

**Koita Center for Digital Health**

**Indian Institute of Technology, Bombay**

**Dual Degree Project 2 (DH 594)**

**Report**

**Data Efficient Machine Learning in Healthcare**

**Atharv Vinayak Savarkar (180010015)**

**Date: 16-06-2023**

**Guide: Prof. Ganesh Ramakrishnan (CSE)**

**Co-Guide: Prof. Kshitij Jadhav (KCDH)**

**Approval Sheet**

This thesis/dissertation/report entitled “Data Efficient Machine Learning in Healthcare” by Atharv Vinayak Savarkar is approved for the degree of B.Tech in Aerospace and Interdisciplinary Dual Degree at Koita Center for Digital Health

Examiners

Prof. Ranjith Padinhateeri (BSBE)

Supervisors

Prof. Ganesh Ramakrishnan (CSE)

Prof. Kshitij Jadhav (KCDH)

Date: 29-07-2023

Place: IIT Bombay

**Declaration**

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed

Signature

Roll no: 180010015

Date: 29-07-2023

**INDEX**

**Chapter 1 - DOGMAS**

1. Abstract
2. Introduction
3. Our Contribution
4. Related Work
5. Preliminaries
   1. Submodular Function
   2. wGANs
   3. t-SNE Plots
6. Our Approach
7. Experiment
   1. Experimental Setup
   2. Datasets Used
8. Results and Discussion
9. Conclusion

**Chapter 2 - Feature Extraction on PET-CT scans**

1. Abstract
2. Introduction
3. Treatment Procedures for HL
   1. ABVD
   2. BEACOPP
4. Preliminaries
   1. PET Scan
   2. Standardized Uptake Value (SUV)
5. Our approach
   1. Shortlisting PET scans for computation
   2. Contouring high SUV uptake regions
   3. Linking PET and CT scans
   4. CT Classification using ResNet18
   5. Nodal location detection using YOLO
6. Future Work

**Chapter 1 : DOGMAS - Diversity induced subset selection based Generation for medical imaging in data constrained setting**

**Abstract**

Deep Learning models have revolutionized the field of radiology and diagnostic medicine, propelling it to new heights by leveraging their exceptional capabilities in identifying subtle pattern differences among various pathology images. These advanced models employ a complex network architecture that allows them to process vast amounts of medical imaging data, extracting intricate features and discerning minute variations that may elude even the keenest human eye. However, the fundamental limitation of Deep Models is the requirement of large amounts of labeled data causing an impediment to their implementation in low resource settings. In this paper, DOGMAS (Diversity induced subset selectiOn based Generation for Medical imaging in datA constrained Setting) we demonstrate that intelligent subset selection based on diversity functions from a huge pool of unlabeled medical data of two pathologies from four open source datasets, followed by human labeling and then Wasserstein GANs (wGANs) based data augmentation performed comparable to large labeled data-based prediction. An important similar methodology after random subset selection did not yield good results. This demonstrates that despite having less amount of labeled data, one can reach the required quantum of large labeled data by data augmentation using wGANs, albeit if a small subset of images selected for human labeled is done using diversity metrics

**Introduction**

Deep neural networks (DNN) have revolutionized numerous fields, including image and video recognition, natural language processing, recommendation systems, finance, gaming, robotics, climate change, and transportation, among others [6]. DNN offers several advantages over traditional machine learning algorithms, primarily due to their high accuracy in complex tasks like image or speech recognition. This is because deep neural networks can learn and represent intricate non-linear relationships between input variables and outcomes. Additionally, they enable automated feature extraction, reducing the need for manual feature engineering and making predictive analysis more efficient. Deep neural networks are also highly scalable, capable of working in high-dimensional feature spaces, and offer the benefit of transfer learning, allowing them to be trained for one task and then retrained for another [23]. Today, the healthcare industry generates vast amounts of data, particularly in the realm of medical imaging. This is partly due to rapid advancements in data acquisition techniques like MRI, CT scans, and whole slide images from pathology sections. Consequently, the medical domain is now considered the next frontier for artificial intelligence and machine learning, with deep neural networks poised to play a significant role [12]. Despite the impressive progress made by deep neural networks, several disadvantages have hindered their widespread adoption. These networks demand high computational power, particularly during training, making them slow and costly when dealing with large models [27]. Additionally, deep neural networks require vast amounts of data for accurate training, necessitating equal representation across all datasets [26]. While deep models have demonstrated huge improvements in several downstream image, video and text tasks, they are associated with several challenges such as higher training complexity, long inference times, difficulty in optimizing hyperparameters. Several previous works have attempted to address the problem by changing the network architecture [10] [11], transfer learning [19], one-shot [31] and zero shot learning [25], core set selection [1] as well as active learning [14]. A critical issue associated with deep neural networks is the need for large quantities of labeled data during training. Data labeling, typically performed by individuals, generates the so-called "ground truth," which is essential for supervised machine learning problems and closely linked to predictive accuracy [21]. In essence, through labeling, we attempt to embed human knowledge and expertise into a machine-learning model, and high quality labels are vital for driving effective predictive algorithms. In the medical domain, this challenge is magnified, as labeling is performed by medical experts, often requiring a consensus from at least three experts. This results in a high cost of labeling, making DNN adoption prohibitively expensive, especially for low- and middle-income countries and low-resource settings [33]. Even in LMICs, rapid digitalization of imaging modalities in the medical domain creates a huge pool of unlabeled data, a subset of which can then be selected for manual labeling from human medical experts to reduce the cost of human labeling [32]. One solution could be labeling a few images with a human expert and then performing data augmentation using Generative Adversarial Networks (GANs) which aim to replicate the underlying data distribution of the few labeled images and generate new images where a generative neural network pits against a discriminator neural network. The generator and discriminator in traditional GANs are trained in a minmax game, the objective function sometimes not being able to provide a meaningful gradient signal [8]. However, traditional GANs hinder training stability due difficulty in, first, the estimation, and then, optimization of the distance between the original real and the fake generated distributions. This potentially results in the generator producing a limited number of samples which do not capture the entire spectrum of diverse original data [22]. To overcome these limitations, wGANs can be utilized where the distance metric is the ‘Earth Mover’s distance’ which is a better representation of data distribution due to the introduction of the Lipschitz constraint on the discriminator network [7]. wGANs were introduced as a variant of the classical GANs network by Martin Arjovsky, Soumith Chintala, and Léon Bottou in 2017 [5] to address some of the limitations of traditional GANs, such as mode collapse and instability during training. Importantly, wGANs provide an objective function that delivers a smoother and more informative gradient signal during training in contrast to traditional GANs, where the discriminator’s binary output does not provide a meaningful gradient [5]. The fundamental requirement to increase the utility of wGANs based generated images in the aforementioned scenario is to identify a subset of diverse images from the huge pool of unlabeled data. Thus, the process of making an intelligent subset selection is of primary importance. Randomly selecting images to label from a huge pool of unlabeled data can create an imbalance in the number of images selected from each class and then the augmented images with wGANs then might suffer from the biases introduced due to random subset selection

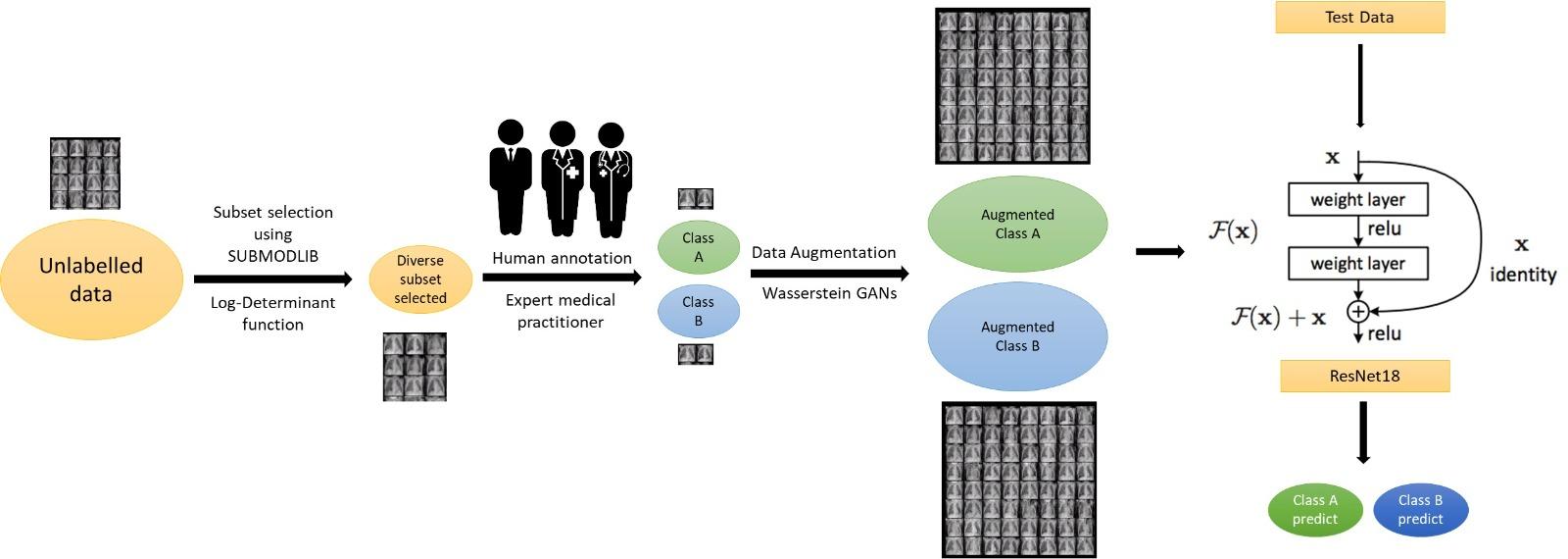


Figure 1: Pictorial presentation of intelligent subset selection driven approach followed by GANs data augmentation

**Our Contributions**

We demonstrate that diversity driven subset selection can form an optimal seed for data augmentation that can then be used to train Deep models (Figure 1 and Section 5). For selection of the smaller subset, we used and compared various submodular functions (Table 2). Submodular functions can model various properties like diversity, representation, importance and coverage etc. The Log-determinant, Disparity sum, Disparity min and Graph cut are different submodular functions that capture diversity. In our approach Log-determinant submodular function potentially provides a higher degree of diversity than rest of the functions since data augmentation post these diversity functions is peaked by Log-determinant driven subset selection (Table 2). Using the Log-determinant function [15], we identify a small subset of the most diverse data points, which were then labeled by human annotation. These were then used to generate increasing number of images using a generative model such as wGAN (c.f. Section 4.2 for the appropriateness of wGAN in this setting). It is observed that, as higher number of images generated using wGANs post diversity based subset selection were used to train RESNET18, the accuracy to predict the labels from the test dataset increased, gradually reaching close to the accuracy of the skyline setting (The skyline being, if we had trained RESNET18 with a high number of labeled images) (Figure 6). Importantly, random subset selection from the large pool of unlabelled data followed by increasing amount of data augmentation by wGANs and then training RESNET18, decreased the predicted accuracy on the test set (c.f. Figure 6). Further, we demonstrate that, data augmentation using wGANs post subset selection shows data distribution similar to original data distribution which is in stark contrast to data augmentation post random subset selection (c.f. Figures 4 and 5). We hypothesize that this discordant data distribution could be the reason for ineffectiveness of data augmentation post random subset selection. This indicates that random subset selection is not ideal to select images for annotation and then follow it up with data augmentation. Thus we conclude that this subset selection should be done in a programmatic intelligent way to identify the most diverse images from a large set of unlabelled images. These can then be used for data augmentation using wGANs or other similar augmentation techniques to improve the predictive ability of deep models.

**Related Work**

GANs, short for Generative Adversarial Networks, are a class of deep learning models that consist of two neural networks, known as the generator and the discriminator, which are trained in a competitive manner. The generator network is responsible for creating synthetic data, such as images, based on random noise input. It learns to generate data that resembles real examples from a given dataset. The goal of the generator is to produce high-quality samples that are indistinguishable from real data. On the other hand, the discriminator network acts as a binary classifier, trained to differentiate between real data and synthetic/generated data. It learns to classify whether a given sample is real or fake. The discriminator's objective is to correctly identify the source of the input data as either real or generated.

The training process of GANs involves a competitive interplay between the generator and the discriminator. The generator tries to improve its ability to generate realistic data by fooling the discriminator, while the discriminator strives to become more adept at distinguishing real data from generated data. This adversarial nature of training leads to both networks continuously improving their performance.

During training, the generator and discriminator are updated iteratively. The generator takes random noise as input and generates synthetic samples, which are then fed to the discriminator along with real samples from the dataset. The discriminator evaluates and provides feedback on the authenticity of the samples. The generator's parameters are updated based on the discriminator's feedback, aiming to generate more realistic samples that can deceive the discriminator. Conversely, the discriminator's parameters are updated to improve its ability to correctly classify the samples.

