algorithms

MDPI

Article

Beyond Pseudo-Labels: Dual-Level Knowledge Distillation for Enhanced Deep Clustering

Rehab Alnefaie [1,2], Mohamed Maher Ben Ismail [1,*] and Ouiem Bchir [1]

[1] Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia; r.alnefaie@psau.edu.sa (R.A.); obchir@ksu.edu.sa (O.B.)
[2] Computer Science Department, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia
* Correspondence: mbenismail@ksu.edu.sa

Abstract

This study introduces a novel clustering approach, namely Teacher–Student-based Deep Clustering (TSDC), that relies on intra- and inter-distillation-based feature representations. In fact, TSDC distils knowledge from (*i*) high- to low-response channels, forcing the latter to mimic the former, and (*ii*) deeper to shallower layers, prompting the transfer of semantic information to enhance representational consistency. Unlike CNN-based deep clustering that relies on pseudo-labels to improve representations, TSDC introduces a newly formulated objective function to simultaneously minimize losses from clustering and intra- and inter-distillation. The proposed approach was rigorously investigated using benchmark datasets and relevant performance measures. In particular, a linear top classifier protocol was adopted to assess TSDC performance. Notably, TSDC outperformed existing deep clustering frameworks, yielding improved classification accuracy. The empirical findings highlight the efficacy of combining intra- and inter-distillation to enrich feature representations. Notably, the most prominent improvement was observed on CIFAR-100, where classification accuracy rose from 54.77 ± 0.07 to 57.58 ± 1.85.

Keywords: deep learning; unsupervised learning; deep clustering; knowledge distillation; self-distillation; representation learning

1. Introduction

Machine Learning (ML) is one of the most active Artificial Intelligence (AI) subfields that contribute significantly to the evolving variety of applications. Unlike other conventional paradigms that require explicit programming, ML exploits available data to learn predictive and descriptive models. Currently, ML approaches are increasingly being employed even in complex domains, highlighting the high-accuracy results obtained [1]. The predominant paradigm within ML is supervised learning which builds predictive models using labelled data. In other words, the use of supervised learning techniques is subject to labelled data availability [1]. In fact, most of the real-world datasets are originally unlabelled, and the manual annotation process is time-consuming, costly, and subjective. This promoted the shift toward unsupervised learning, which aims to model data and uncover hidden patterns without utilizing labelled examples [2].

Unsupervised learning techniques have been introduced as prominent alternatives to capture hidden structures with no supervisory information. In particular, conventional clustering approaches require handcrafted feature extraction to encode data modalities into

Check for updates

Academic Editor: Yu-Chen Hu

Received: 15 June 2026
Revised: 30 August 2026
Accepted: 31 August 2026
Published: 2 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

numeric vectors [2]. On the other hand, deep learning techniques, such as Convolutional Neural Network (CNN)-based approaches [3], have automated the extraction of hierarchical relevant features from raw data [4]. Consequently, deep clustering has been perceived as a successful association of CNN-based feature extraction with clustering algorithms. In addition, pseudo-labelling has been coupled with deep clustering and used to guide the optimization process [4]. Nevertheless, the performance of deep clustering proved to be constrained by the fact that shallower layers do not learn and extract meaningful features [5].

The existing deep clustering paradigm has demonstrated superior efficacy in addressing the challenging representation of high-dimensional data compared to traditional clustering methods. However, large CNN models still suffer from redundant channels that yield insufficient visual concepts [6]. Moreover, shallow layers are the poorest of task-relevance features, or simply low-level, domain-agnostic features, making them less suited for capturing task-specific semantics [5]. On the other hand, the representation of deep layers includes more visual features related to the intended task. Consequently, the information provided by this deeper representation often supports the distillation model [5]. Knowledge Distillation (KD) has emerged as an alternative approach intended to enhance channel information using the teacher–student approach [7]. Initially, it was first introduced to improve the learning capacity of supervised learning models by allowing a lightweight student model to mimic a larger teacher model and relaxing the need for labelled data [7]. One should note that the scarcity of teacher models led to the design of self-distillation approaches in which the teacher is represented by the deeper layers of the same network while the student encloses the network's shallower layers [8]. In addition, self-distillation has been coupled with supervised learning through either distilling knowledge from logits [8] or from feature representations [6]. Self-distillation has been theoretically demonstrated to enhance the regularization properties of learning models [9]. Moreover, only a few researchers have explored the combination of clustering and Knowledge Distillation [10]. In other words, there remains a clear gap in designing an end-to-end, knowledge-base-driven deep unsupervised learning model.

