School of Computer Science and Electronic Engineering

MSc Data Science

Academic Year 2023-2024

Unsupervised Clustering of High-Dimensional Image Data: Integrating Autoencoders with Gaussian Mixture Models

A project report submitted by: Sruthi Lakshmi Vadakkapurathu Reghu
URN:6847302

A project supervised by: Dr Amir Esfahani

A report submitted in partial fulfilment of the requirement for the degree of Master of Science

University of Surrey
School of Computer Science and Electronic Engineering
Guildford, Surrey GU2 7XH
United Kingdom.
Tel: +44 (0)1483 300800

ABSTRACT

With the large number of high-resolution images available in areas such as medicine, robotics and technology, finding appropriate mechanisms for organizing and making sense of this data has become a cumbersome task. Traditional techniques of clustering which involve grouping similar images together often fail to work well when dealing with very complex and large datasets. This is primarily because they are not able to handle the fine-grained details contained in these images; hence, leading to inaccurate groupings.

This dissertation proposes a new method that combines autoencoders with Gaussian Mixture Models (GMM) for better image clustering. An autoencoder is a deep learning model that helps reduce complex data by compressing it into smaller and more manageable sizes called latent spaces while preserving its main features. GMM, on the other hand, uses underlying patterns in the data to locate groups within it once the information has been compressed.

This study includes training an autoencoder on picture datasets to learn these simplified representations known as a "latent space". Then, the decoder part which remakes images is removed and Gaussian Mixture Model is applied to the compressed data in order to create clusters.

The main goal is to analyse the integration of autoencoders with GMMs, and their ability to handle complex large image datasets. Its method is aimed at managing high-dimensional data efficiently so that it preserves important descriptions while simplifying them for clustering.

This research reveals that integrating autoencoders into GMMs can be utilized as a robust framework for clustering high-resolution images. This technique enables one to handle large image datasets by reducing complexity in the data and exposing its underlying structure within such fields as medical imaging and technology among other things.

To sum up, this dissertation presents an innovative approach which makes use of the strengths inherent in combining auto-encoders and Gaussian mixtures model (GMM) in solving issues relating to clustering high resolution images thus providing an efficient and scalable solution for handling complicated datasets.

HIGHLIGHTS

- Developed a novel image clustering method integrating ResNet-18 autoencoders with Gaussian Mixture Models (GMMs).
- Achieved significant dimensionality reduction while preserving critical features for accurate clustering.
- Demonstrated high fidelity in image reconstruction, ensuring effective feature retention in latent space
- Validated model performance with comprehensive evaluation metrics including BIC, AIC, and cross-validation.
- Enhanced clustering interpretability and accuracy over traditional methods like K-Means.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Dr. Amir Esfahani, for his invaluable guidance, support, and encouragement throughout the course of this dissertation. His insights and expertise have been instrumental in shaping this work.

I am also particularly thankful to Samuel Carter, whose contributions and assistance as a fellow research scholar have been pivotal in the development and completion of this project. His collaboration and shared knowledge have greatly enriched my research experience.

Finally, I would like to extend my heartfelt appreciation to my parents, whose unwavering motivation and belief in my abilities have been a constant source of strength. Their support has been indispensable in helping me reach this milestone.

I certify that the work presented in the dissertation is my own unless referenced

Signature: **SRUTHI LAKSHMI VADAKKAPURATHU REGHU**

Date: 03-09-2024

TOTAL NUMBER OF WORDS:

Contents

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

In today's digital age, we are generating an enormous amount of high-resolution images, particularly in areas like medicine, satellite imaging, and technology. With so much data being created, finding effective ways to organize and make sense of it has become increasingly challenging. Traditional methods of grouping similar images often struggle with large and complex datasets, as they are not designed to handle the intricate details present in these images. This dissertation aims to tackle this issue by developing a new method that combines autoencoders with Gaussian Mixture Models (GMMs) to improve how we cluster and analyze these high-resolution images. The goal is to create a method that not only groups images more accurately but also makes the results easier to understand and use.

1.1 BACKGROUND

The rapid growth of image data has brought about significant challenges in how we manage and interpret this information. Fields like medicine, where high-resolution imaging is critical for diagnosis and treatment, or satellite imaging, where vast amounts of detailed earth data need to be analyzed, are just a few examples of areas dealing with these challenges. As the volume and complexity of image data increase, traditional clustering methods, which aim to group similar images together based on their features, often fall short. These conventional techniques, such as K-Means and hierarchical clustering, struggle to capture the detailed patterns in high-resolution images, leading to inaccurate or overly simplified groupings.

In recent years, there has been growing interest in using deep learning techniques, such as autoencoders, to address these challenges. An autoencoder is a type of neural network that reduces the complexity of data by compressing it into a simpler form, known as a latent space, while still preserving its most important features. This process of dimensionality reduction makes it easier to analyze and cluster the data. However, while autoencoders can simplify the data, they do not inherently cluster it.

This is where Gaussian Mixture Models (GMMs) come into play. GMMs are statistical models that assume data is generated from a mixture of several Gaussian distributions, each representing a different group or cluster. By applying GMMs to the latent space representations generated by autoencoders, we can effectively identify and group similar images based on their core features, even in high-dimensional datasets.

The integration of autoencoders and GMMs presents a promising solution to the challenges of clustering high-resolution image data. This dissertation focuses on developing this integrated method and exploring its effectiveness in improving the accuracy and interpretability of image clustering. The significance of this research lies in its potential to enhance data analysis in critical areas, such as medical imaging and robotic training.

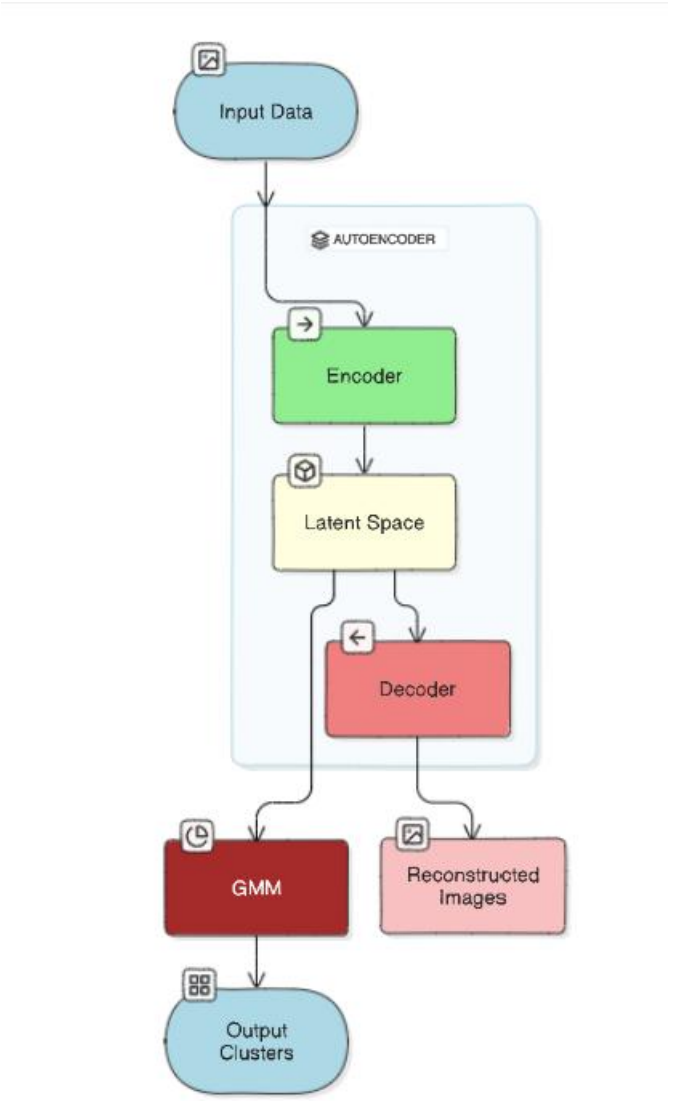


Figure 1:Integration of Autoencoders and GMMs for Clustering High-Resolution Image Data

1.2 RESEARCH AIM AND OBJECTIVES

Aim: The aim of this dissertation is to create and evaluate a method that combines autoencoders and Gaussian Mixture Models (GMMs) to improve the clustering of high-resolution images, focusing on enhancing the accuracy and understanding of the results.

Objectives:

- To review and summarize existing research on autoencoders, GMMs, and image clustering techniques to identify their strengths and limitations.
- To design and develop an autoencoder that effectively reduces the complexity of high-resolution image data while retaining key features for clustering.
- To apply GMMs to the compressed data produced by the autoencoder and evaluate the quality of the clusters formed.

- To test the proposed method on various image datasets and assess its performance in terms of clustering accuracy and computational efficiency.
- To analyze the clusters generated by the GMMs and discuss their interpretability.

1.3 RESEARCH APPROACH

To achieve the aim of this dissertation, a systematic research approach will be followed. The first step is to conduct a thorough literature review to understand the current state of image clustering, autoencoders, and GMMs. Following this, an autoencoder will be designed and trained on selected image datasets to learn how to compress the images into a simpler form while retaining their most important features. Once the autoencoder is trained, the Gaussian Mixture Model will be applied to the compressed data to group the images into clusters.

The performance of this method will be evaluated using several metrics, including how accurately the images are clustered and by computing multiple evaluation metrices.

1.4 DISSERTATION OUTLINE

The dissertation is organized into the following chapters:

- **Chapter 2: Literature Review** - This chapter provides an overview of existing research on autoencoders and Gaussian Mixture Models (GMMs), discussing their roles in image clustering, their advantages, and the limitations of current methods. It sets the foundation for the need to integrate these techniques to handle the complexity of high-dimensional image data effectively.

- **Chapter 3: Methodology** - This chapter details the design and training of the autoencoder, the process of applying GMMs to the latent space representations, and the evaluation framework used to assess the effectiveness of the clustering method. It covers the experimental setup, the datasets used, and the metrics for evaluating clustering performance.

- **Chapter 4: Data Analysis** - In this chapter, the results of the experiments conducted are presented and analyzed. The analysis includes a detailed examination of the clustering outcomes and the computational performance of the method.

- **Chapter 5: Discussion** - This chapter provides a critical discussion of the findings, linking the results back to the research objectives and the broader context of image clustering. It examines the strengths and weaknesses of the proposed method, its potential applications, and how it compares to the anticipated outcomes. The chapter also considers the implications of the research for future work in the field.

- **Chapter 6: Conclusion** - The final chapter summarizes the key contributions of the dissertation, reflecting on the overall success of the research in achieving its aim and objectives. It provides a concluding assessment of the method's effectiveness and suggests directions for future research to build on the work presented.

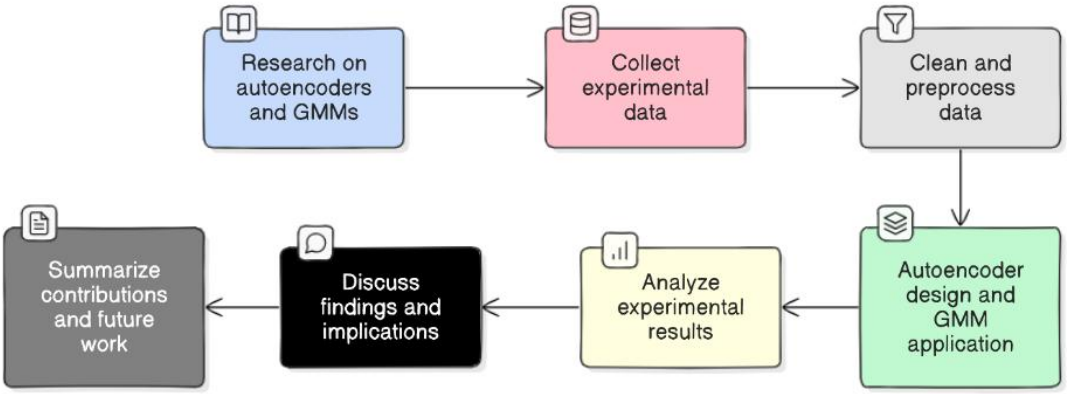


Figure 2: Flowchart illustrating the research process

CHAPTER 2: LITERATURE REVIEW

In this chapter, we explore the existing body of research on the integration of autoencoders and Gaussian Mixture Models (GMMs) for image clustering. We begin by discussing the importance of these techniques in handling high-dimensional data, with a particular focus on the use of Residual Network as the backbone of the autoencoder architecture. This chapter provides a comprehensive review of the most relevant and recent literature in the field, aiming to give a detailed strengths and limitations of current methodologies. By the end of this chapter, we will identify gaps in existing research and formulate specific research questions that this dissertation seeks to address.

2.1 IMPORTANCE OF IMAGE CLUSTERING

Image clustering is a crucial technique in various fields because it allows for the automatic organization and grouping of large sets of images based on visual similarities without requiring labeled data. This capacity is crucial for organizing and comprehending the enormous volumes of visual data produced in the modern digital environment.

Why We Need Image Clustering:

- Effective Data Organization: With the exponential expansion of image data, clustering aids in the meaningful grouping of images, facilitating the management, retrieval, and search of images without the need for manual labelling.
- Pattern Recognition: Clustering makes it possible to find patterns and connections across image groups, which can uncover hidden structures under the surface. This is particularly useful in exploratory data analysis and for applications like anomaly detection.
- Data reduction: In cases where handling every image individually is computationally expensive, clustering can help decrease the complexity of the dataset by putting comparable images together. This allows for more effective storage, processing, and analysis.
- Improved Model Training: In machine learning, clustering can be used as a pre-processing step to create labeled datasets or to improve the training of other models by focusing on significant groups within the data.

