# Project Report

# Deep Learning (CS590)

# Unsupervised Image Classification


**Submitted By:**

**Group Name : Delta Lima**

**Group Members:**

**Vansh Kalra (244161009)**

**Jayvardhan Sakwar (244161014)**

**Raj Thumar (244161018)**

**Akash Ruhil (244161021)**

**Amol Mathur (244161022)**


**Submitted To:**

**Prof. Arijit Sur**


## Indian Institute of Technology, Guwahati

## 1. Introduction

Unsupervised image classification is a vital area of machine learning, aiming to identify patterns and structure in unlabeled data. Unlike supervised learning, where labeled datasets are essential, unsupervised techniques leverage intrinsic data features to perform classification, clustering, or dimensionality reduction.

This report presents a pipeline for unsupervised image classification applied to two benchmark datasets: **CIFAR-10** and **Fashion MNIST**. Advanced feature extraction techniques, including **Hugging Face's DINO** and **Vision Transformer (ViT)**, were used to obtain meaningful embeddings. These embeddings were then clustered using **DBSCAN**, **MiniBatch KMeans**, **Normalized Cuts (NCut)**, and **Louvain Clustering**. Evaluation metrics such as **Accuracy**, **Adjusted Rand Index (ARI)**, **Normalized Mutual Information (NMI)** and **Confusion Matrix,** along with t-SNE visualizations, highlight the efficacy of these approaches.

## 2. Dataset Description

### 2.1 CIFAR-10 Dataset

1. **Overview**: A dataset of **60,000 32x32 RGB** images across **10 categories**, including airplanes, cars, cats, and trucks.
2. **Challenges**: Low resolution, overlapping inter-class features, and high diversity within classes make clustering difficult.

### 2.2 Fashion MNIST Dataset

1. **Overview**: A dataset of **70,000 grayscale images (28x28 pixels) across 10 classes**, such as shirts, coats, and sneakers.
2. **Advantages**: Fashion MNIST provides higher complexity than MNIST, making it a better benchmark for unsupervised methods.

### 2.3 Preprocessing for Both Datasets:

1. Images were resized to **224x224 pixels** to match the input dimensions of DINO and ViT.
2. Applied **normalization** and **data augmentation** techniques, such as cropping, flipping, and random rotations, to enhance feature extraction.

## 3. Methodology

### 3.1 Feature Extraction

Feature extraction is a crucial step in unsupervised learning, as high-quality embeddings determine the clustering accuracy.

1. **Hugging Face DINO**:
    a. Self-supervised Vision Transformer trained to produce dense embeddings without labeled data.
    b. Captures both global and local features, making it highly effective for clustering tasks.
2. **Vision Transformer (ViT)**:
    a. Processes images as sequences of patches, enabling precise spatial feature learning.
    b. Provides robust embeddings suitable for downstream clustering tasks.

