AI@UNICT2024 - Challenge 2 - Round2
VIT + Kmeans

Giuseppe Lombardia
Deep Learning course Data science for management
giuseppe97lomb@gmail.com

15/04/2025

1 Introduction

Image classification is a fundamental task in computer vision, aiming to identify and distinguish objects across different categories. A key challenge in this context is distribution shift, especially when the background or visual context differs significantly between training and testing phases. In the AI-UNICT 2024 Challenge 2 – Round 2, the training set consists of labeled images with heterogeneous and realistic backgrounds, while the test set includes images with uniform backgrounds and centralized objects, creating a domain mismatch that can hinder generalization.

To address this issue, we design a pipeline based on unsupervised clustering, leveraging deep visual features extracted using a pretrained Vision Transformer (ViT-B/16). We begin by cropping the training images using the provided bounding boxes to isolate the objects of interest, thereby reducing the influence of background information. These cropped images are then used to compute class-specific centroids in the embedding space.

On the test side, we extract features using the same ViT model and apply KMeans clustering. To match each test cluster to a known class, we adopt the Hungarian algorithm based on the distance between test and training centroids. By testing multiple random seeds and selecting the clustering result with the most balanced class distribution, we aim to ensure robust and accurate predictions under domain shift.

This method enables classification without relying on direct supervision during testing and proves effective in addressing background variability between training and test domains.

2 Model description

The Vision Transformer (ViT) is a deep learning architecture introduced by Dosovitskiy et al. in 2020 that applies the Transformer model—originally de-

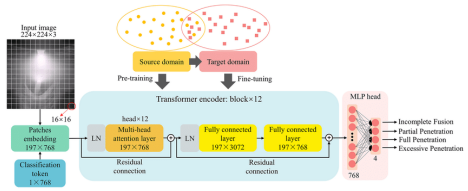

Figure 1: ViT-B/16

veloped for natural language processing—to image data. Unlike traditional convolutional neural networks (CNNs), which process images hierarchically using spatial filters, ViT treats images as sequences of patches, enabling global self-attention across the entire image.

Patch Embedding. An input image is first divided into a fixed number of non-overlapping square patches (e.g., 16×16 pixels). Each patch is then flattened and linearly projected into a vector of fixed size (e.g., 768 dimensions). This results in a sequence of patch embeddings that is analogous to word embeddings in NLP models.

Positional Encoding. Since Transformers lack an inherent notion of order or spatial structure, positional embeddings are added to each patch embedding to retain information about the location of the patch in the original image.

Transformer Encoder. The resulting sequence of vectors is passed through a standard Transformer encoder composed of multiple layers of multi-head self-attention and feedforward neural networks. This allows each patch to attend to all other patches in the image, facilitating a rich global representation.

[CLS] Token and Output. A special learnable classification token ([CLS]) is prepended to the patch sequence. After passing through the Transformer, the embedding corresponding to the [CLS] token is used for downstream tasks such as classification. However, in our approach, we discard the final classification head and use the pre-[CLS] representation of the entire image as a high-level feature vector.

Application in Our Pipeline. In our project, we employ the pretrained `vit_base_patch16_224` model from the `timm` library. We remove the classifier head to use ViT purely as a feature extractor. Each image (cropped around the object of interest using bounding box coordinates) is transformed into a single 768-dimensional embedding vector. These vectors serve as input for our unsupervised classification step based on KMeans clustering and centroid matching.

This approach leverages the ViT's strength in capturing long-range dependencies and contextual information across the entire image, making it particularly effective in scenarios with distribution shifts and background noise.

3 Dataset Description

The dataset used in this project was provided as part of the AI-UNICT 2024 Challenge and is specifically designed to test classification robustness under distribution shift. It is composed of two main parts: the training set and the test set, along with a CSV file containing bounding box annotations.

Training Set. The training set consists of 1600 images, evenly distributed across 8 classes (200 images per class). Each image contains a centered object, but the backgrounds are highly variable, including diverse lighting conditions, colors, and textures. This diversity is intended to encourage models to learn object-specific features rather than background cues.

