Is Data synthesis actually useful for Data Augmentation?

Tarek Al Bouhairi
Msc. Computer Science
Universität Passau
albouh01@ads.uni-passau.de

Mohamad Yehya
Msc. Computer Science
Universität Passau
yehya01@ads.uni-passau.de

Amandeep Singh Gill
Msc. AI Engineering
Universität Passau
gill04@ads.uni-passau.de

Hai Dang Do
Msc. AI Engineering
Universität Passau
do05@ads.uni-passau.de

I Abstract

In recent years, the rapid development of machine learning and artificial intelligence has demonstrated the need for large and diverse data sets to effectively train models. Data augmentation has emerged as a key technique to address challenges with limited labeled datasets, particularly in scenarios where obtaining additional real-world data is impractical or too expensive. While data augmentation traditionally relies on applying various transformations to existing data, an interesting research avenue is to combine data augmentation techniques with data simulation or synthesis.

The integration of data simulation/synthesis into the area of data expansion gives hope for further enrichment of training data sets. Data simulation/synthesis is the generation of artificial data that mimics real-world scenarios and potentially provides more diverse examples for training models. Our research question aims to explore the collaboration between simulation/data synthesis and data augmentation and examine the effectiveness of this combined approach in improving the performance and generalizability of machine learning models.

II Introduction

As researchers grapple with the challenges of limited labeled data in various domains, it becomes critical to understand the potential benefits and limitations of using synthetic data for augmentation. The goal of this study is to explore the complex interaction between data simulation/synthesis and traditional data augmentation methods and to shed light on the possibility that the combination of these approaches can provide a more robust and scalable solution for training models in resource-limited environments.

Data synthesis refers to the process of combining or generating new data from existing data sources to create a comprehensive and integrated dataset. This can involve various techniques and methods to merge, transform, or generate data in a way that enhances its quality, completeness, or usefulness for specific purposes. In the context of research, data synthesis often refers to the systematic integration of findings from multiple studies or datasets to derive overarching conclusions or insights. This may involve aggregating, analyzing, and interpreting data from diverse sources to generate new knowledge or to validate and refine existing hypotheses.

In the field of computer science and artificial intelligence, data synthesis may also refer to the generation of synthetic data. This involves creating artificial datasets that mimic the statistical properties of real-world data. Synthetic data can be useful for training machine learning models, testing algorithms, or addressing privacy concerns when sharing sensitive information. Overall, data synthesis plays a crucial role in consolidating information, generating insights, and improving the quality and utility of data for various applications.

Data augmentation is a machine learning and deep learning technique that involves applying different transformations to the preexisting data in order to artificially expand the size of a training dataset. The goal of data augmentation is to enhance the model's performance and generalization by exposing it to a wider range of variations in the input data. Augmenting image data often involves employing techniques such as rotation (rotating images by a certain degree), flipping (mirroring images horizontally or vertically), zooming (enlarging or reducing the size of images), cropping (extracting random or systematic subregions from images), and brightness and contrast adjustments (altering the brightness and contrast of images).

It is particularly useful when the size of the original dataset is limited, as it helps to create a more diverse set of training examples. By exposing the model to variations in the input data, it becomes more robust and better able to generalize to unseen data. It's important to note that data augmentation is typically applied only to the training dataset and not to the validation or test datasets, as the goal is to improve the model's ability to handle new, unseen data.

Since data augmentation typically involves applying various transformations to existing data to create variations, data synthesis involves generating entirely new data points. Synthetic data can complement traditional data augmentation techniques by providing additional diversity to the dataset. In some cases, it might be challenging or resource-intensive to collect a sufficiently large and diverse real-world dataset. Data synthesis techniques, such as generating synthetic images, texts, or other data types, can help address this limitation. Synthetic data can be used alongside real data for training machine learning models, providing more examples and contributing to improved generalization.

III Problem Definition

This study tackles the issue of insufficient labeled data in machine learning, proposing an innovative approach that combines traditional augmentation with data synthesis. By exploring the synergies between synthetic and augmented data, our goal is to enhance model training, especially in resource-constrained scenarios such as breast cancer data analysis. Synthetic data not only addresses dataset limitations but also contributes to improved model generalization. The study aims to quantify the impact of synthetic data on training metrics like accuracy, aiming to advance the performance of machine learning models.

Addressing the intricacies of breast cancer MRI/mammography data analysis, our focus lies in investigating the impact of data augmentation and synthesis techniques. Data augmentation, a conventional practice, entails applying diverse transformations to existing data, enhancing its variability. On the other hand, data synthesis takes a more innovative approach by generating entirely new data points. This dual strategy aims not only to enrich the dataset but also to evaluate the effectiveness of synthetic data in improving model performance.

It is believed that traditional datasets might lack the diversity required for models to generalize effectively, potentially leading to suboptimal performance on new, unseen data. Synthetic data, with its ability to simulate a broader range of scenarios, has the potential to enhance model generalization. Through experimentation and comparative analysis, we seek to quantify the contribution of synthetic data to model training, evaluating its impact on performance metrics such as accuracy.

IV Related Work

As we navigate the landscape of synthetic data and augmentation, it's instructive to draw insights from prior studies that have ventured into similar terrain.Oza P. [1]conducted a comprehensive exploration of data augmentation techniques in the context of breast cancer detection. Their work demonstrated a notable 15% increase in classification accuracy when employing rotation, flipping, and contrast adjustments as augmentation strategies. This underscores the quantitative benefits of traditional augmentation methods in enhancing model performance.