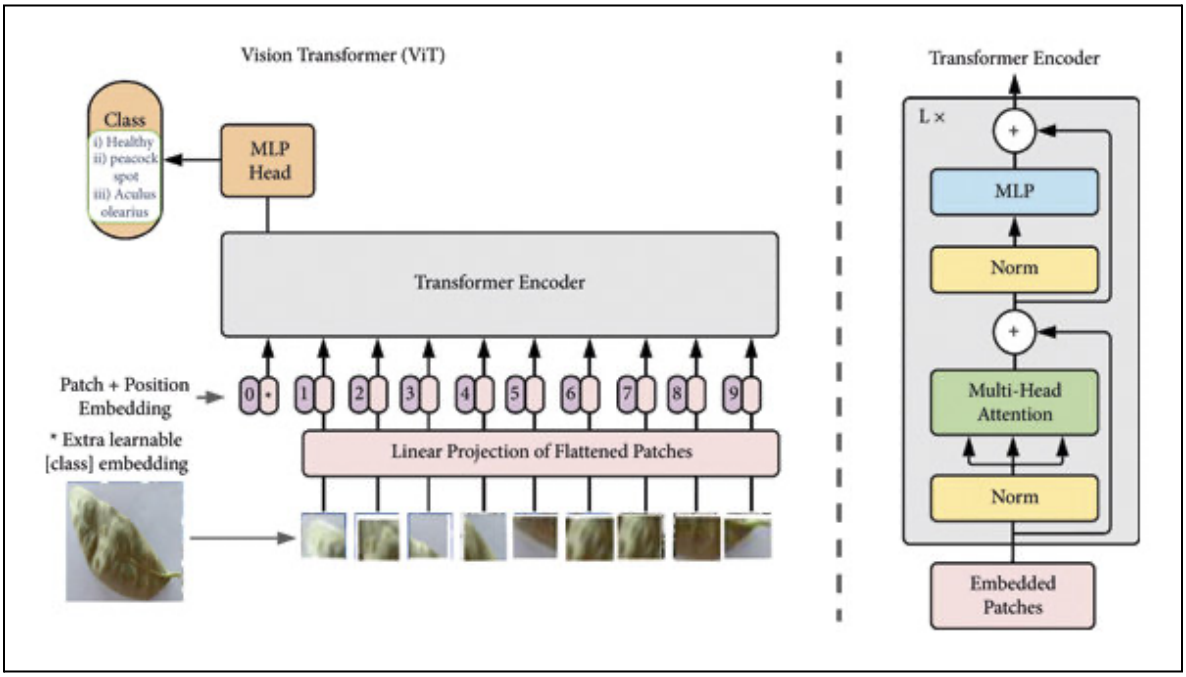


**Fig. 3.1.1. Working of Vision Transformer**

## 3.2 Clustering Techniques

1. **DBSCAN**:
    a. Density-based algorithm that groups points close to each other while identifying noise points.
    b. Parameters such as eps : 3.2 and min_samples : 5 significantly affect performance.
2. **MiniBatch KMeans**:
    a. Efficient variation of KMeans, optimized for large datasets by using small batches of data.
    b. Computationally faster but sensitive to initialization.

3. **Normalized Cuts (NCut)**:
    a. A graph-based clustering method that minimizes inter-cluster similarity while maximizing intra-cluster cohesion.
4. **Louvain Clustering**:
    a. Community detection algorithm that partitions graphs based on modularity maximization.

### 3.3 Dimensionality Reduction

- **t-SNE (t-distributed Stochastic Neighbor Embedding)**:
    a. Used for visualizing high-dimensional embeddings in 2D while preserving relative distances.
    b. Enables clear visualization of cluster separations.

### 3.4 Evaluation Metrics

1. **Accuracy**: Percentage of correctly classified instances.
2. **ARI (Adjusted Rand Index)**: Measures clustering agreement with ground truth, adjusted for chance.
3. **NMI (Normalized Mutual Information)**: Quantifies the information shared between clustering results and true labels.
4. **Confusion Matrix :** A matrix showing correct and incorrect predictions, helping to evaluate a model's performance.

