CentraleSupélec

université PARIS-SACLAY

Big Data Research Project

Enhancing Self-Supervised Learning for Image Clustering Using Geometric Deep Learning

Zhuang XIANYUN

Islam MD KAMRUL

Github Link:

https://github.com/kamrulkonok/bdrp_project [1]

Advisor: Akash MALHOTRA - CentraleSupélec akash.malhotra@centralesupelec.fr

Advisor: Nacera SEGHOUANI - CentraleSupélec nacera.seghouani@centralesupelec.fr

Host Institution: CentraleSupélec

[1] https://github.com/kamrulkonok/bdrp_project

Abstract:

Contemporary self-supervised learning approaches have significantly advanced unsupervised image analysis, yet their application to medical imaging remains constrained by conventional CNN architectures' inherent limitations in handling geometric transformations. This research introduces a novel deep clustering framework that incorporates Group Equivariant Convolutional Neural Networks (G-CNNs) to substantially enhance feature representation learning for medical image clustering. Our methodological framework implements a systematic iterative pipeline extending the DeepCluster architecture through dual feature extraction pathways: (1) an AlexNet implementation incorporating Sobel filtering with random rotation augmentation, and (2) a P4M-equivariant CNN architecture that inherently encodes rotation and reflection invariance properties. Evaluated on the NIH Chest X-ray dataset, our approach leverages k-means clustering with iterative pseudo-labeling to progressively refine latent representations. Experimental results demonstrate that the proposed G-CNN framework achieves improved semantic coherence and clustering quality, converging faster than the conventional CNN-based approach. These findings establish the significant potential of geometric deep learning techniques for developing more robust and clinically relevant solutions for medical image clustering and automated disease detection systems.

Keywords: Self-supervised learning, Deep Clustering, Geometric Deep Learning, Group Equivariant CNNs, Medical imaging, Sobel Filtering

Contents

Chapter 1

Introduction

Deep clustering represents an advanced unsupervised learning paradigm that identifies inherent structural patterns within data by leveraging feature correlations without requiring labeled annotations. In medical imaging analysis interpretation—clustering techniques encounter substantial challenges due to the high-dimensional, complex nature of radiographic data. The absence of labeled annotations compounds these difficulties, necessitating robust self-supervised clustering approaches. However, conventional self-supervised clustering methodologies demonstrate significant limitations in maintaining feature consistency across geometric transformations (rotations, translations, reflections) that frequently occur in medical imaging contexts. These constraints directly impact the accuracy and reliability of automated disease detection systems Asano and et al. (2020).

Traditional convolutional neural networks (CNNs), such as AlexNet Krizhevsky et al. (2012), have been extensively deployed for self-supervised visual feature learning. These networks extract hierarchical feature representations through sequential convolutional and pooling operations. Self-supervised CNN-based methodologies typically employ pretext tasks—including contrastive learning Chen et al. (2020) or pseudo-labeling—to develop discriminative features without manual supervision. Despite their widespread adoption, standard CNNs lack explicit mechanisms to handle geometric transformations, resulting in suboptimal clustering performance when image orientations and scales vary. Consequently, conventional CNN-based clustering approaches often necessitate extensive data augmentation strategies to artificially introduce transformation invariance, which increases computational complexity and data dependency Cohen and Welling (2016).

Recent advancements in geometric deep learning have introduced Group Equivariant Convolutions (G-Convs) as a promising alternative to standard convolutional architectures. These specialized networks enable the learning of feature representations inherently equivariant to geometric transformations Weiler and Cesa (2019). Unlike traditional CNNs, which primarily address translation invariance, Group Equivariant Convolutional Networks (G-CNNs) extend this property to rotations, reflections, and other symmetry transformations, rendering them particularly effective for tasks where spatial orientation plays a critical role Worrall et al. (2017). Studies have demonstrated that Group Equivariant Convolutional Networks achieve superior performance across diverse image recognition tasks, particularly in settings with limited training data and significant intra-class variability Bekkers (2019). By incorporating geometric priors directly into the network architecture, these models reduce dependence on explicit data augmentation and enhance robustness to geometric variations. This makes them particularly promising for medical image clustering applications, where anatomical structures frequently appear in multiple orientations Cohen and Welling (2017).

Medical imaging serves as a cornerstone of disease detection and diagnosis; however, variability in imaging conditions—including acquisition parameters, patient positioning, and

anatomical structures—introduces inconsistencies in pathological manifestations within chest X-ray scans. Traditional clustering techniques, which lack geometric equivariance, often fail to accommodate these variations, resulting in suboptimal clustering performance. Although previous studies have explored deep clustering in medical imaging, few have investigated the integration of Group Equivariant Convolutions to address these challenges. This research aims to fill that gap by evaluating whether Group Equivariant Convolutional Networks (G-CNNs) can improve self-supervised clustering performance without relying on extensive augmentation strategies.
Thus, the following research question arises:

Can the integration of geometric deep learning techniques, specifically Group Equivariant Convolutional Networks, improve deep clustering performance in medical imaging compared to conventional CNN-based approaches that rely on data augmentation?

The primary objective of this study is to determine whether incorporating Group Equivariant Convolutions into a deep clustering framework can enhance self-supervised learning for chest X-ray analysis.
Our contributions are:

- Developing a Group Equivariant Convolutional Networks(G-CNNs) based deep clustering framework tailored for chest X-ray analysis.
- Evaluating the effectiveness of the proposed model in comparison to traditional CNN-based clustering approaches.
- Assessing improvements in clustering accuracy, feature consistency, and generalization capacity.
- Validating the proposed approach using the NIH Chest X-ray dataset Wang et al. (2017a) to ensure robustness across diverse imaging conditions.

This report is organized as follows:

- **Related Work:** A comprehensive review of existing literature on self-supervised clustering and geometric deep learning, highlighting limitations of prior studies and justifying the research gap.
- **Background:** A theoretical foundation of self-supervised learning and Group Equivariant Convolutional Networks, establishing the conceptual framework for the proposed approach.
- **Methodology:** A detailed exposition of the model architecture, data preprocessing techniques, clustering methodology, and optimization strategies.
- **Experiments and Results:** Empirical evaluations demonstrating the effectiveness of the proposed model, along with comparisons to traditional clustering methods.
- **Conclusions and Future Work:** A critical discussion of key findings, potential limitations, and promising avenues for future research aimed at further enhancing clustering performance in medical imaging.

By integrating principles from geometric deep learning, this research seeks to significantly improve the accuracy and robustness of self-supervised clustering in medical imaging. The proposed approach provides a more effective solution for disease detection and classification in chest X-rays while reducing reliance on manual data augmentations. Incorporating Group Equivariant Convolutions into deep clustering frameworks has the potential to substantially enhance feature extraction capabilities and improve model generalization across diverse imaging conditionsCohen and Welling (2017, 2016).

Chapter 2

Related Work

This chapter presents a comprehensive review of classical clustering methods, self-supervised learning approaches, and geometric deep learning techniques. It examines their limitations in medical imaging contexts and evaluates the role of advanced deep learning-based approaches in enhancing clustering accuracy. Furthermore, it investigates the significance of Group Equivariant Convolutional Networks (G-CNNs) and their comparative advantages over conventional CNN-based models for clustering tasks in medical image analysis.

Deep clustering techniques have been extensively applied in visual data categorization, including medical imaging applications. Traditional clustering methodologies such as K-means and fuzzy C-means have been widely implemented for medical image segmentation tasks Zanaty and Abdelhafiz (2016). While these classical approaches offer computational efficiency and methodological simplicity, they demonstrate significant limitations when processing high-dimensional data, handling noise, and accommodating variations in anatomical structures Dhanachandra et al. (2015). Eckhardt et al. Eckhardt et al. (2023) highlight the inherent challenges of unsupervised clustering in healthcare contexts, noting that these methods frequently require extensive manual parameter tuning and exhibit pronounced sensitivity to dataset variations. Hybrid methodologies, exemplified by CDBH Mirzaei et al. (2021), attempt to address these limitations by integrating classical clustering algorithms with density-based approaches, thereby enhancing classification accuracy in imbalanced medical datasets.

Recent advancements in self-supervised learning have demonstrated significant improvements in feature extraction from unlabeled data. Contrastive learning frameworks, such as SimCLR and MoCo, have shown remarkable success in natural image domains Chen et al. (2020); He et al. (2020). However, their applicability in medical imaging necessitates domain-specific modifications due to challenges such as data scarcity and variations in pathology presentation. Studies by Taleb et al. Taleb et al. (2021) and Felfeliyan et al. Felfeliyan et al. (2023) introduce self-supervised learning techniques tailored for multimodal medical imaging, while Sun et al. Sun et al. (2021) and Tiu et al. Tiu et al. (2022) highlight the potential of self-supervised models in achieving expert-level performance in disease detection. Despite these advancements, these approaches depend heavily on augmentation strategies to enforce invariance to geometric transformations. Non-contrastive methods like SimSiam offer alternatives to traditional contrastive approaches. Moreover, Chen et al. showed that transformer-based models like DINO and MAE have demonstrated strong performance in image tasks Chen and He (2021). Integrating these methods with geometric deep learning techniques, which are designed to handle complex geometric structures, could further improve image clustering by preserving meaningful relationships within the data.

Several studies have explored deep clustering as a means of integrating representation

learning and clustering in an end-to-end fashion. Caron et al. Caron et al. (2018) propose DeepCluster, an iterative approach that applies K-means clustering to learned feature representations, using cluster assignments as pseudo-labels for self-supervised training. This approach has demonstrated state-of-the-art performance in unsupervised feature learning on large-scale datasets such as ImageNet. Zhan et al. Zhan et al. (2020) extend this concept with Online Deep Clustering (ODC), which stabilizes training by integrating clustering with network updates in real time. Unlike traditional deep clustering methods that alternate between clustering and network updates, ODC employs dynamic memory modules to store sample features and cluster centroids, improving stability and performance in representation learning.

In the context of medical imaging, Haghighi et al. Haghighi et al. (2022) propose DIRA, a self-supervised learning framework that incorporates discriminative, restorative, and adversarial learning objectives to enhance feature extraction from unlabeled medical images, thereby improving classification and segmentation tasks. Deep learning-based clustering has also gained attention in bioinformatics, where complex biological data such as gene expression and protein interactions necessitate advanced analytical techniques. Karim et al. Karim et al. (2021) provide a comprehensive review of deep clustering methods in bioinformatics, discussing their integration with neural networks to enhance clustering accuracy in biological data analysis.

Geometric deep learning offers a promising alternative to conventional CNNs by incorporating group equivariance, which enables networks to maintain transformation consistency beyond simple translations. Unlike traditional CNNs, which primarily exhibit translational invariance, equivariant architectures extend this property to rotations, reflections, and other geometric transformations Cohen and Welling (2016). Research by Weiler and Cesa Weiler and Cesa (2019) and Worrall et al. Worrall et al. (2017) demonstrates that Group Equivariant Convolutional Networks improve robustness to orientation changes in medical imaging datasets. Bekkers Bekkers (2019) further investigates steerable CNNs for biomedical applications, emphasizing their ability to enhance feature extraction in scenarios where anatomical structures exhibit geometric variability.

