Unsupervised Image Classification

Deep Learning (CS 590)

Prof. Arijit Sur

Vansh Kalra (244161009)

Jayvardhan Sakvar (244161014)

Raj Thumar (244161018)

Sqn Ldr Akash Ruhil (244161021)

Lt Cdr Amol Mathur (244161022)

Introduction

- **What is Image Classification?**
  - **Image Classification** assigns a label to an image based on its content, allowing a model to recognize objects or patterns.
  - For example, a model can classify animal images as "cat," "dog," or "bird" based on their features.

- **What is Unsupervised Learning?**
  - **Unsupervised Learning** involves training a model without labeled data. The algorithm detects patterns and groups similar data points into clusters.
  - **Challenge in Unsupervised Learning :** Handling of unlabeled data.

- **Why Unsupervised Image Classification?**
  - **Real-World Applications**: Clustering large datasets (e.g. medical images, Agriculture).
  - Reducing the need for expensive and time-consuming manual labeling.

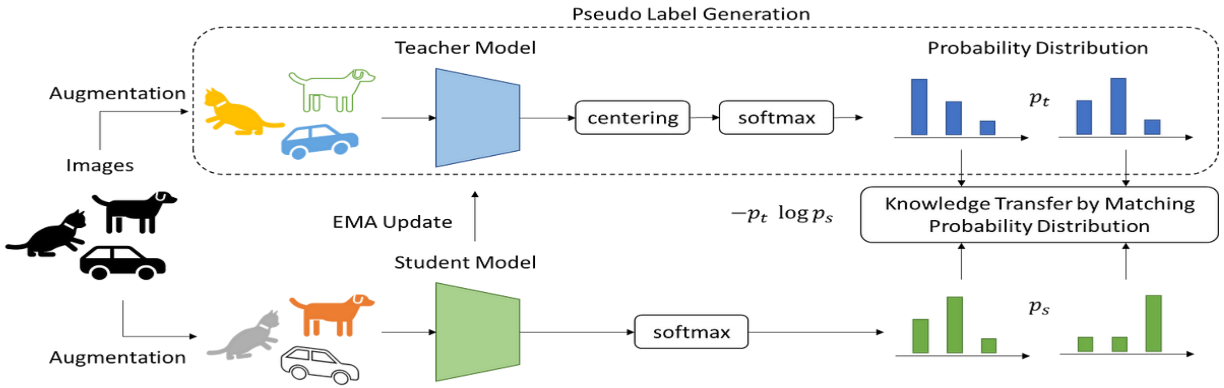
Self-Supervised Learning

- **Why Self-Supervised Learning Introduced?**
  - To tackle the challenge of processing unlabeled data in unsupervised learning.
  - Introduced in paper "A Cookbook of Self-Supervised Learning"(https://arxiv.org/pdf/2304.12210)