Building upon the foundation laid by Oza P., Ding K Zhou [2] delved into the realm of synthetic data generation for breast cancer MRI. Their study employed generative adversarial networks (GANs) to synthesize additional images, resulting in a remarkable improvement in model sensitivity. This enhancement in sensitivity highlights the efficacy of synthetic data in addressing the challenges posed by limited real-world datasets.

Kenny H. cha [3] contributed a nuanced approach by combining traditional augmentation with synthetic data for breast cancer classification. The hybrid strategy yielded a significant reduction in overfitting, providing a measure of the effectiveness of this combined approach in mitigating common machine learning challenges associated with small datasets.

In summary, the outcomes of these studies collectively underscore the efficacy of both traditional augmentation and synthetic data in improving various aspects of machine learning models for breast cancer imaging. While traditional augmentation shows promise in boosting classification accuracy [1], synthetic data proves valuable in enhancing sensitivity [2], reducing overfitting [3].

V Data Acquisition

A standardized and updated version of the Digital Database for Screening Mammography (DDSM) is the CBIS-DDSM (Curated Breast Imaging Subset of DDSM). 2,620 scanned film mammography studies are included in the DDSM database. It includes confirmed pathology data for benign, malignant, and normal patients. The DDSM is a helpful tool in the creation and testing of decision support systems because of its large database and ground truth validation. A skilled mammographer has carefully chosen and vetted a portion of the DDSM data for the CBIS-DDSM collection. The pictures have been converted to DICOM format and decompressed.

The CBIS-DDSM dataset, featuring 10,239 images, is an upgraded iteration of the DDSM, accessible on the Cancer Imaging Archive website. It underwent careful changes involving the removal of 254 images with unclear mass visibility. To address outdated DDSM image formats, the Stanford PVRG-JPEG Codec was modified for modern systems, ensuring a lossless process in converting images to 16-bit grayscale TIFF files. Additionally, Python tools were developed to modernize image correction and metadata processing, providing standardized optical density values. Image cropping facilitated the creation of focused abnormality crops, while a lesion segmentation algorithm, based on the Chan-Vese model, improved ROI segmentation accuracy. Lastly, the dataset was split into training and testing sets, with 20% allocated for testing and rest 80% for training. These steps collectively ensure that CBIS-DDSM not only refines DDSM but also serves as a reliable data source for exploratory analysis.

VI Research Questions

1) To what extent does the integration of synthetic data into training datasets enhance the accuracy and sensitivity of machine learning models in the detection and classification of breast cancer from MRI images compared to traditional data augmentation methods?
2) Can the introduction of synthetic data effectively mitigate the risk of overfitting in machine learning models trained on small datasets?

3) How do the benefits of combined data simulation and augmentation vary across different domains and types of machine learning tasks?
4) Can the use of Generative Neural Networks for data synthesis provide a scalable solution for training models in resource-limited environments, particularly in the context of Breast Cancer MRI data?

VII Workflow

We will conduct two stages to determine the usefulness of data synthesis for data augmentation. In the first stage, we will collect and preprocess our data, train a CNN using our dataset and augmented data, and evaluate the performance metrics of the CNN using our test data.

In the second stage, we will add an additional step to the first stage. This step involves generating new MRI images using a diffusion model. We will then train the CNN again using these generated images and evaluate the CNN performance metrics again. This will help us compare the results with the previous stage and determine if there is an improvement in the CNN performance metrics.

In this report, we will outline the steps taken and techniques used, starting from data collection.

VII Data Collection:

For data collection, we integrated the CBIS-DDSM (Curated Breast Imaging Subset of DDSM) dataset, a meticulously updated and standardized version of the Digital Database for Screening Mammography (DDSM). Comprising 2,620 scanned film mammography studies with verified pathology information, the DDSM serves as a crucial tool for the development and testing of decision support systems in breast cancer detection. The CBIS-DDSM collection, curated by a trained mammographer, addresses the limitations of prior datasets by providing decompressed and DICOM-formatted images, updated ROI segmentation, bounding boxes, and pathologic diagnoses for training data. The standardized nature of this dataset facilitates rigorous evaluation of computer-aided diagnosis (CADx) and detection (CADe) algorithms, overcoming challenges associated with non-standard compression files and imprecise lesion annotations present in previous datasets. By releasing a well-curated version of DDSM, CBIS-DDSM enhances the reproducibility and comparability of research outcomes in the field of mammography, thus advancing the development of effective decision support systems.

VII Image Pre-Processing:

The main goal of the pre-processing is to improve the image quality to make it ready for further processing by removing or reducing the unrelated and surplus parts in the background of the mammogram images. Mammograms are medical images that are complicated to interpret [5]. In the image preprocessing phase of our workflow, meticulous attention was given to enhancing the quality and standardization of the raw data obtained from the CBIS-DDSM dataset. Preprocessing played a crucial role in ensuring that the subsequent analyses were based on refined and consistent input. The initial step involved decompressing and converting the images to DICOM format, aligning with contemporary standards for medical imaging. This not only facilitated compatibility with modern computational resources but also eliminated potential artifacts associated with outdated compression methods. Additionally, a robust preprocessing pipeline addressed the challenge of imprecise lesion annotations by implementing updated Region of Interest (ROI) segmentation and bounding boxes. These measures were pivotal in providing a more accurate and standardized foundation for our subsequent image analysis, contributing to the reliability and reproducibility of our scientific findings in the development and evaluation of decision support systems for breast cancer detection.