The use of graph-based and geometric deep learning approaches has been explored in various biomedical applications. Studies by Li et al. Li et al. (2022) and Ding et al. Ding et al. (2022) discuss how graph representation learning can enhance multi-modality medical imaging, particularly in integrating MRI and CT scans. Furthermore, Atz et al. Atz et al. (2021) and Gerken et al. Gerken et al. (2023) highlight the potential of geometric deep learning in molecular sciences, demonstrating its effectiveness in handling structured biomedical data. While traditional CNNs, such as AlexNet Krizhevsky et al. (2012), have been widely employed in medical image classification and segmentation Yadav and Jadhav (2019); Rana and Bhushan (2023), they often struggle with the geometric variations inherent in medical imaging datasets. Conventional CNN-based clustering approaches rely on extensive data augmentation to artificially introduce transformation invariance, increasing computational complexity and data dependency. Integrating equivariant architectures, such as Group Equivariant Convolutional Networks, has emerged as a potential solution to these challenges by directly encoding geometric priors into the network structure Cohen and Welling (2017). This motivates the present study, which seeks to leverage Group Equivariant Convolutional Network for deep clustering in chest X-ray analysis to enhance

clustering performance without extensive reliance on augmentation.
In summary, while classical clustering techniques offer computational efficiency, they are insufficient for handling the complexity and high dimensionality of medical imaging data. Self-supervised learning has made significant strides in feature extraction from unlabeled images, yet existing methods remain heavily dependent on data augmentation to enforce transformation invariance. Geometric deep learning, particularly through the use of Group Equivariant Convolutional Networks (G-CNNs), presents a more principled approach by inherently encoding equivariance to rotations and reflections, reducing reliance on augmentation. Despite these advances, current methods lack a systematic evaluation of Group Equivariant Convolutions in the context of deep clustering for medical imaging. Therefore, this research proposes a deep clustering framework that integrates Group Equivariant Convolutions to improve feature learning and clustering robustness in Chest X-ray analysis.

Chapter 3

Background

3.1 Self-Supervised Learning and Deep Clustering

Self-supervised learning (SSL) is a machine learning approach that learns meaningful data representations without human-labeled annotations by solving pretext tasks, such as contrastive learning, clustering, or reconstruction. The key idea behind Self Supervised Learning (SSL) is to generate pseudo-labels from the data itself, allowing a model to be trained without requiring external supervision. This is particularly useful in domains such as medical imaging, where annotated datasets are scarce and costly to obtain.

One popular approach to self-supervised learning is contrastive learning, as exemplified by SimCLR Chen et al. (2020). In contrastive learning, the model learns to bring similar samples (positive pairs) closer together in the feature space while pushing dissimilar samples (negative pairs) apart. Positive pairs are typically generated through data augmentation techniques such as cropping, flipping, or color distortion. Negative pairs are drawn from unrelated data samples. This process encourages the model to learn invariant representations that generalize well across various transformations.

Another widely used self-supervised approach is deep clustering, where a neural network is trained in conjunction with a clustering algorithm. A well-known example is DeepCluster Caron et al. (2018), which employs k-means clustering on feature embeddings generated by a neural network. The pseudo-labels obtained from clustering are then used to fine-tune the network iteratively. Although powerful, traditional deep clustering techniques struggle with symmetries and variations in medical images, such as NIH chest radiographs. Methods like SwAV Caron et al. (2020) improve on this by introducing online clustering and using prototypes to create more robust feature representations.

Deep clustering has shown significant potential in medical imaging, where large-scale datasets can be used without manual annotation. Techniques such as BYOL Grill et al. (2020) and MoCo He et al. (2020) have demonstrated effectiveness in extracting transferable representations that improve disease classification and segmentation tasks. By integrating deep clustering into medical imaging workflows, researchers can enhance diagnostic models while reducing the burden of human annotation.

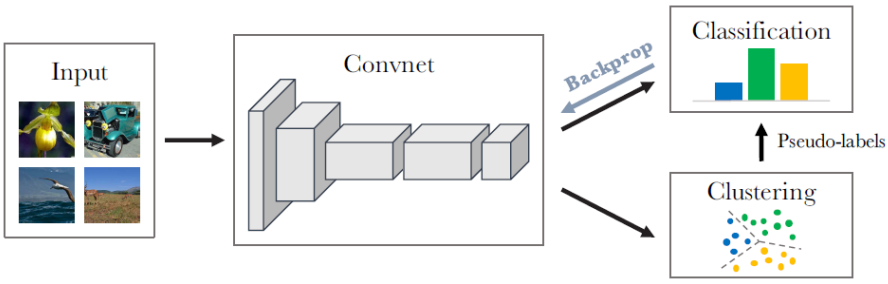

Figure 3.1: Deep Clustering Architecture Caron et al. (2018)

3.2 Group Convolutional Neural Networks (G-CNNs)

Traditional convolutional neural networks (CNNs) are inherently equivariant to translations, making them effective for tasks where spatial invariance is important. However, standard CNNs fail to generalize well to other transformations such as rotations and reflections, which are common in medical imaging. Group Convolutional Neural Networks (G-CNNs) address this limitation by incorporating group convolutions, which generalize the standard convolution operation to a broader set of symmetry transformations Cohen and Welling (2016).
Group Equivariant Convolutional Networks achieve equivariance by convolving inputs over structured groups of transformations, such as $p4$ (translations and rotations) and $p4m$ (translations, rotations, and reflections). This property ensures that a transformation applied to an input image results in a predictable transformation of the output, allowing the model to learn features that are inherently invariant to these transformations Worrall et al. (2017). By sharing weights across multiple orientations, Group Equivariant Convolutional Networks improve generalization and reduce redundancy in the learned representations Weiler and Cesa (2019). By sharing weights across multiple orientations, Group Equivariant Convolutional Networks improve generalization and reduce redundancy in the learned representations Weiler and Cesa (2019). These networks leverage weight-sharing mechanisms to ensure that feature extraction remains consistent across different transformations, thereby enhancing robustness and efficiency.
Figure 3.2 illustrates the $P4M$ kernel and its equivariant feature maps. The $P4M$ kernel extends the $P4$ group by incorporating reflections in addition to 90-degree rotations. The figure demonstrates the hierarchical process of feature extraction using group convolutions. The input image, containing a structured pattern, is first processed under multiple orientations. Through Z2-P4 convolution, the input is mapped to the P4 feature space, where filters capture responses at different rotations and reflections. This results in four feature maps, each corresponding to transformations of the kernel applied to the input and representing a different rotational state. To ensure transformation-invariant representations, the model performs P4-Z2 pooling, which aggregates feature maps across different orientations using an averaging operation. This pooling step ensures that the final output remains consistent, regardless of the rotation or reflection of the input.

The input undergoes Z2-P4 convolution, generating multiple rotated feature maps that preserve rotation and reflection equivariance. The final pooled output ensures transformation consistency across different orientations.
In medical imaging, Group Equivariant Convolutional Networks are particularly beneficial due to the inherent rotational and reflectional symmetries present in datasets like NIH Chest X-rays. Traditional CNNs require extensive data augmentation to learn rotation-invariant features, whereas Group Equivariant Convolutional Networks naturally encode these symmetries within their architecture Bekkers (2019). This leads to improved feature learning, better clustering performance, and reduced computational complexity.
Recent advancements, such as Steerable CNNs Cohen and Welling (2017), further extend Group Equivariant Convolutional Networks by introducing equivariance to continuous transformations. This refinement broadens the applicability of Group Equivariant Convolutional Networks in complex medical datasets, making them a valuable tool for tasks

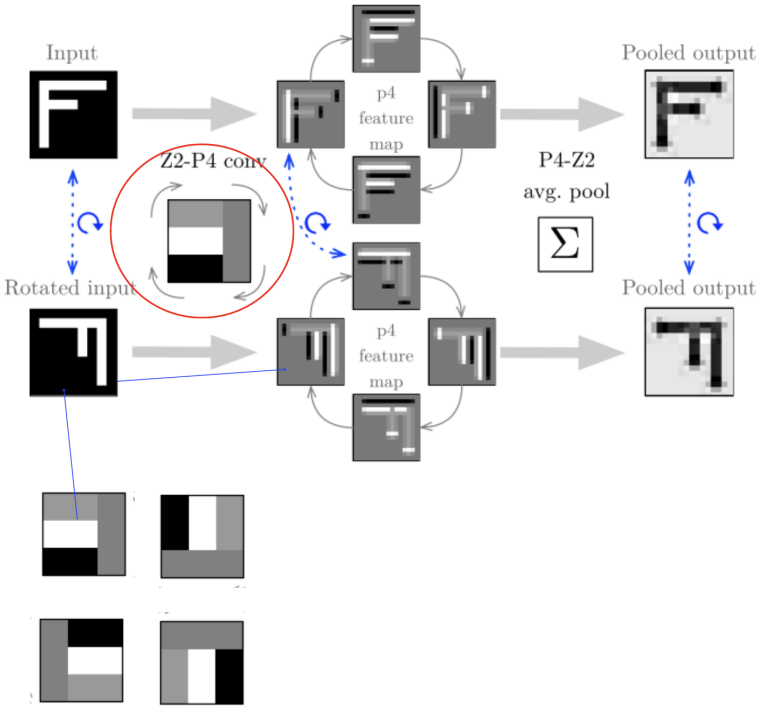

Figure 3.2: Illustration of the P4M kernel and its equivariant feature maps.

such as disease classification, segmentation, and anomaly detection.

3.3 Sobel Filtering for Edge Detection

Sobel filtering is a widely used edge detection technique in image processing that emphasizes regions of high-intensity variation. Since edges often correspond to important structures in images, detecting them helps improve the effectiveness of feature extraction. Sobel filtering enhances high-frequency components in an image, allowing deep learning models to capture key structures while reducing dependency on global brightness variations. This makes it especially useful for medical imaging applications such as chest X-ray analysis, where detecting anatomical boundaries is crucial for disease diagnosis.
Sobel filtering is implemented using two convolutional kernels to approximate the first-order derivatives of an image along the horizontal (x-axis) and vertical (y-axis) directions. The standard Sobel operators are:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

These kernels are convolved with the input image to compute the gradient magnitude, which represents the intensity of edges in the image:

$$G = \sqrt{G_x^2 + G_y^2}$$

where G_x and G_y denote the horizontal and vertical gradient components, respectively. In deep learning, Sobel filtering is often used as a preprocessing step to emphasize texture information and reduce reliance on color features. This technique is particularly valuable in self-supervised learning, as demonstrated in DeepCluster Caron et al. (2018), where Sobel-filtered images improve feature learning by removing color biases and forcing models to focus on structural patterns. In medical imaging, Sobel filtering aids in highlighting anatomical structures such as lung boundaries in chest X-rays, helping radiologists and automated diagnostic systems detect abnormalities like pneumonia or tumors. By enhancing edge-based representations, Sobel filtering improves the interpretability of medical scans, supporting both classification and segmentation tasks Kanopoulos et al. (1988).

3.4 t-SNE for High-Dimensional Data Visualization

t-Distributed Stochastic Neighbor Embedding (t-SNE)Van der Maaten and Hinton (2008) is a non-linear dimensionality reduction technique widely used to visualize high-dimensional data in 2D or 3D space. Unlike Principal Component Analysis (PCA), which preserves global structure, t-SNE focuses on preserving local neighborhood relationships, making it particularly effective for analyzing deep clustering embeddings. By maintaining local similarities, t-SNE provides an intuitive way to explore whether feature representations form distinct clusters, offering qualitative insights into the learned structures of deep models.

t-SNE models the similarity between high-dimensional data points using a Gaussian distribution and their low-dimensional representations using a Student's t-distribution to mitigate the crowding problem. The algorithm consists of the following steps:

1. Compute pairwise similarities in high-dimensional space using a Gaussian distribution:

$$P_{ij} = \frac{P_{j|i} + P_{i|j}}{2n}$$

where

$$P_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}$$

where P_{ji} represents the probability of points x_i and x_j being neighbors in high-dimensional space.