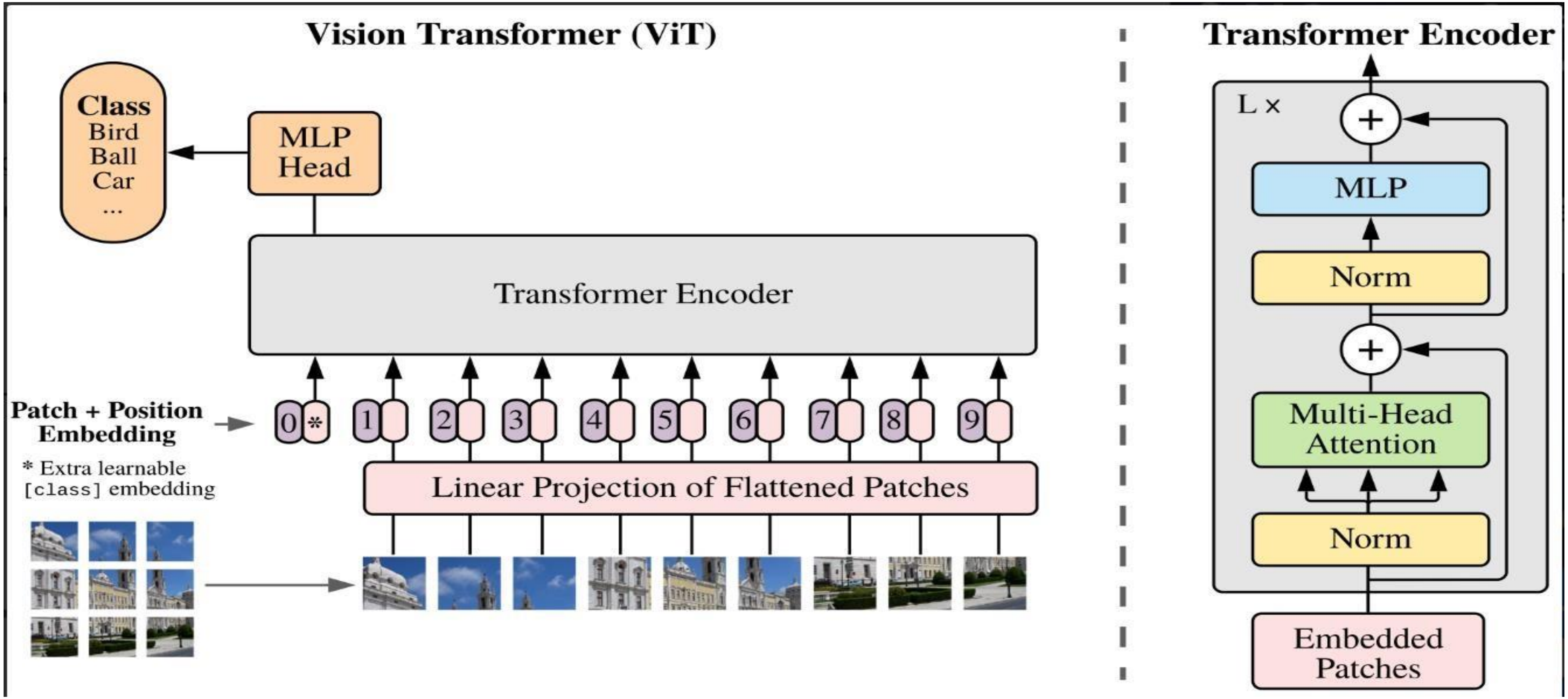
- **What is Self-Supervised Learning?**
  - **Learn from Unlabeled Data**: The model trains on data without labels.
  - **Create Own Tasks**: It makes its own tasks, like guessing parts of data.
  - **Uses DINO Method :** A Unsupervised Learning method to do feature extraction in images.
  - **Extract Features**: The model finds useful patterns in the data.
  - **Improve Performance**: These patterns help with tasks like classification, even without labeled data.

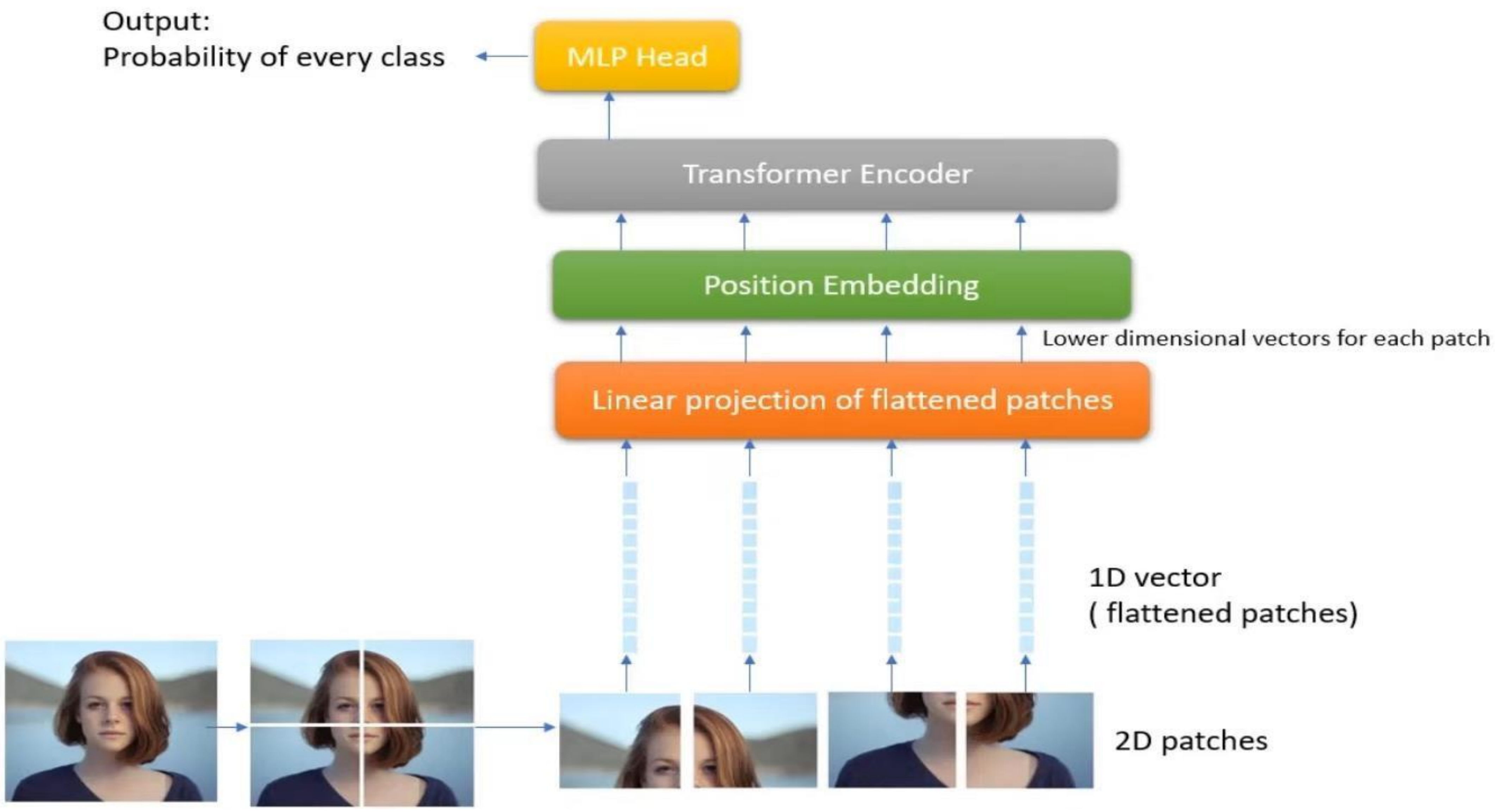
DINO Method (Self-Distillation with No Labels)