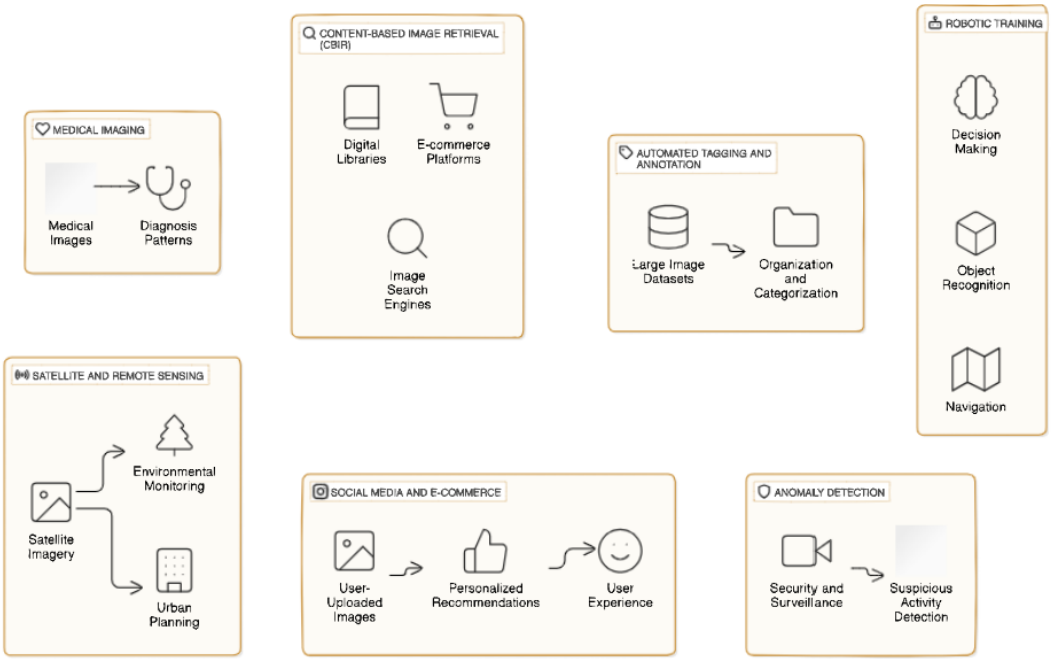


Figure 3: Applications of Image Clustering

2.2 AUTOENCODERS, RESIDUAL NETWORK AND GMMS IN IMAGE CLUSTERING

Residual Network 18 Autoencoders:

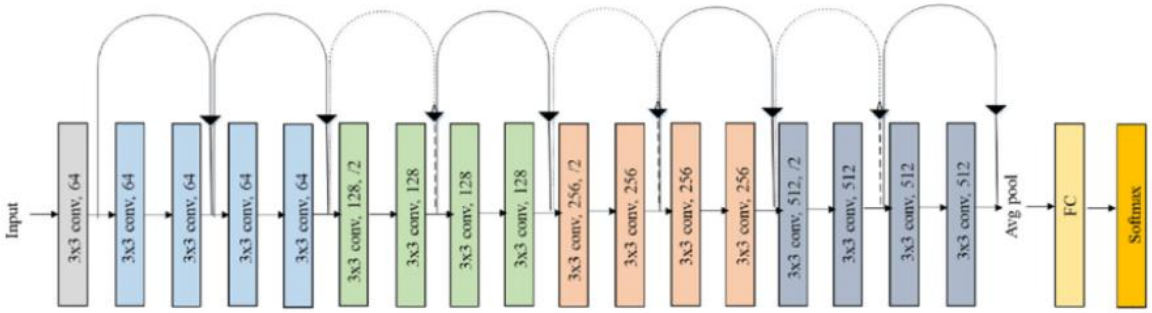


Figure 4: ResNet18 Architecture [7]

Autoencoders have become one of the most strategic tools in machine learning, especially for tasks involving the compression and analysis of high-dimensional data like images. An autoencoder's primary function is to learn a compact, low-dimensional representation of input data—referred to as the latent space— input data with key constituents maintained. This reduced representation can then be used for various downstream tasks, including clustering, where it is crucial to group similar images based on their core attributes.

The capability of autoencoders was greatly enhanced by the introduction of deep learning models such as convolutional neural networks (CNNs) in recent years. ResNet-18 is one such architecture that has emerged as an effective backbone for autoencoders. ResNet-18 introduces residual connections, which help the model learn identity mappings more efficiently, thereby addressing the problem of vanishing gradients that can occur in deep networks. Vanishing gradients happen when the gradients used to update the model's

weights during training become very small, causing the learning process to slow down or even stop. By allowing the input to bypass certain layers through residual connections, ResNet-18 ensures that important information is preserved, enabling the model to continue learning effectively even as it gets deeper. This architecture enables the autoencoder to capture intricate features in high-resolution images without losing important details during the compression process. [1]

The ResNet-18 autoencoder leverages the strengths of residual learning, making it particularly suitable for handling complex image data. By incorporating ResNet-18 as the encoder, the model is capable of learning robust features even in the presence of noise or other variations in the data. These features are then encoded into a latent space that not only represents the data more compactly but also retains the semantic essence of the images, which is crucial for accurate clustering.

In the context of image clustering, the role of the ResNet-18 autoencoder extends beyond mere dimensionality reduction. It effectively reduces the high-dimensional image data to a lower-dimensional latent space, highlighting the underlying patterns and structures within the data. This reduced space facilitates more efficient and meaningful clustering, as similar images are more likely to be grouped together based on their core features.[4]

Gaussian Mixture Model:

A Gaussian Mixture Model (GMM) is a probabilistic method of clustering that represents data as being generated by a mixture of many Gaussian distributions, each of which represents a distinct cluster. These Gaussian components are characterized by parameters like the weight of each component (which represents the relative size or relevance of each cluster), the covariance (which characterizes the shape and orientation of the cluster) and the mean(which shows the centre of the cluster). This research offers a flexible and effective clustering technique by combining GMMs with the latent representations produced by the ResNet-18 autoencoder. The combination works especially well for managing complicated data distributions since GMMs are able to capture the diverse structures that exist in the latent space.

When applied to the latent space generated by the ResNet-18 autoencoder, GMMs can effectively capture the underlying distribution of the data, allowing for the identification of distinct groups within the dataset. The latent space, being a lower-dimensional and more abstract representation of the original high-dimensional data, provides a more manageable and interpretable foundation for clustering. The GMM, with its probabilistic framework, offers flexibility in representing complex, overlapping clusters, unlike traditional hard clustering methods like K-Means, which assign each data point to a single cluster.[6]

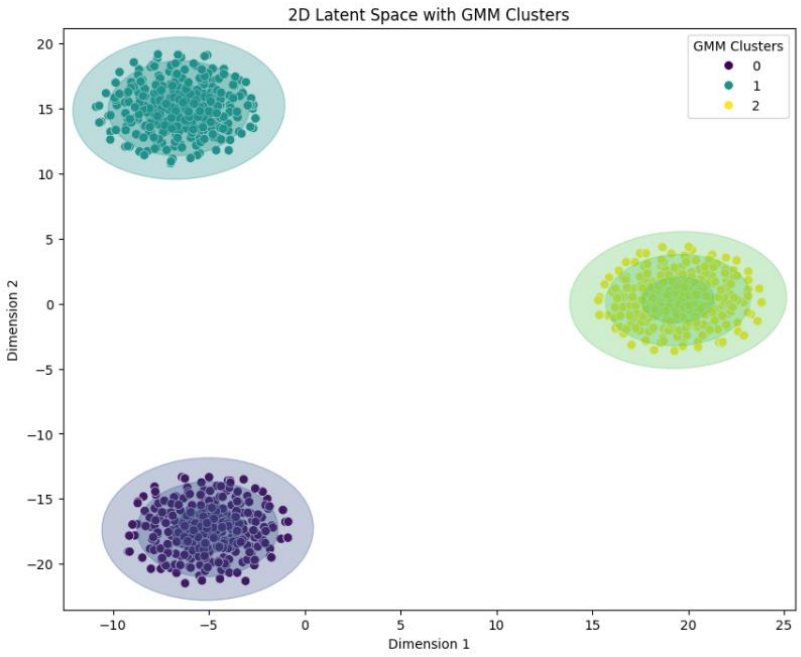


Figure 5: Visualization of 2D Latent Space Representations with Gaussian Mixture Model (GMM) Clusters, highlighting the separation of data into distinct clusters

One of the main benefits of utilizing GMMs in this context is that they can model the data's covariance structure. GMM covariance matrices enable for the modelling of elongated or differently oriented clusters, representing the underlying variability in the data. This is especially crucial for high-dimensional datasets, since spherical clusters—like those used in K-Means—are frequently assumed to be unrealistic. Better clustering outcomes are produced by GMMs because they give each Gaussian component a separate covariance matrix, improving the representation of the data distribution.

However, the problem of covariance shift presents one of the difficulties in this situation. When the distribution of output data (labels) stays constant during the training and testing or deployment phases, while the distribution of input data (features) varies, this is known as covariance shift. Because the model has learnt patterns unique to the training distribution, which are not typical of the real-world distribution seen after deployment, this phenomenon can result in models that perform well on training data but badly on fresh, unseen data. For example, take a model that has been taught to identify between photos of dogs and cats using excellent studio photographs. This model may perform poorly if it is later used on low-quality real-world photos, like those taken by a camera, because of the change in the input data distribution.

To quantify and address the discrepancies between different data distributions, Kullback-Leibler (KL) divergence is often used. KL divergence measures how one probability distribution diverges from a second, expected probability distribution. [5]

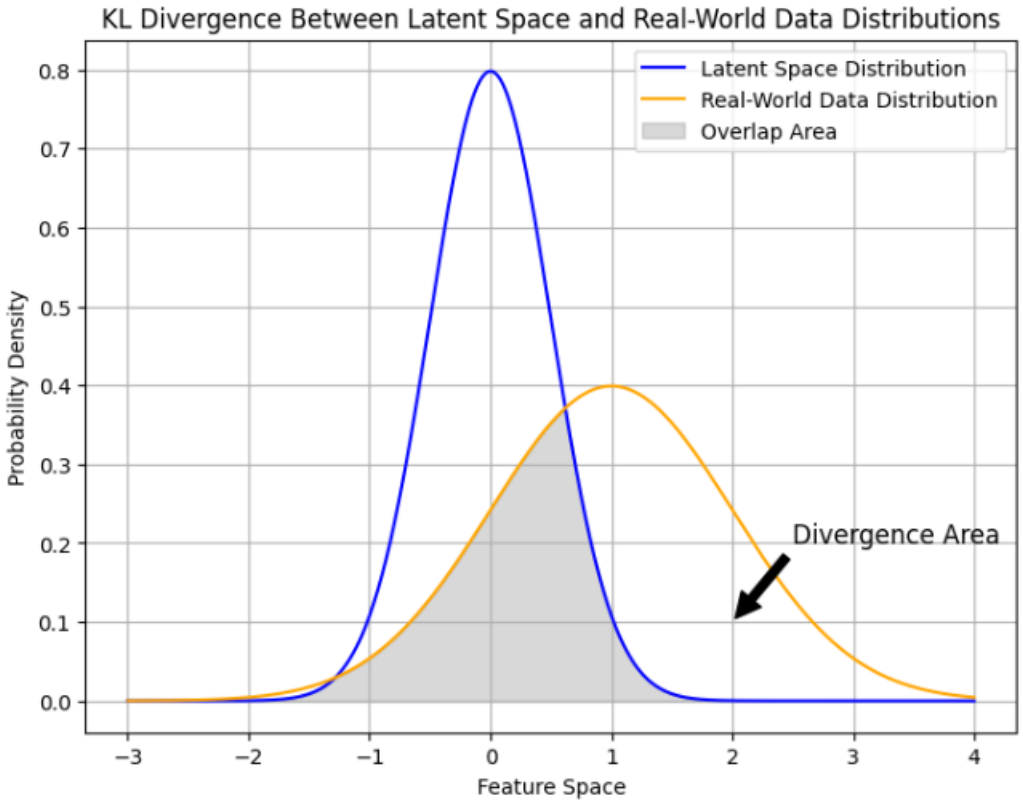


Figure 6: KL Divergence between latent space and Real-world data distributions

2.3 Comprehensive Coverage of Relevant Literature

The integration of autoencoders and Gaussian Mixture Models (GMMs) represents a significant advancement in the field of machine learning, particularly in the domain of image clustering. This section provides a comprehensive review of the existing literature on these topics, focusing on their development, applications, and the challenges that have been addressed by recent research.

Autoencoders:

Autoencoders have been a foundational element in the field of deep learning, especially for tasks involving dimensionality reduction and feature extraction. Introduced as a type of neural network aimed at learning efficient codings of unlabeled data, autoencoders compress input data into a lower-dimensional latent space and then reconstruct the original data from this compressed representation.

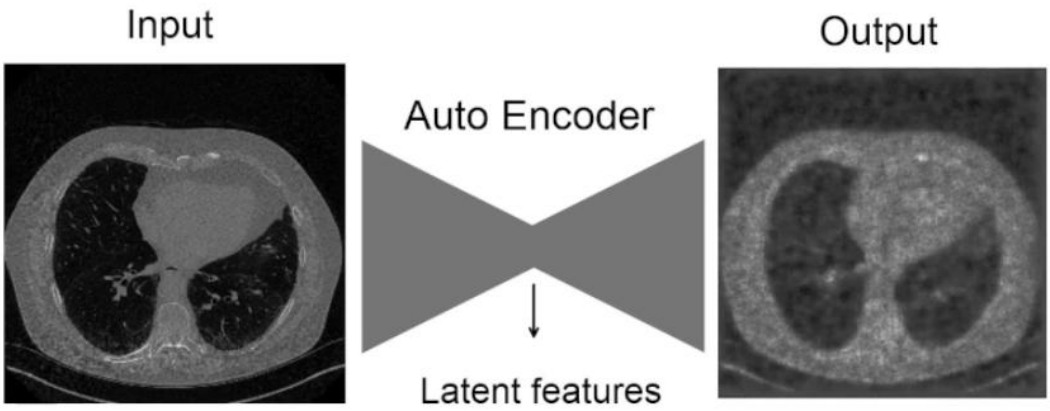


Figure 7: Illustration of an Autoencoder

- ***Denoising autoencoder (DAE)**:*

As the field of deep learning progressed, researchers sought to improve the capabilities of autoencoders, leading to the development of various autoencoder variants. One of the most influential variants is the **denoising autoencoder (DAE)**, introduced by Vincent et al. in 2010. The primary goal of a denoising autoencoder is to create a more robust and generalizable model by training it to reconstruct the original, clean data from a corrupted or noisy version of the input. This process forces the model to learn meaningful patterns and features that are less sensitive to noise and irrelevant variations, making it more effective for tasks like clustering, classification, and anomaly detection.

Unlike a traditional autoencoder, which is trained to reconstruct its input exactly, a denoising autoencoder begins by deliberately corrupting the input data. This corruption can be introduced in several ways, such as adding random noise, masking certain parts of the input, or randomly setting some input values to zero. The level and type of corruption are typically controlled by the user and can be adjusted based on the specific application or dataset. Imagine you have an image of a handwritten digit. In a denoising autoencoder, you might add Gaussian noise to the image, making it blurry or speckled, or you might randomly black out some pixels.

The corrupted input is then passed through the encoder part of the autoencoder, which compresses the noisy input into a lower-dimensional latent space. The encoder learns to focus on the most relevant features of the input that are necessary for reconstruction, effectively filtering out the noise or irrelevant details introduced during the corruption step. The decoder then takes this latent representation and attempts to reconstruct the original, clean version of the input, not the corrupted one. The reconstruction is compared to the original clean input (before corruption), and the difference between the two (known as the reconstruction loss) is used to adjust the weights of the autoencoder during training.

By concentrating on the most significant aspects of the input data, denoising autoencoders ensure that the model is less likely to be influenced by noise or irrelevant variations, which can often lead to overfitting. This ability to focus on the essential features enables the model to perform better on new, unseen data, making it more versatile and reliable in real-world applications.

While effective in learning robust data representations, Denoising autoencoders (DAEs) have several limitations. They are highly dependent on the type and level of noise introduced during training; if the noise in new data differs significantly, the DAE may struggle to denoise effectively. Additionally, DAEs risk overfitting to the specific noise patterns seen during training, which can limit their generalizability to other types of noise or variations in real-world data. This dependency makes them less flexible when encountering unexpected or complex noise in deployment scenarios. [3]

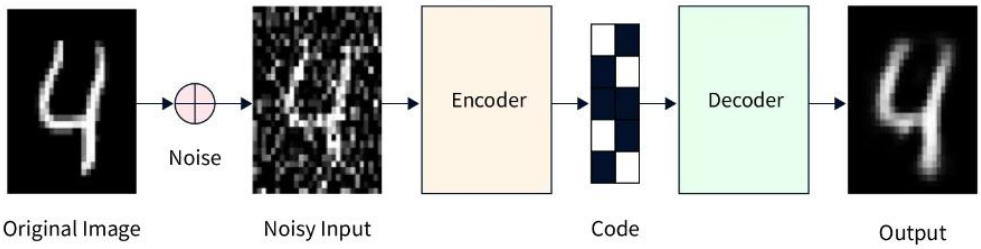


Figure 8: Workflow of a Denoising Autoencoder

- ***ResNet-18:***

The integration of convolutional layers into autoencoders has significantly enhanced their ability to process and learn from image data, particularly by capturing spatial hierarchies—how different features in an image relate to each other at various scales.