2. Compute pairwise similarities in the low-dimensional embedding space using a Student's t-distribution:

$$Q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq i}(1 + \|y_i - y_k\|^2)^{-1}}$$

where Q_{ij} represents the probability of points y_i and y_j being neighbors in the lower-dimensional representation.

3. Optimize the embedding by minimizing the Kullback-Leibler (KL) divergence between the two distributions:

$$C = \sum_{i \neq j} P_{ij} \log \frac{P_{ij}}{Q_{ij}}$$

The optimization process adjusts the positions of points in the low-dimensional space to maintain local similarities while reducing distortions in global structure.
t-SNE is commonly used to analyze clustering performance by visualizing the learned feature representations in deep models. It provides an intuitive way to explore whether the embeddings form well-separated clusters, offering a qualitative validation of deep clustering techniques. In deep learning research, t-SNE is widely applied to self-supervised learning, where it helps visualize how different classes emerge in the latent space without explicit labels Caron et al. (2018); Arora et al. (2018). In medical imaging, t-SNE has been used to explore disease phenotypes by projecting complex multi-modal imaging data into interpretable 2D spaces, facilitating the identification of hidden patterns in radiological scans. For instance, it has been applied to analyze lung disease progression in chest X-ray datasets, allowing researchers to interpret how feature embeddings relate to different pathology types.

3.5 Evaluation Metrics for Deep Clustering

To assess the effectiveness of deep clustering techniques, including those based on Group Equivariant Convolutional Networks, several evaluation metrics are commonly employed. This section provides a structured explanation of key evaluation metrics used in DeepCluster and related approaches.

Normalized Mutual Information (NMI)

Normalized Mutual Information (NMI) evaluates the mutual dependence between predicted clusters and ground truth labels. It is defined as:

$$NMI = \frac{2I(Y, C)}{H(Y) + H(C)} \tag{3.1}$$

where $I(Y, C)$ represents the mutual information between the true labels Y and the predicted clusters C, while $H(Y)$ and $H(C)$ denote their respective entropies. NMI ranges from 0 to 1, with higher values indicating better clustering alignment with ground truth labels Caron et al. (2018).
This metric is particularly useful in deep clustering frameworks like DeepCluster, where the quality of unsupervised cluster assignments is indirectly evaluated by their correlation with existing semantic categories. Two variations of NMI are utilized in this study:

- **NMI between cluster assignments and ground truth labels:** This measures the extent to which the discovered clusters correspond to known class labels (e.g., disease categories in chest X-ray images). A high NMI score suggests that the clustering captures meaningful structure in the data.

- **NMI between consecutive cluster assignments:** Since deep clustering involves iteratively refining feature representations, monitoring NMI across successive epochs helps assess the stability of cluster assignments. A consistently high NMI indicates that the model is learning stable and coherent representations over time.

The use of NMI thus provides both an external validation of clustering quality (against known labels) and an internal measure of training stability, making it a crucial metric in evaluating deep clustering performance.

Linear Probing Accuracy

Linear probing accuracy assesses the quality of learned feature representations by training a linear classifier (e.g., logistic regression) on frozen features extracted from a self-supervised model. Given input features X and corresponding labels Y, the classifier f is trained using a supervised loss, and its accuracy is computed as:

$$ACC_{LP} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{f(X_i) = Y_i\} \tag{3.2}$$

where ACC_{LP} represents the linear probing accuracy, and $\mathbb{1}\{\cdot\}$ is the indicator function. Higher accuracy values indicate that the learned representations are more transferable and meaningful Caron et al. (2018).
Linear probing is essential because deep clustering models are not explicitly trained for classification; instead, they learn to structure the data in ways that may or may not align with supervised labels. A high linear probing accuracy suggests that the clustering process has successfully organized features into a space where semantic distinctions are well-preserved, making them useful for downstream supervised tasks. This metric is particularly relevant when comparing conventional CNN-based clustering with Group Equivariant Convolutional Networks (G-Convs), as it provides insight into whether transformation-equivariant architectures lead to more structured and generalizable representations.
By combining NMI (to assess clustering quality and stability) and linear probing accuracy (to evaluate feature transferability), this study ensures a comprehensive evaluation of deep clustering methods. These metrics provide complementary insights: NMI examines the structure of the learned clusters, while linear probing assesses the usefulness of the extracted features in real-world classification tasks. Together, they enable a systematic comparison of traditional deep clustering against equivariant architectures, ensuring that Group Equivariant Convolutional Networks (G-Convs) not only form robust clusters but also produce meaningful feature embeddings for downstream learning tasks.

Chapter 4

Methodology and Approach

In this study, we reproduce the DeepCluster framework for self-supervised feature learning, adapting it for medical image analysis by training models on medical images instead of natural images. The DeepCluster pipeline follows an iterative learning strategy consisting of four key stages: (1) feature extraction using a deep convolutional network, (2) dimensionality reduction and clustering, (3) pseudo-label assignment, and (4) self-training the CNN using these pseudo-labels. This process is repeated iteratively to refine the learned feature representations over multiple clustering cycles. We implement this framework using two different architectures: a AlexNet, which applies Sobel filtering to enhance edge structures, and a P4M-equivariant CNN, which replaces explicit augmentation with group-equivariant convolutions to encode transformation consistency. A detailed explanation of the P4M-equivariant CNN groups is provided in section 3.2.

Dataset and Preprocessing

Initially, we explored the use of publicly available brain tumor datasets. We found three datasets from different sources: the Brain Tumor MRI Dataset containing 7,023 images classified into four categories Nickparvar (2021), the Brain Tumor Dataset by Jun Cheng with 3,064 images categorized into meningioma, glioma, and pituitary tumors Cheng (2017), and the MRI Brain Tumor Glioma Dataset comprising 9,832 MRI images Khan (2023). Due to the limited size of each dataset, we merged them for training and converted all RGB images to grayscale to maintain consistency. However, after training our model using our proposed method, we observed that the learned representations were clustering images based on the shape of the brain rather than tumor morphology, indicating a strong bias toward structural rather than pathological features.

For this study, we selected the publicly available NIH ChestX-ray dataset, introduced by Wang et al. Wang et al. (2017b). It consists of 112,120 frontal-view chest X-ray images from 30,805 patients, covering 14 different thoracic diseases, including pneumonia, cardiomegaly, and atelectasis. The dataset was collected from clinical sources and includes both normal and abnormal cases, with multiple disease labels per image. Given these variations, preprocessing is required to standardize the input format for deep clustering.

All images are resized to 64×64 pixels using bilinear interpolation and normalized to a pixel intensity range of [-1,1]. This normalization reduces intensity-based biases and stabilizes model training. The baseline AlexNet-based model applies Sobel filtering to enhance structural edge features. The Sobel operator computes first-order gradients along the horizontal and vertical directions, producing a two-channel representation where each channel captures edge intensities in different orientations. This transformation enhances the visibility of anatomical boundaries and reduces reliance on raw pixel intensities.

To handle large-scale training efficiently, we use DistributedDataParallel (DDP) for multi-

GPU processing. The dataset is loaded in a distributed manner across multiple GPUs using PyTorch's DataLoader, optimizing memory usage and reducing I/O bottlenecks. Prefetching and memory pinning techniques are employed to minimize data transfer overhead, ensuring smooth and scalable training.

Model Architecture

To extract meaningful representations for deep clustering, we evaluate two convolutional neural network (CNN) architectures: (1) a **AlexNet** Krizhevsky et al. (2012), which incorporates Sobel filtering for edge-based feature extraction, and (2) a **P4M-equivariant CNN** Cohen and Welling (2016), which leverages group-equivariant convolutions to achieve inherent robustness to geometric transformations. Both models follow a similar deep clustering pipeline, where feature embeddings are extracted, clustered using K-means, and iteratively refined through self-supervised learning. However, their architectural differences significantly impact how features are learned, particularly in handling variations in image orientations and structural patterns.
We selected AlexNet as our baseline model for its established efficacy in self-supervised learning contexts, particularly within the deep clustering framework Caron et al. (2018). Its hierarchical architecture effectively captures multi-scale features without requiring data annotations, while its computational efficiency—stemming from its relatively shallow structure compared to modern networks—facilitates large-scale training in multi-GPU environments. This benchmark selection enables systematic evaluation of group-equivariant convolutions' benefits when incorporated into the P4M model framework.
The AlexNet model used to process two-channel Sobel-filtered images instead of raw intensity images. The architecture consists of five convolutional layers interleaved with max-pooling layers, followed by a fully connected classifier. The final classification layer is replaced with a linear mapping to cluster assignments, enabling the model to learn discriminative representations suitable for unsupervised clustering. Figure 4.1 illustrates the architecture of the AlexNet-based model.

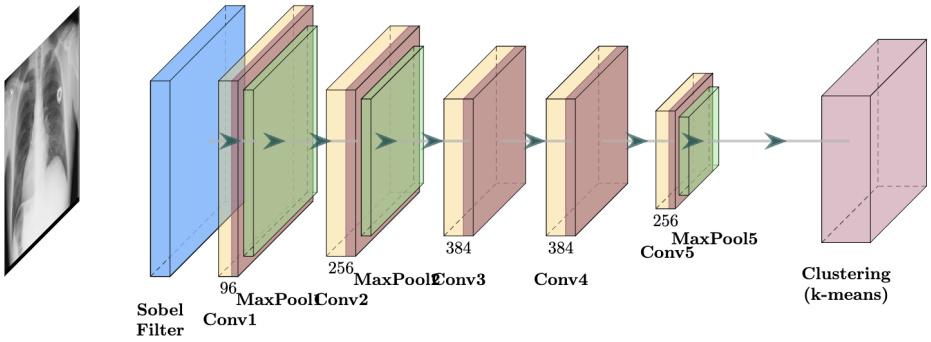

Figure 4.1: Architecture of the AlexNet with Sobel filtering.

For the AlexNet model, five random rotations within ±360° are applied as a data augmentation strategy to improve robustness to positional variations in chest X-ray images. However, the P4M-equivariant CNN eliminates the need for augmentation by employing

group-equivariant convolutions, which inherently encode rotational and reflectional invariance. By leveraging these properties, the equivariant model learns features that remain consistent across different orientations without requiring explicit transformations.
To address the limitations of standard CNNs in handling geometric transformations, the P4M-equivariant CNN employs group-equivariant convolutions (G-Convs). These convolutions operate within the P4M symmetry group, comprising four discrete rotations $(0°, 90°, 180°, 270°)$ and mirror reflections. The network consists of three convolutional blocks with equivariant convolutions, batch normalization, ReLU activation, and anti-aliasing pooling to ensure spatial coherence. A group pooling layer aggregates features across transformation groups, preserving geometric consistency in feature representations. The architecture is illustrated in Figure 4.2.