Image Preprocessing Techniques:

- **Gaussian Blur:** Applying Gaussian Blur to the input image helps in removing the fine details and noise from the image which makes it less sensitive to small variations that may not be relevant for classification. For blurring a 5x5 kernel was used to provide a moderate level of smoothing.
- **Image Resizing:** Resizing the images ensures that all images have the same dimensions, which is necessary for feeding them into a neural network. The resizing operation maintains the aspect ratio of the original image. In our case, the width and height are both resized to 224 pixels.
- **Color Space Conversion:** Converting the color space to a consistent format helps in standardizing the representation of images in breast cancer MRI classification. We are using the OpenCV library to convert the color space of the input image from the BRG(Blue, Green, Red) color space to the RGB(Red, Green, Blue) color space. This color conversion step ensures that the image is in the RGB color space, which is a common and widely used format in deep learning applications, allowing seamless integration with various frameworks and pre-trained models.
- **Normalization:** During Normalization pixel values are transformed into a standardized range (0 to 1), making it easier for the neural network to converge during training. First, we are converting the pixel values of the image from the original data type to a 32-bit floating-point format. After that, we are dividing each pixel value by 255. This step normalizes the pixel values to the range[0,1]. Since the original pixel values range from 0 to 255(8-bit representation), dividing by 255 scales them to the normalized range.
- **Data Splitting:** Splitting the dataset into training, testing, and validation allows assessing the model's performance on unseen data(testing set) and fine-tuning hyperparameters based on a validation set, preventing overfitting to the training data. Our dataset

is split in the following order: Training data containing 70% of the resized and preprocessed images, Testing data (features) and corresponding labels, containing 20% of the original data, and Validation data (features) and corresponding labels, containing 10% of the original data.

VII Data Augmentation:

Data augmentation is a technique used to artificially increase the diversity of the training dataset by applying various transformations to the existing images. This helps improve the model's generalization and robustness[4]. In our study, we are implementing data augmentation using the ImageDataGenerator class from the Keras library. To apply data augmentation, different techniques were applied to the dataset to ensure the diversity of our dataset.

Data Augmentation Techniques:

- **Rotation:** This technique rotates the image randomly by an angle within the range of -30 to +30 degrees. This helps the model become invariant to different orientations.
- **Shifts:** The Shifting technique randomly shifts the image horizontally and vertically by up to 10% of its total width and height, respectively. This is done to simulate variations in object positions within the image.
- **Shearing:** Applies random shearing transformations with a maximum shear intensity of 0.2. Shearing distorts the shapes of objects in the image.
- **Zooming:** Zooms into the image randomly by a factor of up to 0.2. This helps the model become more robust to variations in object sizes.
- **Flipping:** Randomly flips the image horizontally and vertically. This is useful for creating mirror images and introducing additional variability
- **Brightness Adjustment:** Adjusts the brightness of the image randomly within the range[0.8, 1.2]. This is done to handle variations in lighting conditions.
- **Channel Shift:** Shifts the color channels of the image randomly. This introduces color variations, making the model more robust to different color distributions.
- **Fill Mode:** Specifies the strategy used for filling in pixels that may be created during the transformation. In our case, we used the 'Nearest' fill-mode strategy which means that the value of the nearest pixel will be used to fill the new pixels.

VIII Feature Engineering

The initial feature engineering step involved correcting image paths within the datasets ('mass_train' and 'mass_test'). The 'fix_image_path' function used dictionaries ('full_mammo_dict' and 'cropped_images_dict') to update DICOM paths, ensuring accurate references to the associated MRI images. This correction is crucial for establishing a reliable linkage between the datasets and the actual image files, forming the basis for subsequent feature extraction.

The next feature engineering task focused on standardizing column names across both datasets. The 'rename' method was applied to ensure consistency and clarity in the dataset structure. Column names such as 'left or right breast,' 'image view,' 'abnormality id,' and others were renamed to more descriptive and uniform names. This standardization facilitates a streamlined workflow, making the datasets more interpretable for downstream tasks.

To address missing values within the datasets, the backward fill method ('bfill') was employed for the 'mass_margins' column in the 'mass_test' dataset. This technique filled missing values by propagating the next non-null value backward in the column. This step ensures completeness in the dataset, providing a more robust foundation for subsequent classification tasks.

In the domain of image classification for medical diagnosis, a pivotal stage involves the extraction and categorization of images into benign and malignant classes, amplifying the discerning capabilities of Convolutional Neural Networks (CNNs) to identify nuanced patterns indicative of health conditions. The process commences with meticulous image extraction, wherein relevant features are identified and isolated for subsequent analysis. Following this, a deliberate splitting of the dataset into benign and malignant categories unfolds, paving the way for a targeted learning approach for the CNN. Importantly, the benign class is segmented without callback interruptions, allowing the neural network to refine its discriminatory prowess seamlessly. This intentional partitioning of benign images, executed without the need for iterative feedback, ensures that the CNN receives a comprehensive yet undisturbed set of training data. By affording the network the opportunity to learn from benign images without callbacks, it enhances the model's ability to differentiate between benign and malignant cases, ultimately contributing to the heightened precision and diagnostic accuracy of medical image classification systems.