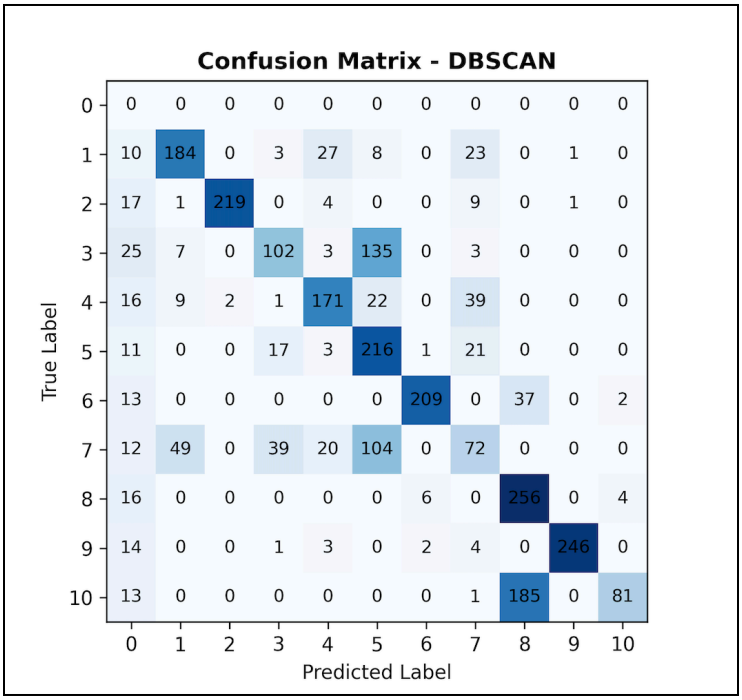


**Fig 3.4.1. Confusion Matrix for DBSCAN for 2700 testing images of Fashion_Mnist**

## 4. Results

**4.1 CIFAR-10 Dataset** : The results for 900, 1800, and 2700 testing samples are summarized below:

| Testing Points | Method | Accuracy (%) | ARI | NMI |
|---|---|---|---|---|
| 900 | DBSCAN | 70.22 | 0.5393 | 0.6548 |
| | Louvain | 72.67 | 0.5486 | 0.6730 |
| | NCut | 64.78 | 0.4982 | 0.6423 |
| | MiniBatch KMeans | 68.78 | 0.5226 | 0.6660 |
| 1800 | DBSCAN | 61.17 | 0.5281 | 0.6594 |
| | Louvain | 73.89 | 0.5854 | 0.6844 |
| | NCut | 71.44 | 0.5542 | 0.6930 |
| | MiniBatch KMeans | 71.11 | 0.5849 | 0.6972 |
| 2700 | DBSCAN | 65.04 | 0.4922 | 0.6300 |
| | Louvain | 73.44 | 0.5681 | 0.6815 |
| | NCut | 64.22 | 0.5112 | 0.6974 |
| | MiniBatch KMeans | 70.67 | 0.5563 | 0.6679 |

