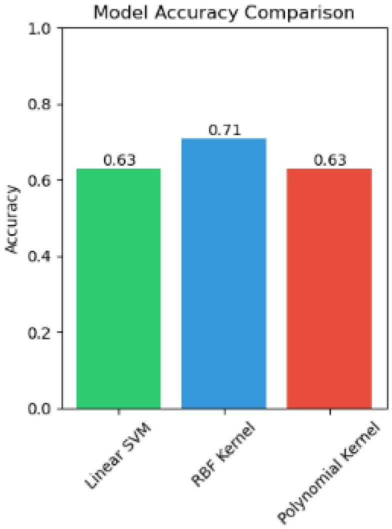


```
RBF Kernel Accuracy: 0.71
              precision    recall  f1-score   support

           0       0.73      0.72      0.72       183
           1       0.68      0.69      0.69       157

    accuracy                           0.71       340
   macro avg       0.70      0.71      0.70       340
weighted avg       0.71      0.71      0.71       340
```

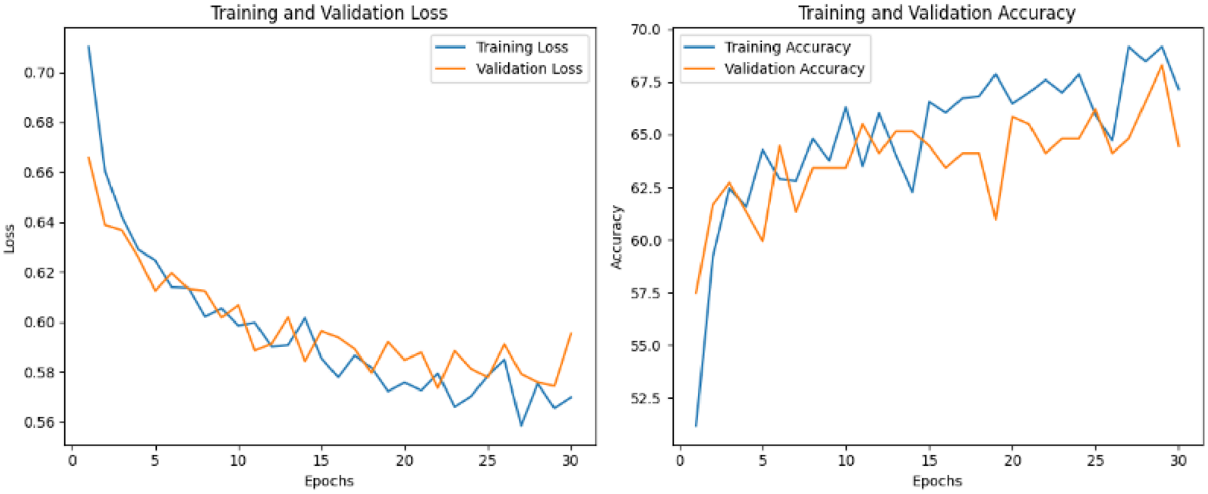
Analysis of SVM



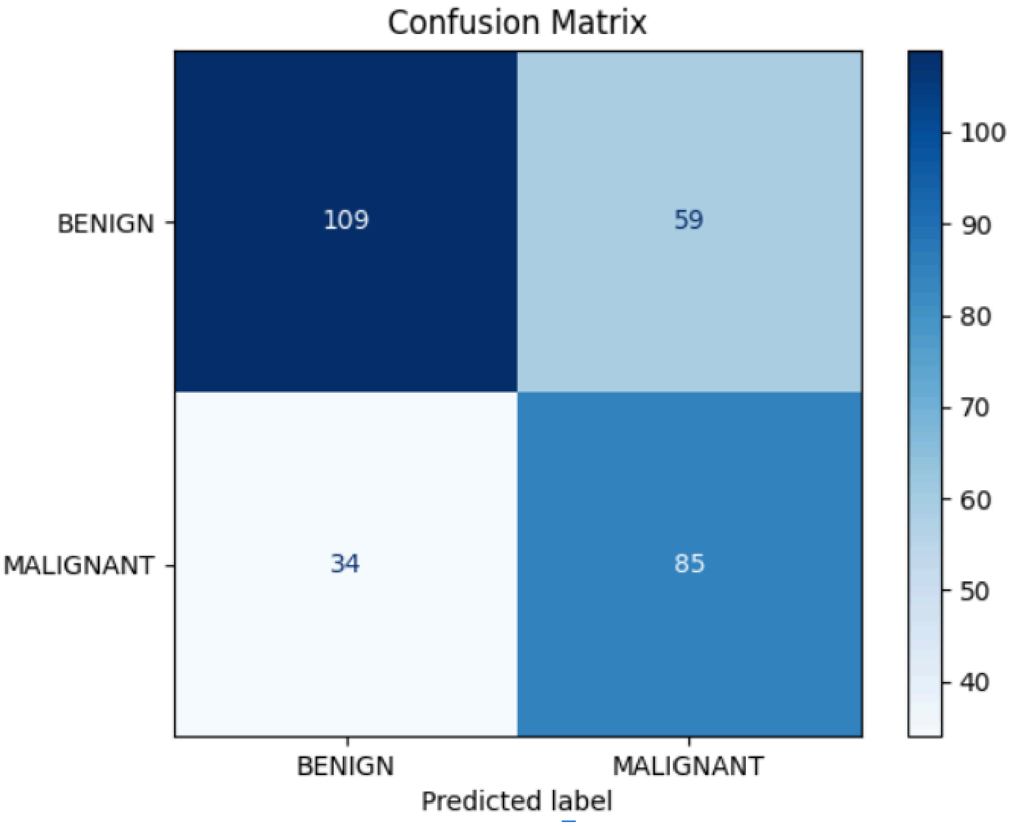
Throughout our tests we saw that RBF performed the best despite what hyperparameters we changed. This was expected due to its ability to handle complex patterns. Mammograms are used to capture very small features. Polynomials worked best with degree 4 with our current hyperparameters. We tried using image augmentation to see whether it would improve the accuracy but as expected there were no improvements but increased the training time slightly. Compared to our other 2 models, SVM only took a few minutes with image augmentation. However, when we didn’t use image augmentation with feature reduction and PCA it only took under a minute.