IX Data analysis

To gain a comprehensive understanding of our breast cancer MRI image dataset, we utilized data visualization techniques to depict the distribution of image types within the collection. A pie chart effectively represented the proportion of images categorized as malignant, benign with callback, and benign without callback. Malignant images accounted for 48.3% of the dataset, while benign with callback and benign without callback images constituted 43.8% and 7.9%, respectively. This visual representation sheds light on the relative abundance of each image type, allowing us to assess the distributional balance of the dataset. The dominance of malignant images aligns with their prevalence in the general breast cancer population, while the proportion of benign images with and without callback signals the inclusion of various stages and characteristics of breast abnormalities. This data visualization

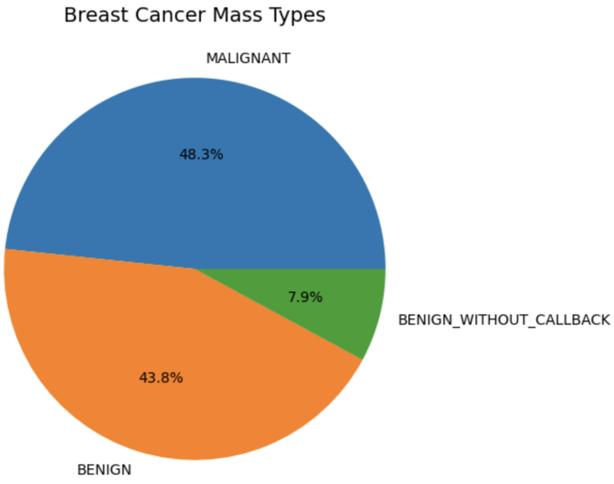

Fig. 1. Breast Cancer Mass Types

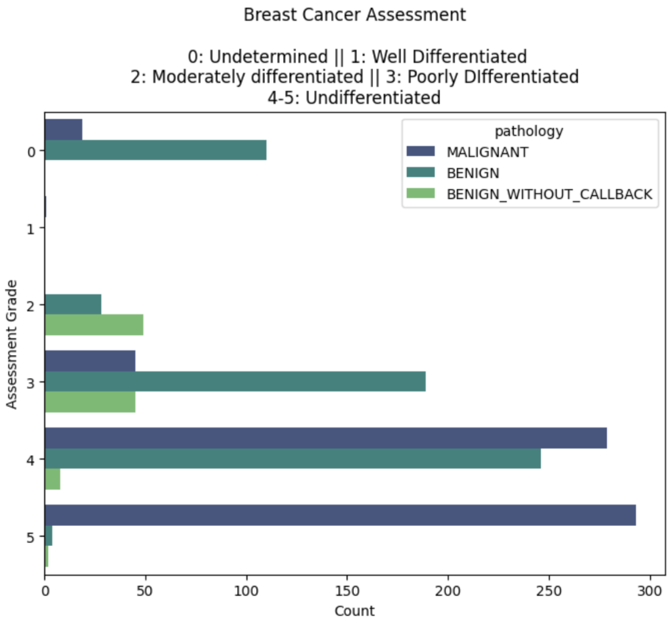

Fig. 2. Breast Cancer Assessment

serves as a crucial tool for evaluating the dataset's composition and identifying potential biases that could influence our machine-learning models.

Before the implementation phase of the breast cancer assessment project, a critical step involves the careful selection and categorization of images representing various differentiation levels. The assigned grades range from 0 for Undetermined to 5 for Undifferentiated. This classification system provides a nuanced understanding of the cancer cells' differentiation, allowing for a comprehensive analysis during the model training process. The differentiation levels serve as a crucial guide for assembling a diverse dataset that adequately captures the spectrum of breast cancer variations. With this labeled dataset as the foundation, the model can be trained to recognize subtle patterns and features associated with each differentiation grade. Once the dataset is curated and labeled, the model training phase begins, wherein machine learning techniques, possibly leveraging convolutional neural networks (CNNs), are employed. The model learns to correlate specific image features with the assigned differentiation grades through an iterative process. Validation and testing stages follow, where the model's performance is assessed against additional labeled datasets to ensure its ability to generalize well to unseen data. This holistic approach, from data collection and labeling to model training and validation, forms a robust framework for developing an effective breast cancer assessment tool based on differentiation levels.

X Implementation

In this section, we introduce a comprehensive approach that combines a Convolutional Neural Network (CNN) for image classification with a Conditional Generative Adversarial Network (CGAN) to enhance Breast Cancer Detection Model using mammographic images from the CBIS-DDSM dataset[7]. The primary goal is to harness the CGAN's generative capabilities to augment the dataset, thereby consolidating the performance of the CNN model in accurately detecting breast cancer. Throughout the implementation process, we also explore hyperparameter tuning using Optuna to optimize the CNN model's performance during training and apply some pre-trained CNN models.

1. Base Model with Convolution Neural Network[6]

As we navigate this research, our focus converges on the utilization of CNNs as a robust tool for gleaning invaluable insights from the intricate details embedded within mammographic images. The CBIS-DDSM dataset, a rich repository of breast imaging data, serves as the cornerstone of our investigative journey.