- **DINO** is an unsupervised learning method that trains a model to recognize images without needing any labeled data by using a teacher-student approach.
- **Architecture**: Teacher-Student framework.
  - **Teacher**: Pre-trained and frozen (Stable), processes global image crops.
  - **Student**: Learns from teacher's outputs, processes both global and local image crops.
- **Self-Distillation**: Student is trained to align its output with the teacher's Output.
- **Exponential Moving Average (EMA)**: Updates the teacher model with student improvements.



Vision Transformer Model




Output:
Probability of every class
MLP Head
Transformer Encoder
Position Embedding
Lower dimensional vectors for each patch
Linear projection of flattened patches
1D vector
(flattened patches)
2D patches

Clustering Techniques:

- **DBSCAN: Density-Based Spatial Clustering of Applications with Noise**

Key Points:

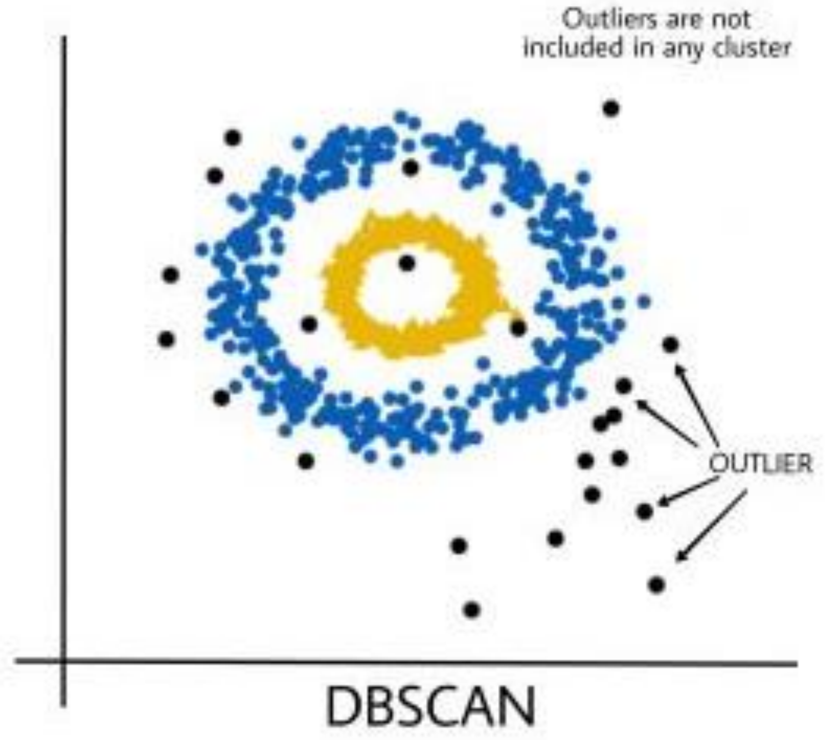
•Groups points based on density and identifies outliers as noise.