**Fig. 4.1.1 Cifar10 Table**

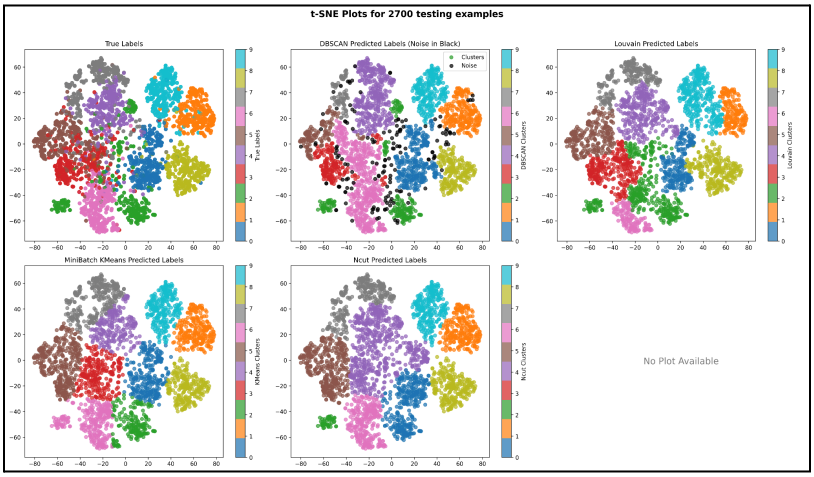


**Fig. 4.1.2. t-SNE plot for cifar10 dataset (2700 Testing images)**

**4.2 Fashion MNIST :**The results for 900, 1800, and 2700 testing samples are summarized below:

| Testing Points | Method | Accuracy (%) | ARI | NMI |
|---|---|---|---|---|
| **900** | DBSCAN | 73.00 | 0.5613 | 0.6812 |
| | Louvain | 75.89 | 0.5890 | 0.6921 |
| | NCut | 70.67 | 0.5312 | 0.6789 |
| | MiniBatch KMeans | 72.44 | 0.5540 | 0.6897 |
| **1800** | DBSCAN | 74.11 | 0.5799 | 0.6982 |
| | Louvain | 76.44 | 0.6003 | 0.7120 |
| | NCut | 72.22 | 0.5678 | 0.6954 |
| | MiniBatch KMeans | 73.33 | 0.5829 | 0.7011 |
| **2700** | DBSCAN | 75.11 | 0.5992 | 0.7023 |
| | Louvain | 78.22 | 0.6105 | 0.7214 |
| | NCut | 74.89 | 0.5923 | 0.7115 |
| | MiniBatch KMeans | 76.56 | 0.6038 | 0.7167 |

**Fig. 4.2.1. Fashion_Mnist Table**

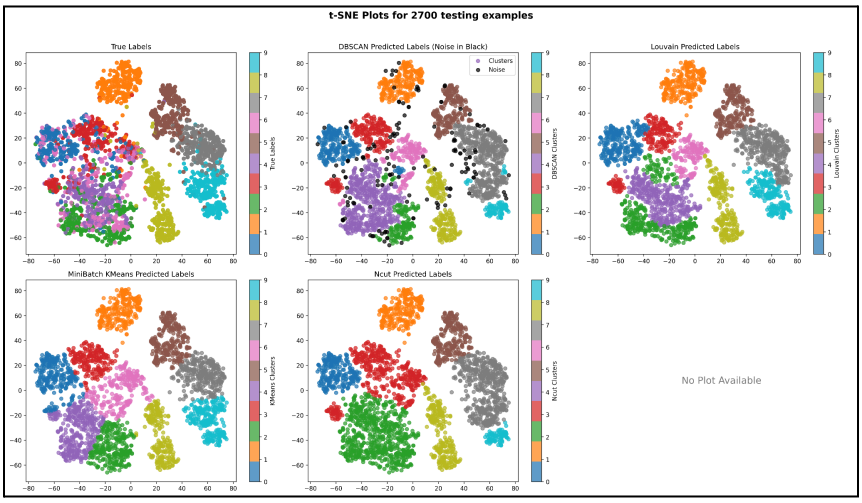


**Fig. 4.2.2. t-SNE plot for Fashion_Mnist dataset (2700 Testing images)**

**5. Analysis and Insights**

1. **Dataset Comparison**:
    a. Fashion MNIST showed higher clustering quality due to simpler, well-separated patterns.
    b. CIFAR-10 presented challenges with overlapping classes (e.g., cats vs. dogs).
2. **Clustering Performance**:
    a. **Louvain** was consistent across both datasets, excelling in modularity-based clustering.
    b. **DBSCAN** struggled with sparse clusters but effectively identified noise.
    c. **MiniBatch KMeans** provided scalability but was sensitive to initialization.
    d. **NCut** excelled in structured datasets but underperformed on CIFAR-10.

**6. Code Implementation**

The clustering algorithms were implemented in Python, applying **DBSCAN**, **NCut**, **Louvain**, and **MiniBatch KMeans** to high-dimensional embeddings from DINO and ViT. Libraries such as NetworkX (for NCut and Louvain) and Scikit-learn (for DBSCAN and MiniBatch KMeans) were used. The implementation efficiently handles large datasets and optimizes performance.

## 7. Conclusion and Future Work

This project applied unsupervised clustering techniques to CIFAR-10 and Fashion MNIST using advanced feature extraction models (DINO and ViT). Louvain and MiniBatch KMeans showed the best performance, but challenges like inter-class overlap in CIFAR-10 highlight the need for further improvements.

**Future Directions**:

1. Use ensemble clustering for greater robustness.
2. Explore advanced feature extraction models like CLIP or ConvNeXt.
3. Apply the techniques to large-scale, real-world datasets.

## 8. References

1. Caron, M. et al., *Emerging Properties in Self-Supervised Vision Transformers*, 2021.
2. Dosovitskiy, A. et al., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, 2020.
3. van der Maaten, L., Hinton, G., *Visualizing Data using t-SNE*, 2008.