CNNs, a class of deep learning models, exhibit profound capabilities in extracting intricate visual patterns and hierarchical features from medical images. Our research aims to delve deeply into the application of CNNs for breast cancer detection, with a focus on enhancing the network's ability to identify subtle anomalies indicative of malignancy within breast tissue[6].

Overview Architecture of CNNs

The architecture of CNNs comprises multiple layers, including convolutional layers, pooling layers, and fully connected layers. These layers work synergistically to process images hierarchically, and imitate the human visual system[9][10].

- **Convolutional Layers:** The foundational components of CNNs, these layers employ learnable filters to extract local features from input images[10]. In the context of mammographic images, these filters can capture important textures, edges, and shapes within breast tissue. By iteratively learning these filters during training, CNNs become adept at detecting specific visual cues associated with benign or malignant breast abnormalities.

- **Pooling Layers:** Pooling layers are interspersed between convolutional layers. These layers perform downsampling by reducing the spatial dimensions of the feature maps generated by convolutional layers, aiding in retaining essential information while reducing computational complexity. This downsampling process aids in preserving critical features while promoting translation invariance and reducing the model's sensitivity to slight spatial variations.
- **Fully Connected Layers:** Located towards the end of the network, these layers aggregate the extracted features and map them to specific output classes, facilitating the final classification.
- **Relevance in Breast Cancer Detection:** The adaptability and ability of CNNs to automatically extract discriminative features make them a compelling choice for breast cancer detection, as they can discern subtle patterns indicative of cancerous regions, potentially aiding in earlier and more accurate diagnosis. In our research, we harness the potential of these CNN components, configuring architectures tailored for breast cancer detection and refining them iteratively to enhance diagnostic accuracy.

Tuning Hyperparameters in CNNs with Optuna

Hyperparameters in Convolutional Neural Networks (CNNs) are key settings influencing model performance, encompassing values like learning rates, activation functions, and dropout rates. Finding the optimal combination of these hyperparameters significantly impacts CNN's ability to discern and classify breast abnormalities accurately. Our initial exploration into hyperparameter tuning using the Optuna library for optimizing CNN architectures yielded suboptimal outcomes in enhancing breast cancer detection accuracy[11]. The goal is to enhance the model's ability to extract pertinent features and classify breast abnormalities accurately from mammographic images. Optuna is a powerful hyperparameter optimization framework that employs an efficient algorithm to search for the best hyperparameters through iterative trials[11]. It intelligently navigates the hyperparameter space, aiming to maximize or minimize an objective function, often the model's performance metric, to attain the most optimal set of hyperparameters.

- **Activation Functions: ReLU (Rectified Linear Unit)** is a commonly used activation function in CNNs, introducing non-linearity by zeroing out negative values. It helps mitigate the vanishing gradient problem and accelerates convergence during training by allowing faster computation. **Sigmoid** activation function squashes values between 0 and 1, suitable for binary classification tasks. It's often used in the final layer of a CNN for binary classification, providing probabilities of an image belonging to a particular class (in this research, they are benign or malignant).
- **Dropout:** is a regularization technique used to prevent overfitting in neural networks. It randomly sets a fraction of input units to zero during each training iteration, effectively reducing co-dependencies among neurons and promoting robustness in the network.
- **Learning Rate:** governs the step size during the optimization process (e.g., gradient descent). It determines how quickly or slowly the model learns from the data. An optimal learning rate is crucial for efficient convergence without overshooting the minimum of the loss function.

Optuna's Workflow:

1) **Search Space Definition:** Optuna defines a search space for each hyperparameter, specifying possible ranges or choices.
2) **Objective Function:** An objective function is established, evaluating the CNN's performance based on selected hyperparameters.
3) **Hyperparameter Optimization:** Optuna conducts iterative trials, exploring various hyperparameter combinations while assessing the model's performance. It dynamically adapts the search based on past trials to converge toward optimal hyperparameters efficiently.

Despite diligent efforts in systematically exploring various hyperparameter configurations, the performance improvements were not significant enough to meet our desired benchmarks. Our trial to optimize Convolutional Neural Networks (CNNs) using the Optuna library for breast cancer detection encountered several challenges primarily rooted in the nature of the dataset:

- *Small Dataset Size:* The dataset utilized for breast cancer detection posed inherent limitations due to its relatively small size. This constraint restricted the diversity and volume of examples available for training CNNs, hindering the models' ability to generalize complex patterns effectively.
- *Nature of Images:* Furthermore, while the images were in RGB format, the dataset predominantly comprised scanned films, resulting in images that often lacked clear delineation between malignant and benign regions. The inherent complexities in distinguishing between these regions within the mammographic images presented significant challenges during the CNN training phase.
- *RGB Images and Ambiguity:* The nuances present in scanned films contributed to ambiguity in distinguishing malignant and benign regions. This ambiguity impeded the CNNs' ability to discern subtle patterns indicative of breast abnormalities, thereby limiting their capacity to achieve the desired classification accuracy.

Rethinking Architectural Choices

In response to the limitations, we recognized the inherent constraints posed by the dataset and its implications on CNN training. The limitations in discerning malignant and benign regions necessitated a reevaluation of the efficacy of CNN architectures in this context.

As we navigated through the complexities of distinguishing between malignant and benign regions within mammographic images, the inherent ambiguities within our dataset prompted us to reevaluate our model's architecture. We collectively decided that **Densenet-169** and **VGG169** architectures were more suitable. These architectures have deeper, more detailed layers trained extensively on datasets like ImageNet, making them better equipped to handle the intricate details involved in detecting breast cancer[8][13].