Visualizations and Quantitative Metrics of ViT

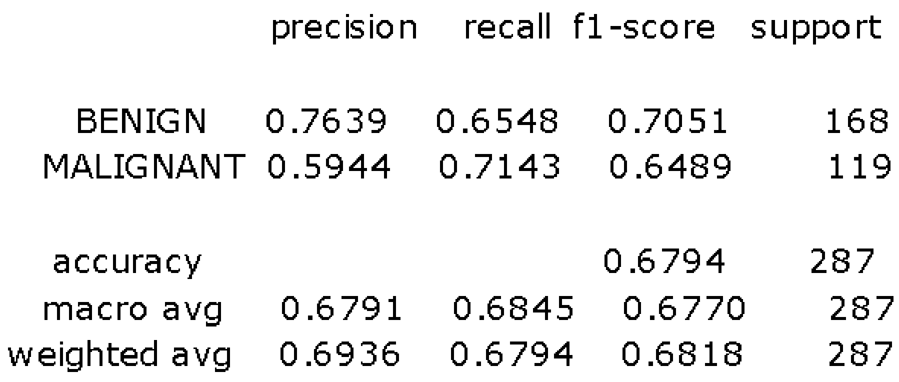
Final Accuracy and Training Loss (ViT)



Confusion Matrix (ViT)



Classification Report (ViT)



```
              precision   recall f1-score   support

     BENIGN     0.7639   0.6548   0.7051       168
  MALIGNANT     0.5944   0.7143   0.6489       119

   accuracy                       0.6794       287
  macro avg     0.6791   0.6845   0.6770       287
weighted avg    0.6936   0.6794   0.6818       287
```

The ViT model demonstrated overall good accuracy, and showed promising performance on the given dataset. It had high precision when identifying benign samples when it predicted them as such, and it had moderate precision with the malignant class, sometimes incorrectly classifying benign samples as malignant.

Analysis of ViT

The Vision Transformer (ViT) was tested with a variety of sample sizes, however we found the model performed best with a larger dataset. When implementing this model, we were faced with severe overfitting. To address this, we implemented dynamic weights adjustments to better align with the data distribution. In addition to overfitting, the model also faced the problem of plateauing. The accuracies would stay consistent throughout each iteration rather than having an increase in performance. To make sure our performance was increasing, we used a scheduler to dynamically control the learning rate. With this, the scheduler would adjust the rate, usually reducing it, when no improvement was detected. For fine-tuning the pretrained model, all layers except the classification head were frozen, focusing adjustments on task-specific features. Despite these efforts, the ViT model’s performance peaked at around 0.73 accuracy but was generally between 0.65-0.68. Training was time intensive due to the model’s complexity, averaging about 75 minutes when using the full dataset.

Comparison of SVM, CNN, and ViT

All three models showed comparable results with similar ranges in their metrics. Each model did have times where images were misclassified and the accuracies of all the models were within 0.1 of each other. Among the SVM models, the SVM with an RBF kernel performed the best. Between the CNN, SVM with an RBF kernel, and the ViT, the CNN achieved the highest accuracy and F1-score for both classes.