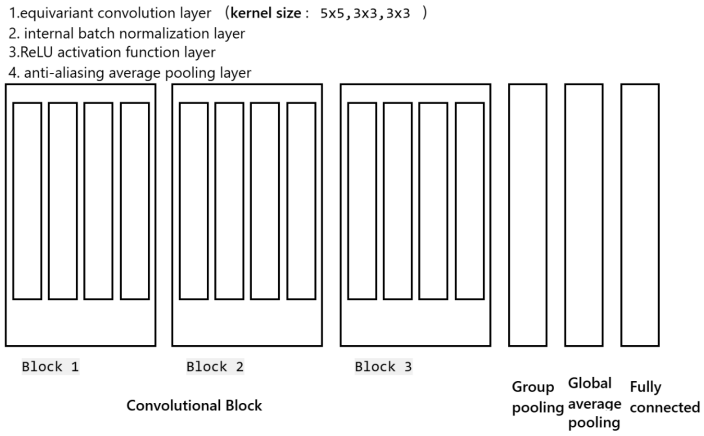

Figure 4.2: Architecture of the P4M-equivariant CNN.

Unlike AlexNet, which relies on explicit augmentation, the P4M model achieves robustness to geometric transformations intrinsically through equivariant convolutions. Table 4.1 summarizes the key differences between the two architectures.

Table 4.1: Comparison of AlexNet and P4M-equivariant CNN.

Feature	**AlexNet**	**P4M-equivariant CNN**
Input Data	Sobel-filtered (2-channel)	Grayscale (1-channel)
Convolution Type	Standard CNN	Equivariant CNN (G-Convs)
Transformation Handling	Requires data augmentation	Intrinsic equivariance
Number of Convolutional Blocks	5	3
Pooling Strategy	Max pooling	Anti-aliasing + group pooling
Fully Connected Layers	3 (including output)	1 (output layer)
Computational Complexity	Higher (data augmentation needed)	Lower (built-in invariance)

Feature Extraction and Clustering Pipeline

Once features are extracted from the CNN, they undergo preprocessing before clustering to ensure that the feature space is compact, normalized, and suitable for clustering. This involves three key transformations: Principal Component Analysis (PCA), whitening, and L2 normalization, followed by K-means clustering to generate pseudo-labels. The primary goal of this preprocessing stage is to transform raw feature vectors into a more compact and normalized representation while preserving essential variance.
To reduce the dimensionality of the feature space, we apply PCA, which projects the high-dimensional data onto a lower-dimensional subspace while retaining maximum variance. Given a set of feature vectors $\mathbf{X} \in \mathbb{R}^{N \times D}$, where N is the number of samples and D is the original feature dimension, PCA computes the eigenvectors of the covariance matrix:

$$\mathbf{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top \tag{4.1}$$

where $\bar{\mathbf{x}}$ is the mean feature vector. The data is then projected onto the top d eigenvectors corresponding to the largest eigenvalues:

$$\mathbf{Z} = \mathbf{X}\mathbf{W} \tag{4.2}$$

where $\mathbf{W} \in \mathbb{R}^{D \times d}$ contains the top d eigenvectors. This transformation ensures that the feature space is reduced to a more compact form while maintaining essential structure. To prevent certain dimensions from dominating due to varying scales, we apply whitening, which decorrelates features and equalizes their variance. Whitening transforms the data such that each feature has unit variance, improving clustering performance by making feature distributions more uniform. This is achieved by scaling the principal components using the eigenvalues $\mathbf{\Lambda}$ from PCA:

$$\mathbf{Z}_{\text{whitened}} = \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{U}^\top \mathbf{Z} \tag{4.3}$$

where $\mathbf{\Lambda}^{-\frac{1}{2}}$ scales each principal component inversely to its variance. This ensures that all dimensions contribute equally to clustering, preventing bias toward features with larger variances.
Following whitening, we apply L2 normalization to rescale each feature vector $\mathbf{z}_i$ to have unit norm:

$$\hat{\mathbf{z}}_i = \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2}, \quad \text{where} \quad \|\mathbf{z}_i\|_2 = \sqrt{\sum_{j=1}^{d} z_{ij}^2} \tag{4.4}$$

where $\|\mathbf{z}_i\|_2 = \sqrt{\sum_{j=1}^{d} z_{ij}^2}$ ensures that all feature vectors lie on a unit hypersphere. This normalization ensures that all feature vectors lie on a unit hypersphere, making clustering results more stable and distance-based comparisons more meaningful.
After preprocessing, clustering is performed using the K-means algorithm, which partitions the feature space into K clusters based on feature similarity. K-means clustering aims to minimize intra-cluster variance by iteratively assigning feature vectors to their nearest

centroids and updating the centroids accordingly. Given a dataset $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N\}$, the objective is to minimize:

$$\mathcal{L}_{\text{k-means}} = \sum_{i=1}^{N} \|\mathbf{z}_i - \mu_{c_i}\|^2 \tag{4.5}$$

where μ_{c_i} represents the centroid of the cluster to which $\mathbf{z}_i$ is assigned.

The K-means algorithm is formulated as follows:

Algorithm 1 K-means Clustering Algorithm

1: **Input:** Feature vectors $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^N \in \mathbb{R}^{N \times d}$, number of clusters K, max iterations T, tolerance ε
2: **Initialize:** Select initial centroids $\mu_k^{(0)}$ randomly from $\mathbf{Z}$ for $k = 1, ..., K$ $t = 1$ to T
3: **Assignment Step:** each feature vector $\mathbf{z}_i$
4: Assign to nearest centroid:

$$c_i^{(t)} = \arg\min_k \|\mathbf{z}_i - \mu_k^{(t-1)}\|^2$$

5: **Update Step:** each cluster k
6: Compute new centroid:

$$\mu_k^{(t)} = \frac{1}{|C_k^{(t)}|} \sum_{\mathbf{z}_i \in C_k^{(t)}} \mathbf{z}_i$$

7: **Convergence Check:** $\sum_{k=1}^K \|\mu_k^{(t)} - \mu_k^{(t-1)}\|^2 < \varepsilon$
8: **Break**
9: **Output:** Cluster assignments $\{c_i\}_{i=1}^N$ and centroids $\{\mu_k\}_{k=1}^K$

By iteratively refining the cluster assignments, the model progressively learns structured feature representations that better separate distinct patterns within the data. The next section describes how these cluster assignments serve as pseudo-labels for self-training, further enhancing the feature learning process.

Self-Training with Pseudo-Labels

The pseudo-labels obtained from K-means clustering are used to retrain the CNN, reinforcing feature representations that are consistent across similar images. This iterative process progressively improves clustering performance by refining the network's ability to differentiate meaningful patterns in the data. During training, the model predicts a probability distribution over the K clusters, and the network is optimized using the cross-entropy loss, which measures the discrepancy between the predicted probability distribution and the assigned pseudo-labels.

Given a batch of N samples with feature representations $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^N$ and corresponding pseudo-labels $\mathbf{y} = \{y_i\}_{i=1}^N$, where each $y_i \in \{1, ..., K\}$ represents the cluster assignment for the i^{th} sample, the model produces a predicted probability distribution $\hat{\mathbf{p}}_i$ using a softmax activation:

$$\hat{p}_{i,k} = \frac{\exp(f_k(\mathbf{z}_i))}{\sum_{j=1}^{K} \exp(f_j(\mathbf{z}_i))} \tag{4.6}$$

where $f_k(\mathbf{z}_i)$ is the network's output for cluster k before softmax normalization.
The cross-entropy loss is then computed as:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{1}[y_i = k] \log \hat{p}_{i,k} \tag{4.7}$$

where $\mathbb{1}[y_i = k]$ is an indicator function that equals 1 if the true pseudo-label y_i corresponds to cluster k, and 0 otherwise. This loss function encourages the model to assign high confidence to the correct pseudo-labels while minimizing incorrect assignments.
By iteratively minimizing the cross-entropy loss, the model refines its feature representations, leading to progressively more structured cluster formations in the learned embedding space.

Training and Optimization

Both models were trained using the Adam optimizer with an initial learning rate of 1×10^{-3} and weight decay of 1×10^{-4}. Learning rate optimization was implemented through a ReduceLROnPlateau scheduler, which adaptively decreases the learning rate upon detection of loss plateaus. To maximize computational efficiency while minimizing memory requirements, automatic mixed precision (AMP) was integrated into the training pipeline. Scalability was ensured through multi-GPU parallelization using the DistributedDataParallel (DDP) framework.
The training protocol consisted of 100 epochs with a fixed batch size of 256. We choose $K = 200$ as the number of cluster, balancing cluster granularity and representation purity, considering that the dataset contains 14 disease labels. The iterative clustering procedure continued across multiple epochs until convergence criteria were satisfied, with regular evaluation intervals to assess both clustering stability and the discriminative quality of the learned feature representations. Comprehensive experimental results and performance evaluations based on established clustering metrics and feature quality assessments are presented in the subsequent section.

Chapter 5

Experiments and Evaluation

The main goal of these experiments is to evaluate and compare the effectiveness of the proposed method or model in solving specific problems. The experiments aim to assess the capability of the self-supervised learning model to extract meaningful features from medical images without labeled data, while testing the effectiveness of enhancing feature learning through the incorporation of Group Convolutional Neural Networks (GCNNs), particularly in tasks requiring invariance to transformations such as rotations.

5.1 Data Exploration Analysis

Due to our data being in image format, particularly chest images, it becomes essential to explore texture features associated with different diseases.

We focused on analyzing several key texture feature distributions, namely contrast distribution, homogeneity distribution, energy distribution, and correlation distribution. As shown in the figure5.1, this is the texture feature distribution of Atelectasis disease. To gain deeper insights, we selected three typical diseases for detailed study, using chest images of individuals without disease as a control group.

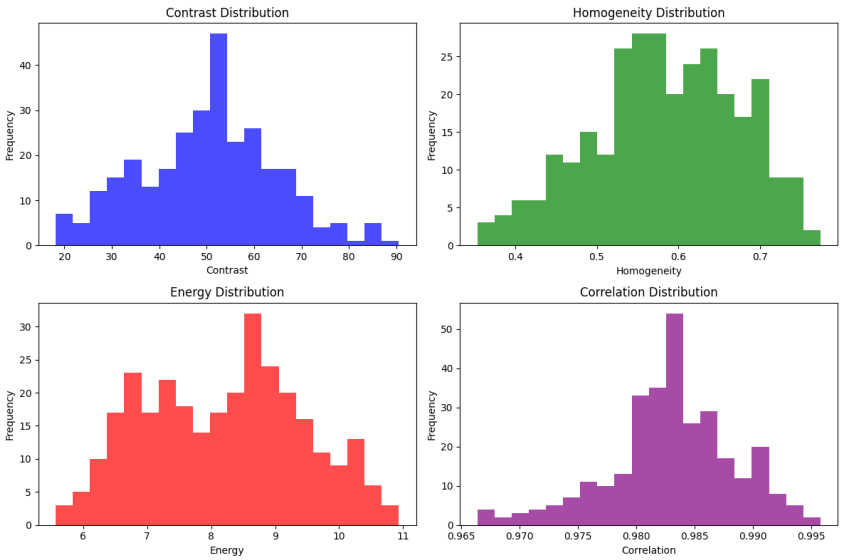

Figure 5.1: Texture feature distribution of Atelectasis disease

Through meticulous analysis, we discovered significant differences in contrast distribution across images of different diseases. As shown in the figure5.2, chest images with diseases show contrast concentrated in low-value regions, while images without diseases

have a more uniform contrast distribution across the entire image. The notable differences in contrast distribution between various diseases suggest that this feature could potentially serve as a powerful discriminative factor in our image clustering task.