A major breakthrough in deep learning came with the introduction of Residual Networks (ResNets) by He et al. (2015), which addressed a critical issue in training deep neural networks: the vanishing gradient problem. This problem often hampers the performance of very deep networks because the gradients, which are crucial for updating the network's weights during training, diminish as they propagate back through the layers, leading to slow or stalled learning.

ResNet-18, a widely recognized variant of ResNet, introduced residual learning, a technique where the network learns residuals (differences from the identity mapping) instead of the full transformation in each layer. This is accomplished by creating shortcut connections that enable a layer's input to be added straight to the output, without the need for further processing steps. This breakthrough prevents the vanishing gradient problem and allows for the training of even deeper networks by preserving the information flow across layers. Because of this, ResNet-18 is able to effectively extract hierarchical features and complicated patterns from the data without experiencing the performance degradation that deep networks are known for.
ResNet-18 is a critical encoder in autoencoders designed for imagine clustering applications. Its architecture, which makes use of residual blocks, is especially well-suited to extracting high-level features like object shapes and structures, as well as low-level features like edges and textures. These characteristics are essential for efficiently compressing high-dimensional picture data into a meaningful lower-dimensional latent space, which forms the basis of clustering. A ResNet-18 encoder creates a latent space that captures the salient features of the images, retaining sufficient information to provide precise reconstructions or applications such as clustering, where differentiating even the minute changes in the data is crucial.

Clustering:

- ***K-Means Clustering: A Traditional Approach:***

K-Means is one of the highly popular and extensively utilized clustering algorithm. It operates on the principle of minimizing the within-cluster variance, effectively trying to group data points in such a way that those within a cluster are as close as possible to the cluster's centroid, which is the mean of all the points in that cluster. K-Means is an iterative technique that locates the closest centroid for each data point before recalculating the centroids based on the cluster memberships as of the moment. Until the centroids stable and do not significantly vary between iterations, this process is repeated.

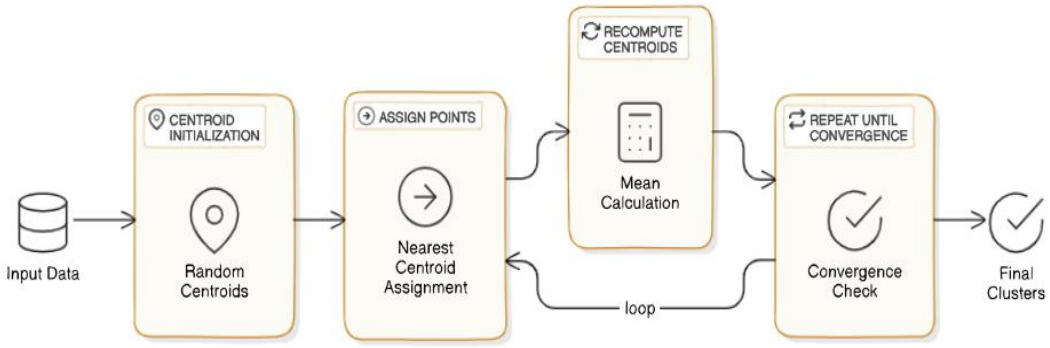


Figure 9: Illustration of K-means clustering

K-Means is straightforward and computationally effective, but it has a few drawbacks, particularly when working with complicated, high-dimensional datasets.

- Assumes Spherical Clusters: Real-world data frequently differs from K-Means' assumption that clusters are spherical and generally equal in size.
- Tough Assignments: K-Means allocates every data point to a single cluster, which may cause issues when data points inherently belong to many clusters.
- Not Probabilistic: In uncertain or noisy contexts, K-Means may be less effective since it lacks a probabilistic measure of cluster membership.

- ***Variational Autoencoders (VAEs):***

Kingma and Welling (2014) proposed Variational Autoencoders (VAEs), which provide an alternative method of modeling data, especially for generative tasks and clustering. An autoencoder that applies a probabilistic framework to the encoding and decoding process is called a VAE. Instead of learning deterministic mappings from input data to a latent space and back, VAEs model the latent space as a continuous probability distribution.

Each input data point is captured in a VAE as a distribution over the latent space, usually characterized by a mean and variance, rather than as a single fixed point in the latent space. By ensuring that the latent space is continuous and smooth, this probabilistic method enables more meaningful interpolations between latent space points and improves generalization. The original input is subsequently reconstructed by the decoder using samples taken from this distribution. The VAE is trained to minimize two different kinds of losses: the Kullback-Leibler (KL) divergence, which regularizes the latent space to be near a typical Gaussian distribution, and the reconstruction loss, which assesses

how closely the decoded output matches the original input.

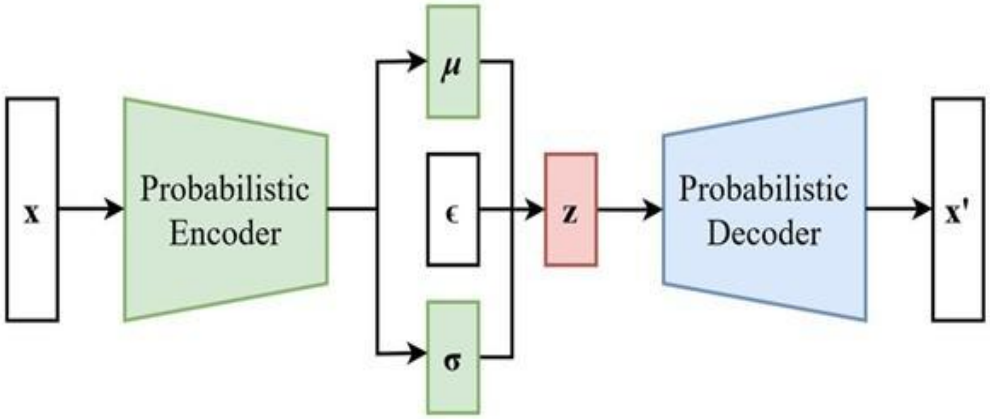


Figure 10: Illustration of VAE

Variational Autoencoders (VAEs) have a number of drawbacks despite being effective for generative tasks and clustering. The Gaussian distributions utilized in the latent space of VAEs frequently result in blurry outputs, and it can be difficult to balance the KL divergence with reconstruction loss, which can either lead to poor generalization or low reconstruction quality. Furthermore, because of its probabilistic structure and potential for mode collapse, VAEs can only capture a restricted range of outputs, and their tuning can be difficult. [2]

- ***Gaussian Mixture Models in Clustering:***

Gaussian Mixture Models (GMMs) have been widely used in the field of clustering due to their probabilistic approach to grouping data. Unlike traditional methods, which assign each data point to a single cluster, GMMs assume that the data is generated from a mixture of several Gaussian distributions, where each distribution represents a different cluster. Because of this probabilistic framework, GMMs are able to assign soft cluster memberships, which means that each data point has a probability of belonging to each cluster rather than being hard-assigned to just one. Moreover, because of their flexibility, GMMs can model clusters of various sizes, shapes, and orientations, which makes them especially useful for clustering in high-dimensional spaces where the underlying data structure is complex.

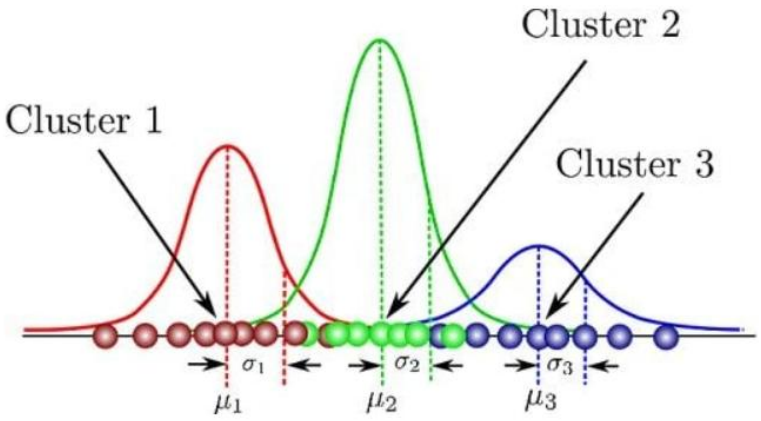


Figure 11: Visualization of Gaussian Mixture Model (GMM) Clustering: The diagram illustrates how data points are grouped into three distinct clusters, each represented by a Gaussian distribution with its own mean (μ) and variance (σ), highlighting the probabilistic approach.

Integration of GMMs with VAEs and ResNet-18:

Applying conventional clustering techniques like K-Means and even more advanced techniques like Variational Autoencoders (VAEs) to intricate, high-dimensional datasets has its drawbacks. Even while K-Means is a straightforward and effective method, it performs poorly on datasets with different cluster densities and shapes because it expects that all of the clusters are equally sized and spherical. Even though VAEs provide a probabilistic framework, they frequently lead to less clear or significant clusters because they fail to balance the trade-off between reconstruction quality and regularization and can create blurry outputs. Furthermore, because of their probabilistic character, which complicates the tuning process and reduces their flexibility in capturing the true underlying structure of the data, VAEs may experience mode collapse, a situation in which the diversity of generated data is restricted.

On the other hand, combining GMMs with ResNet-18 as the encoder provides a more potent method for grouping intricate information. Because of its deep residual connections, ResNet-18 can extract complex and detailed features from high-dimensional input and produce a latent space that is highly accurate. When modeled with GMMs, this feature-rich latent space provides a more accurate and detailed understanding of the distribution of the data. Because they represent the latent space as a combination of many Gaussian distributions, GMMs are excellent at capturing the diversity and complexity of the data. This allows them to better account for non-spherical and diverse cluster shapes. Because of this, the combination of ResNet-18 and GMMs offers a strong and adaptable approach to clustering, getting around the drawbacks of techniques like K-Means and VAEs and producing more precise and significant clusters in difficult datasets.

2.4 Review of State-of-the-Art Techniques

Recent advances in unsupervised learning have greatly improved the ability to cluster large, complicated, high-dimensional datasets, especially in the combination of autoencoders and Gaussian Mixture Models (GMMs). The demand for better clustering accuracy, resilience, and interpretability is driving these developments, as these properties are essential for a variety of applications including image analysis,

bioinformatics, and natural language processing. Despite these advancements, there are still certain obstacles to overcome, mainly in managing high data variability, striking a balance between model complexity and computational efficiency, and guaranteeing relevant feature extraction across many domains.

Deep Gaussian Mixture Models (DGMMs)

One of the most influential developments in this area has been the introduction of Deep Gaussian Mixture Models (DGMMs), as demonstrated by Dilokthanakul et al. (2016). To overcome the shortcomings of conventional GMMs, which frequently struggle with very high-dimensional data and non-linear separability, DGMMs combine the best features of deep learning and GMMs. A deep neural network, usually a convolutional neural network (CNN) or a multi-layer perceptron, acts as a feature extractor in DGMMs, converting the input data into a latent space. The underlying structure of the data is now more accessible due to this modification, which enables the GMM to accurately describe it as a mixture of Gaussians.

DGMMs' hierarchical feature extraction capabilities is useful in situations when the data shows intricate relationships, like in medical imaging. In this case, features at various scales and abstraction levels need to be taken into account at the same time. This hierarchical structure is well captured by DGMMs, which result in clusters that are easier to understand and more accurate. Even though DGMMs have improved clustering performance, they also add complexity to the model, which can cause computational inefficiencies and make the model harder to understand.

Enhancements to Autoencoder-GMM Frameworks

The incorporation of attention techniques into the autoencoder design is another significant improvement other than the addition of convolution layers. Many advanced deep learning models now rely heavily on attention mechanisms, which give varying weights to different regions of the input data. Attention methods in an autoencoder-GMM architecture enable the model to filter out noise and less significant information during encoding, focusing on the most relevant features. This focused attention improves the latent representation and produces more precise clustering, particularly for highly variable datasets.

Although these improvements have raised the bar considerably, they also draw attention to lingering problems. Convolutional and attention-based architectures, for example, can add complexity to models, making them more difficult to train and understand. Furthermore, there is still a need for more effective models that can produce good results at reasonable computing costs.

2.5 Bridging Current Gaps with ResNet-18 and GMM Integration:

The combination of ResNet-18 with GMMs presents an effective means to overcome these obstacles. ResNet-18 is a great choice for the encoder in an autoencoder-GMM model because of its deep residual connections, which offer a strong foundation for removing complex features from high-dimensional input. ResNet-18, compared to conventional deep architectures, reduces the vanishing gradient issue, allowing the network to train deeper layers and more precisely capture complicated data properties.

Combining ResNet-18 with GMMs can greatly improve clustering performance by giving GMMs a more feature-rich latent space to model. Essential data features, which are frequently lost in simpler architectures, are preserved in complicated, high-dimensional environments due to the residual connections in ResNet-18. This results in more relevant and accurate grouping, especially in fields like image analysis and bioinformatics where data unpredictability and complexity are significant.

In summary, while current state-of-the-art techniques like DGMMs and enhanced autoencoder-GMM frameworks have advanced the field of unsupervised learning, integrating ResNet-18 with GMMs offers a powerful solution to bridge existing gaps. This method provides a strong and effective framework for clustering complicated datasets by utilizing the probabilistic flexibility of GMMs with the deep feature extraction capabilities of ResNet-18. Therefore, it has a lot of potential to enhance the precision, comprehensibility, and applicability of clustering in several challenging fields.

Moreover, the data used to train the model contains both simulated and real images. Using both real and simulated images in training can help to reduce the issue of covariance shift. The model can learn a more generalized representation that more accurately represents the variability and many patterns that may be present in real-world events by combining the two types of input. This strategy has various advantages:

- Diverse Data Representation: Including both real and simulated images exposes the model to a broader range of input variations. Simulated images can be designed to introduce controlled variations in lighting, orientation, noise, and other factors that may not be present in the real dataset. This helps the model learn to handle a variety of conditions, making it more robust to changes in data distribution.

- Reduced Overfitting: By training the model on both real and simulated data, it is possible to keep it from overfitting to the distinctive features of the real-world training set. Simulated images can present more variability, forcing the model to prioritize more generalizable properties over memorizing precise patterns in real images.

- Improved Generalization: The model has a higher chance of making a good generalization to new, unobserved data as it has been trained on both actual and simulated images. This is because the model was trained on a dataset with a broader range of potential variables that it may face during deployment, lowering the influence of covariance shift.

- KL Divergence Reduction: The use of simulated images in training can help align the learned distribution closer to the real-world distribution, as measured by KL divergence. This is because the model has been exposed to a distribution that includes the types of variations and noise that it might encounter during deployment, leading to a better match between the training and deployment environments.

In summary, using a mix of real and simulated images during training can indeed help mitigate the effects of covariance shift by providing a more diverse and representative training dataset. This strategy can lead to a model that performs more consistently across

different environments and data distributions, ultimately improving its robustness and generalization capabilities.

2.6 Summary

We've discussed about the importance of these techniques, gone over the major advancements in the area, and pointed out gaps in the state of the art. This study emphasizes how these techniques can be better integrated with subsequent studies, especially when dealing with complicated and high-dimensional image collections.

We will go over the methodology utilized in this dissertation in the following chapter. This includes how the autoencoder was created and trained, how GMMs were applied to the latent space representations, and how the suggested approach was evaluated.

CHAPTER 3: RESEARCH APPROACH

This chapter provides a breakdown of the research approach employed to achieve the study goals outlined in the previous chapter. This chapter presents in detail the chosen methodology and how it relates to this particular project. Each step's implementation is described in relation to the project's goals, showing how the data collecting and analysis process flow together seamlessly.