As the training progresses, the generator learns to produce samples that increasingly resemble real data, while the discriminator becomes more skilled at distinguishing between real and generated data. Ideally, this competition drives both networks towards convergence, resulting in a generator that can produce high-quality synthetic data that is difficult to differentiate from real data. GANs have demonstrated remarkable success in various applications, such as image synthesis, style transfer, image-to-image translation, and data augmentation. They have also been extended to conditional GANs, where the generator is conditioned on additional input information, enabling control over the generated output.

However, training GANs can be challenging and unstable, often requiring careful tuning of hyperparameters and architectural choices. Issues like mode collapse (when the generator fails to explore the full range of possible outputs) and training instability can arise. Researchers have proposed numerous techniques to mitigate these challenges, such as using different loss functions, regularization methods, and architectural modifications.

Despite the challenges, GANs have significantly advanced the field of generative modeling, enabling the generation of realistic and diverse synthetic data. They continue to be an active area of research and hold immense potential for applications in various domains, including computer vision, natural language processing, and generative art.

Uzunova et al [29], discuss how generative models are essential for generating realistic and diverse medical images, which can be utilized in various applications such as data augmentation, and training deep learning models, as well as assist in medical research. Importantly, they touch upon specific limitations of generative models such as the scarcity of labeled data, class imbalance, the former is being specifically tackled in our research. Similarly, Skandarani et al [24] explores the effectiveness and limitations of GANs for synthesizing medical images. The paper provides insights into the limitations and potential biases associated with GAN-based medical image synthesis. It emphasizes the importance of careful evaluation and validation of the generated images to ensure their reliability and usefulness in real-world medical applications. In our paper we have addressed this by training a ResNet18 model and then testing the predictive accuracy on an empirical test set. Mukherjee et al [20] introduces a novel approach for generating realistic brain tumor images through the combination of GAN models and style transfer techniques. The generation of synthetic brain tumor images is crucial for various medical applications, such as training deep learning algorithms and augmenting limited datasets. They utilized multiple GAN models trained on different subsets of images and then aggregated them to increase the diversity and quality of images. They demonstrated that the aggregated GAN models with style transfer achieve superior performance in terms of image quality, diversity, and realism compared to state of the art models. However, Mukherjee et al [20] did not utilize optimal subset selection techniques to then apply GANs for data augmentation. Ahmad et al [2] have introduced the novel Generative Adversarial Network (GAN) specifically designed for super-resolution of medical images. Super-resolution refers to the process of generating high-resolution images from low-resolution counterparts, a task critical in medical imaging for enhancing image details and improving diagnostic accuracy.A novel perceptual loss function is introduced, which encourages the generator to produce visually similar images to the ground truth high-resolution images. Additionally, an attention mechanism is incorporated into the generator network to enhance the reconstruction of fine details in medical images. However, there is no method incorporated to determine which low resolution images to select that would give the best outcome. Further, the assumption holds that the experimenters have access to large amounts of labeled data albeit low resolution images.

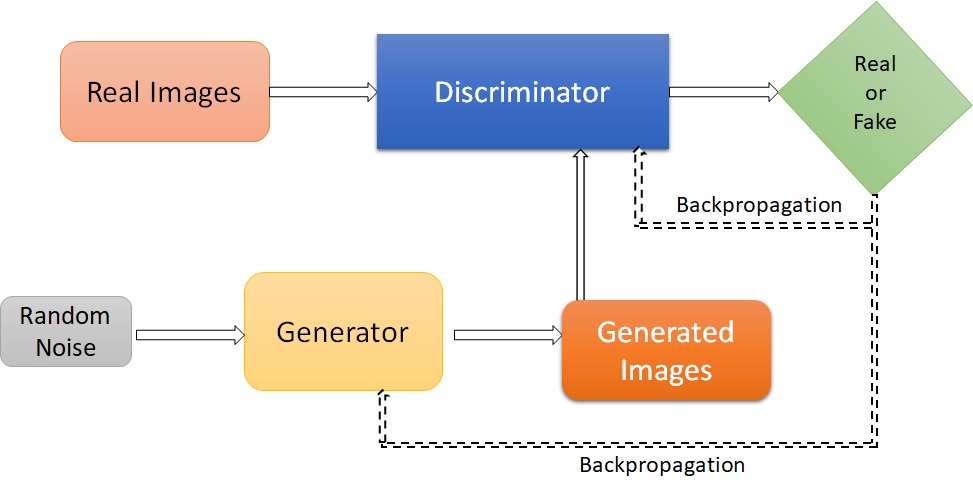


Figure 2: Representation of the Generative adversarial network architecture

**Preliminaries**

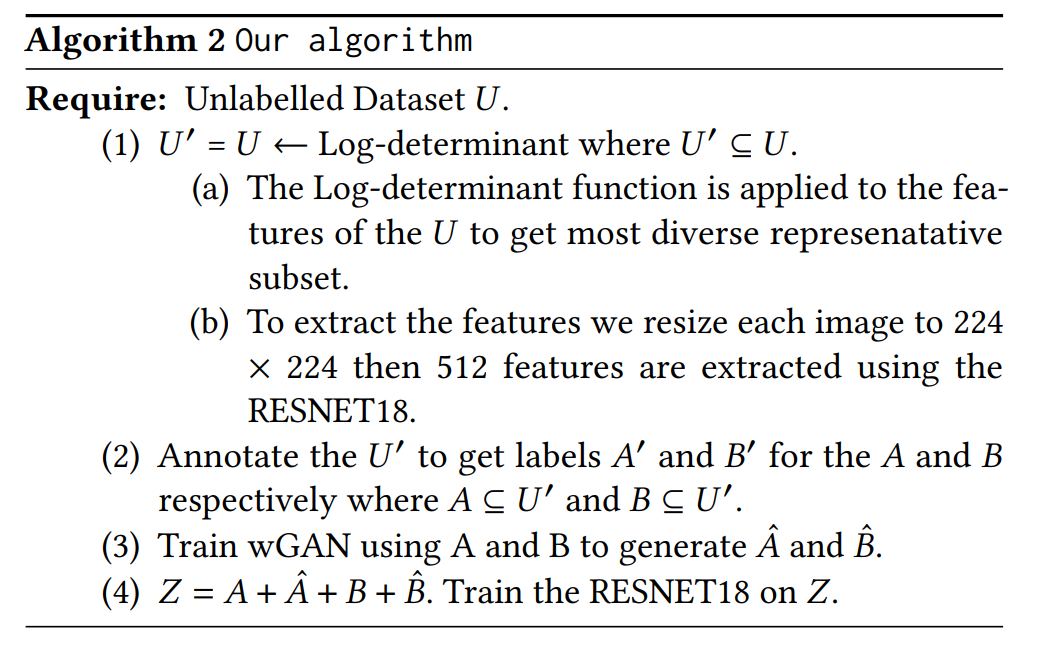
***Submodular functions***

Submodular functions are a mathematical concept that describes functions with diminishing returns or decreasing marginal gains. These functions are defined over sets and satisfy a property called submodularity, which states that the marginal gain of adding an element to a smaller set is greater than or equal to the marginal gain of adding the same element to a larger set. This property captures the idea that the value obtained from adding an item to a set diminishes as the set grows larger.

Submodular functions have diverse applications in optimization, set cover problems, and machine learning. They are used to model problems where there is a limited benefit to adding additional elements to a set once certain elements have already been included. This property allows for efficient optimization algorithms and enables the use of greedy algorithms, which iteratively select elements that provide the maximum marginal gain.

In the field of machine learning, submodular functions have proven useful for tasks such as feature selection, data summarization, active learning, and recommendation systems. They capture notions of diversity, coverage, and information gain, making them valuable for solving problems that involve selecting a representative subset or making efficient choices among a large set of options. Overall, submodular functions provide a mathematical framework for understanding and solving problems with diminishing returns. Their applications span various domains, offering efficient optimization techniques and enabling intelligent decision-making in complex settings.

To select a subset from a dataset we used submodular functions, each of which has different properties. Examples of submodular functions include facility location, graph cut, log-determinants, etc. These submodular functions are part of the submodlib library. (<https://github.com/decile-team/submodlib>) [18]. After having tested several diversity based submodular functions we zeroed on the Log-determinant function to select the diverse set of images from the given dataset given its higher efficacy in our algorithm. The assumption is that diversity based subset selection will be able to capture all the selective but distinct subsets from the data.



Log-Determinant Algorithm

***WGANs***

The Wasserstein GAN (WGAN) is a variant of the popular Generative Adversarial Network (GAN) framework that addresses some of the limitations of traditional GANs. The main idea behind WGAN is to use the Wasserstein distance, also known as Earth Mover's distance, as a measure of the difference between the generated and real data distributions. The authors argue that the Wasserstein distance provides a more meaningful and stable metric for training GANs.

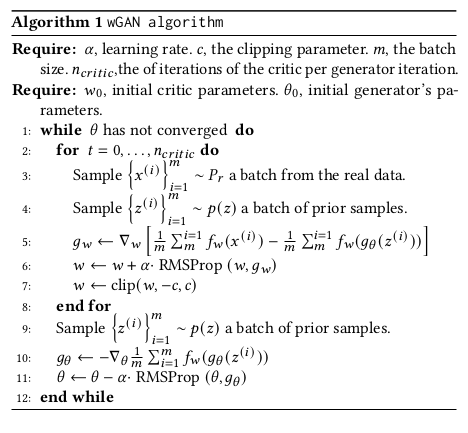
In traditional GANs, the discriminator objective is to classify the generated and real samples accurately. However, this can lead to training issues, such as mode collapse, vanishing gradients, and instability. WGAN tackles these problems by introducing a novel approach for training the discriminator and generator networks.

Instead of training the discriminator to output probabilities, WGAN encourages the discriminator to learn a function that approximates the Wasserstein distance between the generated and real data distributions. This is achieved by constraining the Lipschitz continuity of the discriminator function. The authors propose using weight clipping to enforce this constraint, although subsequent research has suggested alternative methods like gradient penalty.

By optimizing the Wasserstein distance, WGAN offers several advantages over traditional GANs. It provides a more meaningful measure of the discrepancy between the real and generated distributions, which correlates better with the visual quality of the generated samples. WGAN also tends to produce more stable training dynamics, mitigating issues like mode collapse and vanishing gradients.

The WGAN framework also has implications for the generator's training process. In WGAN, the generator is trained to minimize the negative Wasserstein distance, effectively encouraging it to generate samples that can gradually move the generated distribution closer to the real data distribution. This formulation provides better gradients for the generator to learn from and results in improved convergence properties. There are experimental results on various datasets, including synthetic data and real-world images, demonstrating the effectiveness of WGAN in generating high-quality samples and avoiding mode collapse. Since the publication of the original WGAN paper, numerous advancements and variations have been proposed to enhance the stability and performance of WGAN, such as gradient penalty, spectral normalization, and different ways of enforcing Lipschitz continuity. These modifications have further improved the training dynamics and quality of generated samples in WGAN-based approaches.

In summary, the Wasserstein GAN (WGAN) introduces a new approach to training GANs by using the Wasserstein distance as a measure of the difference between the generated and real data distributions. This formulation addresses several challenges associated with traditional GANs, leading to more stable training dynamics and higher-quality generated samples.



wGAN Algorithm

**t-SNE Plots**

t-SNE (t-Distributed Stochastic Neighbor Embedding) is a popular dimensionality reduction and visualization technique that is commonly used to transform high-dimensional data into a lower-dimensional space while preserving the inherent structure and relationships among data points. It is particularly useful for visualizing complex datasets with many variables or features.

The t-SNE transformation begins by calculating pairwise similarities or distances between all data points in the high-dimensional space. The similarities are typically computed using a Gaussian kernel or other similarity measures such as cosine similarity. These pairwise similarities capture the local relationships between data points, emphasizing nearby neighbors.

Next, t-SNE constructs a probability distribution that represents the pairwise similarities between points in the high-dimensional space. This distribution is called the similarity distribution or the joint probability distribution. The similarity distribution is defined as a symmetrical version of the pairwise similarities, where the similarities are normalized and transformed into probabilities. The normalization ensures that the sum of probabilities for each data point is equal to one.

In the lower-dimensional space, t-SNE constructs a similar probability distribution, known as the Student's t-distribution, that represents pairwise similarities between points. The goal of t-SNE is to minimize the divergence between the high-dimensional similarity distribution and the low-dimensional similarity distribution. It does this by iteratively optimizing the positions of the points in the low-dimensional space to find a configuration that best matches the similarities in both spaces.

During the optimization process, t-SNE computes the Kullback-Leibler (KL) divergence between the two distributions and adjusts the positions of the points to minimize this divergence. The optimization is performed using gradient descent, where the gradient of the KL divergence with respect to the positions of the points is computed. The points are iteratively moved in the low-dimensional space to find a configuration that minimizes the KL divergence.

Once the optimization is complete, t-SNE produces a low-dimensional representation of the data points. This lower-dimensional representation can be visualized using scatter plots or other visualization techniques. In the t-SNE plot, each point represents a data point, and the distances between the points in the plot reflect the similarities or dissimilarities between the original high-dimensional data points.