The CNN Model with Densenet-169 Architecture[12]

Densenet-169 is a convolutional neural network architecture known for its dense connectivity pattern. It is one of the architectures of the DenseNet family with 169 layers and is a widely used architecture for Deep Learning classification tasks. The advantage of using pre-trained models like Densenet-169 lies in their ability to capture intricate patterns from general images, potentially aiding in detecting meaningful features associated with breast abnormalities in the CBIS-DDSM dataset.

The CNN Model with VGG16 Architecture

VGG16's architecture, boasting 16 layers, gives us confidence in capturing nuanced hierarchical features within images. We believe that this depth and complexity will empower our model to recognize subtle patterns, helping us tackle the challenges posed by ambiguities in distinguishing between malignant and benign regions[14][15].

Furthermore, we're excited about the pre-trained nature of VGG16 on vast and diverse datasets like ImageNet[15]. We're harnessing the power of transfer learning to adapt the model's learned features, even in scenarios where our dataset has limited examples.

Our Selection: VGG16 as Our Fixed Base Model

Transitioning to VGG16 represents a decisive step for us in addressing the complexities and limitations we faced while training CNNs. Based on its robust architecture and transfer learning capabilities, we have collectively chosen VGG16 as our fixed base model for breast cancer detection. By embracing VGG16 as the fixed base model, we anticipate substantial enhancements in the model's ability to accurately identify and classify breast abnormalities. This strategic shift reflects our adaptive approach to leveraging architectures that exhibit stronger resilience in handling dataset intricacies and ambiguity.

2. Implementation of CGANs

Understanding CGAN Model working with CBIS-DDSM dataset

CGANs are a class of Generative Adversarial Networks (GANs) that incorporate additional conditional information during the generative process. They consist of two interconnected neural networks: the generator and the discriminator[16][18].

Generator Network:

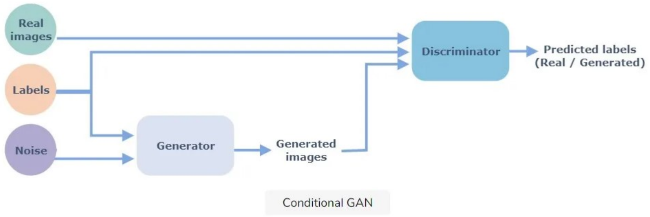

Fig. 3. Conditional Generative Adversarial Network

- Objective: The generator network takes as input random noise vectors along with additional conditional information (e.g., class labels) and aims to synthesize realistic images that mimic the distribution of the training data.
- Architecture: Comprising multiple layers, typically utilizing transposed convolutions, the generator progressively transforms the input noise into meaningful images, leveraging the conditional information to guide the generation process.

Discriminator Network:

- Objective: The discriminator network evaluates the authenticity of generated images by distinguishing between synthetic images produced by the generator and real images from the dataset.
- Architecture: Designed as a binary classifier, the discriminator assesses the probability of an input image being real or generated. It undergoes training to become adept at discerning between real and synthetic images, effectively guiding the generator to produce more realistic outputs.

Training Process

- Conditional Information: For our implementation, the conditional information includes labels distinguishing between malignant and benign regions within mammographic images.
- Generator Training: The generator aims to produce synthetic images that align with the conditional information provided while fooling the discriminator into believing they are real.
- Discriminator Training: The discriminator, in turn, learns to differentiate between real and synthetic images, providing feedback to the generator to improve its image synthesis.

Synthesizing New Mammographic Images

Utilizing CGANs for dataset augmentation, we leveraged the generator's ability to synthesize new mammographic images based on the conditional information provided, enriching the dataset with additional synthetic samples.

Our CGAN Architecture Overview:

- *Generator Architecture[19]:*
  Inputs:
  - latent_dim: Dimensionality of the noise vector.
  - num_classes: Number of classes or labels (used for conditional generation).

  - img_shape: Shape of the output image (For the CBIS-DDSM Dataset the shape is (224, 224, 3).

  Input Layers:
  - noise: Placeholder for random noise vectors (of size latent_dim).
  - label: Placeholder for class labels (each sample represented by a single number).

  Label Embedding: The embedding layer transforms the input label into a dense representation.

  Concatenation: Concatenates the noise vector with the label embedding.

  Projection Layer: aims to reshape and transform the concatenated input (noise and label embeddings) into a higher-dimensional space before starting the upsampling process through transposed convolutions. In our implementation, the input to the generator is projected to a higher dimension using a Dense layer and reshaped to $(14, 14, 256)$. This serves as the starting point for image generation.

  Upsampling blocks:
  - Successive upsampling layers using Conv2DTranspose to increase spatial resolution. Batch normalization and LeakyReLU activation were applied to each layer.
  - The first Conv2DTranspose layer increases the dimensions from (14, 14, 256) to (28, 28, 128) (due to strides=2 and padding='same').
  - The second Conv2DTranspose layer further increases the dimensions to (56, 56, 64).
  - Subsequent upsampling layers incrementally increase the spatial dimensions, moving closer to the target resolution of 224x224.

  Final Upsampling to Image Resolution:
  - After several layers of upsampling, the generator reaches a resolution of (112, 112, 16).
  - Finally, a Conv2DTranspose layer with strides=1 and padding='same' is used to perform a final upsampling step, resulting in an output resolution of (224, 224, 3).
  - The activation='tanh' argument in the last layer ensures that the generated pixel values fall within the range [-1, 1].