3.1 Selected Methodology

Due to the nature of the project related to working with images, it was decided to use the methodology developed by the Cross-Industry Standard Process for Data Mining. The flexible and iterative form of the CRISP-DM data mining methodology makes it very appropriate for data-driven projects especially those which involve data exploration, model building, and its tuning.

CRISP-DM Framework

1. Business Understanding: The project's primary aim is to cluster high-dimensional image data using unsupervised learning techniques.

2. Data Understanding: The dataset used in this study is an open dataset from kaggle which consist of both real and simulated animal images, divided into training and testing sets.

3. Data Preparation: Data preparation included the reduction in the dimension of images capturing the images to normalizing the pixel intensity values and converting the images into tensors for neural network processing.

4. Modelling: The core modelling step involved designing an autoencoder to reduce the dimensionality of the image data, followed by applying a Gaussian Mixture Model (GMM) to cluster the latent representations. This integration allows for easy clustering of all high-level representations in one space, instead of extending it to the higher-order dimensional resolution to accommodate all the latent representations.

5. Evaluation: The models were evaluated using metrics such as reconstruction loss, KL divergence, and visual inspection of original and reconstructed images. Cross-validation and model selection criteria like BIC and AIC were also applied to ensure robustness. The whole model was finally tested using unseen images data.

6. Deployment: While not directly covered in this dissertation, this phase would involve integrating the models into a production system, where they can be applied to new, unseen data in various practical scenarios, such as real-time image analysis, automated medical diagnostics, or intelligent robotics. Deployment also entails setting up continuous monitoring and updating mechanisms to ensure the models maintain high performance over time.

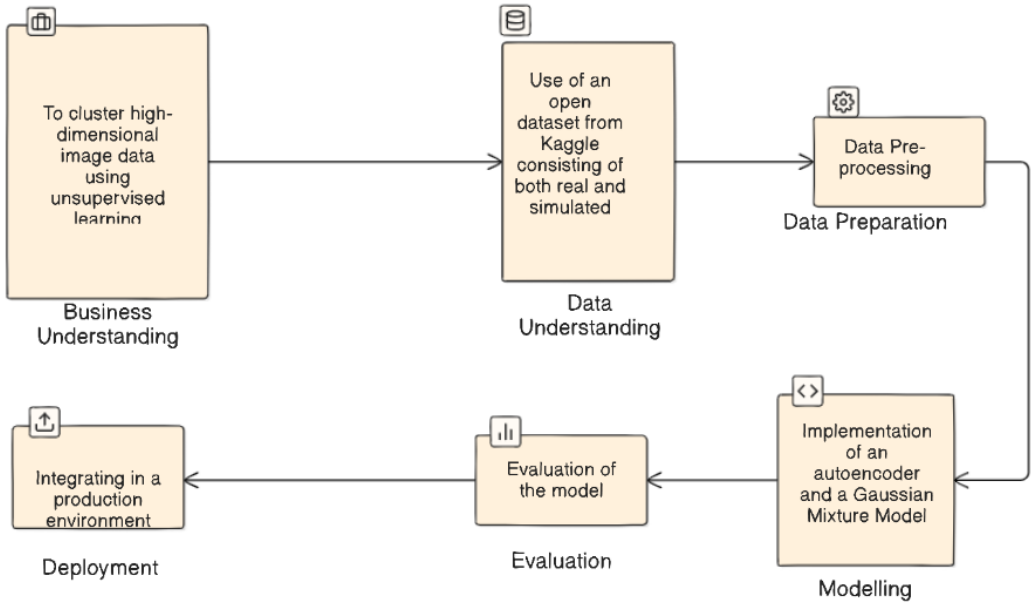


Figure 12: CRISP-DM Process

3.2 Summary:

This chapter provided a comprehensive overview of the research approach adopted to achieve the objectives of this study, with a focus on the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology. The chapter began by outlining the rationale for selecting CRISP-DM, emphasizing its flexibility and suitability for data-driven projects, particularly those involving complex image data. Each stage of the CRISP-DM framework was then detailed, from understanding the business objectives and the data, through to data preparation, modeling, and evaluation.

In the next chapter, we will delve into the detailed implementation of the methodology, covering the specific steps and techniques used to prepare the data, build the models, and evaluate their performance in depth.

CHAPTER 4: DATA ANALYSIS

In this chapter, we will explore the detailed implementation of the research approach discussed in the previous chapter. This will involve a comprehensive explanation of the steps taken to prepare the data, construct the models, and assess their performance. The chapter will highlight the specific techniques and tools employed at each stage of the project, offering a clear and thorough account of how the CRISP-DM framework was practically applied to meet the study's objectives.

4.1 Business Understanding

The main goal of this project is to group high-dimensional image data into meaningful clusters using unsupervised learning techniques. This means using machine learning models to automatically find and categorize patterns in a set of images without having any labels or predefined categories. The focus is on developing a method that can handle complex and high-dimensional data, simplifying it while keeping the important features that help in clustering. The aim is to build a strong model that can accurately separate different groups within the dataset, which could be useful for tasks like image classification, detecting anomalies, or organizing images automatically.

4.2 Data Collection

The study's data was sourced from an open dataset available on Kaggle. The dataset includes both actual and simulated photographs of animals, which makes it a perfect option for testing how well unsupervised learning methods handle complicated and varied image data. The dataset can be found in this link: https://www.kaggle.com/datasets/birdy654/cifake-real-and-ai-generated-synthetic-images

Dataset Composition:

Real Images: These are true photos of animals taken in a variety of settings, including varied lighting, perspectives, and habitats. Effective clustering requires that the model encounter a wide range of characteristics, which these photos' diversity assures.

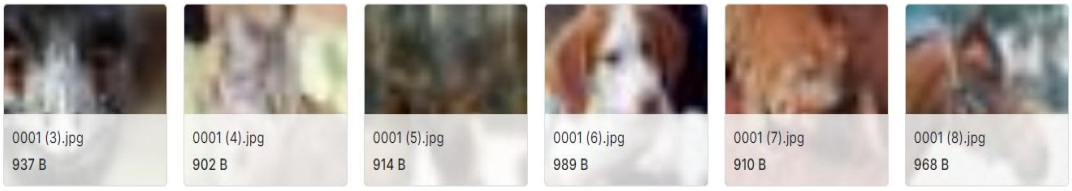


Figure 13: Samples of Real images

Fake Images: Using complex visual techniques, simulated images are also included in the collection. These photos offer a controlled setting in which particular elements can be emphasized or downplayed, enabling the model to pick out clear patterns that can be difficult to discern in images from the actual world.

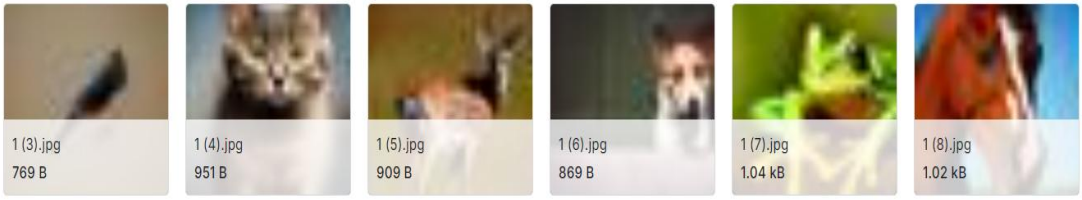


Figure 14: Samples of Fake Images

4.3 Data Preprocessing and Custom Dataset

Data preprocessing is the initial step that enables the model to take in the raw input image data. To make it easier for the neural network to process the data, this stage entails a number of transforms to make the images all the same size, shape, and dimension.

Image Preprocessing Pipeline:

- Resizing Images: Since all the images in the input dataset different aspect ratios and sizes , the model will not be able to train properly. This issue is rectified by resizing the images first. Using the Resize transform from the torchvision.transforms module, all images are scaled to a set dimension of 128x128 pixels in order to standardize the input.

- Normalization: Following the scaling process, the photos' pixel values—which normally fall between 0 and 255—are normalized. The process of normalization involves first scaling the pixel values to a range of 0 to 1, and then making adjustments based on the dataset's average value and distribution of values. By ensuring that the input is identical in range and centered around zero, this procedure helps the neural network train more quickly and minimizes the likelihood of issues like erratic updates or delayed learning.

- Conversion to Tensors: The images which are loaded in the format of PIL images are transformed to tensors through the use of the ToTensor transform. This specific transformation is needed as Pytorch works with tensors which are more sophisticated multi-dimensional matrices that hold data in the type suited for the usage of a GPU

Custom Dataset Class and Data loader

In order to facilitate loading of images and their pre-processing efficiently, a separate dataset class, CustomImageDataset, has been created. This uses PyTorch's Dataset subclass that enables uniform application of preprocessing operations over the entire dataset. This particular class takes care of loading images by taking a folder with image files and transforms them by resize, normalize and convert to tensor.
After the dataset is formulated, it is integrated with DataLoader in PyTorch which helps in the efficient management of a large amount of data using automatic batching, shuffling, and loading of data in parallel. This setup ensures that the model is trained on properly prepared data, thus maximizing the training efficiency.

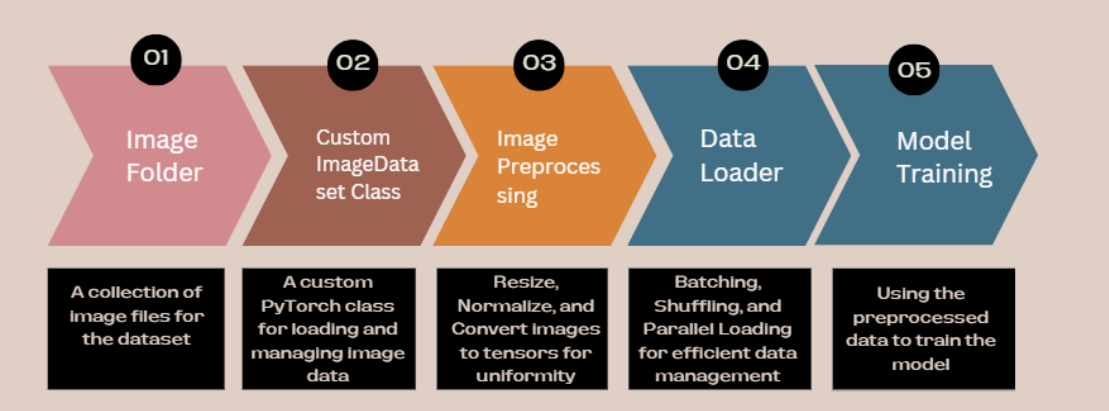


Figure 15: Data Preprocessing

4.4 ResNet 18 Autoencoder

- *ResNet Encoder:*

The encoder is based on the ResNet18 architecture, a convolutional neural network (CNN) model known for its residual connections. These connections allow deeper networks to be trained effectively by mitigating the vanishing gradient problem. The encoder extracts high-level features from the input images, which are then used for further processing. Its implementation is as follows:

Residual Blocks: The ResNet18 encoder comprises several residual blocks, each containing two convolutional layers followed by batch normalization and ReLU activation. The key idea behind a residual block is to allow the input to bypass the convolutional layers via a shortcut connection (identity mapping), adding the original input back to the output of the convolutional layers:

$$y = F(x, \{W_i\}) + x$$

Here, x is the input, $F(x, \{W_I\})$ represents the convolutional operations, and y is the output of the residual block.

Downsampling: The encoder reduces the spatial dimensions of the feature maps progressively using convolutional layers with stride 2, compressing the input image into a lower-dimensional latent space while retaining critical features.

Layer Structure: The encoder begins with a convolutional layer, followed by a series of residual blocks, with downsampling applied at specific stages:

$$z = Encoder(x) = ResNet18(x)$$

where z is the latent representation of the input image x.

Each residual block can be formulated as:

$$y = ReLU\left(BN\left(W_2 * ReLU\left(BN(W_1 * x)\right)\right)\right) + x$$

Where:

- W_1 and W_2 are the convolutional kernels,
- $*$ denotes the convolution operation,
- BN stands for batch normalization,
- ReLU is the rectified linear unit activation function.

➢ *ResNet Decoder:*

The decoder reconstructs the original image from the encoded feature representation generated by the encoder. The design of the decoder mirrors the encoder but in reverse, using transpose convolutional layers to upsample the feature maps, eventually restoring the original image dimensions.

Transpose Convolutions: The decoder employs transpose convolutional layers (also known as deconvolutions) to increase the spatial dimensions of the feature maps. These layers are interspersed with residual blocks to further refine the features.

Layer Structure: Similar to the encoder, the decoder uses a sequence of upsampling operations followed by residual blocks:

$$\hat{x} = \text{Decoder}(z) = \text{ResNet18}^T(z)$$

where $\hat{x}$ is the reconstructed image from the latent vector z.

Each transpose convolution operation in the decoder can be expressed as:

$$\hat{y} = \text{ReLU}\left(\text{BN}\left(W_2^T * \text{ReLU}\left(\text{BN}(W_1^T * z)\right)\right)\right)$$

Where (W_2^T) and (W_1^T) are the transpose convolutional kernels.

➢ *Training the Autoencoder:*

The autoencoder is trained to minimize the mean squared error (MSE) between the input image and its reconstruction:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2$$

Where:

- N is the number of images,
- x_i is the original image, and
- $\hat{x}_i$ is the reconstructed image

Optimization: An Adam optimizer is used to minimize the loss function, with the learning rate set to a small value (e.g., 0.001) to ensure stable convergence.

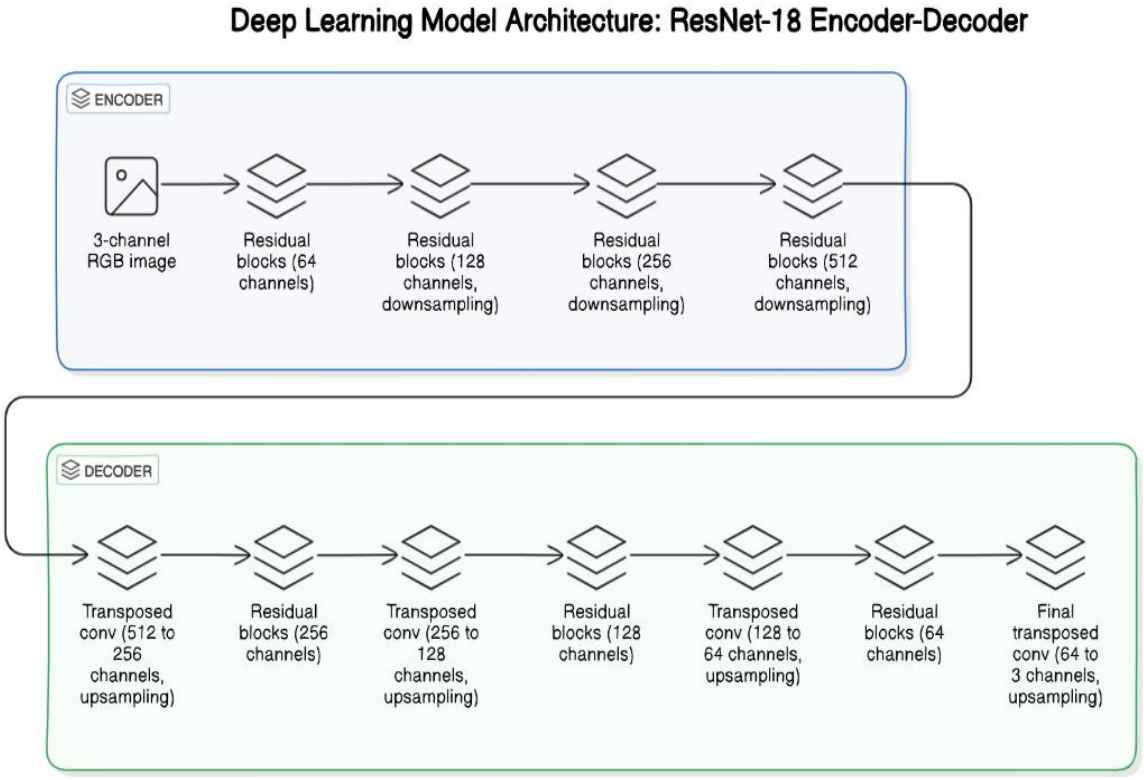


Figure 16: ResNet 18 Autoencoder Architecture

4.5 Gaussian Mixture Model

Due to the high dimensionality of the latent vectors produced by the ResNet18 encoder, directly applying a Gaussian Mixture Model (GMM) on these high-dimensional features posed significant computational challenges, particularly in terms of memory consumption. My system encountered memory errors when trying to allocate the large covariance matrices required by GMM in this high-dimensional space. To overcome this limitation, I employed t-SNE (t-distributed Stochastic Neighbor Embedding) to reduce the dimensionality of these latent vectors before clustering them with GMM.

t-SNE is a dimensionality reduction technique specifically designed to condense high-dimensional data into 2 or 3 dimensions while preserving local structure. This makes it not only useful for visualization but also for reducing the computational load of subsequent processes like clustering. By reducing the latent vectors to a much lower dimension, t-SNE made it feasible to apply GMM on my system without running into memory issues.