The t-SNE plot provides a visual summary of the relationships and structure in the data. It helps identify clusters or groups of similar data points, as they tend to be positioned close to each other in the t-SNE plot. Additionally, t-SNE can reveal global patterns and local structures in the data, making it useful for exploring and interpreting complex datasets.

It is important to note that t-SNE is a stochastic algorithm, meaning that each run of t-SNE may produce slightly different results. The outcome can depend on various factors, such as the initialization of the points and the hyperparameters chosen, including the perplexity (a parameter that controls the balance between local and global structure in the plot).

In summary, t-SNE is a dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space while preserving local and global relationships among data points. It computes pairwise similarities, constructs probability distributions in both high- and low-dimensional spaces, and optimizes the positions of points to minimize the divergence between the distributions. The resulting t-SNE plot provides a visual representation of the data, aiding in the identification of clusters and patterns in complex datasets.

**Our Approach**

This section proposes our approach and demonstrates how intelligent subset selection was applied to the large pool of unlabelled medical images. We begin with the assumption of the entire dataset being unlabelled. We applied RESNET18 to each image to obtain 512 features and then the Log-determinant submodular function was utilized for identification of the most diverse images from the large unlabelled dataset. We select only 5-10% images of the entire set of unlabelled images. These were then labeled by a domain expert. These newly labeled images were then used to train a wGANs model to generate images in increasing numbers. The originally labeled subset of images and the wGANs generated images were then used to train RESNET18 deep neural network [9]. The details are shown in Table 1. For each dataset we create 5 sets of images with the increasing numbers of generated images. We train the RESNET18 model on these five different sets and evaluate the accuracy on an independent test dataset. The outline of our approach is shown in Figure 1. Also the detailed procedure of our approach is shown in Algorithm 2. In our experiment we used 1 python library that is open source and available on github. [15].

**Experiment**

This section discusses the different models used along with their parameters. We tested our algorithm’s effectiveness on four different open access medical datasets, namely chest X-ray (pneumonia MNIST) [16], DermaMNIST [28], BreastMNIST [3] and APTOS datasets [13]. We compare the accuracy of generated images using GANs with submodular Log-determinant images with randomly selected images. To check the accuracy, we train the RESNET18 using generated images and assess its performance on the test dataset. Our experiments demonstrate that data generated using submodular images outperforms data generated on randomly selected images. The details of the experimental setup, deep learning models, hyperparameter and dataset used is given in section experimental setup. We compare the performance of subset images selected using different diversity submodular functions. We use t-SNE plot to visualize the representation of randomly selected images and images selected using submodular functions along with generated images of set 2.

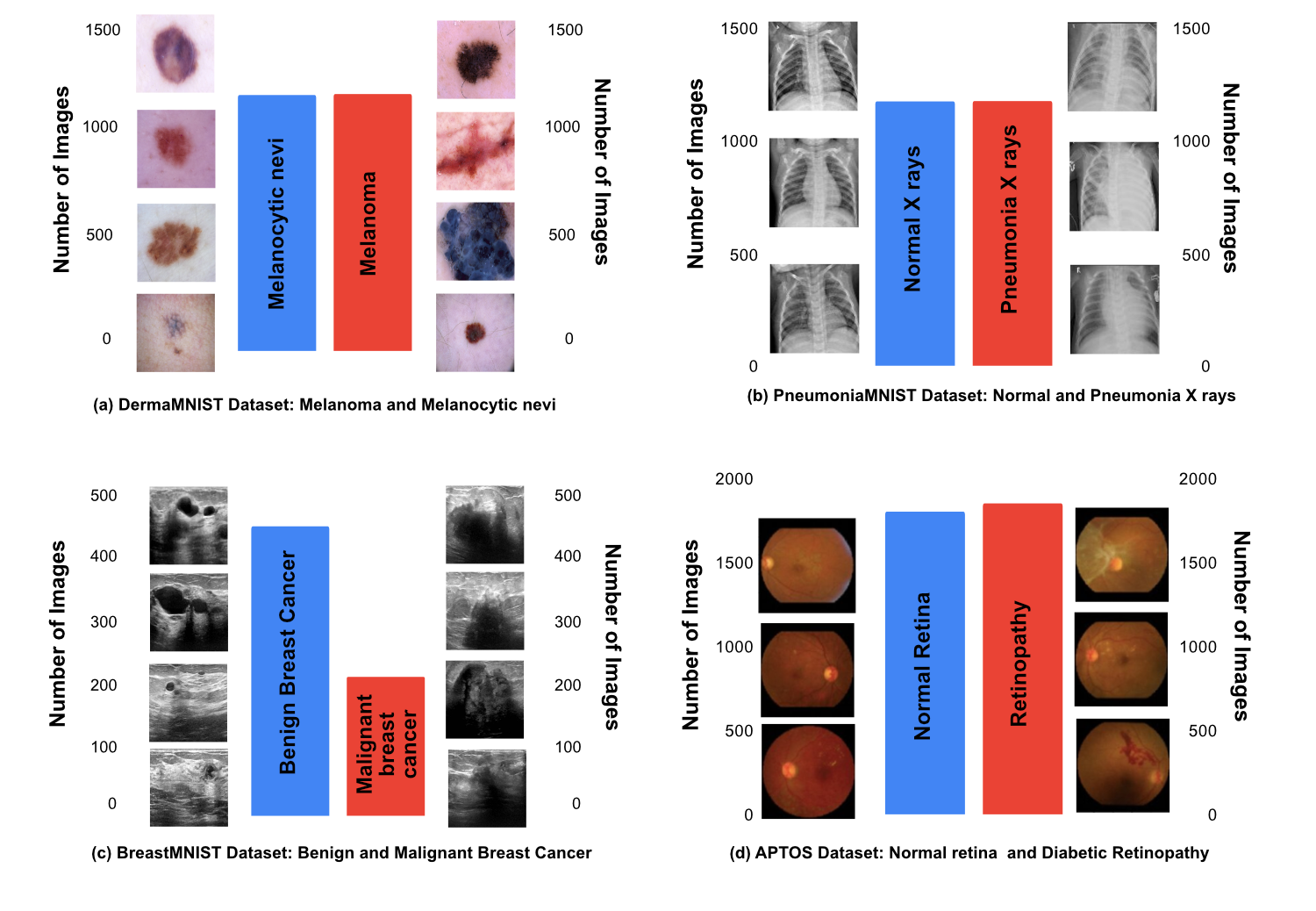


Figure 3: Pictorial presentation of intelligent subset selection driven approach followed by wGANs data augmentation

**Experimental Setup**

For all the experiments, we selected the same training procedure and hyperparameters. For feature extraction, we used the pretrained RESNET18 model [9]. We kept the size of the feature extracted to 512. For subset selection, we used the Log-determinant submodular function. The size of the subset was approximately 5-10% of the original dataset. We trained a RESNET18 from scratch on generated images combined with subset selected images, and a test dataset was used to assess the predictive accuracy. We used a Stochastic gradient descent optimizer, and the learning rate was set to 0.01. The batch size was set to 128, and RESNET18 was trained for 50 epochs. For wGANs, the network was trained for 20000 epochs, and the last epoch was used to generate images. The learning rate of generator and discriminator was set to 0.00005. We generated five sets of images with the same number of images per class. The number of images generated is detailed in Table 1. Also, We plot t-SNE plot for randomly selected and Log-determinant selected images along with images generated (Set 2) by wGAN for pneumoniaMNIST dataset. The t-SNE plots are generated from the features extracted using RESNET18.

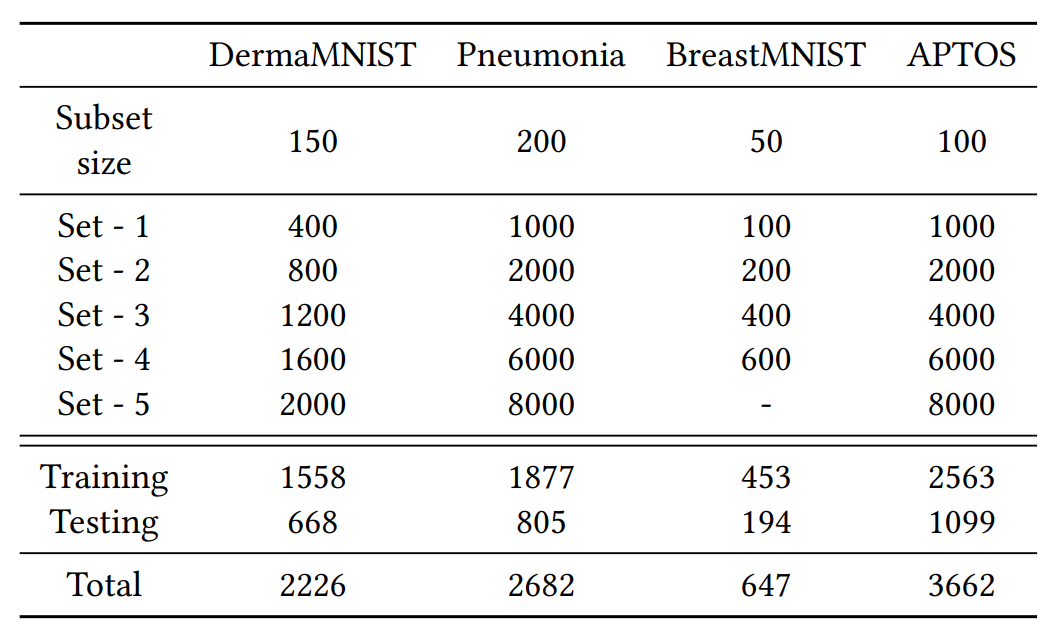


Table 1: Details of the datasets used and images generated. Same number of images are generated by random as well as Log-determinant driven subset selection

**Datasets Used :**

We used four different datasets for our experiment namely PneumoniaMNIST [17] , BreastMNIST [3], DermaMNIST [28] and APTOS [13]. These dataset contains different modalities of images such as skin lesions images, chest X-rays, breast ultrasound images and retina fundus photography. All the datasets are converted into binary classification for purposes of our experiment. Each dataset is then divided into 70-30% split i.e. training dataset and testing dataset. The details of the dataset are given in Table 1. Figure 3 shows some sample images from each dataset.

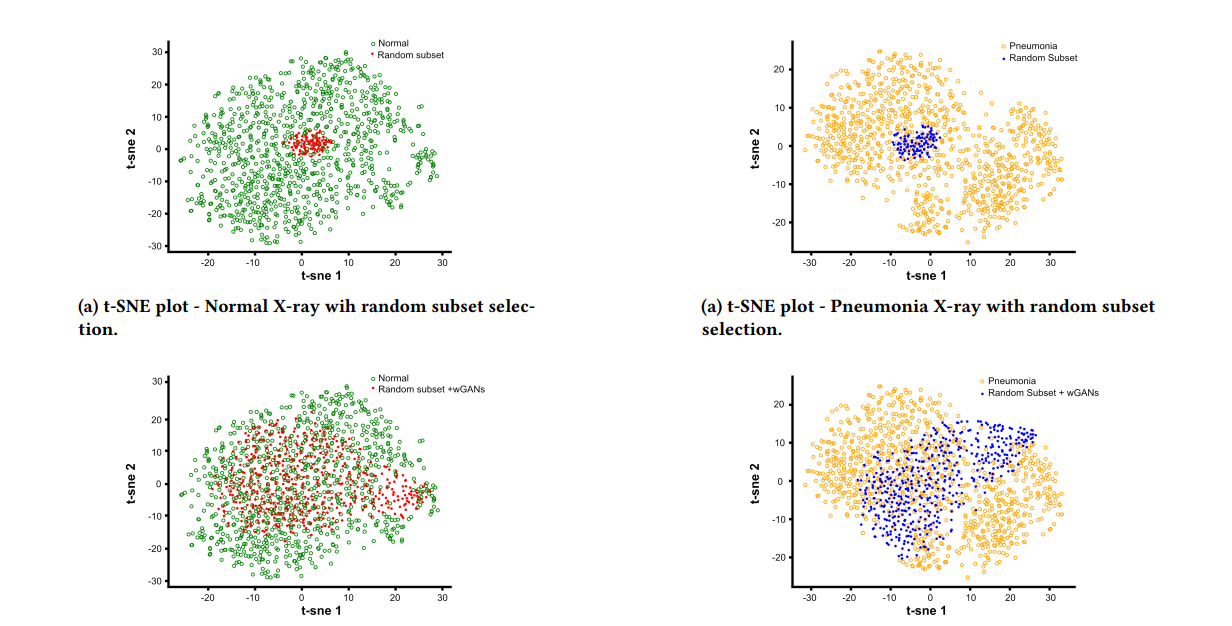
**DermaMNIST** - Dataset is a publicly available benchmark dataset commonly used in the field of dermatology and computer vision. It serves as a simplified version of the original HAM10000 dataset, which consists of clinical images of skin lesions captured with a dermatoscope. Dataset comprises a collection of 70,000 28x28 grayscale images, where each image corresponds to a skin lesion. The dataset is divided into two parts: 60,000 training images and 10,000 test images. Each image is labeled with one of ten different classes, representing different types of skin lesions, including melanoma, nevus, seborrheic keratosis, and others. The class distribution is relatively balanced, ensuring a fair representation of each category.