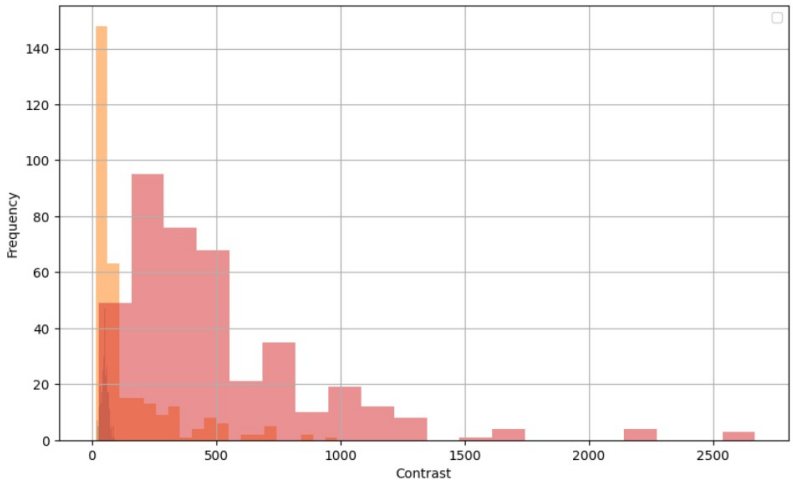

Figure 5.2: Contrast distribution across images of different diseases

Based on these findings, we decided to apply Sobel filtering to image processing. This preprocessing step is expected to further highlight discriminative features, enhance the quality of subsequent analysis, and establish a more solid foundation for image clustering algorithms based on self-supervised learning.

Sobel filtering was applied to emphasize edge features in the chest X-ray images. This process enhances the model's ability to extract structural information, which is crucial for identifying medical features such as boundaries and textures. Figure 5.3 illustrates the effect of Sobel filtering, comparing the original chest X-ray image with its Sobel-filtered counterpart. The filtered image highlights significant edges and boundaries, aiding the model in learning discriminative features during training.

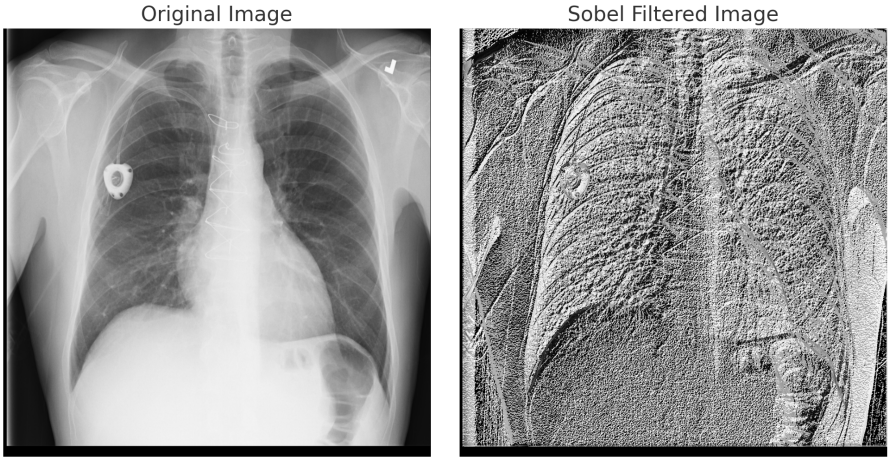

Figure 5.3: Effect of Sobel filtering on chest X-ray images. The original image (left) and its Sobel-filtered version (right).

5.2 Feature Embedding Projection Using t-SNE

We used t-distributed Stochastic Neighbor Embedding (t-SNE)Van der Maaten and Hinton (2008) to visualize the feature distributions extracted from two different training models. t-SNE is a powerful dimensionality reduction technique that can map high-dimensional feature vectors to low-dimensional space (typically 2D or 3D) while preserving local data structure, facilitating intuitive observation of feature distribution patterns.

We used the same dataset of chest images with different chest diseases and individuals without diseases for both training models. After training the two models, we extracted features from the images in the dataset and then applied t-SNE to project these high-dimensional features into 3D space for visualization in the embedding projector Smilkov et al. (2016).

The t-SNE visualization in the embedding projector revealed significant differences in feature distributions between the two models. As shown in the figure5.4,the feature points of the first model were largely clustered together, without any clear cluster separation. This suggests that the features extracted by this model lacked sufficient discriminative power to effectively differentiate between different data classes.

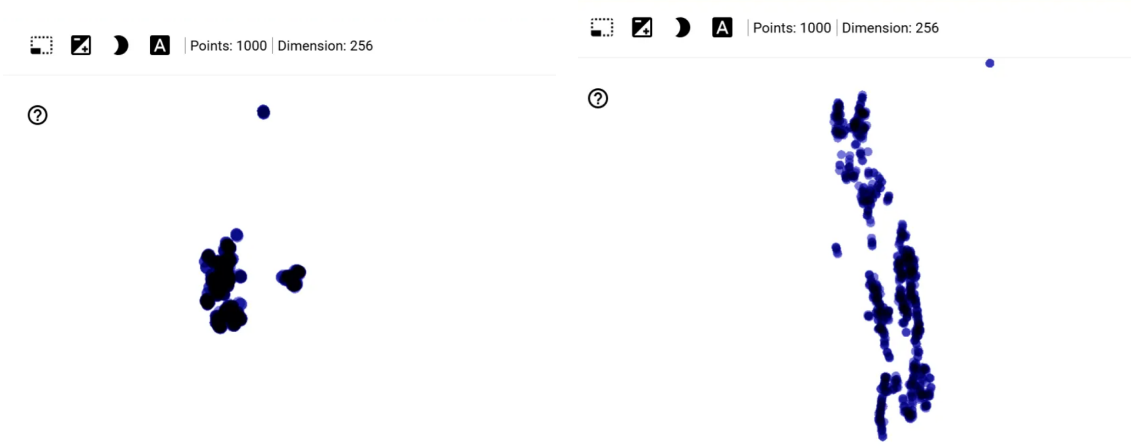

Figure 5.4: t-SNE visualization of AlexNet model(left) and GCNN model(right)

In contrast, the feature points of the second model were divided into multiple distinct clusters,as shown in the figure5.4,. These clusters represent the discriminative features extracted by the model, indicating that the second model was better able to capture and distinguish the features of different data classes, resulting in a more separated feature space.

These results suggest that the second model performed better in extracting features with high discriminative capability. This implies that the second model may exhibit superior performance in subsequent classification or clustering tasks, as it can better separate samples from different classes.

5.3 Results of Experiment

The performance of the model was evaluated over 20 epochs, and the metrics tracked included the Average Loss and Normalized Mutual Information (NMI).

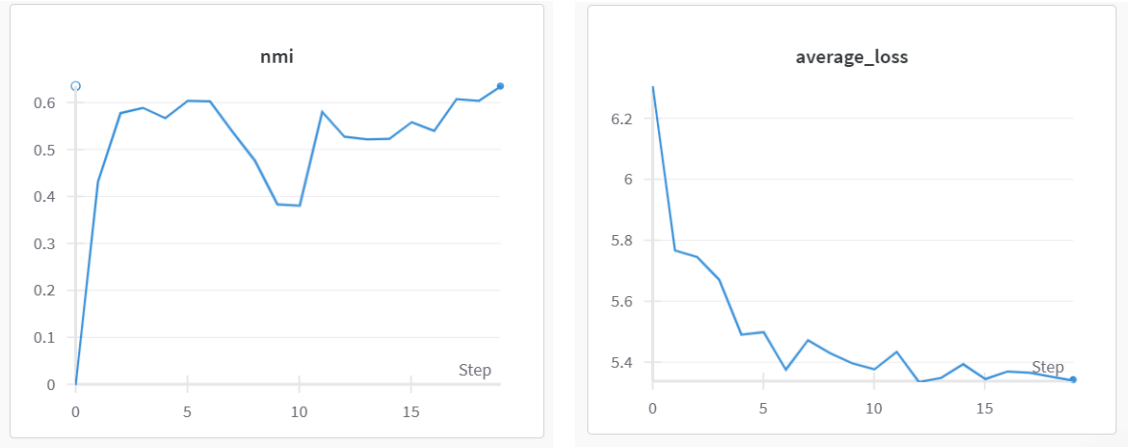

Figure 5.5: Training metrics for the AlexNet model over 20 epochs, showing the Average Loss and NMI scores

For the AlexNet model, as shown in Figure 5.5, the NMI score increases steadily over the epochs, indicating that the clustering quality improves as training progresses. Both of these models learn to create clusters that align more closely with the underlying structure of the data, reflecting the effectiveness of iterative training and pseudo-labeling in enhancing the model's clustering performance. The initial epochs show a rapid increase in NMI, demonstrating that the model quickly learns meaningful cluster structures, while later epochs exhibit smaller but consistent improvements, culminating in a final NMI score of approximately 0.6, signifying good clustering performance. Concurrently, the Average Loss consistently decreases over the epochs, indicating effective optimization of the model during training. By the 20th epoch, the loss stabilizes, suggesting that the model has converged and effectively learned to minimize the reconstruction or clustering error.

While the AlexNet model demonstrated progressive improvements over training, the P4M-equivariant CNN exhibited a notably different behavior. The results for the P4M model, shown in Figure 5.6, reveal that its NMI score stabilizes much earlier in training, indicating that the model converges quickly. This suggests that the inherent equivariance of the P4M model enables it to learn more stable representations within fewer training iterations, reducing the reliance on augmentation-based regularization.

As observed in Figure 5.6, the NMI curve for the P4M model remains stable throughout training, indicating that the model does not experience drastic fluctuations in clustering assignments. This suggests that the learned feature representations are already robust to transformations, reducing the need for iterative refinement through pseudo-labeling. In contrast to the gradual improvements seen in the AlexNet model, the P4M model converges within the first few epochs, highlighting its stability and efficiency in feature learning. Additionally, the Average Loss curve confirms that the P4M model effectively minimizes

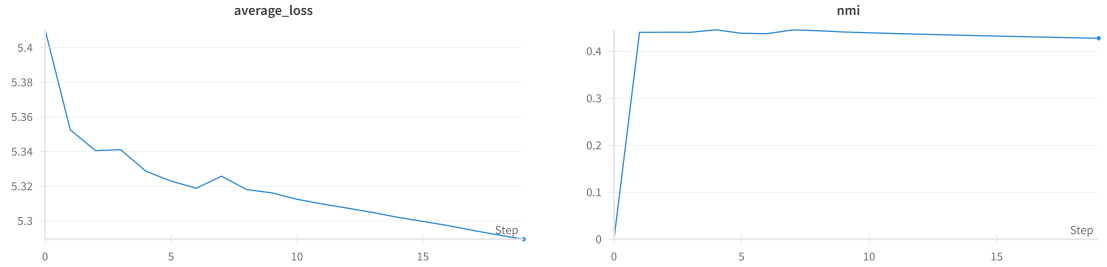

Figure 5.6: Training metrics for the P4M model over 20 epochs, showing the Average Loss and NMI scores

the clustering loss early in training, reinforcing its capability to extract consistent and meaningful representations.