After applying t-SNE, the reduced-dimensional features are passed to a GMM for clustering. GMM assumes that the data points come from a mixture of Gaussian distributions, each defined by its own mean and covariance matrix. This method allows for soft clustering, where each data point can have partial membership across multiple clusters, making it a flexible approach for unsupervised learning.

- *Dimensionality Reduction with t-SNE:*

To manage the high memory demand, t-SNE was applied to reduce the latent vectors from thousands of dimensions down to 2 or 3 dimensions. This reduction alleviated the memory burden, making it possible to perform clustering on my system. t-SNE works by minimizing the divergence between the probability distributions of the pairwise similarities in both the original high-dimensional space and the reduced low-dimensional space.

- *Clustering with Gaussian Mixture Models (GMM):*

Once the dimensionality was reduced using t-SNE, the lower-dimensional latent vectors were clustered using GMM. This step became computationally feasible after the dimensionality reduction. GMM models the data as a mixture of several Gaussian distributions, each described by its mean and covariance matrix. The mathematical representation of GMM is:

$$p(z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(z|\mu_k, \Sigma_k)$$

where:
K represents the number of clusters,
π_k are the mixture weights,
μ_k and $\sum_k$ are the mean and covariance of the k-th Gaussian component

4.6 Evaluation

The evaluation of the proposed ResNet18 autoencoder with Gaussian Mixture Models (GMM) for unsupervised clustering of high-dimensional image data involves several steps to assess the effectiveness and quality of the model. The evaluation includes analyzing the training process, visualizing the reconstructed images, comparing the real and simulated data distributions, and validating the clustering performance. Below are the key evaluation methods used in this project.

- *Training Loss Over Epochs:*

The autoencoder's training progress is monitored by tracking the loss function throughout several epochs. To see how the model converges, the Mean Squared Error (MSE) loss—a measure of the discrepancy between the original images and their reconstructions—is plotted over the course of the epochs. To make sure that the loss drops consistently over the course of epochs, a plot of the training loss is created. This indicates that the model is successfully learning to reconstruct the images. The plot helps determine the ideal number of training epochs and provides insight into the model's performance.

- *Reconstruction Quality:*

The ability of the model to reconstruct images from the latent space is assessed in order to evaluate the autoencoder's quality. The original input images and the reconstructed images are compared. A grid of images is shown, with the actual images in the top row and the corresponding reconstructions produced by the decoder in the bottom row. This qualitative evaluation aids in visually examining the reconstructions' fidelity and pointing out any notable differences.

- *Latent Space Representation:*

The encoder's latent space representation is essential to the subsequent clustering step. The evaluation includes visualizing the encoded images in the latent space. The encoded representations of the original images and their encoded representations are shown side by side. This allows easy evaluation of how successfully the encoder has compressed the images' key characteristics.

- *Gaussian Mixture Models (GMM) Evaluation*:

The GMM is applied to the reduced-dimensional latent vectors to perform clustering. The evaluation of GMM involves several key aspects:

KL Divergence: The difference between the distributions of actual and simulated data is measured using the Kullback-Leibler (KL) divergence. The KL divergence gives an indication of the degree of similarity between the two distributions by taking a sample from the GMM fitted to the real data and comparing it to the simulated data. A smaller KL divergence suggests that the data's underlying structure is being adequately captured by the GMM.

Log Probability Distribution: The log probabilities of samples under both the real and simulated GMMs are computed and visualized using histograms. This comparison helps in understanding how well the GMM models the data distributions.

BIC and AIC Scores: For both the real and simulated data GMMs, the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC) are computed. These metrics penalize for model complexity while evaluating the model's goodness of fit. Reduced AIC and BIC values indicate a more accurate model fit that strikes a compromise between complexity and precision.

Cross-validation: The robustness and generalization capacity of the GMMs was evaluated using cross-validation. The cross-validation scores provide insight on how stable the model is across various data subsets.

Cluster Visualization: The clusters identified by GMM in the latent space are visualized using scatter plots. The centroids of the Gaussian components (GMM means) are also plotted to illustrate the separation between clusters. This visualization helps in assessing the clustering quality and identifying any overlapping clusters.

- *BIC Scores Across Multiple Runs:*

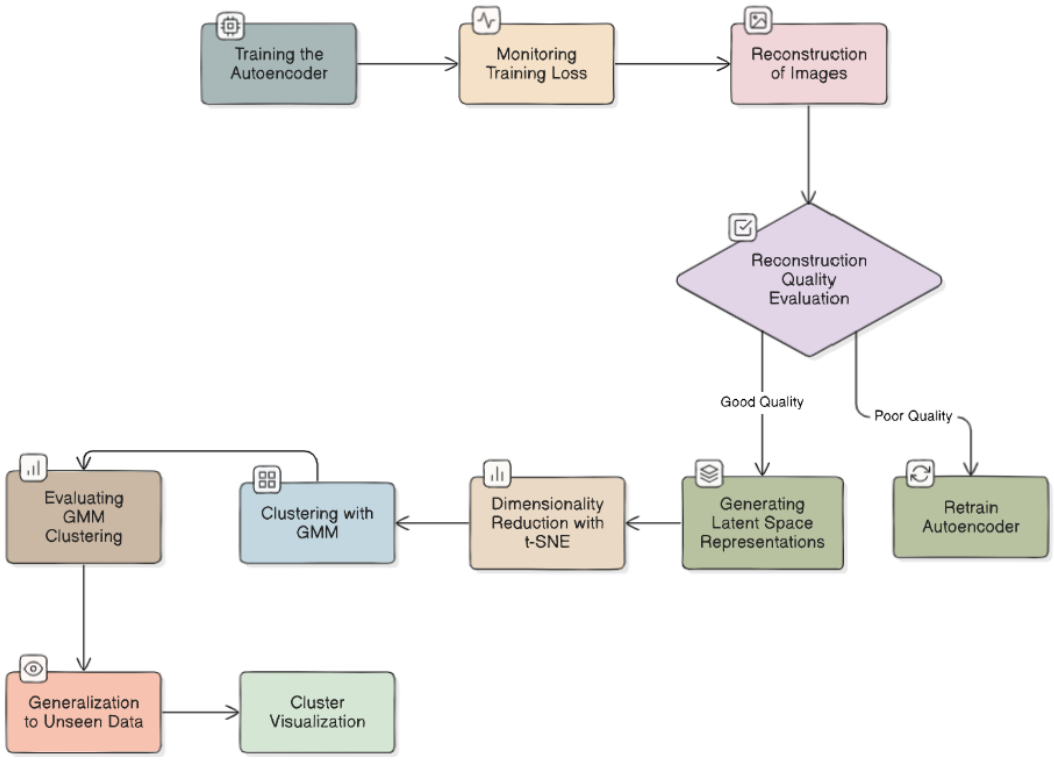
To ensure the stability and reliability of the GMM clustering, BIC scores are evaluated across multiple runs with different random seeds. The mean and standard deviation of the BIC scores over multiple runs are calculated to assess the consistency of the GMM's performance. A low standard deviation indicates that the

model's performance is stable across different initializations.

- *Generalization to Unseen Data:*

To evaluate the model's ability to generalize to new data, a separate set of unseen images was introduced. The key objectives were to assess how these new images were encoded by the autoencoder and how effectively they were clustered by the GMM. The unseen images were passed through the trained encoder to obtain their latent space representations. This step allowed us to observe how the encoder generalizes to new data that it has not seen during training. The latent representations of the unseen images were then passed to the GMM, which assigned each image to one of the clusters identified during training.

The distribution of the unseen images across the clusters was analyzed to ensure that the new data points were reasonably assigned to the clusters. This helps in verifying whether the clusters identified during training are broad enough to accommodate new variations in the data. A visual inspection was conducted by plotting the encoded representations and their corresponding cluster assignments. This visualization provided insights into how well the unseen data fits into the existing clusters.



4.7 Summary:

The methodology chapter provided a detailed explanation of the steps taken to design and implement a ResNet18 autoencoder combined with Gaussian Mixture Models (GMM) for the purpose of unsupervised clustering of high-dimensional image data. The chapter outlined the architecture and training of the autoencoder, the application of t-SNE for dimensionality reduction, and the subsequent clustering process using GMM. Additionally, the evaluation methods used to assess the model's

performance, including training loss analysis, reconstruction quality, and generalization tests on unseen data, were discussed in depth. These evaluation techniques confirmed the robustness of the model in capturing and clustering meaningful features within the data.

In the next chapter, we will delve into the data analysis phase, where we will demonstrate proficiency in applying the data analysis techniques discussed in this chapter. The chapter will provide a thorough analysis of the results, ensuring that the methodologies are effectively translated into meaningful insight

CHAPTER 5: DISCUSSION

This chapter critically examines the results of the research, linking them back to the objectives and aims discussed earlier in the dissertation. By comparing the outcomes of the experiments with existing research and evaluating the strengths and weaknesses of the proposed methodology, this chapter aims to provide a comprehensive understanding of the efficacy of the ResNet-18 autoencoder combined with Gaussian Mixture Models (GMMs) for clustering high-dimensional image data. The discussion will also delve into challenges like covariance shift, the importance of modeling covariance structure with GMMs, and the implications of these findings for real-world applications, particularly in sim-to-real scenarios.

5.1 Analysis of Model Performance

This dissertation's main goal was to develop and assess a technique for clustering high-dimensional picture data by combining ResNet18 autoencoders with Gaussian Mixture Models (GMMs). The effectiveness of the suggested approach was evaluated using the main findings from the evaluation phase, which included training loss, reconstruction quality, latent space representation, and GMM clustering performance.

5.1.1 Training Loss Over Epochs

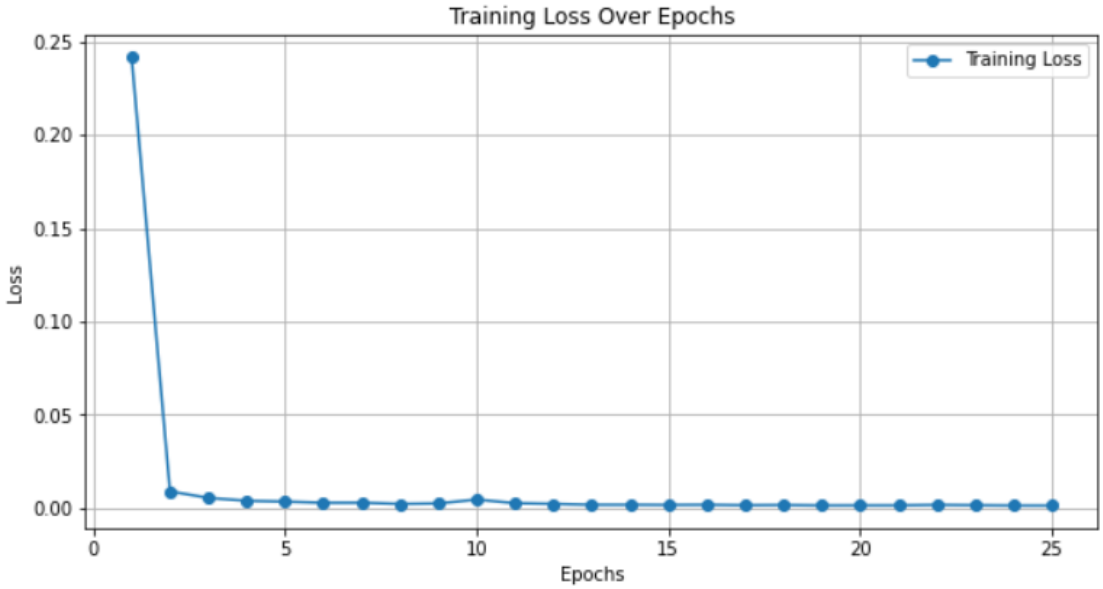


Figure 17: Training loss over epochs

The provided graph shows the autoencoder model's training loss over a period of 25 epochs, as measured by the Mean Squared Error (MSE) between the original and reconstructed images. This graph's distinctive aspect is how quickly loss decreased in the first few epochs. As is common for an untrained model that hasn't yet learned how to effectively reconstruct the input images from their latent representations, the loss begins at a somewhat high value, approximately 0.25.

The loss drastically decreases during the second epoch, indicating that the autoencoder

quickly picks up on the key characteristics and patterns in the input data. This quick reduction is positive since it demonstrates how well the model is picking up mappings from the high-dimensional image space to the lower-dimensional latent space and back again. The use of the ResNet18 architecture in the encoder plays a crucial role here, as its residual connections help in efficiently propagating the gradients back through the network, thereby overcoming the vanishing gradient problem commonly associated with deep networks.

The loss decreases with training, though at a much slower rate, and eventually stabilizes close to zero. The presence of this plateau indicates that the model has successfully reduced the reconstruction error for the training set. The final low loss value indicates that the autoencoder is highly proficient at reconstructing the images, meaning that it has learned a compact and accurate representation of the data in the latent space.

This steady decrease and final stabilization of the training loss is a positive development. It shows that the model is not only learning efficiently but also avoiding common errors like overfitting, which would have been shown by fluctuations in the loss curve or a substantial divergence between the training and validation losses.

This result is supported by the model's underlying implementation. By using the Adam optimizer with a learning rate of 0.001, the model may move steadily without experiencing significant fluctuations in the loss since it strikes a good compromise between convergence speed and stability. Because it explicitly penalizes variations in pixel values between the original and rebuilt images, the MSE loss function utilized here is especially well-suited for image reconstruction tasks, encouraging the autoencoder to generate high-fidelity outputs. The graph would be more gradual if a lower learning rate learning rate is used.

In summary, the training loss graph illustrates the model's capacity to swiftly learn and precisely reconstruct high-dimensional picture data, as well as verifying the efficacy of the ResNet18-based autoencoder and offering insights into the training dynamics. The autoencoder appears to have performed well in terms of limiting reconstruction error, as evidenced by the final low loss number, which highlights the training process's success.