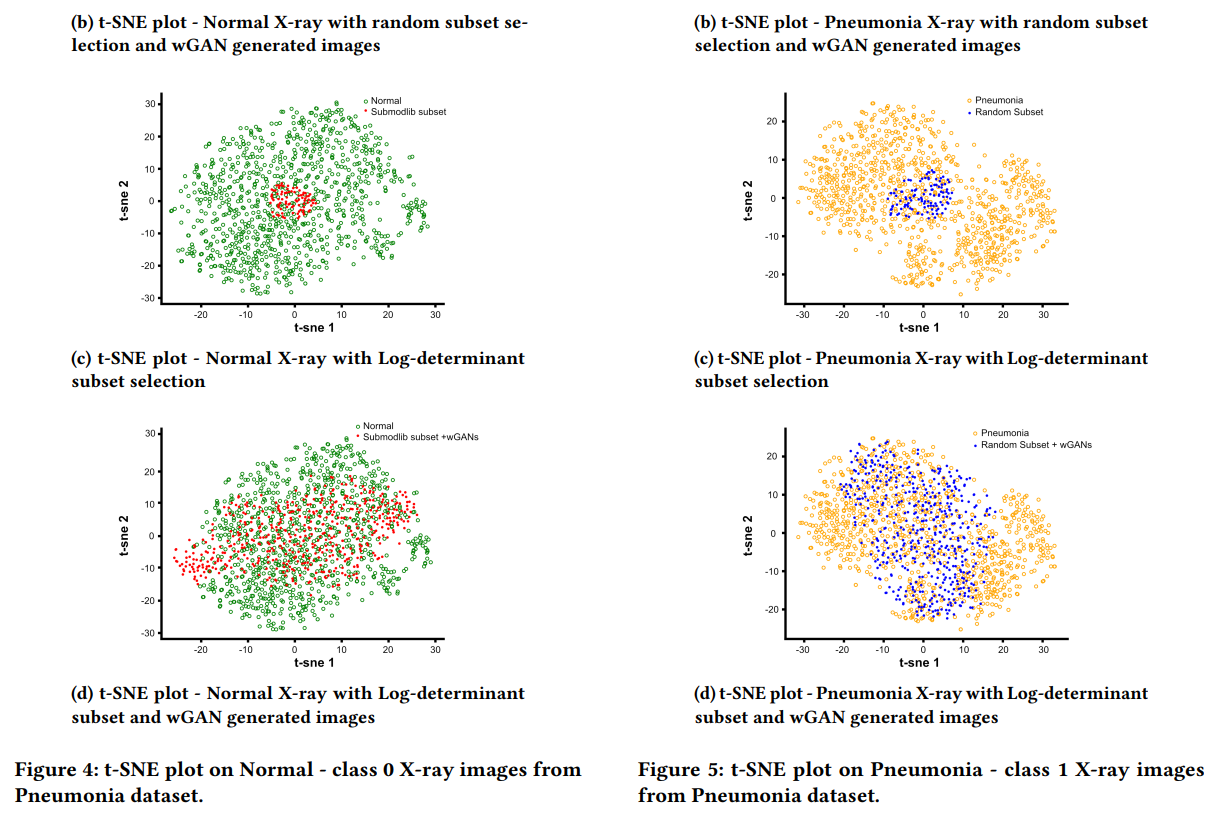
We used images from class melanoma (1113) and class melanocytic nevus (1113).

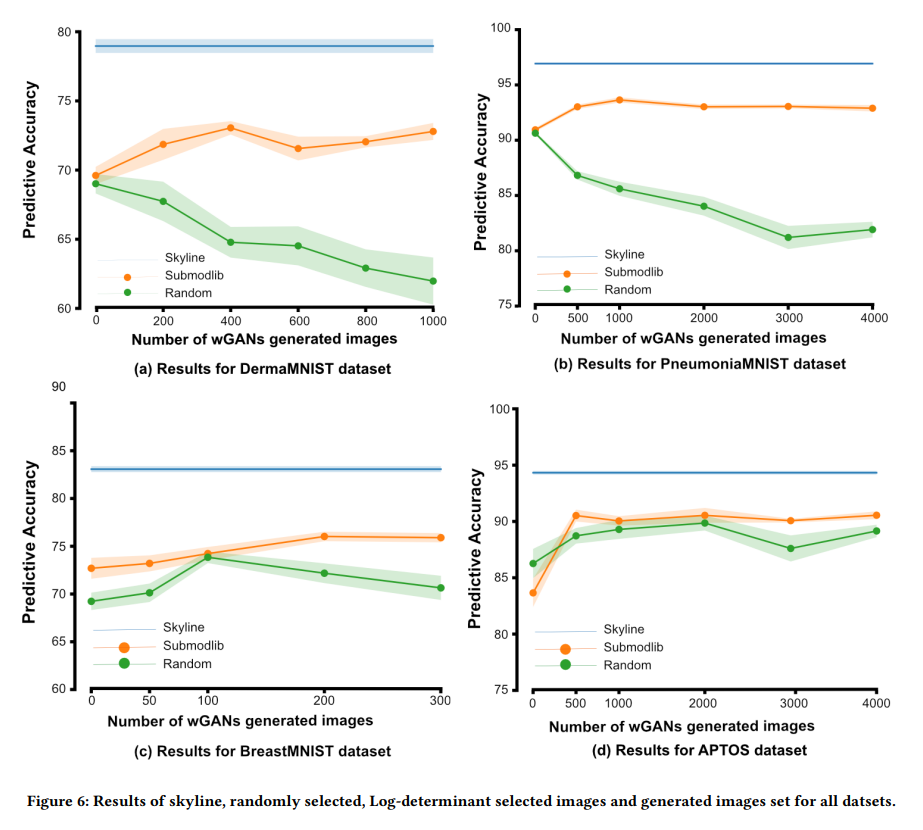
**PneumoniaMNIST** - Dataset consists of chest X-ray images collected from various sources, including public repositories and hospital databases. It contains both normal and pneumonia-affected images, making it suitable for binary classification tasks. The dataset is often used to train machine learning models to classify X-ray images as either showing signs of pneumonia or being normal. For our experiment we used 1341 normal X-ray while 1341 pneumonia X-ray images

**BreastMNIST** - This dataset contains the 780 images of breast ultrasound categorized into normal, benign and malignant class. The average image size was 500 × 500 pixels. We used images from benign class (437) and malignant class (210) as complete unlabelled dataset.

**APTOS** - We used a benchmark dataset of APTOS 2019 blindness detection challenge from Kaggle platform. [13]. This dataset consists of retinal images taken using fundus photography. The images are divided into five classes namely normal - class 0 (1805), mild - class 1 (370), moderate - class 2 (999), severe - class 3 (193); and proliferative - class 4 (295). We combined images from class 1, 2, 3 and 4 to create a single class - 1 (1857) and class 0 for our experiment.







**Results and Discussion**

Our experimental setup was to determine if intelligent subset selection for labeling followed by data augmentation performs better than random subset selection for labeling medical images. On the PneumoniaMNIST dataset we demonstrate that Log-determinant based subset selection performs better than other subset selection methods namely DisparitySum, DisparityMin and GraphCut, all of which beat random subset selection driven data augmentation as mentioned in Table 2. Hence, for all subsequent experiments with other datasets we applied the Log-determinant subset selection method. We hypothesized that this Log-determinant based subset selection was able to select a diverse set of images compared to random subset selection. This could be the driving force behind better representation of the entire dataset post wGANs based data augmentation.

We assessed the aforementioned observation by plotting the t-SNE plots as demonstrated in Figures 4 and 5. Figure 4(a) and 5(a) as well as Figure4(c) and 5(c) all demonstrate that the subset selection driven by Log-determinant function identifies in an unsupervised way diverse images compared to random selection. Random subset selection followed by wGANs data augmentation does not represent the entire spectrum of data diversity of the original distribution as demonstrated in 4(b) and 5(b). However, Log-determinant based subset selection followed by wGANs data augmentation demonstrated a wider and more overlapping representation of the original distribution as becomes evident in Figure 4(d) and 5(d). This better representation of the original data distribution is potentially the driver of our subsequent results demonstrating the efficacy of Log-determinant based subset selection.

We have performed another experiment which demonstrates the power of Log-determinant subset selection algorithm over random selection. Table 3 shows the number of random selections required to reach comparable accuracy to diverse subset selection along with data augmentation using wGANs. We can infer that approximately twice the number of images are required to get similar accuracy scores. This potentially has a huge impact in the medical domain as data labeling is much more involved and commercially costly.

It is expected that having a larger labeled dataset performs better in predictive accuracy than smaller datasets due to the diversity of the images present. This is apparent in Figure 6(a) and 6(d) where it is represented as skyline predictive accuracy. Thus, we obtain an accuracy of 78.97%, 96.93%, 83.07% and 94.33% on the skyline for DermaMNIST, PneumoniaMNIST, BreastMNIST and APTOS datasets respectively. Further, driven by the better representation of the original distribution by the Log-determinant based subset selection followed by wGANs data augmentation as seen in Figures 4 and 5, we demonstrate that this outperforms the random subset selection driven data augmentation in predictive accuracy. As the number of generated images using wGANs post Log-determinant submodular function increases the predictive accuracy increases. With the DermaMNIST dataset we obtain 69.61% accuracy after only Log-determinant based subset selected images are used to train RESNET18, while 72.79% accuracy on 1000/class generated images combined with subset images, thereby obtaining an improvement of 3%. With the PneumoniaMNIST dataset we achieve 90.94% predictive accuracy for Log-determinant based subset selected images training RESNET18 and 92.90% accuracy for 4000/class generated image, thus an improvement of 2% accuracy. Similarly for BreastMNIST dataset we observe 72.69% and 75.90% accuracies and, in the APTOS dataset 83.66% and 90.55% accuracies respectively by training RESNET using only subset selected images and then highest generated wGANs augmented images. Thus, it is evident that for Log-determinant based subset selection followed by data augmentation we observe an improvement in accuracy as we increase the number of images generated. This is potentially driven by better representation of the original dataset by wGANs generated images in this setting as observed in Figure 4 and 5.

However, random subset selected images followed by data augmentation and training RESNET18 to test the predictive accuracy on the Test dataset, exhibit a decrease in the accuracy as we increase the number of generated images. In the DermaMNIST dataset we observe 69.01% accuracy on randomly selected images and 61.98% accuracy on wGANs generated 1000/class images. In PneumoniaMNIST dataset accuracy decreases from 90.64% to 81.92% for subset dataset to 4000/class wGANs generated images. In the BreastMNIST dataset we observe 69.23% accuracy on random subset and 70.64% accuracy on wGANs generated images . Finally, in the APTOS dataset we achieve accuracy of 86.27% on randomly selected dataset and 89.16% accuracy on generated images. Thus, we either observe a fall or a maintenance of predictive accuracy in randomly selected images combined with wGANs augmented images after RESNET18 training indicating the non-utility of random subset selection for labeling unlabelled data

**Conclusion**

Medical domain is accelerating at a breathtaking speed especially in the imaging field owing to advances made in the other technology domains of image acquisition and image processing. However, this has now created scenarios where large reservoirs of unannotated data are being created. While, these could be a boon to the application of Deep neural networks for predictive analysis, they require high quality of annotated images. Large amounts of unlabelled data could thus be a hindrance in the development of deep models for medical datasets in low resource settings. Here, we demonstrate that one can overcome this resource constraint by subset selection for expert data labeling followed by data augmentation methods like wGANs. However, this needs to be performed intelligently and diversity driven so that a wider subset of images are selected which perform close to scenarios where we would have had the entire dataset labeled. This demonstrates the utility of subset selection based on diversity which could be a solution for low and middle income countries that do not have extensive resources for data labeling which by itself is an expensive process. This would not only decrease the labeling cost but also contribute to the requirement of large annotated data requirements for training Deep models.

**Chapter 2 : Feature extraction from PET-CT scans for prognostic analysis of Hodgkin-Lymphoma**

**Abstract**

Hodgkin lymphoma (HL) is a complex hematological malignancy that requires accurate prognostic assessment for appropriate treatment planning and patient management. In this project, we propose a novel approach which will eventually lead to prognostic analysis of HL by combining PET (Positron Emission Tomography) and CT (Computed Tomography) scans and leveraging the power of deep learning architectures like ResNet and YOLO for extracting clinically significant features. The integration of PET and CT scans allows us to extract comprehensive features related to tumors, providing a more complete and accurate representation of the disease. The YOLO architecture, known for its object detection capabilities, enables us to efficiently and accurately locate nodal regions of interest in PET scans, facilitating subsequent analysis and characterization. Through advanced image processing techniques, we extract relevant quantitative features from the identified nodal regions, such as metabolic activity, size, shape, and spatial distribution. These features serve as input to our prognostic model, which employs machine learning algorithms and statistical methods to predict disease progression, treatment response, and overall patient prognosis. The project's findings and validation will be based on a comprehensive dataset of PET and CT scans from patients diagnosed with Hodgkin lymphoma, with accompanying clinical data and long-term follow-up. The results will be evaluated and compared against established prognostic factors and existing clinical guidelines, demonstrating the potential value of our proposed approach in enhancing prognostic assessment for HL. Overall, this project contributes to the field of lymphoma research by combining multimodal imaging and deep learning techniques for improved prognostic analysis of Hodgkin lymphoma, offering potential advancements in personalized medicine and treatment strategies for patients with this disease.