•**Key Parameters:**

- **ε (epsilon):** Radius for neighborhood search.
- **MinPts:** Minimum points to form a dense region.

•Handles clusters of arbitrary shapes and is robust to outliers.

•Sensitive to ε and MinPts settings.



• MiniBatch KMeans: Scalable Clustering for Large Datasets

Key Points:

•Processes data in **mini-batches** for faster and memory-efficient clustering.

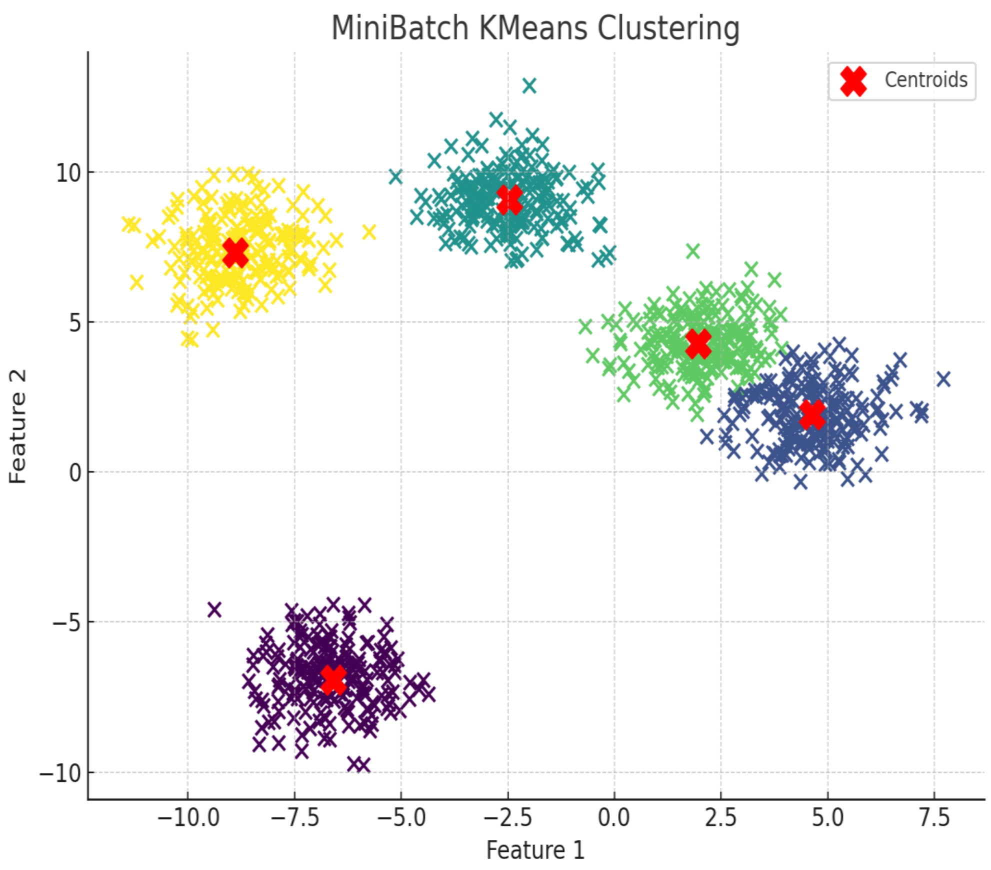
•Retains the simplicity and interpretability of standard KMeans.

•**Advantages:**

- Scalable to massive datasets.
- Efficient for streaming data.

•**Limitations:**

- May produce less accurate clusters compared to KMeans.
- Sensitive to parameter choices (e.g., number of clusters, batch size).



- **NCut: Normalized Cut for Graph-Based Clustering**

Key Points:

- Represents data as a graph where:
  - **Nodes** = data points.
  - **Edges** = similarity between points.
- Partitions the graph into clusters by minimizing the normalized cut value, ensuring balanced and connected clusters.
- **Advantages:**
  - Handles complex structures and non-convex clusters.
  - Effective for image segmentation and network analysis.
- **Limitations:**
  - Computationally expensive for large datasets.
  - Depends on graph construction and similarity metric.

• Louvain Algorithm: Modularity-Based Clustering

Key Points:

•**What is the Louvain Algorithm?**

A graph-based algorithm that detects communities by optimizing modularity, measuring the density of edges within communities versus between them.