Test Set. The test set contains 800 unlabeled images. Unlike the training set, all test images share a nearly identical background. This was deliberately done to simulate a domain shift and to evaluate the model's ability to generalize beyond background-dependent patterns. Importantly, all test images belong to the same class distribution as the training set.

Bounding Boxes. The file `train.csv` provides bounding box annotations for each training image. Each row contains the image filename, class label, and the coordinates of the bounding box (top-left and bottom-right corners). These annotations are crucial for isolating the objects from the background during preprocessing and are a key component of our strategy.

Submission Format. A sample file named `submission.csv` is provided to illustrate the expected submission format. Each row should include an image filename from the test set and the predicted class label.

This setup provides a challenging yet realistic scenario, where background invariance becomes essential for achieving high generalization accuracy. Our methodology explicitly addresses this by leveraging the bounding box annotations and advanced feature extraction techniques.

4 Preprocessing and Data Augmentation

To address the distribution shift between the training and test sets, we apply a tailored preprocessing pipeline focused on reducing the influence of the background and enhancing the model's ability to focus on object-related features.

Bounding Box Cropping. The first step of our preprocessing pipeline involves cropping each training image based on the bounding box coordinates provided in `train.csv`. This step isolates the object of interest and removes most of the surrounding background, which is highly variable in the training set. The cropped images are then saved in class-specific directories, forming a new training dataset with minimal background noise.

Resizing and Normalization. All cropped images are resized to a fixed resolution of 224×224 pixels to match the input size expected by the Vision Transformer (ViT) model. Additionally, we normalize the pixel values using the mean and standard deviation of the ImageNet dataset, which the ViT model was originally trained on.

Data Augmentation. To improve generalization and prevent overfitting, we apply a set of standard data augmentation techniques:

- **Random Horizontal Flip** with probability 0.5
- **Random Rotation** up to 15 degrees
- **Color Jittering** with slight variations in brightness, contrast, and saturation

These augmentations encourage the model to learn invariant representations of the objects, making the classifier more robust to small changes in pose, orientation, and lighting.

Reproducibility. All random operations are seeded using a fixed random seed. We also configure deterministic behavior in PyTorch to ensure full reproducibility of the training process.

This preprocessing pipeline is essential to enforce background invariance, reduce noise, and improve the robustness of the downstream clustering and classification stages.

5 Feature Extraction with Vision Transformer (ViT)

To extract high-quality semantic features from images, we employ a pretrained Vision Transformer model (ViT-B/16) as a feature extractor. The classification head of the model is removed, and the transformer encoder is used to generate a 768-dimensional feature vector for each image.

This process is performed on both the training and test sets. The resulting vectors form the basis for downstream clustering and classification tasks.

5.1 t-SNE Visualization of Feature Space

To gain insight into the separability and structure of the extracted features, we apply t-distributed Stochastic Neighbor Embedding (t-SNE), a dimensionality reduction technique that projects high-dimensional data into a 2D space while preserving local relationships between points.

Figures 2 and 5 show the 2D projections of the training and test features, respectively. The training set is color-coded by class to visualize how well the

ViT encoder separates different categories, while the test set, lacking labels, is shown in grayscale.

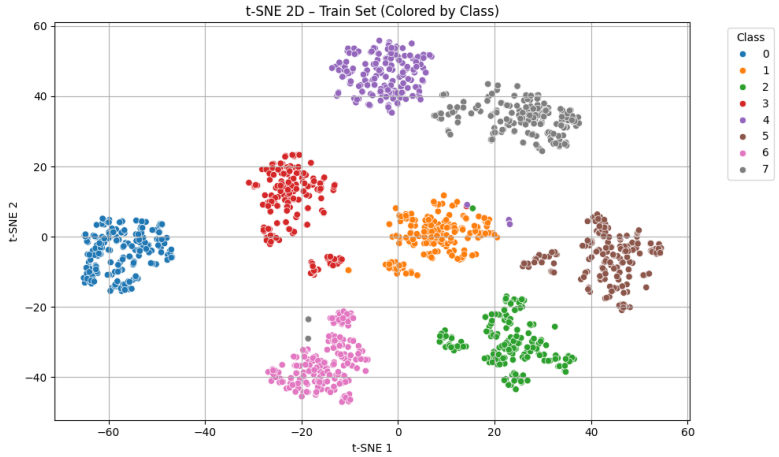

Figure 2: t-SNE 2D projection of training features (colored by class)