**Introduction**

Hodgkin lymphoma (HL), also known as Hodgkin's disease, is a type of cancer that affects the lymphatic system. It is characterized by the presence of specific abnormal cells called Reed-Sternberg cells. HL typically begins in the lymph nodes, which are part of the lymphatic system and play a crucial role in the body's immune response. However, the disease can also spread to other lymphoid tissues, such as the spleen, bone marrow, liver, and lungs. There are two main types of Hodgkin lymphoma: classical Hodgkin lymphoma (cHL) and nodular lymphocyte-predominant Hodgkin lymphoma (NLPHL). The exact cause of Hodgkin lymphoma is unknown, but it is believed to result from a combination of genetic factors, alterations in the immune system, and exposure to certain infections, such as the Epstein-Barr virus (EBV). HL can occur at any age, but it is most commonly diagnosed in young adults between the ages of 15 and 35, as well as in older adults over the age of 50.

Diagnosis of Hodgkin lymphoma involves various diagnostic procedures, including a thorough medical history review, physical examination, blood tests, imaging studies (such as CT scans, PET scans, or MRI scans), and the collection of a tissue sample for microscopic examination (biopsy). The biopsy helps confirm the presence of Reed-Sternberg cells and determine the specific type and stage of the disease. Treatment for Hodgkin lymphoma depends on several factors, including the stage of the cancer, the subtype, the patient's age, and overall health. Common treatment approaches include chemotherapy, radiation therapy, targeted therapies, and in some cases, stem cell transplantation. The goal of treatment is to achieve remission, which may involve a combination of these treatment modalities.

Prognosis for Hodgkin lymphoma has significantly improved over the years, with high cure rates, particularly for early-stage disease. The five-year survival rate for HL is generally favorable, with the majority of patients achieving long-term remission. Ongoing monitoring and follow-up care are important to detect any potential relapses or late effects of treatment. Research efforts in Hodgkin lymphoma focus on understanding the underlying mechanisms of the disease, refining diagnostic techniques, identifying potential biomarkers for risk stratification, and developing targeted therapies with fewer side effects. These advancements aim to further improve outcomes and enhance the quality of life for individuals affected by Hodgkin lymphoma.

**Treatment Procedures for HL**

**ABVD**

ABVD is a commonly used chemotherapy regimen for the treatment of Hodgkin lymphoma. It is considered a standard treatment option for patients with early-stage Hodgkin lymphoma. This treatment also has components similar to BEACOPP. ABVD treatment is typically administered in cycles, with each cycle lasting approximately 28 days. The number of cycles depends on the stage of the disease, the response to treatment, and the recommendation of the healthcare team.

ABVD is generally well-tolerated by most patients, but it can cause side effects. Common side effects include nausea, hair loss, fatigue, bone marrow suppression (resulting in decreased blood cell counts), increased susceptibility to infection, and potential long-term effects on fertility and secondary malignancies. Close monitoring and supportive care are essential to manage and alleviate these side effects. ABVD has shown high cure rates for patients with Hodgkin lymphoma, especially those with early-stage disease. However, for patients with advanced-stage or high-risk disease, alternative chemotherapy regimens, such as BEACOPP, may be considered for a more intensive treatment approach. The decision to use ABVD as the treatment regimen for Hodgkin lymphoma depends on several factors, including the stage and risk stratification of the disease, the patient's overall health status, and discussions with the medical oncology team. Treatment plans should be tailored to the individual's specific circumstances to optimize outcomes and minimize potential side effects.

**BEACOPP**

BEACOPP is an intensive chemotherapy regimen commonly used for the treatment of Hodgkin lymphoma, particularly for patients with advanced-stage disease or those with unfavorable prognostic factors. BEACOPP is a multi-agent chemotherapy regimen that combines several drugs to target and destroy cancer cells. The BEACOPP regimen is administered in multiple cycles over a period of several months. The treatment is typically divided into two phases: the escalated phase and the standard phase.

During the escalated phase, the dosage of certain drugs is higher to achieve a more intensive treatment effect. This phase is often recommended for patients with advanced-stage disease or those with unfavorable risk factors. Following the escalated phase, patients transition to the standard phase, which involves lower drug dosages but still maintains the effectiveness of the treatment. The duration and number of cycles in each phase may vary based on individual patient factors and the response to treatment.

BEACOPP treatment for Hodgkin lymphoma is associated with higher rates of complete remission and improved long-term outcomes compared to other chemotherapy regimens. However, it can also cause significant side effects due to its intensity, including bone marrow suppression, increased risk of infection, nausea, hair loss, and potential long-term effects on fertility and secondary malignancies. Close monitoring and supportive care are crucial during treatment to manage and mitigate these side effects. The decision to use the BEACOPP regimen for Hodgkin lymphoma treatment is typically based on various factors, including the stage and risk stratification of the disease, overall health status of the patient, and consultation with a specialized oncology team. Treatment plans should be tailored to the individual's specific circumstances to ensure optimal outcomes and minimize potential side effects.

**Using ML to assist the process**

Feature extraction, combined with machine learning (ML) techniques, can play a crucial role in assisting clinicians in determining the most appropriate treatment option, such as ABVD or BEACOPP, for better treatment outcomes in Hodgkin lymphoma. Here's how ML can contribute to this process:

1. Improved Patient Stratification: Feature extraction techniques can identify relevant and informative features from various medical imaging modalities, such as PET scans and CT scans. These features can capture important characteristics of the tumor, such as size, shape, and metabolic activity. ML models can then analyze these extracted features and identify patterns or subgroups of patients who may respond differently to different treatments. This can help clinicians stratify patients into distinct groups, allowing for personalized treatment decisions.
2. Treatment Response Prediction: ML models can be trained on a combination of patient data, including demographic information, clinical factors, and extracted imaging features, along with treatment outcomes. By leveraging these data, ML models can learn to predict the response to different treatment regimens, such as ABVD or BEACOPP, for individual patients. This can assist clinicians in assessing the potential effectiveness of each treatment option and making informed decisions tailored to each patient's needs.
3. Prognostic Assessment: ML models can analyze a variety of patient-related factors, including clinical data, genetic markers, laboratory results, and imaging features, to predict the prognosis and overall survival rates for different treatment approaches. By incorporating these models into clinical decision-making, clinicians can gain insights into the potential long-term outcomes of ABVD or BEACOPP treatments for individual patients. This information can guide discussions with patients regarding treatment options, potential risks, and expected benefits.
4. Treatment Planning and Monitoring: ML models can aid in treatment planning by providing a quantitative assessment of the predicted response to ABVD or BEACOPP. By considering the patient's individual characteristics and predicted treatment outcomes, clinicians can make informed decisions about the optimal treatment regimen, dosage adjustments, or the need for additional therapies. ML models can also be utilized for treatment monitoring, tracking the response to treatment over time and enabling early intervention or modification if necessary.
5. Clinical Decision Support: ML models can serve as decision support tools for clinicians, providing evidence-based recommendations based on large datasets and analysis of historical treatment outcomes. These models can integrate information from multiple sources, including clinical guidelines, medical literature, and patient-specific data, to assist clinicians in making treatment decisions that maximize efficacy while minimizing potential risks.

**Preliminaries**

**PET Scan**

PET-CT (Positron Emission Tomography-Computed Tomography) is a powerful imaging technique that combines the functional information from PET scans with the anatomical details provided by CT scans. It plays a significant role in the assessment of patients with various diseases, including cancer, and aids in determining appropriate treatment strategies. These are some of the facets of PET-CT scans:

1. Imaging Technique Combination: PET-CT integrates two imaging modalities into a single examination. PET utilizes radiotracers, which are radioactive substances that emit positrons. These radiotracers are injected into the patient's body and accumulate in tissues based on their metabolic activity. CT, on the other hand, uses X-rays to create detailed anatomical images of the patient's body. By combining these two techniques, PET-CT provides both functional and structural information, enabling a comprehensive assessment of the patient.
2. Accurate Localization: The fusion of PET and CT images in PET-CT scans allows for precise localization of functional abnormalities within the anatomical context. It helps identify the exact location of areas with abnormal metabolic activity, such as tumors or areas of inflammation. This localization is crucial for accurate diagnosis, staging, and treatment planning.
3. Disease Detection and Staging: PET-CT scans are highly sensitive in detecting various diseases, particularly cancer. They can reveal the presence and extent of tumors, identify metastases or spread to other regions of the body, and determine the stage of the disease. This information is vital in developing an appropriate treatment strategy and assessing the prognosis of the patient.
4. Treatment Response Assessment: PET-CT scans play a crucial role in monitoring the response to treatment. By comparing pre- and post-treatment PET-CT images, clinicians can assess changes in metabolic activity within tumors. A decrease in metabolic activity suggests a positive response to treatment, while persistent or increased activity may indicate treatment resistance or disease progression. This information helps guide treatment decisions, including modifications in therapy or the need for additional interventions.
5. Treatment Planning: PET-CT scans provide valuable information for treatment planning in cancer patients. They assist in determining the target volumes for radiation therapy, helping to accurately deliver radiation to tumor sites while minimizing damage to healthy tissues. PET-CT can also guide surgical planning by identifying the precise location and extent of tumors, aiding surgeons in determining the optimal surgical approach.
6. Prognostic Assessment: PET-CT findings, such as the intensity of metabolic activity in tumors (measured by SUV) and the extent of disease involvement, have prognostic implications. Higher metabolic activity or a larger tumor burden on PET-CT scans is often associated with a poorer prognosis. This information helps in risk stratification and predicting patient outcomes, influencing treatment decisions and follow-up strategies.

**Standardized uptake value (SUV)**

Standardized Uptake Value (SUV) is a quantitative measure used in medical imaging, particularly in Positron Emission Tomography (PET) scans. It provides a standardized measurement of the uptake or concentration of a radiotracer within a region of interest, typically a tumor or specific tissue.

The calculation of SUV involves several steps:

1. Image Acquisition: A patient is injected with a radiotracer, which is a radioactive substance that emits positrons. The radiotracer is taken up by tissues, including tumor cells, based on their metabolic activity.
2. PET Scan Imaging: A PET scanner detects the emitted positrons and creates images that represent the distribution and concentration of the radiotracer within the body.
3. Region of Interest (ROI) Selection: A specific region of interest, usually corresponding to the tumor or a specific tissue, is selected on the PET images.
4. SUV Calculation: The SUV is calculated by normalizing the radioactivity concentration within the selected ROI to the injected dose of the radiotracer and the patient's body weight. The formula for calculating SUV is typically as follows: SUV = (Tissue radioactivity concentration in kBq/g) / (Injected dose in kBq / Patient's body weight in grams)
5. The clinical significance of SUV lies in its ability to provide a quantitative measure of radiotracer uptake in tumors or tissues. It helps in assessing the metabolic activity of the tissue, which can be indicative of various physiological processes, including glucose metabolism, cell proliferation, and tumor aggressiveness.

SUV has several roles in determining clinical features related to tumors:

1. Diagnosis and Staging: SUV can assist in differentiating between malignant and benign lesions. Higher SUV values are often associated with a higher likelihood of malignancy. In addition, SUV measurements can be used for staging tumors and assessing the extent of disease spread.
2. Treatment Response Assessment: Changes in SUV over time can be used to evaluate the response to therapy. A decrease in SUV after treatment may indicate a positive response, while an increase or persistent high SUV may suggest resistance or disease progression.
3. Prognostic Indicator: SUV can serve as a prognostic indicator, providing information on the aggressiveness and potential outcome of the disease. Higher SUV values have been associated with poorer prognosis in various types of cancer.
4. Treatment Planning: SUV values can guide treatment planning by helping to identify target volumes for radiation therapy and selecting appropriate treatment modalities. Areas with high SUV may require additional radiation doses or targeted therapies.

It's important to note that SUV values should be interpreted in the context of clinical and imaging findings, as various factors can influence SUV measurements, such as patient characteristics, the type of radiotracer used, and technical factors related to imaging protocols.

**Our Approach**

We have implemented a 6 step approach for extracting features related to tumors, using a combination of deep learning tools. Figure 7 shows a complete flowchart of feature extraction incorporated for PET-CT scans. Features extracted include various properties of tumor like intensity of uptake measured using SUV (standardized uptake value), area, lesion glycolysis, metabolic tumor volume. Also spread of tumor at all the nodal sites is computed automatically.