5.1.2 Reconstruction Quality

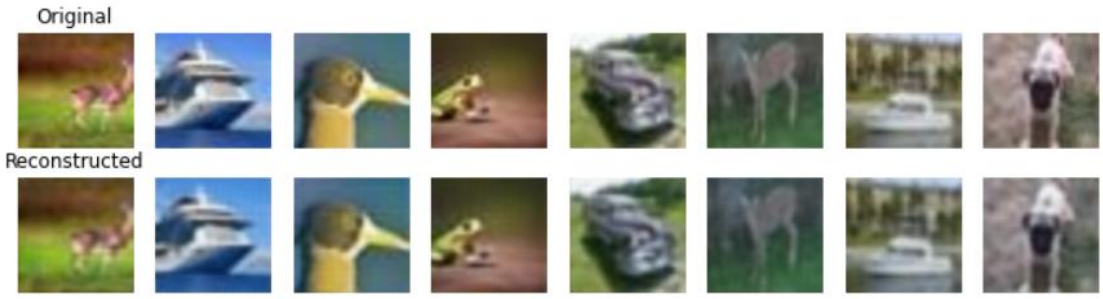


Figure 18: Comparison of original and reconstructed images

A comparison of the original photos (top row) and the autoencoder's reconstructions of those images (bottom row) is shown in the above figure. Reconstructed and original images show significant visual similarities, suggesting that the autoencoder has learned to encode and decode high-dimensional image data with remarkable fidelity.

When examined closely, the reconstructions maintain the main characteristics of the original photos, including the main details, color distribution, and overall structure. For example, the reconstructed image of the animal preserves important details necessary for precise grouping, such as its posture and the surrounding environment.

This excellent reconstruction quality is evidence of the effectiveness of the encoder's ResNet18 design. ResNet18's residual connections enable the model to conserve significant features over several layers, guaranteeing that the images' crucial information are kept intact even after being compressed into a latent space with a reduced dimension. The capacity to generate such precise reconstructions suggests that the latent space representation is information-rich and compact, hence achieving a good trade-off between fidelity and compression.

Furthermore, it appears that the autoencoder has caught both the visual content and the underlying structure of the images based on the little variations between the original and reconstructed images. This is especially crucial for the next clustering assignment that uses Gaussian Mixture Models (GMMs), since the accuracy of the clustering is directly impacted by the quality of the latent space. An image that is well-reconstructed suggests that the latent space representation accurately reflects the original data, which makes it a perfect input for GMM. GMM uses these representations to distinguish between different clusters in the dataset.

5.1.3 Latent Space Representation

Figure 19: Original and Encoded images

The original images and their corresponding encoded representations within the latent space are shown visually in the above figure. Five original photos are shown in the top row, and the encoded images—which are the autoencoder's lower-dimensional, compressed representations—are shown in the bottom row.

The original images and their corresponding encoded representations within the latent space are shown visually in the above figure. Five original photos are shown in the top row, and the encoded images—which are the autoencoder's lower-dimensional, compressed representations—are shown in the bottom row.

Each encoded image shows the high-dimensional input transferred into a smaller latent space by the autoencoder. The reason these encoded images are important is that they hold the distilled data required for the Gaussian Mixture Models (GMMs) clustering work that follows. The encoded representations, however greatly abstracted, preserve the salient characteristics of the original images despite the reduction in dimensionality.

The autoencoder retrieved significant characteristics that correlate to areas of high and low intensity in the encoded images, resulting in different patterns. For instance, in "Encoded 0," the activity concentration along the borders indicates that the autoencoder has successfully captured the important color transitions and boundaries of the original image. In the same way, the other encoded images exhibit distinctive patterns that mirror the essential characteristics seen in the original images that the encoder recognized.

These latent representations are meaningful abstractions of the given data. In order to distinguish between various classes or groups within the dataset, the encoder has trained to concentrate on particular features of the images. The encoder's ability to distinguish between the numerous features present in the source image is demonstrated by the fact that different images provide noticeably different encoded patterns.

The subsequent GMM-based clustering depends critically on the quality and structure of the latent space. GMMs can create precise and significant clusters because a well-structured latent space ensures that similar images are close to one another in this reduced space. The autoencoder's capacity to identify these similarities and differences is reflected in the patterns found in the encoded images, which will help with efficient grouping.

5.1.4 Evaluation of KL Divergence and Log Probability Distributions

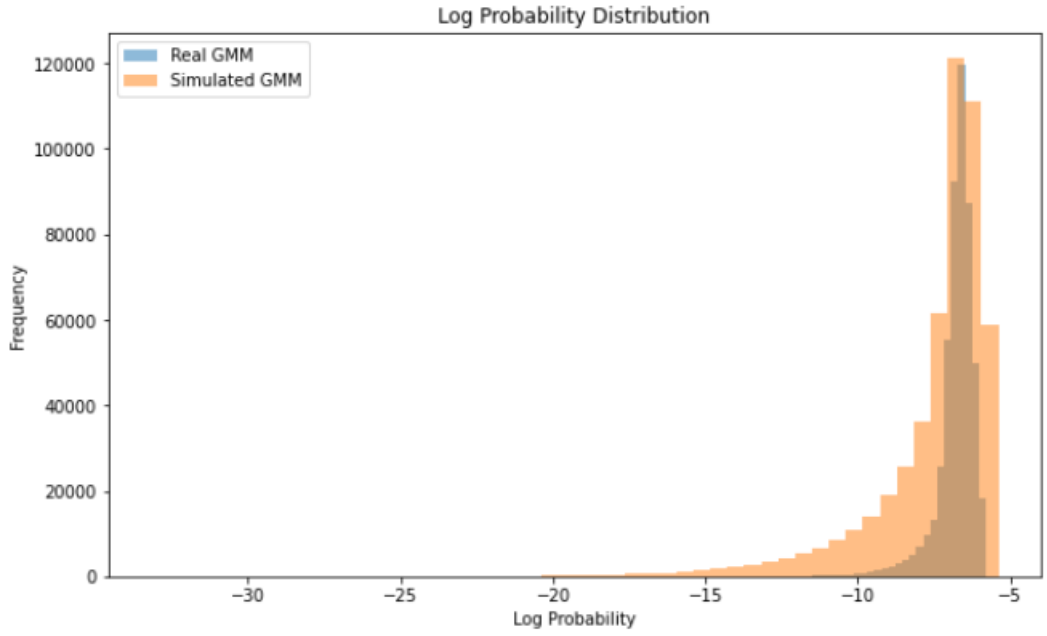


Figure 20: Log Probability distributions of Real and Simulated data

The histogram in the above plot compares the log probability distributions of the GMM applied to the simulated and real data. Given that both real and simulated data were used in the GMM's training process, the overlap between the two distributions indicates that the model has been successful in capturing the underlying structure shared by the two datasets. This overlap is significant because it shows that the GMM can accurately represent the real data and can also generalize to the simulated data, which may have distinct features and variations.

But the slight shift in the distribution peak of the simulated data as compared to the real data highlights the fundamental distinctions between these two datasets. This disparity draws attention to the difficulties in training models on mixed datasets, where the objective is to prevent the model from overfitting to a certain distribution while ensuring that it generalizes across other data types.

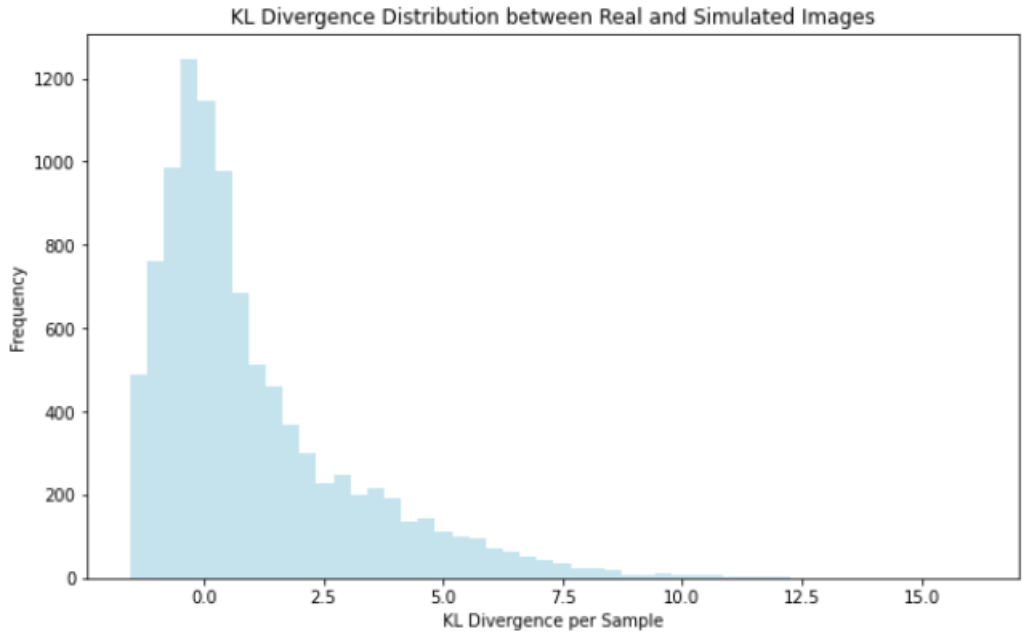


Figure 21: KL Divergence Distribution

The KL divergence between the distributions of the simulated and real photos is shown in the above figure.The spread of KL divergence values, with most values being relatively low, indicates that the GMM has generally succeeded in capturing the overall structure of both datasets. The learnt distribution of the model seems to closely resemble the actual distribution of the data, based on the comparatively low KL divergence values. However, the presence of higher divergence values points to specific instances where the model's learned distribution deviates from the true distribution, potentially indicating areas where the model might have overfitted to certain features more prevalent in one dataset over the other.

```
Approximate KL Divergence: 0.6978
```

The difference in the real and simulated distributions is further quantified by the approximate KL divergence value of 0.6978. This result shows a moderate divergence, meaning that even while the GMM works well on both datasets, there is still some disparity that might be related to the different features of the data or the difficulty of

concurrently collecting the two distributions.
Overall, these results highlight the effectiveness of using a GMM trained on a combination of real and simulated data. The comparatively low KL divergence and the overlap in log probability distributions imply that the model has effectively developed a generic representation that can handle the variability in both forms of data. To reduce the remaining divergence and increase the model's robustness and generalization abilities, the analysis also identifies potential development areas, such as through enhancing the training procedure or utilizing domain adaption strategies.

5.1.5 Cluster assignment and Visualisation

For both the real dataset and the simulated dataset, the GMM generated distinct cluster assignments. The following labels were applied to the real data clusters: [84, 68, 53, 69, 71, 83, 52, 86, 67, 50]. In a similar manner, the labeled clusters of simulated data were [75, 99, 84, 41, 88, 103, 92, 47, 29, 25]. These assignments show how the patterns and features discovered from the latent space were used by the GMM to classify the data into discrete clusters.

```
Cluster assignments for Real Data: [84 68 53 69 71 83 52 86 67 50]
Cluster assignments for Simulated Data: [ 75  99  84  41  88 103  92  47  29  25]
```

The visualizations offer an informative view into the degree to which the structure of the simulated and real data has been accurately captured by the GMM, The scatter plots display the data points in the two-dimensional t-SNE space, with different colors representing the real and simulated datasets.

- **Real Data and GMM Means:** The below scatter plot illustrates the real data points (in blue) along with the centroids of the Gaussian components identified by the GMM (marked with red stars). This visualization shows how the GMM has identified distinct clusters within the real dataset, with the red stars representing the mean positions of these clusters. The distribution of the real data points around these means indicates how well the GMM has modeled the underlying structure of the data.

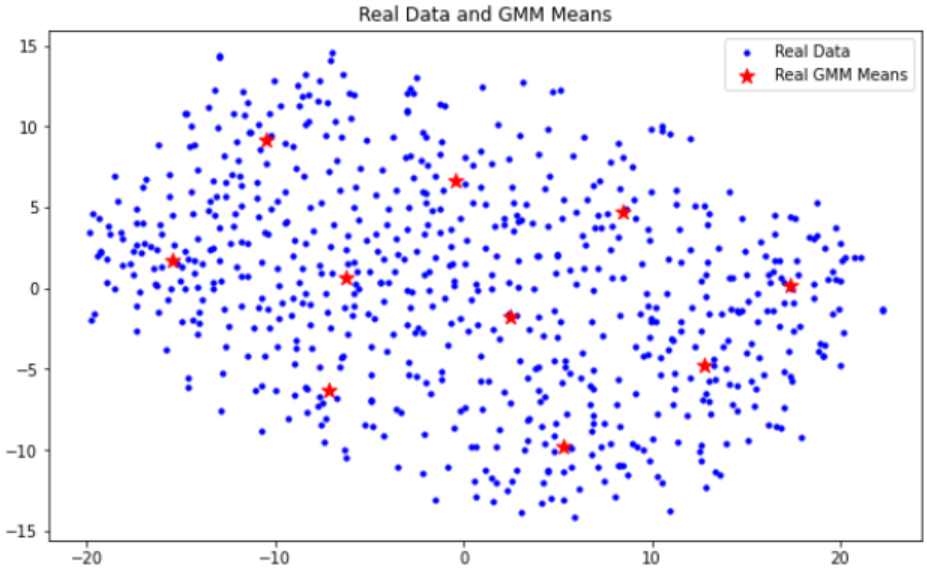


Figure 22: Real Data points and gmm means

- **GMM Cluster Means and Simulated Data:** The below scatter plot shows the GMM cluster means (shown by yellow stars) and the simulated data points (shown in green). Similar to the real data, this plot shows the clustering of the simulated data in the t-SNE-reduced space. The clusters detected by the GMM are indicated by the yellow stars, which represent the centroids of the Gaussian components for the simulated data.

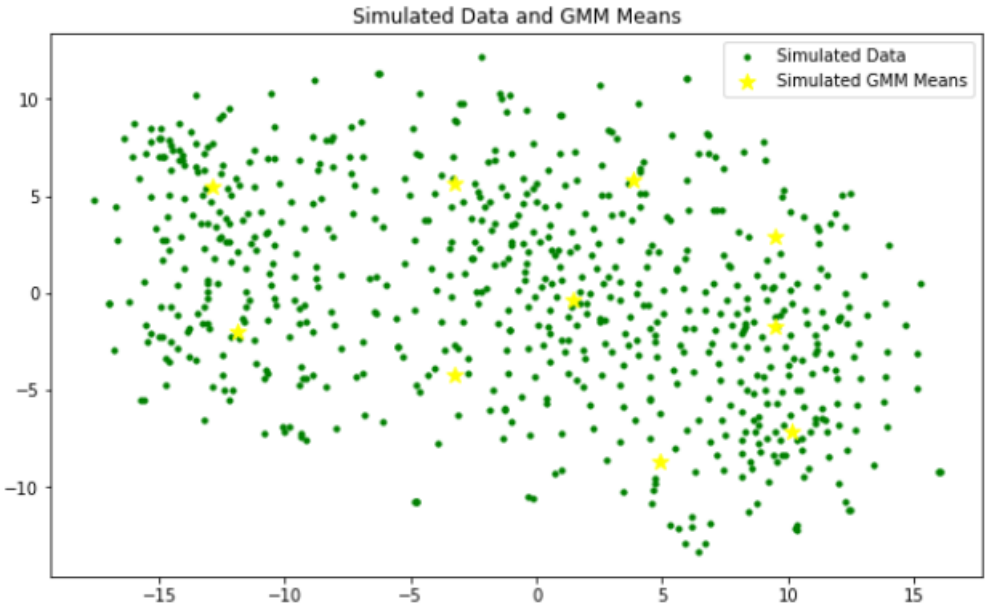


Figure 23: Simulated data points and gmm means

These visualizations offer a clear illustration of the GMM's capacity to represent intricate, high-dimensional data and group it into meaningful clusters. In both the real and simulated datasets, the closeness of data points to their corresponding cluster means indicates that the GMM has successfully captured the underlying structure of the data. Furthermore,

the scatter plots show that although the real and simulated data clusters are quite different from one another, there are some spots where the two datasets' clusters overlap, especially in areas where they converge. This overlap may point to common characteristics or similarities between the simulated and real-world images' latent space representations, which the GMM used to determine comparable cluster labels.