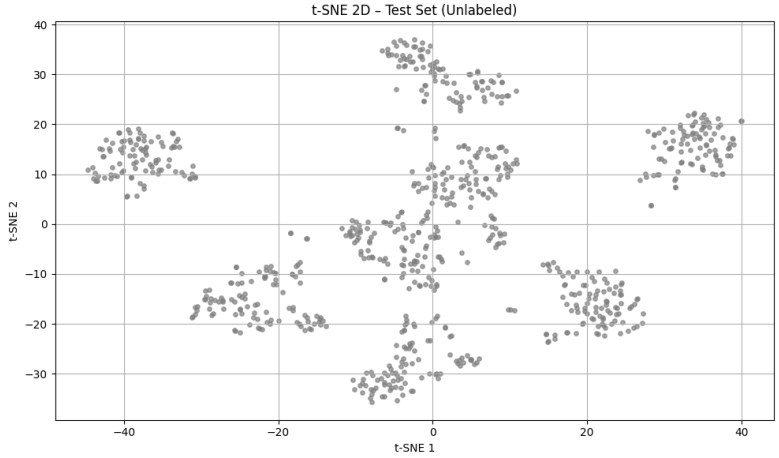

Figure 3: t-SNE 2D projection of test features (unlabeled)

As shown in Figure 2, the training features cluster clearly according to class, confirming the ViT encoder's ability to separate semantic content effectively. In Figure 5, the test features also form visible clusters, even in the absence of labels, indicating consistent feature extraction across domains despite the background shift.

These visualizations validate the effectiveness of the ViT-based encoder and motivate our use of clustering techniques in the classification stage.

6 Clustering and Class Assignment Strategy

After extracting features from both the training and test sets using the Vision Transformer encoder, we adopt an unsupervised clustering strategy to predict the classes of the test images. This approach does not require any labels from the test set, relying entirely on feature similarity.

6.1 Fixed Centroids from Training Set

We compute the centroids of each of the 8 training classes by averaging the extracted feature vectors for all images belonging to each class. These centroids represent the semantic centers of the training classes in the feature space and will serve as references for class assignment.

6.2 KMeans Clustering on Test Set

To cluster the feature vectors from the unlabeled test set, we employ the **KMeans algorithm** with $k = 8$. KMeans is an iterative unsupervised clustering algorithm that partitions the data into k clusters by minimizing the within-cluster variance.

Given a dataset $\{x_1, x_2, \ldots, x_n\}$ and a fixed number of clusters k, the KMeans algorithm proceeds as follows:

1. Initialize k cluster centroids (either randomly or using heuristics).
2. **Assignment step:** Assign each data point x_i to the nearest centroid c_j using Euclidean distance:
$$\text{assign } x_i \to \arg\min_j \|x_i - c_j\|^2$$
3. **Update step:** Recompute each centroid as the mean of all points assigned to it:
$$c_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$
4. Repeat steps 2 and 3 until convergence (i.e., cluster assignments no longer change significantly).

To mitigate randomness, we repeat this process for multiple random seeds (from 40 to 49), each time producing a unique partition of the test set into 8 clusters and computing their centroids.

6.3 Optimal Matching via Hungarian Algorithm

Since KMeans cluster indices are arbitrary and do not inherently correspond to the true class labels, we must determine the best way to match clusters to known class centroids from the training set. To accomplish this, we use

the **Hungarian algorithm** (also known as the Kuhn-Munkres algorithm), a combinatorial optimization algorithm that solves the *assignment problem* in polynomial time.