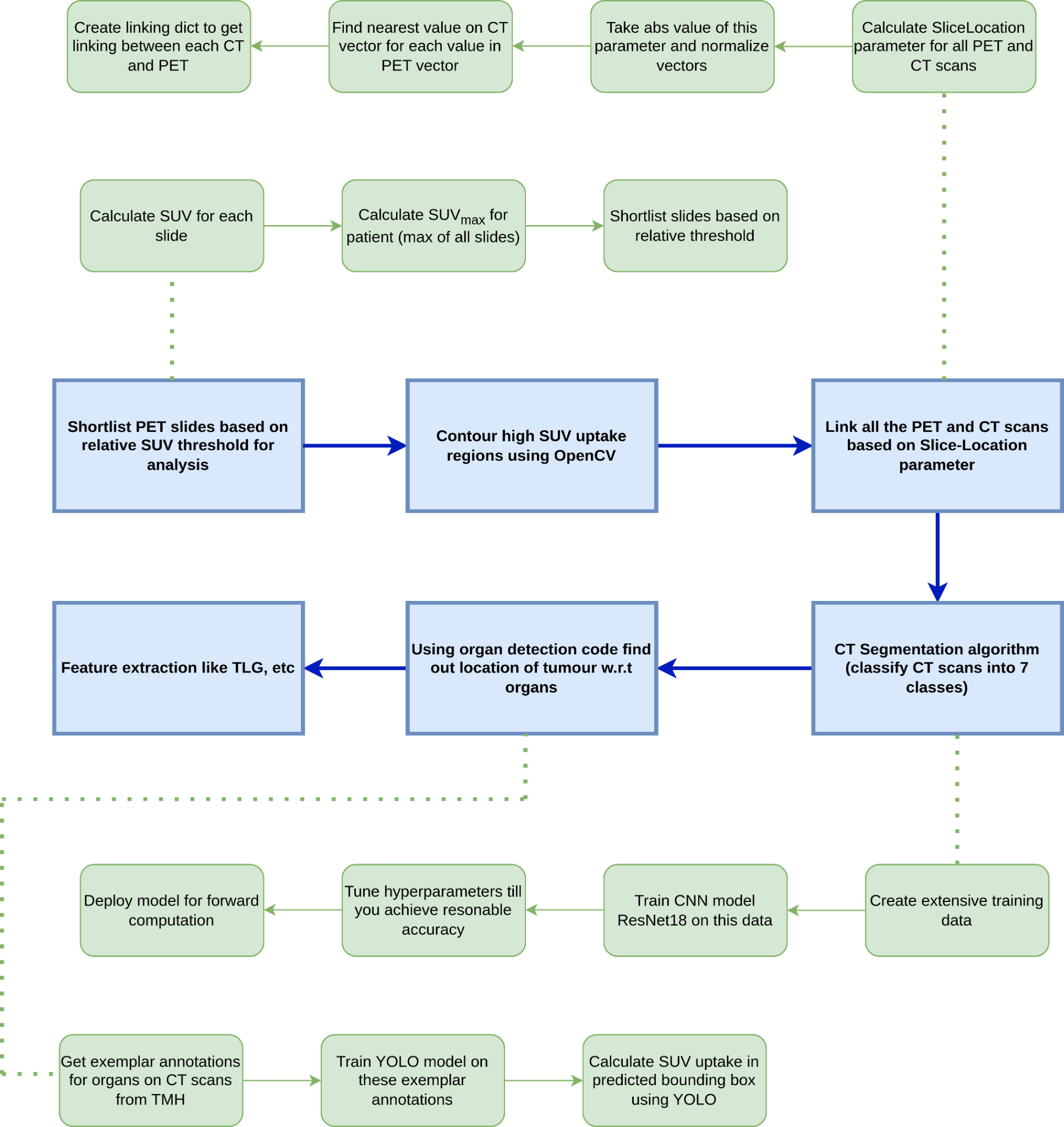


Figure 7: Flowchart for feature extraction for PET-CT scans

1. **Shortlisting PET scans for computation**

Specifically for HL whole body PET scan is taken into consideration for analysis by clinicians. Total information extracted by scan is in the form of images from head to toe. Analyzing slices with high SUV uptake help in extracting more meaningful and clinically relevant information compared to analyzing all the data available. So we incorporate algorithms to shortlist slices from PET scans which have more uptake compared to others.

In this process initially SUV for each slice is calculated. This gives us distribution of uptake throughout the body. Taking max of these values gives us SUVmax for the patient (this also is used as a feature in the final predicting algorithm). A relative threshold is set to shortlist slides for further analysis, for example if relative threshold is 0.9 then slice will be selected for further computation if uptake of slide is greater than 0.9 times SUVmax

SUVslide > Relative-Threshold \* SUVmax

Figure 8 shows pictorial representation of how slides are selected based on their uptake values

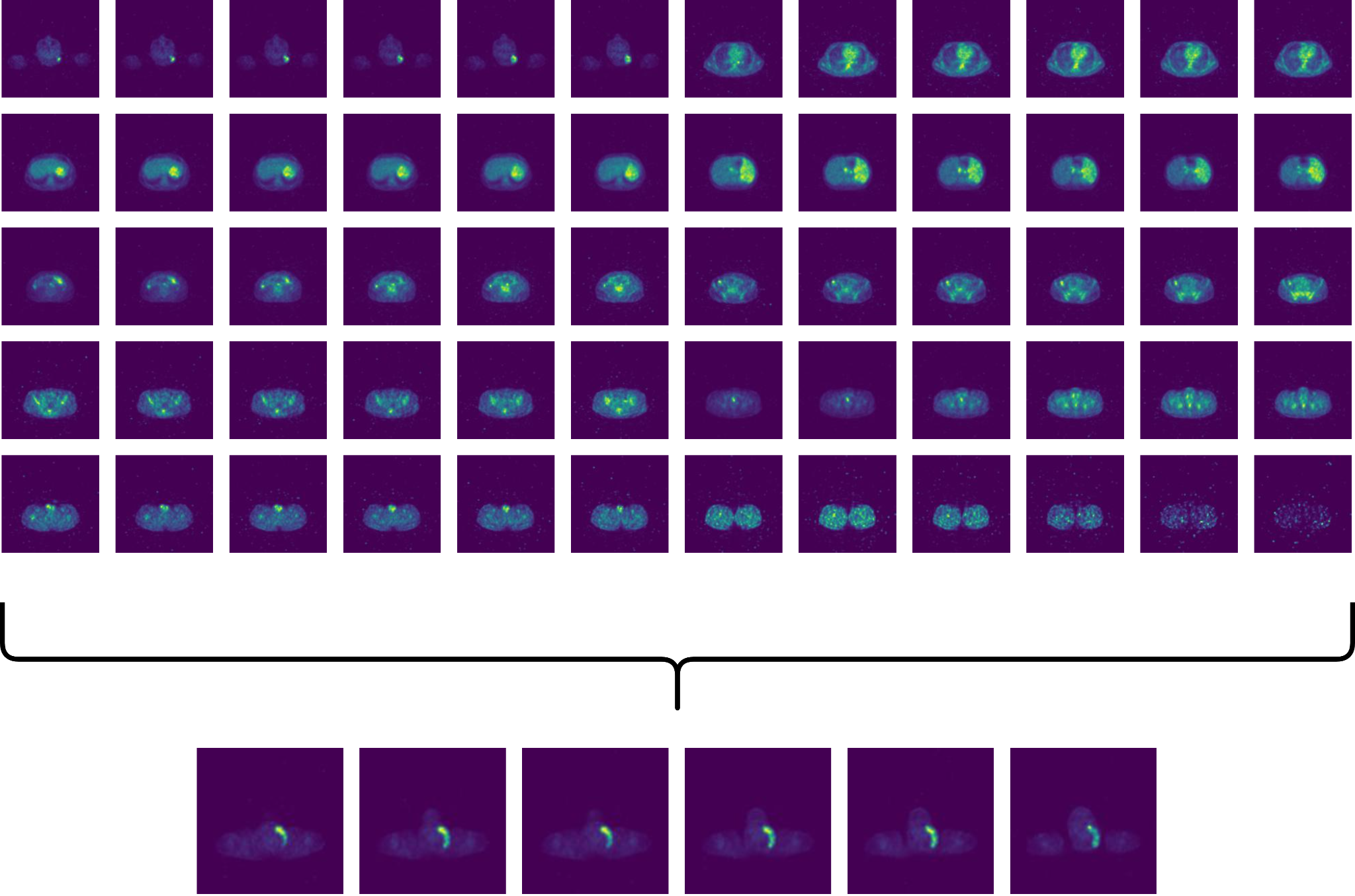
****

Figure 8: Selected PET scans for analysis from pool of scans

1. **Contouring high SUV uptake regions**

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. It provides a comprehensive set of tools and functions for image and video processing, feature detection and extraction, object recognition, and more. OpenCV is widely used in various domains, including robotics, augmented reality, surveillance, and medical imaging.

We have used findContours function which is inbuilt in the OpenCV library. The cv2.findContours() function is a part of the OpenCV library and is used for detecting contours in images. Contours can be understood as continuous curves or boundaries that enclose the areas of similar intensity or color. These contours are often used for various image processing tasks such as object detection, shape analysis, and recognition.

These are the parameters function

1. image: The input image from which contours are to be extracted. It should be a binary image (e.g., thresholded or edge-detected).
2. mode: It specifies the contour retrieval mode. It has three possible values:
3. cv2.RETR\_EXTERNAL: Retrieves only the external contours.
4. cv2.RETR\_LIST: Retrieves all the contours without establishing any hierarchical relationships.
5. cv2.RETR\_TREE: Retrieves all the contours and creates a full hierarchy of nested contours.
6. method: It specifies the contour approximation method. It has four possible values:
7. cv2.CHAIN\_APPROX\_NONE: Stores all the contour points.
8. cv2.CHAIN\_APPROX\_SIMPLE: Compresses horizontal, vertical, and diagonal segments into their endpoints only, discarding the intermediate points.
9. cv2.CHAIN\_APPROX\_TC89\_L1: Applies Teh-Chin chain approximation algorithm.
10. cv2.CHAIN\_APPROX\_TC89\_KCOS: Applies Teh-Chin chain approximation algorithm with the additional K cos similarity test.
11. contours (optional): Output parameter that contains the detected contours as a Python list.
12. hierarchy (optional): Output parameter that represents the hierarchical relationships between detected contours as a Python list.
13. offset (optional): An optional offset by which every contour point is shifted. This is useful when working with ROIs (regions of interest) within larger images.

These are return values of the function

1. contours: A Python list of detected contours. Each contour is represented as a numpy array of shape (N, 1, 2), where N is the number of contour points.
2. hierarchy: A Python list that represents the hierarchical relationships between contours. It has four elements [Next, Previous, First Child, Parent], where each element is an index to the corresponding contour in the contours list.

This function is individually applied to every slide in selected slide and all the tumors are extracted from the slide

Figure 9 demonstrates how this function efficiently extracts high intensity contours from image