Next Steps

While this project has come to an end, it is clear that there is still work that would need to be done before these machine learning models can be integrated in to the healthcare process. Increasing the accuracy of the models would be the most important next step to help avoid false positives and false negatives. In their current states, these models could not safely diagnose a patient. However, as the field of machine learning continues to improve, these models can also improve with it and become a valuable tool for healthcare providers. While they should not be used independently of a medical professional diagnosis, they could one day help catch something that doctors may miss.

Gantt Chart

https://docs.google.com/spreadsheets/d/1zWXrbGhmEucC5g5ZZteOZa2LxzIpYh3c/edit?pli=1&gid=1825123831#gid=1825123831

Contribution Table

Name	Proposal Contributions
Carina Copeland	Final Report, Vision Transformer(coding & implementation, preprocessing, visualization, analysis)
Ezra Kim	Final Report, CNN + SVM Backbone
Kenny Lin	Final Report, preprocessing, CNN and parameter tuning
Sena Korkaya	Final Report, Vision Transformer(coding & implementation, preprocessing, visualization, analysis)
Peter Wang	Final Report, SVM(Coding + Implementation, preprocessing, Data Visualization, analysis)

References

- [1] U.S. Cancer Statistics Working Group, “Cancer Trends.” cdc.gov. https://gis.cdc.gov/Cancer/USCS/#/Trends/ (accessed Oct. 2, 2024)

- [2] U.S. Cancer Statistics Working Group, “U.S. Cancer Statistics Breast Cancer Stat Bite.” cdc.gov. https://www.cdc.gov/united-states-cancer-statistics/publications/breast-cancer-stat-bite.html (accessed Oct. 2, 2024)
- [3] S. A. Joshi, A. M. Bongale and A. Bongale, "Breast Cancer Detection from Histopathology images using Machine Learning Techniques: A Bibliometric Analysis," Library Philosophy and Practice, pp. 1-29, 2021. Available: https://www.proquest.com/scholarly-journals/breast-cancer-detection-histopathology-images/docview/2552127219/se-2.
- [4] M. Darwich and M. Bayoumi, “An evaluation of the effectiveness of machine learning prediction models in assessing breast cancer risk,” Informatics in Medicine Unlocked, vol. 49, pp. 101550-, 2024. Available: https://doi.org/10.1016/j.imu.2024.101550.
- [5] U.S. Cancer Statistics Working Group, “What Causes Hereditary Breast and Ovarian Cancers.” cdc.gov. https://www.cdc.gov/breast-ovarian-cancer-hereditary/causes/index.html (accessed Oct. 2, 2024)
- [6] American Cancer Society, “Limitations of Mammograms | How Accurate Are Mammograms?,” www.cancer.org, Jan. 14, 2022. https://www.cancer.org/cancer/types/breast-cancer/screening-tests-and-early-detection/mammograms/limitations-of-mammograms.html
- [7]“Use rolling-ball algorithm for estimating background intensity — skimage 0.23.2 documentation,” scikit-image.org. https://scikit-image.org/docs/stable/auto_examples/segmentation/plot_rolling_ball.html
- [8]“Canny edge detector — skimage 0.22.0 documentation,” scikit-image.org. https://scikit-image.org/docs/stable/auto_examples/edges/plot_canny.html
- [9]A. R. Beeravolu, S. Azam, M. Jonkman, B. Shanmugam, K. Kannoorpatti, and A. Anwar, “Preprocessing of Breast Cancer Images to Create Datasets for Deep-CNN,” IEEE Access, vol. 9, pp. 33438–33463, 2021, doi: https://doi.org/10.1109/access.2021.3058773.
- [10]“1.17. Neural network models (supervised) — scikit-learn 0.23.1 documentation,” scikit-learn.org. https://scikit-learn.org/stable/modules/neural_networks_supervised.html
- [11]H. Mechria, Khaled Hassine, and Mohamed Salah Gouider, “Effect of Denoising on Performance of Deep Convolutional Neural Network For Mammogram Images Classification,” Procedia Computer Science, vol. 207, pp. 2345–2352, Jan. 2022, doi: https://doi.org/10.1016/j.procs.2022.09.293.
- [12]M. O. Khairandish, M. Sharma, V. Jain, J. M. Chatterjee, and N. Z. Jhanjhi, “A Hybrid CNN-SVM Threshold Segmentation Approach for Tumor Detection and Classification of MRI Brain Images,” IRBM, Jun. 2021, doi: https://doi.org/10.1016/j.irbm.2021.06.003.
- [13]X. Jiang, S. Wang, and Y. Zhang, “Vision transformer promotes cancer diagnosis: A comprehensive review,” Expert Systems with Applications, vol. 252, pp. 124113–124113, May 2024, doi: https://doi.org/10.1016/j.eswa.2024.124113.