Table 5.1: Model Accuracy Comparison on the ChestX-ray Dataset

Model	**Accuracy**
AlexNet-based Model	0.5640
P4M-equivariant CNN	0.5673

As shown in Table 5.1, both models achieve similar accuracy scores, with the P4M-equivariant CNN slightly outperforming the AlexNet-based model. The relatively low accuracy can be attributed to several factors: (1) an insufficient number of training epochs, which may have prevented the models from fully converging; (2) label noise in the ChestX-ray14 dataset, where multiple conditions are often labeled for a single image; and (3) inherent model limitations, particularly in feature extraction from high-variance medical imaging data.

To further evaluate the classification performance of the learned features, we assess the Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) for the AlexNet model. Figure 5.7 shows the micro-average ROC curve, which summarizes the model's ability to distinguish between positive and negative cases across multiple disease labels. The AUC score of 0.94 indicates that the model's feature representations are effective for classification, even though clustering accuracy remains relatively low. This suggests that while the model successfully learns useful features, the clustering process may require further refinement to improve label alignment.

Overall, the results demonstrate that while AlexNet benefits from iterative refinement through pseudo-labeling and augmentation, the P4M model's built-in transformation equivariance enables it to reach stable clustering performance more efficiently. This highlights the advantage of leveraging geometric deep learning techniques for medical image clustering, as they inherently preserve structural consistency without requiring extensive augmentation strategies.

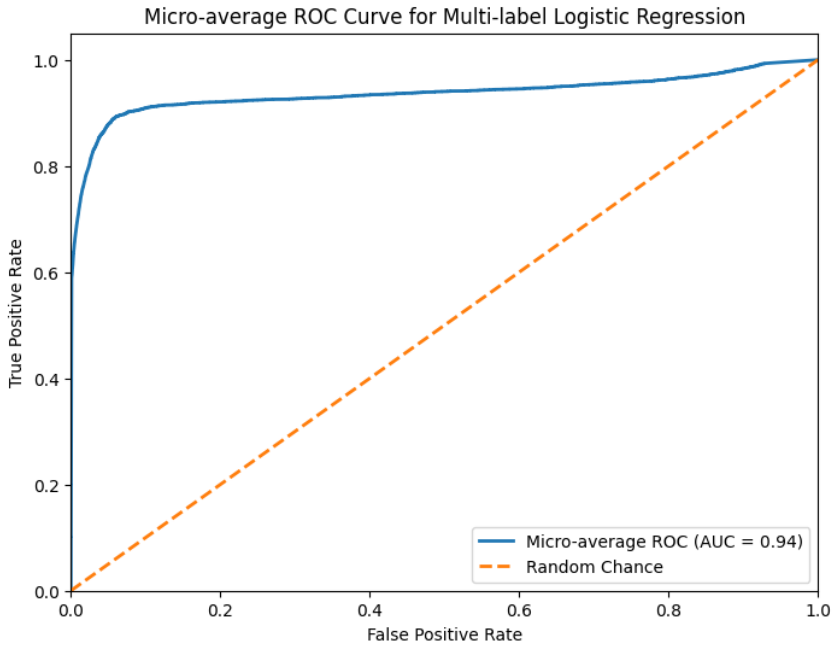

Figure 5.7: Micro-average ROC curve for the AlexNet model on the ChestX-ray dataset.

Chapter 6

Conclusion and Perspectives

In this report, we developed a robust pipeline inspired by DeepCluster to address the challenges of self-supervised learning for image clustering. Using the NIH Chest X-ray dataset, we trained a custom convolutional neural network (CNN) to extract meaningful features through iterative pseudo-labeling and clustering. The evaluation results showed a steady increase in Normalized Mutual Information (NMI) and a consistent decrease in Average Loss, highlighting the effectiveness of our model in learning meaningful and semantically coherent representations. Additionally, instance-level image retrieval results further demonstrated the discriminative power of the learned features, showing that the model could group similar images based on their intrinsic characteristics.

To enhance the model's generalization ability and reduce the reliance on large amounts of labeled data, we incorporated recent advancements in geometric deep learning, particularly Group Equivariant Convolutions (G-Convs). Unlike traditional CNNs, G-Convs enable the network to automatically capture features invariant to transformations like rotation and reflection by introducing geometric symmetries into the learning process. This approach reduced the need for extensive data augmentation and significantly improved the model's robustness to geometric transformations in medical imaging data. By training a model that included G-Convs, we enhanced the feature learning capabilities, particularly in handling images with different orientations and perspectives.

Despite the promising results, there are still some challenges. One of the main limitations is the scalability of k-means clustering when handling very large datasets, especially when the data is complex and contains many categories. Additionally, the model's reliance on the quality of pseudo-labels during iterative training may limit the stability and reliability of the final results. To address these issues, future work could explore more scalable clustering techniques, such as hierarchical clustering or graph-based methods, which may perform better on large datasets. Additionally, integrating more robust pseudo-labeling strategies, such as Generative Adversarial Networks (GANs) in self-supervised learning or more sophisticated label generation mechanisms, could improve the model's adaptability and performance. Moreover, incorporating additional modalities, such as multi-view or multi-label learning, may further enhance the model's generalization ability and adaptability to complex datasets.

Finally, this work was made possible through effective teamwork and collaboration. Each member contributed their expertise, whether in designing the model architecture, implementing the training pipeline, or analyzing the results.

Bibliography

Arora, S., Cohen, N., Golowich, N., and Hu, W. (2018). A convergence analysis of gradient descent for deep linear neural networks. arXiv preprint arXiv:1810.02281. (Cited on page 11.)

Asano, Y. and et al. (2020). Self-labelling method integrating clustering and representation learning. Retrieved from [Your source here]. (Cited on page 1.)

Atz, K., Grisoni, F., and Schneider, G. (2021). Geometric deep learning on molecular representations. Nature Machine Intelligence, 3(12):1023–1032. (Cited on page 5.)

Bekkers, E. J. (2019). B-spline cnns on lie groups. Advances in neural information processing systems, 32:74–85. (Cited on pages 1, 5 and 8.)

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2018). Deep clustering for unsupervised learning of visual features. European conference on computer vision, pages 132–149. (Cited on pages 5, 7, 10, 11, 12 and 14.)

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. NeurIPS. (Cited on page 7.)

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. International conference on machine learning, pages 1597–1607. (Cited on pages 1, 4 and 7.)

Chen, X. and He, K. (2021). Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 15750–15758. (Cited on page 4.)

Cheng, J. (2017). brain tumor dataset. (Cited on page 13.)

Cohen, T. S. and Welling, M. (2016). Group equivariant convolutional networks. International Conference on Machine Learning. (Cited on pages 1, 3, 5, 8, 14 and 29.)

Cohen, T. S. and Welling, M. (2017). Steerable cnns. International Conference on Learning Representations. (Cited on pages 1, 3, 5 and 8.)

Dhanachandra, N., Manglem, K., and Chanu, Y. J. (2015). Image segmentation using k-means clustering algorithm and subtractive clustering algorithm. Procedia Computer Science, 54:764–771. (Cited on page 4.)

Ding, K., Zhou, M., Wang, Z., Liu, Q., Arnold, C. W., Zhang, S., and Metaxas, D. N. (2022). Graph convolutional networks for multi-modality medical imaging: Methods, architectures, and clinical applications. arXiv preprint arXiv:2202.08916. (Cited on page 5.)

Eckhardt, C. M., Madjarova, S. J., Williams, R. J., Ollivier, M., Karlsson, J., Pareek, A., and Nwachukwu, B. U. (2023). Unsupervised machine learning methods and emerging applications in healthcare. Knee Surgery, Sports Traumatology, Arthroscopy, 31(2):376–381. (Cited on page 4.)

Felfeliyan, B., Forkert, N. D., Hareendranathan, A., Cornel, D., Zhou, Y., Kuntze, G., Jaremko, J. L., and Ronsky, J. L. (2023). Self-supervised-rcnn for medical image segmentation with limited data annotation. Computerized Medical Imaging and Graphics, 109:102297. (Cited on page 4.)

Gerken, J. E., Aronsson, J., Carlsson, O., Linander, H., Ohlsson, F., Petersson, C., and Persson, D. (2023). Geometric deep learning and equivariant neural networks. Artificial Intelligence Review, 56(12):14605–14662. (Cited on page 5.)

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Avila, P., Piot, B., Valko, M., et al. (2020). Bootstrap your own latent: A new approach to self-supervised learning. NeurIPS. (Cited on page 7.)

Haghighi, F., Taher, M. R. H., Gotway, M. B., and Liang, J. (2022). Dira: Discriminative, restorative, and adversarial learning for self-supervised medical image analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 20824–20834. (Cited on page 5.)

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. IEEE/CVF Conference on Computer Vision and Pattern Recognition. (Cited on pages 4 and 7.)

Kanopoulos, N., Vasanthavada, N., and Baker, R. L. (1988). Design of an image edge detection filter using the sobel operator. IEEE Journal of solid-state circuits, 23(2):358–367. (Cited on page 10.)

Karim, M. R., Beyan, O., Zappa, A., Costa, I. G., Rebholz-Schuhmann, D., Cochez, M., and Decker, S. (2021). Deep learning-based clustering approaches for bioinformatics. Briefings in bioinformatics, 22(1):393–415. (Cited on page 5.)

Khan, K. (2023). $\mathrm{Mri}_b raintumor_g lioma_d ataset. (Citedonpage\ 13.)$

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25. (Cited on pages 1, 5 and 14.)

Li, M. M., Huang, K., and Zitnik, M. (2022). Graph representation learning in biomedicine and healthcare. Nature Biomedical Engineering, 6(12):1353–1369. (Cited on page 5.)

Mirzaei, B., Nikpour, B., and Nezamabadi-Pour, H. (2021). Cdbh: A clustering and density-based hybrid approach for imbalanced data classification. Expert Systems with Applications, 164:114035. (Cited on page 4.)

Nickparvar, M. (2020). Brain tumor mri dataset. Retrieved from `https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset`. (Cited on page 30.)

Nickparvar, M. (2021). Brain tumor mri dataset. (Cited on page 13.)

Rana, M. and Bhushan, M. (2023). Machine learning and deep learning approach for medical image analysis: diagnosis to detection. Multimedia Tools and Applications, 82(17):26731–26769. (Cited on page 5.)

Smilkov, D., Thorat, N., Nicholson, C., Reif, E., Viégas, F. B., and Wattenberg, M. (2016). Embedding projector: Interactive visualization and interpretation of embeddings. arXiv preprint arXiv:1611.05469. (Cited on page 21.)

Sun, L., Yu, K., and Batmanghelich, K. (2021). Context matters: Graph-based self-supervised representation learning for medical images. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 4874–4882. (Cited on page 4.)

Taleb, A., Lippert, C., Klein, T., and Nabi, M. (2021). Multimodal self-supervised learning for medical image analysis. In International conference on information processing in medical imaging, pages 661–673. Springer. (Cited on page 4.)

Tiu, E., Talius, E., Patel, P., Langlotz, C. P., Ng, A. Y., and Rajpurkar, P. (2022). Expert-level detection of pathologies from unannotated chest x-ray images via self-supervised learning. Nature Biomedical Engineering, 6(12):1399–1406. (Cited on page 4.)

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. Journal of machine learning research, 9(11). (Cited on pages 10 and 21.)