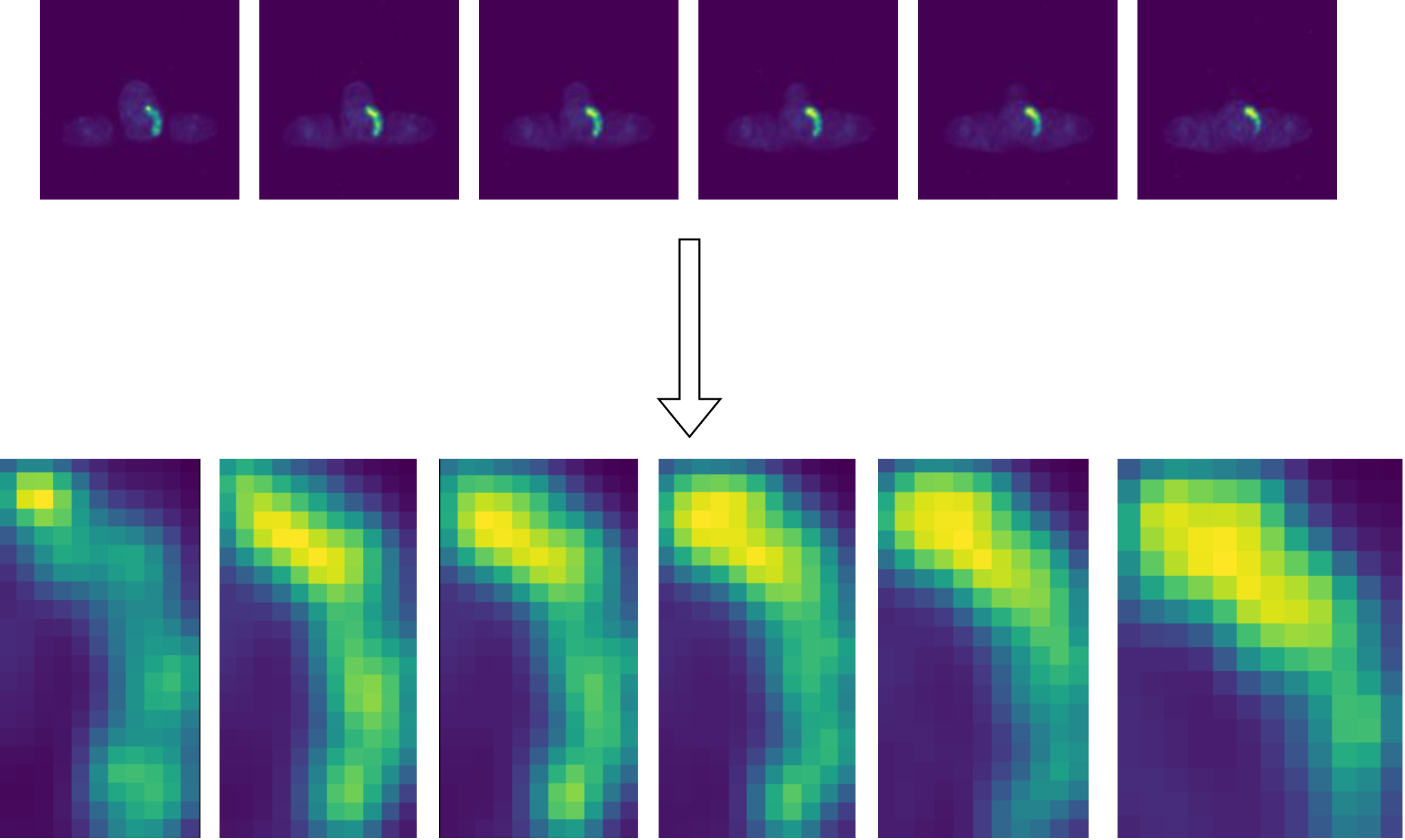


Figure 9: Extracted contours from PET slides

1. **Linking PET and CT scans**

CNN models like ResNet18 are widely used for classification tasks. After contouring the exact tumor from the PET slide, the next task is to identify the exact location of the tumor i.e. it’s nodal location. We have trained and deployed ResNet18 along with the YOLO v5 model for this task (which will be explained subsequently in further sections). Training ResNet on PET scans is not recommended as PET scan has tumor related information but less information related to location of slide. Sections can be easily distinguishable in CT scans compared to PET and can be used for classification tasks. SliceLocation parameter can be used to create a linking between all the PET and CT scans

Slice Location parameter in DICOM (Digital Imaging and Communications in Medicine) files is a metadata attribute that provides crucial information about the physical position of an image slice within a three-dimensional volume. It indicates the location of the image slice along the z-axis or the longitudinal axis of the patient's body. Slice Location is typically expressed in millimeters and is measured relative to a predefined reference point, often referred to as the origin or the coordinate system's origin. The reference point can vary depending on the scanner manufacturer and imaging protocol but is generally consistent within a specific imaging study.

The Slice Location parameter is an essential attribute for correctly ordering and aligning image slices in a series, enabling accurate visualization and reconstruction of volumetric data. It helps establish the spatial relationship between different slices, allowing for proper interpretation and analysis of medical images, such as CT scans or PET volumes. By leveraging the Slice Location attribute, radiologists and medical professionals can identify and navigate specific anatomical regions within a volume. It aids in precise localization, measurement, and comparison of structures or abnormalities present in multiple slices of the same study. It is worth noting that the Slice Location parameter does not provide absolute anatomical coordinates or precise distance measurements within the patient's body. It only represents the relative positioning of image slices within the imaging series. To derive accurate spatial information, additional DICOM attributes such as Pixel Spacing, Image Position (Patient), and Image Orientation (Patient) can be utilized in conjunction with the Slice Location parameter.

Figure 10 shows how we have used slice location parameter to create a linking between PET scans and corresponding CT scans

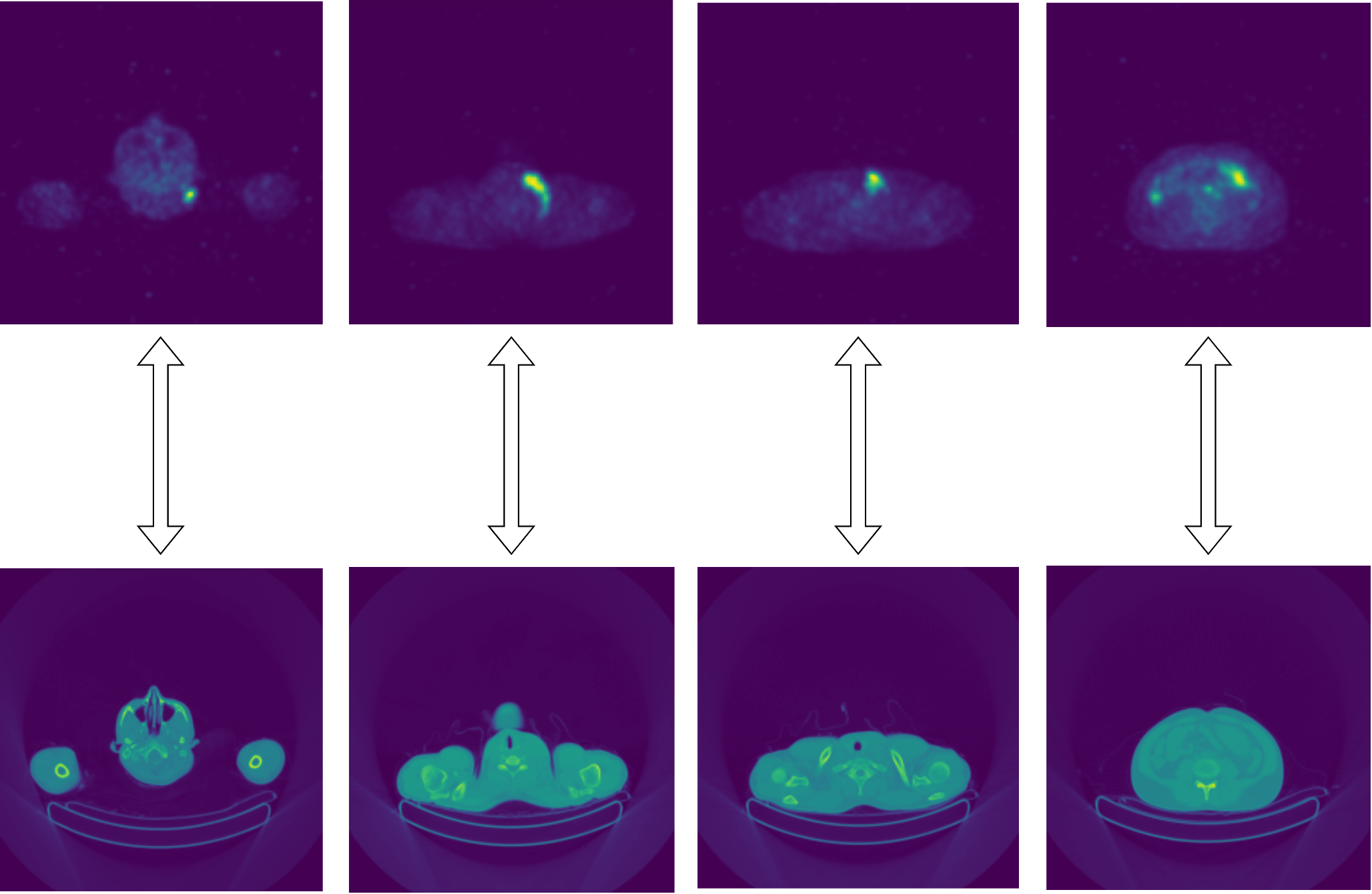


Figure 10: PET-CT Linking

1. **CT Classification using ResNet18**

ResNet-18 is a convolutional neural network (CNN) architecture that belongs to the ResNet (Residual Network) family. ResNet was introduced by Microsoft Research in 2015 and has since become one of the most influential and widely used deep learning architectures for various computer vision tasks. The "18" in ResNet-18 refers to the depth of the network, which means it consists of 18 layers. Figure 11 shows the representative structure of ResNet. The key innovation in ResNet is the introduction of residual connections or skip connections, which help address the degradation problem in deep neural networks. The degradation problem refers to the observation that as the network depth increases, the performance of the network starts to degrade instead of improving. ResNet solves this problem by using skip connections that allow information to flow directly from earlier layers to later layers, bypassing a few layers in between. This helps in alleviating the vanishing gradient problem and enables easier optimization of deep networks.

Here is a high-level overview of the ResNet-18 architecture:

Convolutional Layer: The input to ResNet-18 is an RGB image of size 224x224 pixels. The first layer is a 7x7 convolutional layer with a stride of 2, followed by a max-pooling layer with a 3x3 window and a stride of 2. This reduces the spatial dimensions of the image.

Residual Blocks: ResNet-18 consists of four stages, each containing multiple residual blocks. Each residual block consists of two or three convolutional layers followed by a skip connection that adds the input to the output. The skip connection allows the network to learn the residual mapping, making it easier to optimize. ResNet-18 has a total of 18 convolutional layers, which are divided into 4 stages with 2, 2, 2, and 2 residual blocks, respectively.

Global Average Pooling: After the last residual block, a global average pooling layer is applied to reduce the spatial dimensions to a vector of features. This pooling operation computes the average value of each feature map, resulting in a fixed-size feature vector regardless of the input image size.

Fully Connected Layer: The global average pooled features are then fed into a fully connected layer, followed by a softmax activation function for classification tasks. The number of neurons in the fully connected layer depends on the specific classification problem.

ResNet-18 has been pre-trained on large-scale image classification datasets like ImageNet, which contains millions of labeled images from thousands of categories. Due to its depth and skip connections, ResNet-18 has demonstrated excellent performance in image classification, object detection, and other computer vision tasks. It strikes a good balance between model size and performance, making it a popular choice for various applications.

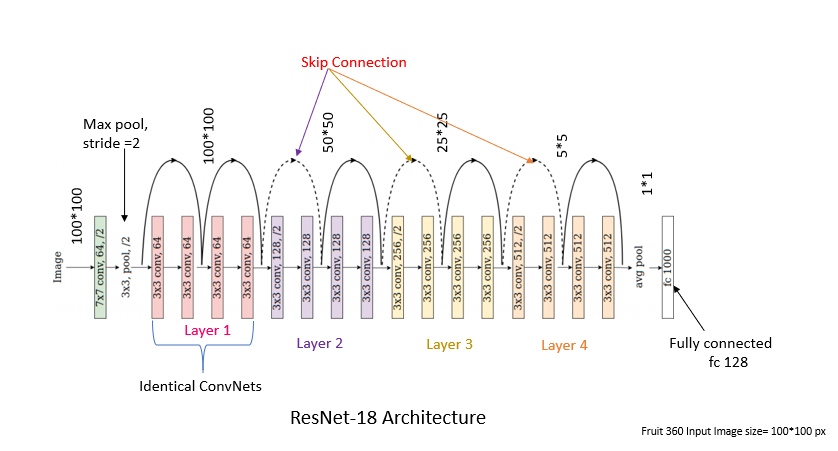


Figure 11: ResNet18 Architecture

1. **Nodal location detection using YOLO**

It is important to find SUV in each nodal location in PET scan, this information can be further fed into a predictive model as features. Clinically important nodal features are pointed out by doctors of TMH. We got some examplar annotations for nodal locations in the form of bounding boxes. 7 YOLO models are trained for each section starting from head to legs.

YOLOv5 is an upgraded version of the YOLO (You Only Look Once) object detection algorithm. It was developed by Ultralytics and released in 2020. YOLOv5 builds upon the concepts introduced in previous versions but introduces several architectural improvements and training techniques to achieve better performance.

Here's a detailed overview of how YOLOv5 works:

Network Architecture: YOLOv5 employs a deep convolutional neural network architecture. It follows a similar principle as previous versions, where the network predicts bounding boxes and class probabilities directly from the input image.

Backbone: YOLOv5 uses a CSPDarknet53 backbone, which is a modified version of the Darknet53 backbone used in previous YOLO versions. CSPDarknet53 incorporates Cross Stage Partial connections to improve the network's ability to capture both low-level and high-level features.

Feature Pyramid: YOLOv5 incorporates a Feature Pyramid Network (FPN) to handle objects at different scales. FPN consists of multiple convolutional layers that merge high-resolution features from earlier layers with low-resolution features from deeper layers. This helps in detecting objects of various sizes.

Prediction Head: The prediction head of YOLOv5 consists of a series of convolutional layers that process the features from the FPN. It predicts bounding boxes and class probabilities at different spatial scales. The bounding box predictions include coordinates (x, y, width, height) and objectness scores (confidence that the box contains an object).

Anchor Boxes: YOLOv5 utilizes anchor boxes, which are pre-defined bounding boxes of various sizes and aspect ratios. These anchor boxes are used as reference points for predicting the final bounding boxes. YOLOv5 applies a process called anchor box assignment, where the predicted boxes are matched with the most suitable anchor boxes based on IoU.