In our case, we consider the problem of matching the 8 cluster centroids from the test set to the 8 class centroids from the training set. The goal is to minimize the total cost, where the cost is defined as the Euclidean distance between centroids. The Hungarian algorithm takes as input the 8×8 distance matrix D where $D_{i,j} = \|c_i^{\text{test}} - c_j^{\text{train}}\|$, and finds the assignment (a permutation of labels) that minimizes the total cost:

$$\min_{\pi \in S_k} \sum_{i=1}^{k} D_{i,\pi(i)}$$

This results in a one-to-one mapping between clusters and classes that minimizes the total dissimilarity in feature space.

6.4 Label Assignment

Once the optimal mapping between clusters and training classes is established, we assign a class label to each test image according to its cluster. This yields a complete set of predicted labels for the test set.

6.5 Multi-Seed Evaluation

To identify the most effective clustering, we evaluate each seed using the following criteria:

- **Standard deviation of class counts:** Lower values indicate a more balanced distribution of predicted labels across the 8 classes.
- **Average intra-cluster distance:** Measures the compactness of clusters by computing the average distance of points to their respective cluster centroid.
- **Average centroid matching distance:** Quantifies the overall alignment between test cluster centroids and training class centroids.

The best-performing seed is selected based on the lowest standard deviation, with additional metrics used for support.

6.6 Distance Matrix and Matching Visualization

To support the cluster-to-class assignment process, we compute the full 8×8 Euclidean distance matrix between the centroids of the test clusters and the centroids of the training classes. Each cell (i, j) in the matrix represents the distance between the i-th test cluster and the j-th training class.

The Hungarian algorithm is then applied to this matrix to find the optimal assignment. Figure 4 visualizes the matrix, highlighting how well clusters align with class centroids. Darker cells indicate closer (more similar) pairs.

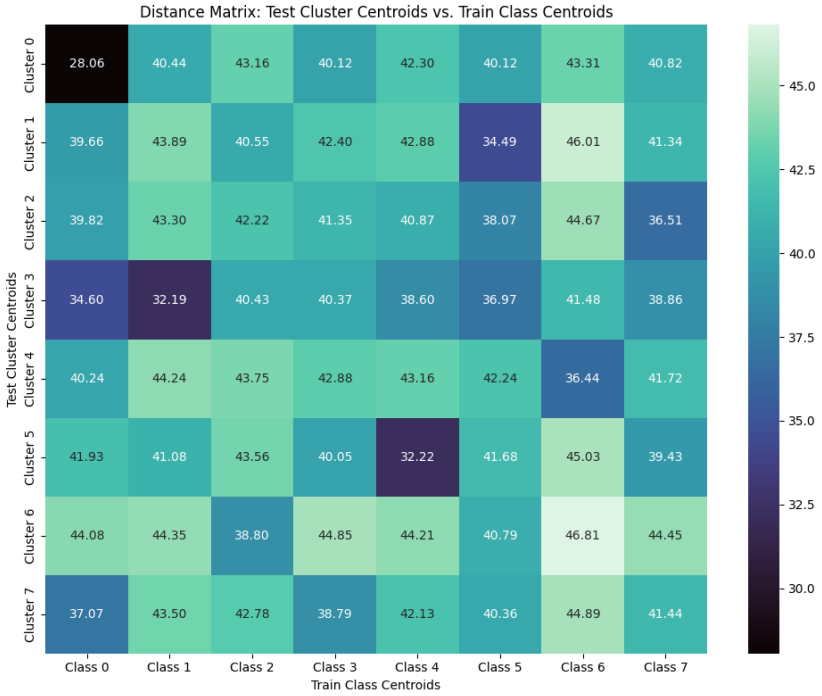

Figure 4: Distance Matrix: Euclidean distances between test cluster centroids and training class centroids. Darker values represent stronger similarity.

7 Results

7.1 t-SNE Visualization of Predictions

To qualitatively assess the quality of the clustering and class assignment strategy, we apply t-SNE dimensionality reduction on the extracted features from the test set and visualize the predicted classes obtained using the best-performing seed (Seed 42).

t-SNE (t-distributed Stochastic Neighbor Embedding) is a non-linear dimensionality reduction technique that maps high-dimensional data to a lower-dimensional space (typically 2D or 3D), while preserving local neighborhood structure. It is widely used in visualizing feature embeddings and understanding class separability after transformation.