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. (2017a). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2097–2106. (Cited on page 2.)

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. (2017b). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2097–2106. (Cited on page 13.)

Weiler, M. and Cesa, G. (2019). General e(2)-equivariant steerable cnns. Advances in Neural Information Processing Systems. (Cited on pages 1, 5 and 8.)

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. (2017). Harmonic networks: Deep translation and rotation equivariance. IEEE/CVF Conference on Computer Vision and Pattern Recognition. (Cited on pages 1, 5 and 8.)

Yadav, S. S. and Jadhav, S. M. (2019). Deep convolutional neural network based medical image classification for disease diagnosis. Journal of Big data, 6(1):1–18. (Cited on page 5.)

Zanaty, E. and Abdelhafiz, W. M. (2016). A performance study of classical techniques for medical image segmentation. Intl. J. Informatics and Medical Data Processing, 1(2). (Cited on page 4.)

Zhan, X., Xie, J., Liu, Z., Ong, Y.-S., and Loy, C. C. (2020). Online deep clustering for unsupervised representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 6688–6697. (Cited on page 5.)

Appendix A

Appendix

Progress Report

In our research project titled **"Enhancing Self-Supervised Learning for Image Clustering Using Geometric Deep Learning"**, we aim to develop a novel approach that leverages geometric deep learning to improve self-supervised image clustering. This progress report outlines the work completed since our last update, the feedback received during our recent presentation, and our plans moving forward.

Literature Review and Problem Definition

In the first week of our research, we laid the groundwork by conducting an extensive literature review and setting up collaborative tools for efficient workflow management. Under the guidance of our supervisor, Akash Malhotra, we initiated our study by exploring foundational papers and conducting a comprehensive literature search using Google Scholar. Keywords such as "self-supervised learning," "geometric deep learning," and "image clustering" were used to ensure that we captured the latest advances in the field. We specifically targeted publications from leading journals and conferences, such as *Knowledge-Based Systems*, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, and the *International Conference on Machine Learning (ICML)*.

Our literature review identified key limitations in current convolutional neural networks (CNNs). Traditional CNNs are inherently equivariant to translations due to their convolutional structure but fail to generalize to rotations. This limitation arises from the use of inner vector dot products in standard convolutions, which are not rotation-invariant.

To address this gap, we explored **Group Equivariant Convolutional Networks (G-CNNs)** as introduced by Cohen and Welling (2016) Cohen and Welling (2016). G-CNNs extend traditional CNNs by incorporating group symmetries into the convolutional layers, enabling equivariance to a broader set of geometric transformations, including rotations and reflections. This aligns with our goal of enhancing self-supervised learning by extracting more robust geometric features from images regardless of their orientation.

The review included several influential works:

- *Cohen and Welling (2020)* introduced G-CNNs, laying the foundation for our exploration of geometric deep learning in image clustering.
- *Asano et al. (2020)* proposed a self-labeling method integrating clustering and representation learning, aligning with our goal to minimize manual labeling.
- *Caron et al. (2020)* presented an unsupervised learning approach through contrasting cluster assignments, enhancing feature representations without labels.

- *Romero et al. (2020)* demonstrated the importance of data symmetries using G-CNNs with attention mechanisms.
- *Wang et al. (2021)* proposed techniques to accelerate training convergence, relevant for scaling experiments.
- *Ntelemis et al. (2022)* introduced a multi-view self-labeling approach, improving feature representation accuracy in clustering.
- *Fini et al. (2023)* presented a hybrid approach blending supervised and unsupervised learning, reducing reliance on labeled data.

Methodology

We have designed a detailed architecture that integrates G-CNNs within a self-supervised learning framework for image clustering. The architecture consists of the following components:

1. **Unlabeled Data Input**: We begin with unlabeled images, emphasizing the reduction of dependency on labeled datasets.
2. **Geometric Feature Extraction with G-CNNs**: Utilizing G-CNNs, we extract geometric features that are invariant or equivariant to rotations and other transformations. This step ensures that the model learns features based on the inherent structure of the images rather than their orientation.
3. **K-Means Clustering**: The extracted features are clustered using the K-Means algorithm. This unsupervised clustering groups similar data points based on the learned feature representations.
4. **Pseudo-Label Assignment**: Clusters are assigned pseudo-labels, effectively transforming the unlabeled data into a pseudo-labeled dataset.
5. **Classification with Multi-Classifier Network**: A multi-classifier network is trained using the pseudo-labeled data to perform classification tasks. This step leverages the learned features and pseudo-labels to refine the model's predictive capabilities.

Dataset

To initiate our experiments, we selected the **Brain Tumor MRI Dataset** from Kaggle Nickparvar (2020). This dataset contains 7,023 MRI images categorized into four classes: glioma, meningioma, no tumor, and pituitary tumor. For our self-supervised learning approach, we are using the images without their labels to simulate an unlabeled dataset. This choice allows us to test our model's ability to learn and cluster images based solely on intrinsic patterns within the data.

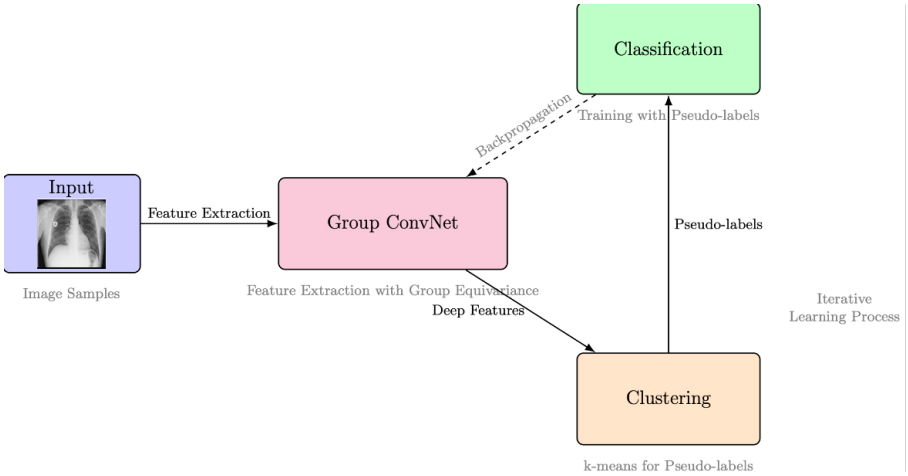

Figure A.1: Proposed Methodology

Implementation and Preliminary Experiments

- **Data Preprocessing**: MRI images were standardized in size and normalized for consistency.
- **Model Implementation**: We implemented the G-CNN architecture in PyTorch, incorporating group convolutions to account for rotational symmetries.
- **Initial Training**: Preliminary training was conducted on a subset of the dataset, validating the model's capability to extract meaningful features without labels.

Feedback from Last Presentation

During our recent presentation, we received some insightful and encouraging feedback:

- It was suggested that we begin our presentation by clearly defining the problems identified in our literature review, providing a solid foundation and context for our research objectives.
- We were encouraged to explicitly identify any research gaps that our project aims to address, highlighting what makes our work unique and necessary within the field.
- Given that our dataset is relatively small, it was recommended that we consider pretraining our model on a larger dataset, such as ImageNet, before applying it to our specific task. This could improve our model's performance and robustness.
- Our architecture was described as high-level, and we were advised to provide a more detailed design in future presentations, specifying layers, configurations, and a clearer breakdown of each component.
- Lastly, there was positive feedback on the potential of our work for publication, motivating us to refine our research further with a view towards achieving publication-worthy results.

Collaboration and Workflow Management

To ensure efficient collaboration throughout the project, we created a shared OneNote drive for organizing research papers, notes, and presentation materials. This provides a centralized repository for our collective knowledge. Additionally, a private Microsoft Teams channel was created to facilitate real-time communication, enabling quick discussions and problem-solving sessions. For code development, we set up a GitHub repository for version control, allowing efficient contributions to the project's codebase.

18/02/2025

Dataset Preprocessing and Training Subset Selection

- Modified both the training script to use only the first 20% of the dataset for unbiased comparison.

Integration of Training Time Tracking

- Implemented tracking of total training time across all epochs.
- Added per-epoch training time measurement for better performance analysis.
- Separated tracking for feature extraction and clustering time to compare computational overheads.
- Logged timing results to a CSV file for reporting and analysis.

Learning Rate Scheduling using CosineAnnealingLR

- Added learning rate scheduler with `CosineAnnealingLR` for smooth and adaptive learning rate decay.
- Ensured that learning rate updates are logged and saved in CSV for later analysis.
- Verified that the model optimizes training while avoiding premature convergence.

19/02/2025 (Challenges)

FAISS + PyTorch Multi-GPU

Integrating FAISS with PyTorch in a multi-GPU environment presents several challenges, particularly regarding version compatibility, GPU memory management, and multiprocessing handling. Below are the key challenges encountered and the solutions applied.

1. Version Compatibility

FAISS, CUDA, cuBLAS, and PyTorch versions need to be compatible to avoid runtime errors such as `CUBLAS_STATUS_SUCCESS` failures. Incompatible versions can lead to resource allocation issues and unexpected crashes.

Solution: Ensure that FAISS, CUDA, and PyTorch versions are aligned. The following table provides the recommended version compatibility:

Library	**Recommended Version**
FAISS-GPU	1.7.4 or higher
CUDA Toolkit	11.7
cuBLAS	11.x (bundled with CUDA)
PyTorch	1.12.1 or 1.13.x

Table A.1: Recommended Library Versions

Check Versions: If a mismatch occurs, reinstall the correct version using:

```
conda install -c conda-forge faiss-gpu=1.7.4 cudatoolkit=11.7
```

2. GPU Memory Management

FAISS does not gracefully handle out-of-memory (OOM) situations, leading to `cublas failed (13)` errors. PyTorch and FAISS may compete for GPU resources, causing memory exhaustion.

Solution:

- Reduce batch sizes to lower GPU memory consumption.
- Clear GPU memory before FAISS operations using:

```
import torch
torch.cuda.empty_cache()
```

3. Data Parallelism and Multiprocessing

Using `torch.multiprocessing` with FAISS can lead to memory fragmentation and resource exhaustion. FAISS may not function reliably with PyTorch's multiprocessing setup.

Solution:

- Avoid `torch.multiprocessing` where FAISS is used.
- Use `torch.nn.DataParallel` or `torch.distributed` for safer multi-GPU training.
- If `torch.multiprocessing` is necessary, set the start method to `spawn`:

```
import torch.multiprocessing as mp
mp.set_start_method('spawn', force=True)
```

21/02/2025

Error while after 17 epochs:

```
Traceback (most recent call last):
  File "/gpfs/workdir/islamm/ex1.py", line 415, in <module>
    preprocessed_features = preprocess_features(features, pca_dim=PCA_DIM)
  File "/gpfs/workdir/islamm/ex1.py", line 250, in preprocess_features
    eigenvalues, eigenvectors = torch.linalg.eigh(cov)
torch._C._LinAlgError: cusolver error: CUSOLVER_STATUS_EXECUTION_FAILED,
when calling `cusolverDnXsyevd( handle, params, jobz, uplo, n, CUDA_R_32F,
reinterpret_cast<void*>(A), lda, CUDA_R_32F, reinterpret_cast<void*>(W),
CUDA_R_32F, reinterpret_cast<void*>(bufferOnDevice), workspaceInBytesOnDevice,
reinterpret_cast<void*>(bufferOnHost), workspaceInBytesOnHost, info)`.
```

Solution

The error occurs due to the CUDA solver failing during eigen decomposition, often caused by **NaN** or **Inf** values in the covariance matrix or numerical instability in the CUDA backend. To resolve this, first **check for NaN or Inf** in the covariance matrix using `torch.isnan()` and `torch.isinf()`, and replace them with finite values using `torch.nan_to_num()`. Alternatively, bypass CUDA-specific issues by using **Scikit-learn's PCA** (`sklearn.decomposition.PCA`) with whitening and L2 normalization, which offers better numerical stability. If the problem persists, consider **switching the CUDA backend** using `torch.backends.cuda.preferred_linalg_library("cublas")` to avoid compatibility issues with the default solver.