•**Steps:**

- Assign each node to its own community.
- Iteratively merge communities to maximize modularity.
- Builds a hierarchy of communities for multi-scale analysis.

•**Advantages:**

- Scalable to large graphs.
- Produces meaningful and hierarchical clusters.

•**Applications:**

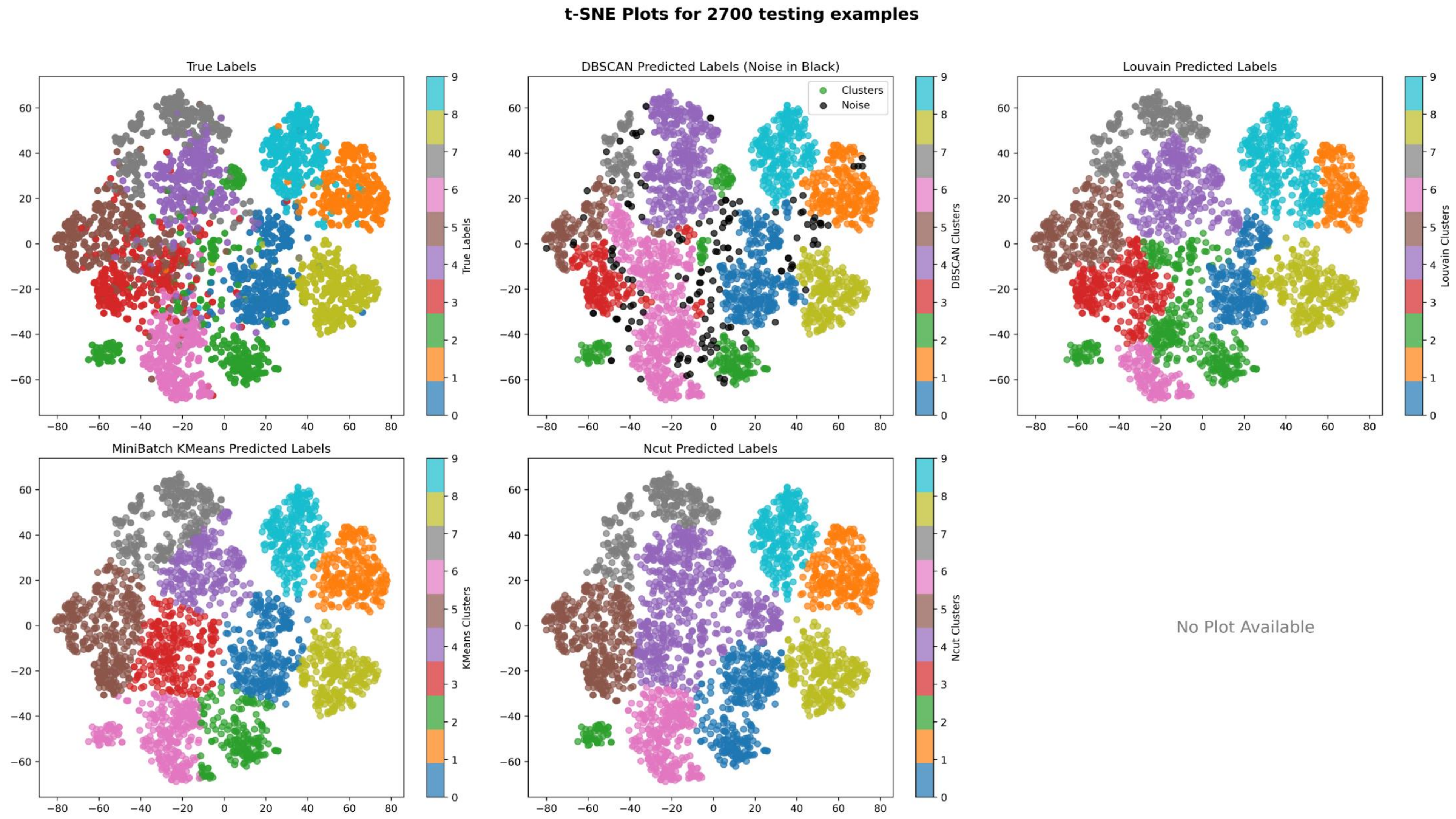
- Social network analysis.
- Recommendation systems.
- Biological networks.