As shown in Figure 5, the test instances form well-separated and dense clusters in the 2D space, with minimal overlap between predicted classes. This

indicates that the extracted features are highly discriminative, and the unsupervised class assignment based on KMeans and optimal matching has been effective.

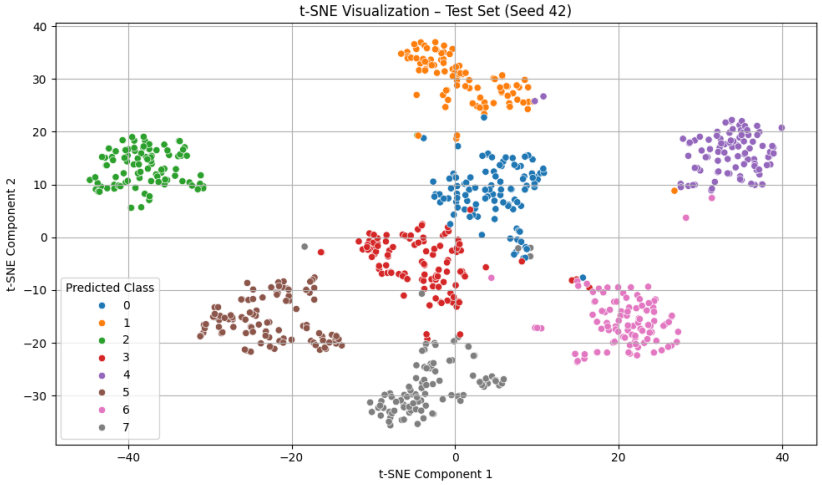

Figure 5: t-SNE Visualization – Test Set colored by predicted classes (Seed 42). Each point corresponds to a test image represented in 2D t-SNE space. The visualization shows clear separation between clusters, supporting the quality of predictions.

7.2 Quantitative Evaluation Across Seeds

To select the optimal random seed for KMeans clustering, we perform a comprehensive evaluation using three quantitative metrics:

- **Standard Deviation of Class Counts:** Measures the variation in the number of images assigned to each predicted class. A lower value implies more balanced predictions, which is desirable for avoiding class bias.
- **Average Intra-Cluster Distance:** For each cluster, we compute the average Euclidean distance of its members to the cluster centroid. We then average this value across all clusters. Lower values indicate that the samples within each cluster are more tightly packed and consistent.
- **Centroid Matching Distance:** After matching each test cluster to a class centroid using the Hungarian algorithm, we compute the average Euclidean distance between each test centroid and its matched training class centroid. Smaller distances suggest a better alignment between test clusters and the learned class distributions.

These metrics were calculated for seeds ranging from 40 to 49. Table 1 summarizes the results.

Table 1: Evaluation Metrics Across KMeans Seeds

Seed	STD Class Counts	Intra-Cluster Dist.	Centroid Matching Dist.
42	**4.58**	26.32	34.69
45	**4.58**	26.32	**34.68**
46	4.80	**26.30**	34.69
48	4.95	26.31	34.69
49	5.00	26.32	34.69
44	6.04	26.31	34.67
41	9.06	26.32	34.63
40	11.62	26.31	34.60
43	12.03	26.32	34.60
47	13.40	26.31	34.58

From the table, we observe that Seeds 42 and 45 yield the lowest standard deviation, indicating a highly balanced classification. Seed 46 achieves the lowest intra-cluster distance, suggesting the most compact clusters, while Seed 45 also provides the closest centroid alignment.

Overall, Seed 42 was selected for the final predictions due to its combination of balanced class distribution, low intra-cluster dispersion, and competitive centroid alignment distance, demonstrating the robustness of the proposed unsupervised assignment pipeline.

References

[1] Yuning Xu, Weiping Cao, Junlin Zheng, Yifan Yang, and Yu Bai. *The overall architecture of the pretrained ViT-B/16 for penetration recognition and the proposed ViT-B/16-SPN.*
Available at: `https://www.researchgate.net/figure/The-overall-architecture-of-the-\\pretrained-ViT-B-16-for-penetration-recognition-and-the_fig5_357859326`.
Accessed: April 2025.