22/02/2025

```
[rank0]:[W221 23:52:55.488902976 ProcessGroupNCCL.cpp:1250]
WARNING: process group has NOT been destroyed before we destruct ProcessGroupNCCL.
On normal program exit, the application should call destroy_process_group to ensure
that any pending NCCL operations have finished in this process. In rare cases,
this process can exit before this point and block the progress of another member
of the process group. This constraint has always been present, but this warning
has only been added since PyTorch 2.4 (function operator())
```

```
AttributeError: type object 'torch._C._profiler.ProfilerActivity'
has no attribute 'xCPU'
```

- The **NCCL process group** was not properly destroyed, leading to potential deadlocks.
- Incorrect usage of the `ProfilerActivity` attribute. The correct attribute is `CPU`, not `xCPU`.

Solution

1. Ensure the distributed process group is always destroyed, even if exceptions occur. Wrap the training code in a `try-finally` block and call `dist.destroy_process_group()` in the `finally` section.

2. Replace incorrect profiler activity attribute:

```
activities = [ProfilerActivity.CPU, ProfilerActivity.CUDA]
```

While using Faiss for GPU-based K-Means clustering, the following error occurred during matrix multiplication using cuBLAS:

```
Faiss assertion 'err == CUBLAS_STATUS_SUCCESS' failed
cublas failed (13): CUBLAS_STATUS_ALLOC_FAILED
```

Error

`RuntimeError: Could not infer dtype of NoneType` occurs when `torch.tensor()` receives a `None` value, preventing PyTorch from inferring a valid data type.

Cause

In the DeepCluster pipeline, the PyTorch PCA step failed, resulting in `features_np` being `None`.

Solution

Replaced **PyTorch PCA** with `sklearn.decomposition.PCA` to ensure proper dimensionality reduction and avoid `NoneType` issues.

22/02/2025

During training, the following error is raised:

```
ValueError: array is not C-contiguous
```

This error occurs in the FAISS k-means clustering step when converting a PyTorch tensor to a NumPy array. FAISS requires the input array to be stored contiguously in memory (C order).

Solution

Ensure that the NumPy array is C-contiguous before passing it to FAISS. Two common fixes are:

1. **Using `.contiguous()`:**

```
features_tensor = features_tensor.cpu().contiguous()
x = features_tensor.numpy()
```

2. **Using `np.ascontiguousarray`:**

```
x = np.ascontiguousarray(features_tensor.cpu().numpy())
```

Could not solve the problem using the above solution, thus using the PyTorch Kmeans Algorithm:

PyTorch K-Means Algorithm

Given:

- $X \in \mathbb{R}^{N \times D}$: Feature matrix with N samples and D-dimensional features.
- K: Number of clusters.

Algorithm Steps:

1. Initialize K centroids by randomly selecting samples from X.
2. For each iteration:
   - Compute pairwise distances using:

     $$d_{ij} = \|x_i - c_j\|_2$$

     where x_i is a data point and c_j is a centroid.
   - Assign each sample to the nearest centroid.
   - Update centroids by computing the mean of assigned samples:

     $$c_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i$$

     where S_j is the set of samples assigned to cluster j.
3. Stop when the centroid shift is less than a tolerance ε or after a fixed number of iterations.

1. What makes DeepCluster good at clustering?

DeepCluster excels at clustering datasets with certain patterns and distributions in their visual features, especially when image features exhibit **clear category distinctions** or **latent grouping patterns**. Below are the characteristics of tasks where DeepCluster performs well:

Tasks Suited for DeepCluster

1. **Tasks with Clear Visual Features:**
   - Suitable for images where differences between features are significant.
   - *Examples:* Natural image datasets like CIFAR-10 and STL-10, where categories have relatively clear boundaries.
2. **Data with Latent Grouping Structures:**
   - DeepCluster is adept at discovering hidden patterns in unlabeled data.
   - *Applications:* Unsupervised scene classification, object detection.
3. **Large-Scale Datasets:**
   - Relies on the diversity and scale of data to learn meaningful representations.
   - Larger datasets like the unlabeled portion of ImageNet significantly enhance its performance in self-supervised learning.
4. **Dimensionality Reduction and Representation Learning for High-Dimensional Data:**
   - Effectively reduces the dimensionality of high-resolution image data.
   - Extracted deep features through CNNs significantly improve clustering compared to clustering raw data.
5. **Feature Learning for Cross-Domain Transfer:**
   - Features learned on one domain (e.g., unlabeled ImageNet data) can be transferred to other domains or downstream tasks (e.g., smaller datasets).

Image Types Where DeepCluster Performs Well

1. **Images with Clear Classification Boundaries:**
   - Categories with significant visual differences.
   - *Examples:*
     - CIFAR-10: 10 categories of natural objects (dogs, cats, airplanes, etc.).
     - STL-10: Higher resolution, fewer images.
2. **Images with Repetitive Patterns:**

- Contain recurring textures or structures (e.g., faces or landscapes).
- *Examples:*
  - Face datasets (grouped by expressions or angles).
  - Scene classification (e.g., urban vs. forest scenes).

3. **Large-Scale Unlabeled Image Datasets:**
   - Unlabeled datasets like ImageNet, which have rich semantic features.

Limitations: Scenarios in which the Deep Cluster Struggles

1. **Overlapping Visual Features Between Categories:**
   - When differences between categories are minimal, clustering may fail.
   - *Example:* CIFAR-100, where many categories are visually similar.
2. **Small Datasets:**
   - Performs poorly on small datasets due to reliance on data diversity.
3. **Noisy Data:**
   - Sensitive to noise or artifacts, e.g., noisy medical images like MRI scans.
4. **High-Dimensional Data Without Clear Patterns:**
   - Struggles when visual patterns are absent (e.g., random noise images).
5. **Non-Image Data:**
   - Designed for image data; requires modification for other types like text or time series.

Summary

DeepCluster is most effective for tasks with:

- Significant visual differences between categories.
- Large, unlabeled datasets with clear classification boundaries.
- Potential patterns or grouping structures in data.

It is ideal for self-supervised learning tasks to generate pseudo-labels for unlabeled data or as a pre-training model for downstream tasks. However, for noisy or highly overlapping data, preprocessing or combination with other methods may be necessary.

2. Can DeepCluster Perform Well on MRI Images of Brain Tumors?

DeepCluster's performance on MRI images of brain tumors depends on the feature distribution and specific challenges.

Strengths: When It May Perform Well

1. **Distinct Tumor Features:**
   - If different tumor types show clear differences in shape, boundary, intensity, or texture, DeepCluster can cluster effectively.
   - *Examples:*
     - Significant differences in T1, T2, or FLAIR modes.
     - Morphological or locational differences (e.g., frontal lobe vs. parietal lobe).
2. **High-Quality Data:**
   - High signal-to-noise ratio (SNR), no artifacts, and well-labeled datasets like BraTS can help the model perform well.
3. **Large Dataset Scale:**
   - Adequate tumor MRI data, including multimodal data, enables discovering latent grouping patterns and generating pseudo-labels.

Challenges: Scenarios Where Performance May Suffer

1. **Minimal Tumor Differences:**
   - If tumor types have overlapping features, clustering might fail.
   - *Examples:* Small or early-stage tumors may be overshadowed by background noise.
2. **Noisy Data and Artifacts:**
   - Motion artifacts or varying scanning parameters interfere with feature extraction.
3. **Insufficient Dataset Size:**
   - Small datasets result in unstable feature extraction and clustering results.
4. **Fusion of Multimodal Data:**
   - Brain tumor analysis often relies on multiple MRI modalities (e.g., T1, T2, FLAIR), and single-modality information may be insufficient to differentiate complex categories.
   - DeepCluster is designed to process single-modality images by default, and additional design is required for multimodal data (such as multimodal feature fusion).
5. **Complexity of Medical Features:**
   - Medical data often contain complex contextual relationships (e.g., the relationship between tumors and surrounding brain tissue).
   - The simple K-means clustering in DeepCluster may not be sufficient to capture these relationships.

Strategies to Improve Performance

1. **Data Preprocessing:**
    - Noise removal, artifact correction, and parameter standardization to enhance image quality.
2. **ROI Extraction:**
    - Use segmentation algorithms (e.g., UNet) to isolate tumors and reduce background interference.
3. **Multimodal Integration:**
    - Fuse T1, T2, and FLAIR modalities for richer feature representation and improved clustering performance.
4. **Enhanced Clustering Algorithms:**
    - Replace K-means with more advanced algorithms such as Gaussian Mixture Models (GMM) or DBSCAN to better capture complex distributions.
5. **Semi-Supervised/Self-Supervised Methods:**
    - Incorporate a small number of labeled samples or other self-supervised techniques such as SeLa to improve feature learning.

3. Are the Visual Features of Gliomas, Meningiomas, and Pituitary Tumors Sufficiently Distinct?

Gliomas, meningiomas, and pituitary tumors have certain differences in MRI features, but the clarity depends on factors such as modality, size, and aggressiveness. Below is a summary:

Glioma

Features:

- **Location:** In brain parenchyma.
- **Shape:** Irregular, especially high-grade gliomas.
- **Boundaries:** Fuzzy, invasive.

Signal:

- **T1:** Low signal.
- **T2/FLAIR:** High signal, especially in edema areas.

Clarity:

- High-grade gliomas: Easier to detect on T2/FLAIR.
- Low-grade gliomas: Harder to distinguish due to subtle features.

Meningioma

Features:

- **Location:** Near meninges.
- **Shape:** Round or oval.
- **Boundaries:** Clear and well-defined.

Signal:

- **T1:** Isointense or hyperintense.
- **T2:** Isointense or hyperintense.
- **Contrast:** Strong enhancement.

Clarity:

- Visual features are highly distinct, especially with contrast-enhanced scans.

Pituitary Tumor

Features:

- **Location:** In the pituitary gland near the sella turcica.
- **Shape:** Small, well-defined.

Signal:

- **T1:** Isointense or hypointense.
- **T2:** Hyperintense.
- **Contrast:** Strong enhancement.

Clarity:

- Small tumors may be hard to detect, but larger ones are more distinguishable.

4. Figshare Brain Tumor Dataset

Description:
A publicly available 2D brain tumor dataset suitable for classification tasks. The data includes Meningioma, Glioma, and Pituitary Tumor.
Modalities Included: T1-weighted.
Task: Tumor classification.
Size: 3064 MRI images, divided into three categories:

- Meningioma: 708 images
- Glioma: 1426 images

- Pituitary Tumor: 930 images

Use: Suitable for preliminary research, especially for image classification or simple feature extraction.
Link: https://figshare.com/articles/dataset/brain_tumor_dataset/1512427