Training Techniques: YOLOv5 introduces several training techniques to improve performance. It uses a technique called "mosaic data augmentation" where multiple images are combined into a single mosaic image during training. This helps the model learn to detect objects in complex scenes. Another technique used is "self-adversarial training," where the model learns from its own incorrect predictions to improve performance.

Loss Function: YOLOv5 uses a combination of different loss components to train the model. The loss function includes terms for bounding box regression loss, objectness loss, and classification loss. The loss components are weighted to ensure a balance between different objectives.

During inference, YOLOv5 processes the input image in a single pass through the network. The prediction head generates bounding box predictions and class probabilities at different scales. Non-Maximum Suppression (NMS) is then applied to remove redundant and overlapping detections, resulting in the final set of detected objects.

YOLOv5 has gained popularity due to its improved accuracy and speed compared to previous versions. It is widely used for a variety of computer vision tasks, including object detection, instance segmentation, and pedestrian detection, among others.

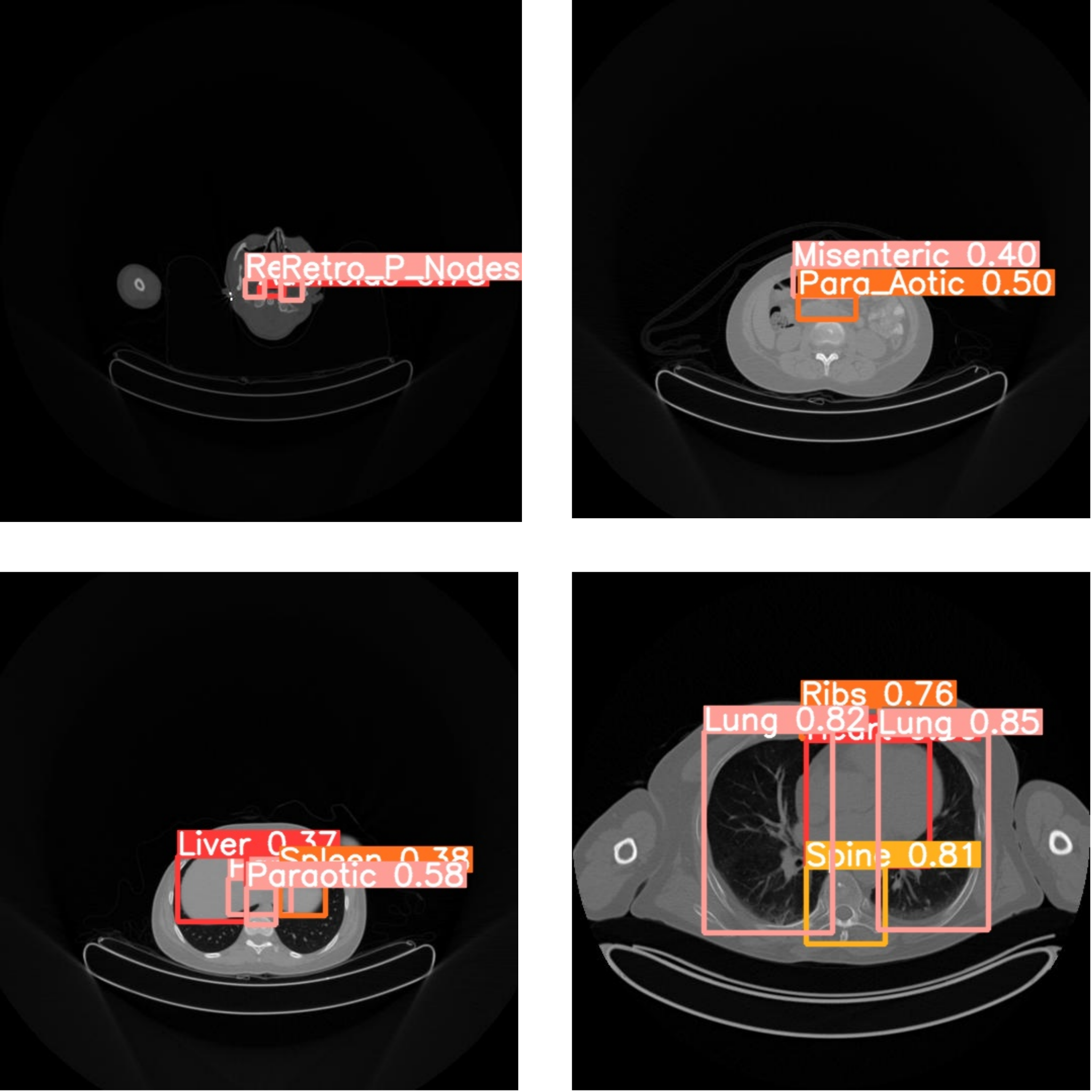


Figure 12: YOLO detections

For each section classified by CT, the corresponding YOLO model is used to generate bounding boxes in CT scans, further these coordinates are mapped after appropriate scaling to corresponding PET scans for further analysis.

Combining features extracted using YOLO bounding boxes and tumor contours give final pool of features that will be used in predictive model

**Future Work**

In this study, we have successfully developed a pipeline for extracting features from PET-CT scans of Hodgkin lymphoma patients using ResNet18 and YOLO models. Once we receive the data from Tata Memorial Hospital, this pipeline will allow us to efficiently extract features in bulk, enabling further analysis.

Moving forward, the next steps involve utilizing these extracted features in a predictive algorithm to determine the most suitable treatment option between BEACOPP and CBVD. Our plan is to employ the XGBoost algorithm, known for its effectiveness in regression and classification tasks, for this purpose.

To continue this work, we propose the following steps:

1. Data Collection: Continue gathering PET-CT scan data from Tata Memorial Hospital, ensuring adherence to privacy regulations and proper anonymization.
2. Feature Extraction: Utilize the pretrained ResNet18 and YOLO models to extract relevant features from the PET-CT scans. Carefully preprocess the data to match the input requirements of these models.
3. Data Preprocessing: Prepare the extracted features for further analysis, which may involve normalization, scaling, or encoding to ensure compatibility with the predictive algorithm.
4. Data Split: Divide the dataset into training, validation, and testing subsets. The training set will be used to train the XGBoost model, the validation set will assist in tuning hyperparameters, and the testing set will evaluate the final model's performance.
5. Model Training: Train an XGBoost model using the training dataset. Optimize the model's hyperparameters using techniques such as cross-validation and grid search.
6. Model Evaluation: Evaluate the trained XGBoost model using the testing dataset, considering metrics such as accuracy, precision, recall, and F1-score. Additional evaluation techniques like ROC curves or confusion matrices can provide further insights.
7. Treatment Prediction: Utilize the trained XGBoost model to predict the most suitable treatment option (BEACOPP vs. ABVD) for new patients based on their extracted features. These predictions will serve as valuable guidance for medical professionals in making informed treatment decisions.
8. Validation and Iteration: Continuously validate the predictive algorithm's performance on new data and iterate on the model as necessary to improve its accuracy and reliability.

Collaboration with medical professionals and experts in the field is essential to ensure the validity and clinical relevance of the findings. Moreover, ongoing monitoring and evaluation of the predictive algorithm's performance on new data will contribute to its further refinement.

By following these steps, we aim to develop a robust predictive algorithm that can assist in determining the most appropriate treatment option for Hodgkin lymphoma patients based on their PET-CT scan features.

**References :**

[1] Pankaj K Agarwal, Sariel Har-Peled, Kasturi R Varadarajan, et al. 2005. Geometric approximation via coresets. Combinatorial and computational geometry 52, 1 (2005), 1–30.

[2] Waqar Ahmad, Hazrat Ali, Zubair Shah, and Shoaib Azmat. 2022. A new generative adversarial network for medical images super resolution. Scientific Reports 12, 1 (2022), 9533.

[3] Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy. 2020. Dataset of breast ultrasound images. Data in Brief 28 (2020), 104863. https: //doi.org/10.1016/j.dib.2019.104863

[4] Martin Arjovsky and Léon Bottou. 2017. Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862 (2017).

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In International conference on machine learning. PMLR, 214–223.

[6] Mihalj Bakator and Dragica Radosav. 2018. Deep learning and medical diagnosis: A review of literature. Multimodal Technologies and Interaction 2, 3 (2018), 47.

[7] Gérard Biau, Maxime Sangnier, and Ugo Tanielian. 2021. Some theoretical insights into Wasserstein GANs. The Journal of Machine Learning Research 22, 1 (2021), 5287–5331.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. Commun. ACM 63, 11 (2020), 139–144.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. 770–778.

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015).

[11] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360 (2016).

[12] Fei Jiang, Yong Jiang, Hui Zhi, Yi Dong, Hao Li, Sufeng Ma, Yilong Wang, Qiang Dong, Haipeng Shen, and Yongjun Wang. 2017. Artificial intelligence in healthcare: past, present and future. Stroke and vascular neurology 2, 4 (2017).

[13] Kaggle. 2019. Aptos 2019 blindness detection. https://www.kaggle.com/c/ aptos2019-blindness-detection/data

[14] Vishal Kaushal, Rishabh Iyer, Suraj Kothawade, Rohan Mahadev, Khoshrav Doctor, and Ganesh Ramakrishnan. 2019. Learning from less data: A unified data subset selection and active learning framework for computer vision. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 1289–1299.

[15] Vishal Kaushal, Ganesh Ramakrishnan, and Rishabh Iyer. 2022. Submodlib: A Submodular Optimization Library. arXiv preprint arXiv:2202.10680 (2022).

[16] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. 2018. Identifying medical diagnoses and treatable diseases by image-based deep learning. cell 172, 5 (2018), 1122–1131.

[17] Daniel S. Kermany, Michael Goldbaum, Wenjia Cai, Carolina C.S. Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, Justin Dong, Made K. Prasadha, Jacqueline Pei, Magdalene Y.L. Ting, Jie Zhu, Christina Li, Sierra Hewett, Jason Dong, Ian Ziyar, Alexander Shi, Runze Zhang, Lianghong Zheng, Rui Hou, William Shi, Xin Fu, Yaou Duan, Viet A.N. Huu, Cindy Wen, Edward D. Zhang, Charlotte L. Zhang, Oulan Li, Xiaobo Wang, Michael A. Singer, Xiaodong Sun, Jie Xu, Ali Tafreshi, M. Anthony Lewis, Huimin Xia, and Kang Zhang. 2018. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. Cell 172, 5 (2018), 1122–1131.e9. https://doi. org/10.1016/j.cell.2018.02.010

[18] Suraj Kothawade, Akshit Shrivastava, Venkat Iyer, Ganesh Ramakrishnan, and Rishabh Iyer. 2022. DIAGNOSE: Avoiding Out-of-Distribution Data Using Submodular Information Measures. In Medical Image Learning with Limited and Noisy Data: First International Workshop, MILLanD 2022, Held in Conjunction with MICCAI 2022, Singapore, September 22, 2022, Proceedings. Springer, 141–150.

[19] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In International conference on machine learning. PMLR, 97–105.

[20] Debadyuti Mukherkjee, Pritam Saha, Dmitry Kaplun, Aleksandr Sinitca, and Ram Sarkar. 2022. Brain tumor image generation using an aggregation of GAN models with style transfer. Scientific Reports 12, 1 (2022), 1–16.

[21] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. 2015. Deep learning applications and challenges in big data analytics. Journal of big data 2, 1 (2015), 1–21.

[22] Zhaoqing Pan, Weijie Yu, Xiaokai Yi, Asifullah Khan, Feng Yuan, and Yuhui Zheng. 2019. Recent progress on generative adversarial networks (GANs): A survey. IEEE access 7 (2019), 36322–36333.

[23] Ricardo Ribani and Mauricio Marengoni. 2019. A survey of transfer learning for convolutional neural networks. In 2019 32nd SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T). IEEE, 47–57.

[24] Youssef Skandarani, Pierre-Marc Jodoin, and Alain Lalande. 2023. Gans for medical image synthesis: An empirical study. Journal of Imaging 9, 3 (2023), 69.

[25] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. Advances in neural information processing systems 26 (2013).

[26] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. 2016. Convolutional neural networks for medical image analysis: Full training or fine tuning? IEEE transactions on medical imaging 35, 5 (2016), 1299–1312.

[27] Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. 2020. The computational limits of deep learning. arXiv preprint arXiv:2007.05558 (2020).

[28] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. 2018. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific data 5, 1 (2018), 1–9.

[29] Hristina Uzunova, Matthias Wilms, Nils D Forkert, Heinz Handels, and Jan Ehrhardt. 2022. A systematic comparison of generative models for medical images. International Journal of Computer Assisted Radiology and Surgery 17, 7 (2022), 1213–1224.

[30] Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing Data using t-SNE. Journal of Machine Learning Research 9 (2008), 2579–2605.

[31] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. Advances in neural information processing systems 29 (2016).

[32] Douglas Williams, Heiko Hornung, Adi Nadimpalli, and Ashton Peery. 2021. Deep learning and its application for healthcare delivery in low and middle income countries. Frontiers in Artificial Intelligence 4 (2021), 553987.

[33] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. 2018. Convolutional neural networks: an overview and application in radiology. Insights into imaging 9 (2018), 611–629.