Results - CIFAR-10

Performance Metrics for 900,1800 and 2700 testing points

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

Results - CIFAR-10

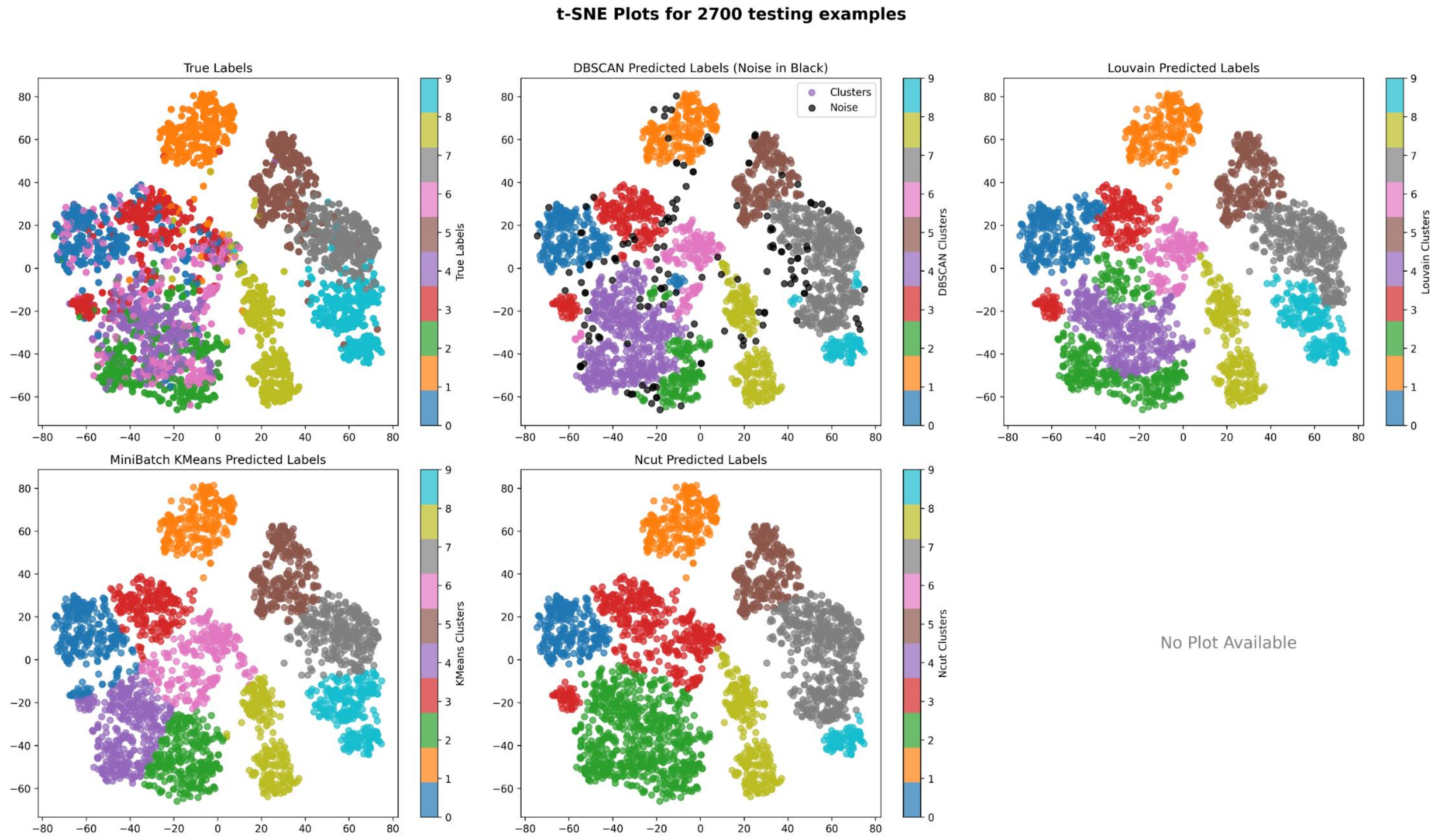


Results - Fashion MNIST

Performance Metrics for 900,1800 and 2700 testing points

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

Results - Fashion MNIST



Conclusion

- **Achievements**:
    - Successfully clustered CIFAR-10 and Fashion MNIST.
    - Louvain and MiniBatch KMeans emerged as top performers.

- **Challenges**:
    - Overlap in CIFAR-10 classes.
    - DBSCAN’s sensitivity to hyperparameters.

- **Learnings**:
    - DINO and ViT provide robust embeddings.
    - t-SNE is effective for visualizing high-dimensional clusters.

THANK YOU