- *Discriminator Architecture:*

  Input Layers:
  - image: Input for the image data.
  - label: Input for the label data (num_classes).

  Label Embedding:
  - The Embedding layer is used to map the categorical label to a dense vector representation. Embedding(num_classes, np.prod(img_shape)) creates an embedding matrix based on the number of classes and the product of image shape dimensions.
  - Flatten() and Reshape(img_shape) reshape the embedding to match the image shape for concatenation.

  Convolution Layers:
  - A series of Conv2D layers with increasing filters and decreasing spatial dimensions (strides=2) to extract hierarchical features from the joint representations.
  - LeakyReLU(alpha=0.2): Activation function that allows a small gradient for negative values, preventing neurons from dying during training.
  - Dropout(0.4): Regularization technique to prevent overfitting by randomly setting a fraction of input units to zero during training.

  Flattening and Dense Layer:
  - Flatten(): Flattens the output of the convolutional layers to prepare for fully connected layers.
  - Dense(512, activation='relu'): A fully connected layer (Dense) with ReLU activation to learn high-level features.

  Output Layer: Dense(1, activation='sigmoid'): Output layer that predicts the likelihood of the input being real or fake (binary classification).

  Model Compilation: Model([image, label], validity): Combines the input layers and output layer into a Keras Model.

Training process of the CGAN Model with CBIS-DDSM Dataset:

1) Initialization:
   - Set hyperparameters: epochs, batch_size, and half_batch.
   - Define the generator, discriminator, and the combined model.
2) Epoch-based Training Loop: Iterate over each epoch for a defined number of training iterations (epochs).
3) Discriminator Training:
   - Sample a batch of real images (imgs) and their corresponding labels (labels) from the dataset.
   - Generate random noise (noise) and choose labels for generated images (labels_for_gen) from the sampled labels.
   - Use the generator to produce fake images (gen_imgs) based on the generated noise and labels.
   - Train the discriminator: Compute the loss for real and fake images (d_loss_real and d_loss_fake) separately.
   - Update the discriminator's weights by averaging the losses (d_loss).
4) Generator Training:
   - Generate a new batch of noise and random labels (sampled_labels) for the generator.
   - Train the combined model (generator + discriminator) by producing fake images with the generator.
   - The objective is for the generator to deceive the discriminator by training with a label of '1' (indicating real images).

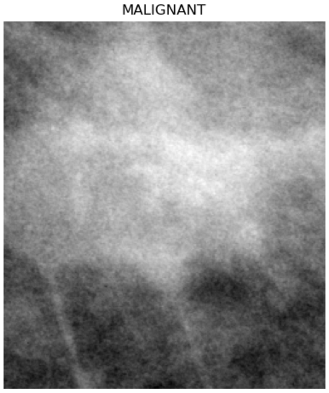

Fig. 4. Original Image from CBIS-DDSM Dataset

Apply Trained CGAN Model to synthesize new images: The function **generate_images** encapsulates this process, enabling the creation of synthetic images conditioned on random noise and categorical labels.

Inputs:

- n_rows and n_cols: Set the grid size for the generated images.
- generator_model: A trained model that generates images from noise and labels.

Process:

- Noise Generation: Random noise is created using a specific pattern. This noise matrix defines the basis for image creation.
- Label Assignment: Labels are randomly chosen to guide the generator in creating specific types of images.
- Image Generation: The generator model uses the noise and labels to produce new synthetic images.
- Image Adjustment: The created images are adjusted to fit within the visual range for better display.

Output: The function gives back a grid of new images. These images are unique and differ based on the provided noise and labels.

3. Results

The implementation of a Conditional Generative Adversarial Network (CGAN) for generating images within the CBIS-DDSM dataset posed several challenges and limitations. Despite efforts, the generated images did not closely resemble the original data within the CBIS-DDSM dataset.

Image Comparison: A generated image from the CBIS-DDSM dataset, shows the challenges faced in mimicking the original dataset's characteristics. This image, when compared to the original data, highlights the disparities and discrepancies between the generated and actual images, illustrating the limitations encountered when working with complex medical imaging datasets.

Challenges Encountered:

- Discrepancy from Real Images: The generated images from the CGAN did not closely match the characteristics of the original CBIS-DDSM images. This discrepancy might be due to the complexity of translating scanned films, mostly observed as grayscale images, into RGB images of size 224x224x3. This transformation might have led to information loss and affected the network's ability to learn and reproduce authentic images.
- Training Time: Training the CGAN model was notably time-consuming (epochs can be up to 30000 to reproduce the image above). The complexity and size of the model, coupled with the large image dimensions and limited dataset size, contributed to extensive training times.
- Data Size Limitation: X_train has shape is (1187, 224, 224, 3), meaning that during training, we only have more than a thousand samples. The size of the training dataset might not have been sufficient to capture the full diversity and complexity of the images, leading to limitations in the model's ability to generalize and create realistic images.