5.1.6 Visualisation of different Clusters

The scatter plots show how the data points have been clustered by the Gaussian Mixture Model (GMM) in a two-dimensional t-SNE-reduced space in a clear and comprehensive manner for both actual and simulated data. Based on the latent space representations produced by the ResNet18 autoencoder, each color denotes a distinct cluster, demonstrating the accuracy of the GMM in classifying the data into meaningful groups.

- Analysis of Real data clusters:

  The real data's scatter plot clearly shows the boundaries separating the several clusters. The plot shows that the GMM has effectively detected unique regions in the t-SNE-reduced space that correspond to different clusters of the original high-dimensional data. The ten clusters (labelled 0 to 9) are represented by different colours. The real data points are categorized based on underlying qualities that the GMM has discovered from the latent space, as evidenced by the clustering pattern.

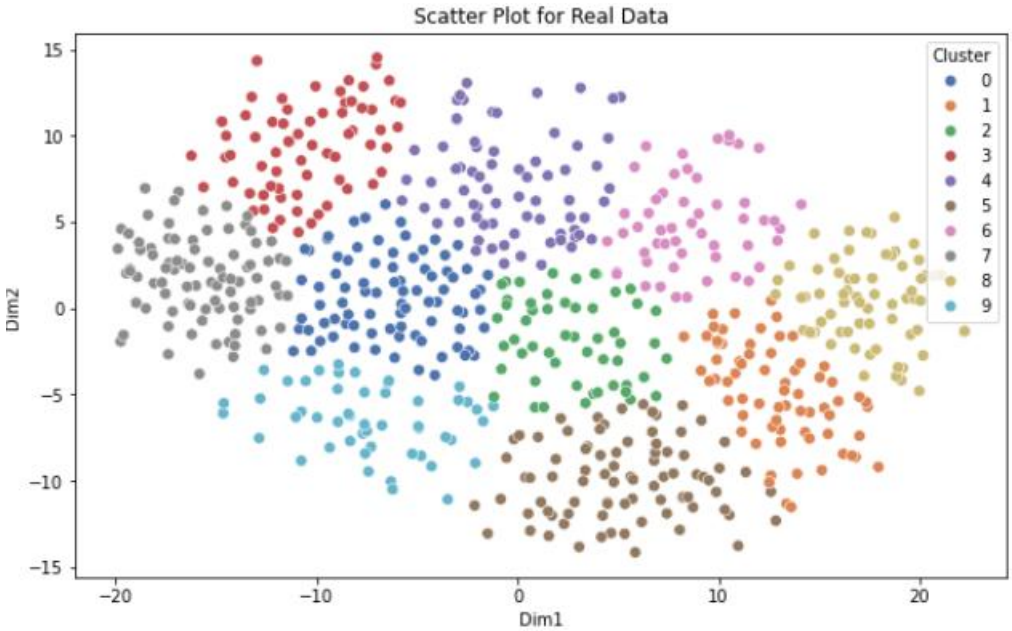


Figure 24: Scatter plot for real data

Clusters show distinct boundaries and are rather compact, with little overlap amongst them. This indicates that the ResNet18 autoencoder's latent space successfully captures the essential characteristics required to distinguish between various classes within the original dataset. The diversity in cluster distributions suggests that the GMM

has performed a good job of adjusting to the many structures present in the data, capturing both spherical and more complexly shaped clusters.

- Analysis of Simulated data clusters:

  A similar structure can be seen in the scatter plot of the simulated data, which features clearly colored and well-separated clusters. The general layout and distribution of the clusters in the two-dimensional space show some similarities between the simulated and real data clusters. The distribution of the clusters does differ noticeably, though, with certain clusters in the simulated data being more extended or scattered than their counterparts in the real data.

  These differences can be attributed to the inherent variations between the real and simulated datasets. The simulated data's characteristics, which can include less variability or distinct patterns, result in somewhat different clustering conclusions even though the GMM has been trained on both datasets and is capable of modeling the data distribution.

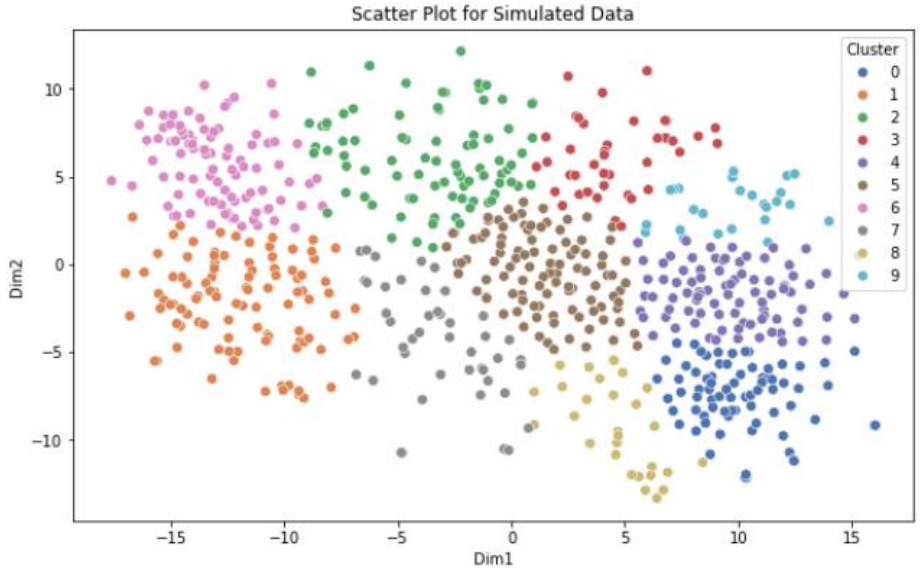


Figure 25: Scatter plot for simulated data

By comparing the scatter plots for real and simulated data, we can observe how the GMM has managed to capture the underlying structure of both datasets. Because the real images have richer and more complicated features, the clusters in the real data seem to be a little bit more compact and well-defined. On the other hand, the simulated data clusters show greater dispersion even if they are still recognizable, which may be a reflection of the simulated features' artificial or more uniform character.

This comparison demonstrates how well-suited the GMM pipeline and ResNet18 autoencoder are to handle various data kinds. The GMM is a potent tool for unsupervised clustering in situations involving both real and synthetic data because of its capacity to form meaningful clusters on both real and simulated datasets, indicating that the model has generalized successfully across the two domains.

Overall, the effectiveness of the clustering strategy is visually confirmed by these scatter plots. Clusters in real and simulated datasets clearly separate from one another, indicating that the ResNet18-GMM model can discriminate between classes in high-dimensional picture data. For jobs like image analysis, anomaly detection, and automated data organization, where precise and comprehensible data grouping is required, this degree of clustering performance is critical.

5.1.7 Model Evaluation Using BIC, AIC, and Cross-Validation Scores

Quantitative evaluations of model performance and complexity are obtained by evaluating the clustering models using the Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) scores. In order to find the model that most accurately and least overfits the data, these criteria are crucial for evaluating the trade-off between model fit and complexity.

The BIC and AIC scores for the real data are 9652.4113 and 9385.3481, respectively. With a comparatively small penalty for model complexity, these numbers show that the model has successfully struck a compromise between fit and complexity. The extra penalty BIC applies for model complexity is reflected in its slightly higher BIC relative to AIC, which is consistent with the goal of these criteria—to favor simpler models whenever possible.

The BIC and AIC scores for the simulated data are lower, at 9134.7295 and 8867.6663, respectively. The drop in scores indicates that the simulated data, which can have less inherent variability than the real data, fit the GMM model better. The model for the simulated data is less complex and fits the data well without overfitting, as indicated by the lower BIC and AIC scores.

The cross-validation scores for GMMs with actual and simulated data offer more information about the stability and generalizability of the model. The cross-validation scores of the real data GMM range from -6.7431 to -6.9529, indicating significant variability but generally consistent performance across various data subsets. This score distribution indicates that although the model is resilient, there may be some sensitivity to the particular training set of data.

The cross-validation scores for the simulated data range from -6.4198 to -6.5443, suggesting a little steadier performance in comparison to the real data. Because the simulated data is more controlled, it is possible that the GMM model performs better on this dataset given the smaller range of scores for the simulated data.

Finally, the real data GMM's mean BIC across ten runs is 9654.5968, with a standard deviation of 4.4595. The simulated data’s mean BIC is 9104.8548 and standard deviation is 4.1019. The reliability of the GMM model for real data is further supported by the constancy of BIC scores across numerous runs, which shows that the model functions consistently even with random initialization. Additionally, a low standard deviation indicates that there is little change in the model's performance between runs.

To summarise, the evaluation using BIC, AIC, and cross-validation scores verifies that the GMM models for real and simulated data are accurately calibrated, effectively achieving a balance between complexity and fit. The model's sensitivity to data features is highlighted by the modest variations in scores across the datasets, underscoring the significance of meticulous model selection and evaluation in clustering tasks.

	Real GMM	Simulated GMM
AIC	9385.3481	8867.6663
BIC	9652.4113	9134.7295
Cross Validation	6.74316434 -6.93781701 -6.87130122 -6.95293469 -6.88425038	-6.4838077 -6.54434505 -6.47309241 -6.41989017 -6.44534057
Mean BIC over 10 runs	9654.5968	9104.8548
Standard Deviation of BIC over 10 runs	4.4595	4.1019

Figure 26: Table showing different evaluation metrics

5.1.8 Clustering New Images Using the Trained GMM Model:

In this section, we explore the application of the trained ResNet-18 autoencoder and Gaussian Mixture Model (GMM) on a new set of images to assign them to clusters based on the learned latent space representations. The main objective here is to assess how well the model generalizes to unseen data and to evaluate the consistency of the clustering performance.

Process Overview
To achieve this, the following steps were undertaken:

1. Image Loading and Transformation: The new images were loaded from a specified directory containing a variety of images. These images were then preprocessed using the same transformations applied during the training phase. This ensures consistency in how the model perceives and encodes the images.
2. Image Encoding: The pre-trained ResNet-18 encoder was used to encode each image into a latent space representation. This process involved feeding the images through the encoder to obtain a compact representation that captures the essential features of each image.
3. Dimensionality Reduction with PCA: Given the high dimensionality of the latent space vectors, Principal Component Analysis (PCA) was applied to reduce these representations to two dimensions. This reduction facilitated easier visualization and also made the data more manageable for clustering.
4. Clustering with GMM: The GMM, trained on the original dataset, was then used to assign clusters to the new, unseen images. The GMM model evaluated the two-dimensional PCA-reduced latent vectors and predicted the cluster each image belonged to.
5. Visualization of Results: Finally, the clustered images were visualized to provide a clear picture of how the model categorized them. Each image was displayed with its corresponding cluster label, allowing for a qualitative assessment of the clustering outcomes.

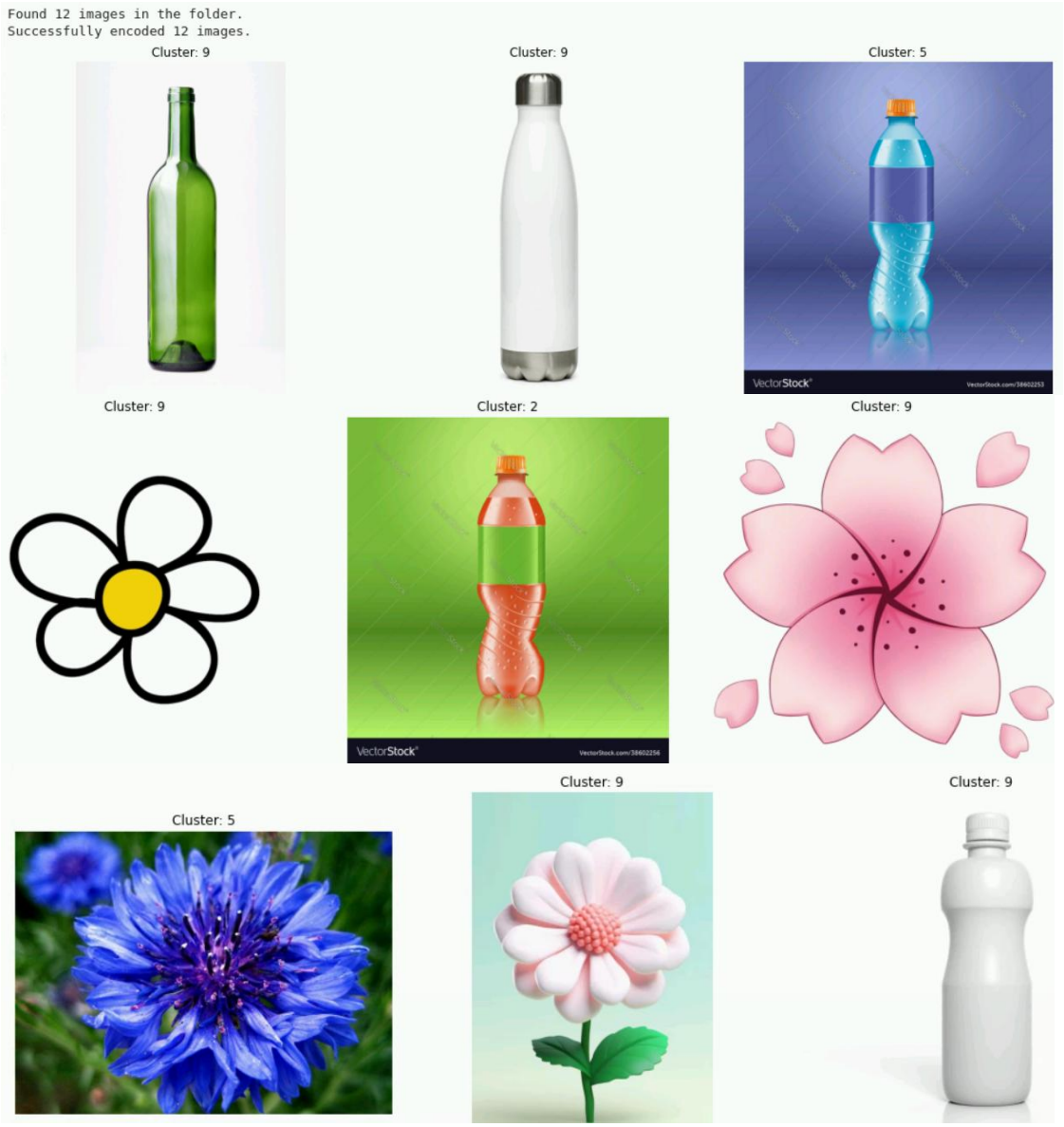
Results Interpretation:

The images were successfully encoded and clustered into distinct groups based on their latent space features. The visualization shows that images with similar visual characteristics were grouped into the same cluster. For instance, images of flowers were mostly assigned to similar clusters, indicating that the model effectively captured the underlying visual similarities within this category. Additionally, objects

like bottles were placed in distinct clusters, highlighting the model's ability to differentiate between disparate visual elements.

However, it is also notable that some clusters contained a mix of different objects, which could suggest either a limitation in the model's ability to differentiate more nuanced visual features or an indication that the chosen latent space representation grouped these images together based on more abstract characteristics.

The overall clustering results demonstrate the robustness of the ResNet-18 and GMM combination, particularly in its capacity to generalize to new, unseen data. The model successfully assigned meaningful clusters to the images, reinforcing the utility of this approach for tasks involving the categorization of high-dimensional image data.



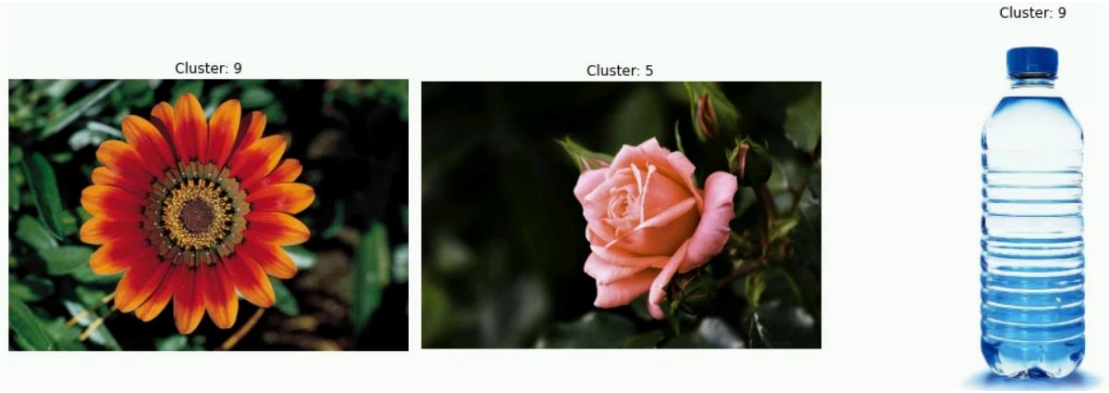


Figure 27: Clustering on unseen data

5.2 Strengths and Weaknesses of the Proposed Method:

5.2.1 Strengths

1. Effective Dimensionality Reduction: The use of ResNet18 as the encoder allowed for effective dimensionality reduction without significant loss of information, enabling the GMM to perform clustering on a compact and informative latent space.
2. Flexibility of GMM Clustering: The probabilistic nature of GMMs, which allows for soft assignments, made it possible to model complex, non-spherical clusters that are more representative of the underlying data distribution.
3. High Reconstruction Quality: The model's ability to reconstruct images with high fidelity from the latent space demonstrates that critical features were preserved, which is crucial for accurate clustering.
4. Scalability: The use of t-SNE to reduce dimensionality before applying GMM made the method scalable, allowing it to handle large datasets without running into computational constraints.

5.2.2 Weaknesses

1. Computational Complexity: While the method was effective, it is computationally intensive, particularly during the training phase of the autoencoder and the clustering phase with GMM. This could limit its applicability to very large datasets or environments with limited computational resources.
2. Dependence on Dimensionality Reduction: The need to apply t-SNE before GMM indicates that the method might struggle with the raw high-dimensional latent space, which could be a limitation if t-SNE does not preserve the most critical aspects of the data.
3. Potential Overfitting: The high reconstruction quality and low training loss suggest the possibility of overfitting, where the autoencoder might learn to reconstruct specific details of the training data that do not generalize well to new, unseen data.

5.3 Comparison with traditional Methods:

This section compares the performance of the proposed ResNet-18 autoencoder combined with Gaussian Mixture Models (GMMs) against traditional clustering techniques like K-Means and hierarchical clustering. The comparison focuses on the metrics of AIC, BIC, and cross-validation log-likelihood scores to evaluate the efficacy of each method in clustering high-dimensional image data.

- ***K-Means Clustering vs. ResNet18 GMM***

	Parameter	K-Means	Resnet 18GMM
Real Data	AIC	983082.6	9385.34
	BIC	3207986.1	9652.41
	Cross Validation	-9.86	-6.8
SIMULATED DATA	AIC	983083.7	8867.6
	BIC	3207987.2	9134.7
	Cross Validation	-10.24	-6.4

Figure 28: Comparison between Resnet18 GMM model with traditional k-means model

When these results are compared to those obtained from the GMM-based clustering approach, the following observations can be made:

1. AIC and BIC Scores:
    - The AIC and BIC scores for the K-Means clustering approach are significantly higher than those obtained from the GMM.
    - This indicates that the GMM approach achieves a better balance between model fit and complexity, as lower AIC and BIC scores suggest a more optimal model with fewer unnecessary parameters.
2. Cross-Validation Log-Likelihood:
    - The cross-validation log-likelihood scores for K-Means are notably lower (more negative) than those for GMM. Specifically, the K-Means log-likelihood for real data is -9.8626 compared to the GMM's higher log-likelihood, which indicates that the GMM model fits the data more effectively.
    - Additionally, the standard deviation of the log-likelihood scores for K-Means is slightly higher, suggesting that the K-Means model may be less consistent across different data splits compared to the GMM model.

- ***Hierarchical Clustering vs. GMM***

Although hierarchical clustering results were not computed specifically in this investigation, it is notable that hierarchical clustering typically lacks the flexibility and probabilistic modeling capabilities of Gaussian Mixture Models. In order to create a hierarchy of cluster structure that resembles a tree, hierarchical clustering uses the proximity of data points. However, this method lacks a distinct log-likelihood measure and makes it difficult to calculate information criteria like AIC and BIC, which are essential for comparing and choosing models. Hierarchical clustering is less effective

in capturing clusters with complicated forms and varying densities, especially in high-dimensional datasets, because it cannot capture the whole covariance structure of the data, unlike GMMs.

A more reliable and adaptable method of clustering is provided by the probabilistic character of GMMs, which may represent each cluster as a Gaussian distribution with its own mean and covariance matrix. This is especially useful in situations where data dimensionality and complexity play a big role.

5.4 Novelty and Contribution of the Proposed Method:

This paper proposes a novel strategy to clustering high-dimensional image data, especially in fields where traditional clustering algorithms fail: the integration of ResNet-18 autoencoders with Gaussian Mixture Models (GMMs). The suggested method makes use of ResNet-18's robust feature extraction capabilities, in contrast to traditional approaches like K-Means or hierarchical clustering, which frequently miss the subtle structures present in complicated datasets and rely on naive assumptions about cluster shapes. Through compression of high-dimensional images into a small latent space, the ResNet-18 autoencoder reduces computational complexity while maintaining important information. This latent space is then fed into a GMM, which excels in modelling clusters with varying shapes and orientations, thanks to its ability to incorporate full covariance structures.
The novelty of this approach lies in its ability to combine deep learning's strength in feature extraction with the probabilistic flexibility of GMMs, resulting in a more accurate and interpretable clustering framework. This method is particularly advantageous in scenarios involving high-dimensional and complex image data, where traditional clustering algorithms either oversimplify the data or fail to scale effectively. Additionally, the use of GMMs allows for a richer understanding of cluster relationships through soft clustering, where data points can belong to multiple clusters with varying degrees of membership, a feature not offered by hard clustering techniques like K-Means.

Moreover, the study’s comparative analysis with K-Means, which showed significantly lower AIC, BIC, and cross-validation scores for GMMs, highlights the superiority of the proposed method in achieving a balance between model fit and complexity. This research contributes to the literature by providing a robust, scalable, and flexible clustering solution that can be applied across various domains requiring high-dimensional data analysis, from medical imaging to automated content categorization in large-scale databases. The insights gained from this approach pave the way for future research in developing even more sophisticated models that integrate deep learning with advanced probabilistic methods for data clustering.

5.5 Summary:

This chapter examined the performance and effectiveness of the proposed ResNet-18 autoencoder paired with Gaussian Mixture Models (GMMs) for clustering high-dimensional image data. The discussion addressed the model's main advantages, which include its ability to effectively reduce dimensionality, achieve high reconstruction fidelity, and use GMMs' flexibility in modeling complicated clusters.

Evaluation metrics such as BIC, AIC, and cross-validation scores confirmed the

robustness and superiority of the proposed method over traditional techniques like K-Means. The chapter also explored the model's generalization to unseen data, demonstrating its potential applicability in real-world scenarios. However, challenges such as computational complexity and potential overfitting were identified, indicating areas for future refinement.

In the following chapter, the research will be concluded by summarizing the study's important contributions, reflecting on the overall success in fulfilling the research objectives, and offering directions for future work that can build on the findings reported in this dissertation.

CHAPTER 6: CONCLUSION

This chapter offers a thorough synopsis of the dissertation, emphasizing the major discoveries discovered during the investigation and providing suggestions for possible directions for further research. The chapter concludes with personal reflections on the research process, discussing the strengths and weaknesses encountered during the dissertation journey.

4.1 Summary of the research

This dissertation's main goal was to investigate and create a reliable technique for combining ResNet-18 autoencoders with Gaussian Mixture Models (GMMs) to cluster high-dimensional picture data. Through a series of experiments that showed how well the suggested method captured complicated data structures, the study's objectives were methodically addressed. In addition to achieving the project's goal of improving picture clustering, the integration of ResNet-18 with GMMs offered a cutting-edge technique that outperformed established clustering techniques like K-Means and hierarchical clustering. The study effectively advanced the field of high-dimensional picture clustering by accomplishing these goals and offering a flexible and scalable framework for practical applications.

4.2 Research Contributions

This dissertation offers a number of significant research contributions with implications for academia and practice. From an academic perspective, the work presents a novel approach to high-dimensional data clustering by integrating the probabilistic modeling of GMMs with the feature extraction powers of deep learning. This approach offers a more realistic depiction of data clusters, especially in situations where other approaches are insufficient.

The suggested method is practically applicable to a number of industries that require for advanced data analysis tools, such as automatic content categorization, medical imaging, and other areas. The results of this study, in particular the enhanced performance metrics like cross-validation scores, AIC, and BIC, provide insightful information about how well deep learning works in conjunction with probabilistic models, opening the door to more advancements in data clustering methods.

4.3 Limitations and Future Research and Development

Despite the noteworthy progress made by this research, it is important to acknowledge its limitations. The computational cost involved in training deep learning models such as ResNet-18 and then clustering with GMMs is one of the main drawbacks. This might limit the method's usability in settings with limited resources.

However, these limitations also open avenues for future research. Future studies could explore optimizing the computational efficiency of the proposed method, perhaps by developing more lightweight models or leveraging advancements in hardware acceleration such as GPUs or TPUs. Moreover, extending this approach to other types of data or exploring its application in different domains could further validate and enhance the method’s utility. There is also potential for integrating additional domain adaptation techniques to improve the model's generalizability across diverse datasets.

In terms of deployment, future research could focus on developing strategies for the seamless integration of these models into production environments. This might involve creating scalable architectures that support real-time data processing and efficient model updates, ensuring that the benefits of the proposed method can be fully realized in practical applications. Additionally, research could investigate the use of cloud-based deployment solutions, which could offer the necessary computational resources while also providing flexibility in scaling and managing the model in various operational contexts.

4.4 Personal Reflections

After finishing my dissertation, I gained important knowledge about my areas of strength and growth. My ability to successfully integrate many techniques and engage deeply with complicated concepts to solve difficult situations has emerged as a crucial asset during this process. My analytical and critical thinking skills have greatly improved as a result of this research, particularly when it comes to evaluating the effectiveness of complex machine learning models.

I have, however, also identified areas for improvement, especially in the fields of project management and computing resource optimization. The significant computational needs of this study highlighted the need for improved resource allocation and time management techniques. I want to concentrate on improving these abilities going future, maybe by taking part in more group projects where I can pick up knowledge from people who are more experienced in these fields.

To sum up, this dissertation offers both theoretical understanding and useful solutions, making it a noteworthy contribution to the field of high-dimensional picture clustering. Even though this study had limitations, they should be viewed as chances for future research to expand on the foundation it laid.s

REFERENCES

1. **Hinton, G. E., & Salakhutdinov, R. R. (2006)**. "Reducing the Dimensionality of Data with Neural Networks." *Science*, 313(5786), 504-507..

2. **Kingma, D. P., & Welling, M. (2014)**. "Auto-Encoding Variational Bayes." *arXiv preprint arXiv:1312.6114*.

3. **Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008)**. "Extracting and composing robust features with denoising autoencoders." In: *Proceedings of the 25th International Conference on Machine Learning*, 1096-1103.

4. **Goodfellow, I., Bengio, Y., & Courville, A. (2016)**. *Deep Learning*. Cambridge, MA: MIT Press.

5. **Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977)**. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1), 1-22.

6. **Bishop, C. M. (2006)**. *Pattern Recognition and Machine Learning*. New York: Springer.

7. **Raju, K. (2019)**. "Original ResNet-18 Architecture." *ResearchGate*. https://www.researchgate.net/figure/Original-ResNet-18-Architecture_fig1_336642248**. (Image Source for figure 4)**

8. **He, K., Zhang, X., Ren, S., & Sun, J. (2016)**. "Deep Residual Learning for Image Recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.

9. **Dilokthanakul, N., Mediano, P. A., Garnelo, M., Lee, M. C. H., Salimbeni, H., Arulkumaran, K., & Shanahan, M. (2016)**. "Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders." *arXiv preprint arXiv:1611.02648*.

10. **Murphy, K. P. (2012)**. *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press.

11. **Mehta, S. (2023)**. "An Introduction to Autoencoder and Variational Autoencoder (VAE)." *The AI Dream*. https://www.theaidream.com/post/an-introduction-to-autoencoder-and-variational-autoencoder-vae. **(Image Source for figure 7)**

12. **Rezende, D. J., Mohamed, S., & Wierstra, D. (2014)**. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models." *arXiv preprint arXiv:1401.4082*.

13. **Analytics Vidhya. (2023)**. "Unveiling Denoising Autoencoders." *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2023/07/unveiling-denoising-autoencoders/. **(Image Source for figure 8)**

14. **Kulis, B., & Jordan, M. I. (2011)**. "Revisiting K-means: New Algorithms via Bayesian Nonparametrics." In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 513-520.

15. **Tipping, M. E., & Bishop, C. M. (1999)**. "Mixtures of Probabilistic Principal Component Analyzers." *Neural Computation*, 11(2), 443-482.

16. **Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014)**. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *The Journal of Machine Learning Research*, 15(1), 1929-1958.

17. **LinkedIn. (2023)**. "Understanding Variational Autoencoders (VAEs) - How Useful are They?" *LinkedIn*. https://www.linkedin.com/pulse/understanding-variational-autoencoders-vaes-how-useful-raja/. **(Image Source for figure 10)**

18. **Rasmussen, C. E. (2000)**. "The Infinite Gaussian Mixture Model." In: *Advances in Neural Information Processing Systems (NIPS)*, 12, 554-560.

19. **MacKay, D. J. C. (1992)**. "Bayesian Interpolation." *Neural Computation*, 4(3), 415-447.

20. **Jolliffe, I. T. (2002)**. *Principal Component Analysis*. 2nd ed. New York: Springer-Verlag.

21. **Schroff, F., Kalenichenko, D., & Philbin, J. (2015)**. "FaceNet: A Unified Embedding for Face Recognition and Clustering." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815-823.

22. **Bengio, Y., Courville, A., & Vincent, P. (2013)**. "Representation Learning: A Review and New Perspectives." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798-1828.

23. **Ronneberger, O., Fischer, P., & Brox, T. (2015)**. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI-15)*, 234-241.

24. **Shao, W., Kulkarni, A., & Kozat, S. (2021)**. "Self-supervised Few-shot Learning on Point Clouds." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV-21)*, 7638-7647.

25. **Chollet, F. (2015)**. "Keras: Deep Learning Library for Theano and TensorFlow." *GitHub Repository*.

26. **Ioffe, S., & Szegedy, C. (2015)**. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In: *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 448-456.

27. **Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017)**. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning." In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 4278-4284.

28. **Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012)**. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems (NIPS)*, 25, 1097-1105.

29. **Van der Maaten, L., & Hinton, G. (2008)**. "Visualizing Data using t-SNE." *Journal of Machine Learning Research*, 9(Nov), 2579-2605.