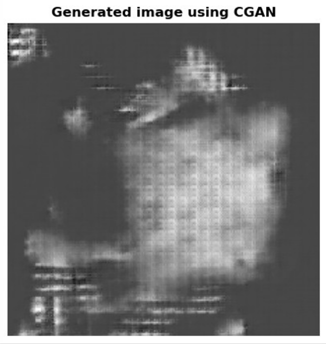

Fig. 5. The generated image from the CGAN model

Comparative Analysis: Additionally, experimenting with the MNIST dataset showed promising results with CGAN's synthesized images. The MNIST dataset, comprising simpler handwritten digits, might have been more amenable to the CGAN's learning process, resulting in more satisfactory and coherent generated images compared to the CBIS-DDSM dataset[17].

In summary, while CGANs demonstrate potential for image generation, the challenges encountered with the CBIS-DDSM dataset influenced the model's ability to generate accurate and representative images. Further refinement and exploration are necessary to overcome these obstacles and achieve better results with complex medical imaging datasets.

References

[1] Oza, Pratik and Sharma, Prashant and Patel, Samir and Adedoyin, Folashade and Bruno, Agostinho, *Image Augmentation Techniques for Mammogram Analysis*, *Journal of Imaging*, vol. 8, no. 5, pp. 141, May 20, 2022, https://www.mdpi.com/2313-433X/8/5/141/htm

[2] Ding, Kaiyue and Zhou, Ming and Wang, Hao and others, *A Large-scale Synthetic Pathological Dataset for Deep Learning-enabled Segmentation of Breast Cancer*, *Scientific Data*, vol. 10, p. 231, 2023, https://www.nature.com/articles/s41597-023-02125-y

[3] Cha, Kuo Han and Petrick, Nicholas and Pezeshk, Aria and Graff, Christian G. and Sharma, Deep and Badal, Andreu and Sahiner, Berkman, *Evaluation of Data Augmentation via Synthetic Images for Improved Breast Mass Detection on Mammograms Using Deep Learning*, *Journal of Medical Imaging (Bellingham)*, vol. 7, no. 1, p. 012703, 2020, https://doi.org/10.1117/1.JMI.7.1.012703, Epub 2019 Nov 22, PMID: 31763356, PMCID: PMC6872953

[4] Rebuffi, Sylvestre-Alvise and Gowal, Sven and Calian, Dan Andrei and Stimberg, Florian and Wiles, Olivia and Mann, Timothy A, *Data Augmentation Can Improve Robustness*, In *Advances in Neural Information Processing Systems*, pp. 29935–29948, Curran Associates, Inc.,2021.
https://proceedings.neurips.cc/paper_files/paper/2021/file/fb4c48608ce8825b558ccf07169a3421-Paper.pdf

[5] Author, A. (2013). *I.J. Image, Graphics and Signal Processing, 5*(6), 47-54. Published Online April 2013 in MECS (http://www.mecs-press.org/). DOI: 10.5815/ijigsp.2013.05.06.

[6] *Breast Cancer CNN Model*. By JOSHUA AMPOFO YENTUMI. https://www.kaggle.com/code/joshuaampofoyentumi/breast-cancer-cnn

[7] CBIS-DDSM: Breast Cancer Image Dataset. Kaggle. https://www.kaggle.com/datasets/awsaf49/cbis-ddsm-breast-cancer-image-dataset/code

[8] DenseNet169 Model. By Hithesh M R from Kaggle https://www.kaggle.com/code/hitheshmr/densenet169-cbis-ddsm

[9] Dive into Deep Learning - Chap 7: Convolutional Neural Networks by Aston Zhang (Author), Zachary C. Lipton (Author), Mu Li (Author), Alexander J. Smola (Author) https://d2l.ai/index.html

[10] Deep Learning by Ian Goodfellow and Yoshua Bengio and Aaron Courville MIT Press http://www.deeplearningbook.org 2016

[11] Optuna: A Next-generation Hyperparameter Optimization Framework By Akiba, Takuya and Sano, Shotaro and Yanase, Toshihiko and Ohta, Takeru and Koyama, Masanori. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* https://optuna.readthedocs.io/en/stable/

[12] Fine-Tuned DenseNet-169 for Breast Cancer Metastasis Prediction Using FastAI and 1-Cycle Policy. By Adarsh Vulli, Parvathaneni Naga Srinivasu, Madipally Sai Krishna Sashank, Jana Shafi, Jaeyoung Choi, and Muhammad Fazal Ijaz. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9025766/

[13] Very Deep Convolutional Networks for Large-Scale Image Recognition. By Karen Simonyan and Andrew Zisserman. 2015. https://arxiv.org/abs/1409.1556

[14] Understanding VGG16: Concepts, Architecture, and Performance. https://datagen.tech/guides/computer-vision/vgg16/

[15] Step-by-step VGG16 implementation in Keras for beginners. By Rohit Thakur. https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c

[16] What is a Conditional Generative Adversarial Network (cGAN)? https://datascientest.com/en/what-is-a-conditional-generative-adversarial-network-cgan

[17] Conditional GAN. Keras. https://keras.io/examples/generative/conditional_gan/

[18] How to Develop a Conditional GAN (cGAN) From Scratch. By PhD.Jason Brownlee. 2020. https://machinelearningmastery.com/how-to-develop-a-conditional-generative-adversarial-network-from-scratch/

[19] A guide to convolution arithmetic for deep learning. By Vincent Dumoulin and Francesco Visin. March 24, 2016. https://arxiv.org/pdf/1603.07285v1.pdf

[20] OpenAI's GPT (Generative Pretrained Transformer) Models. 2021. https://chat.openai.com/