This research introduces a novel teacher–student deep clustering framework that simultaneously leverages pseudo-labels to optimize the objective function and Knowledge Distillation (KD) to refine feature representations, thereby combining the strengths of both paradigms to enhance feature representations. In particular, the main research contribution consists in designing an objective function that combines the clustering loss along with the KD terms in order to optimize the parameters of the proposed deep clustering model. Specifically, the unsupervised intra-distillation terms let low-response channels learn from high-response channels with no supervision information. In addition, an unsupervised inter-distillation term is incorporated to incentivise the shallower layers of the proposed network to function as student models. Moreover, the deeper layers of the network thereby serve as teacher models. Accordingly, TSDC is designed as a clustering-based self-supervised representation learning framework, where cluster assignments act as intermediate supervision signals instead of serving as final semantic outputs. The novelty of TSDC resides in how knowledge transfer is positioned and structured within a single backbone, in contrast to single-backbone approaches that pass both feature and prediction knowledge from a primary teacher head to auxiliary student heads.

The rest of this article is organized as follows: Section 2 discusses related works relevant to deep clustering and its association with KD; Section 3 details the proposed Teacher–Student-based Deep Clustering (TSDC) approach; the experimental results are presented and thoroughly discussed in Section 4; finally, Section 5 presents a detailed discus-

sion and interpretation of the results obtained, while Section 6 concludes this investigation and outlines the intended future research directions.

2. Related Work

Deep learning algorithms have recently demonstrated effectiveness in advancing unsupervised learning methodologies, particularly in addressing challenges associated with constructing and optimizing feature representations [11]. In the following, we explore relevant deep clustering and deep clustering-based KD works. Several surveys [11–13] have systematically reviewed deep clustering approaches that address different architectures, open challenges, and future directions. The authors in [12] conducted a review of the literature and categorized the selected publications into four architectural groups: autoencoders, feed-forward networks, Generative Adversarial Networks (GANs), and Variational AutoEncoders (VAEs). They investigated and analyzed the optimization of network and clustering loss functions. The researchers in [13] investigated several training strategies, including multi-stage, iterative, generative, and simultaneous schemes. Moreover, they analyzed the effectiveness of these approaches with respect to representation learning and clustering performance. A more recent survey [11] addressed data configurations not only in the traditional single-view setting, as in earlier work, but also in more advanced paradigms such as semi-supervised, multi-view, and transfer-based deep clustering.

In [14], K-means was employed to pre-train convolutional networks in a bottom-up fashion. It was not optimized over all parameters in an end-to-end fashion; instead, each layer was trained sequentially. In the current layer, K-means clustering is applied to image patches to construct a dictionary of centroids, which are subsequently employed as convolutional filters, followed by a pooling operation. In the subsequent layer, only the pooled responses from the previous layer are utilized, and the model parameters across layers are not jointly optimized. The Deep Embedded Clustering (DEC) approach proposed in [15] includes an autoencoder that serves as a backbone responsible for learning relevant features. During the fine-tuning stage, the encoder maps observations into a latent space. The DEC network is optimized by minimizing an objective function that matches the soft assignment probabilities learned in the embedding space with a predetermined target distribution. One should mention that DEC struggles in scenarios dealing with overlapping clusters. To address this limitation, the authors in [16] suggested Discriminatively Boosted Clustering (DBC) which combines a fully convolutional autoencoder with a soft K-means clustering algorithm. Another line of research has focused on agglomerative clustering [17], which does not require the pre-definition of the number of clusters. Moreover, the framework depicted in [18] relied on iterative collaboration learning between the agglomerative clustering of images and CNNs to generate pseudo-labels in order to refine the intended model.

An iterative learning framework where CNN features are partitioned via K-means clustering was proposed in [4]. The resulting cluster assignments are subsequently employed as pseudo-labels to supervise the training of the network. Specifically, ResNet [19] is employed as a feature extractor. In addition, the supervision signals provided by the prototype layer are updated along with the new centroids. The model optimization is performed using the Kullback–Leibler (KL) divergence between the assignments and the Softmax output of the network. Let x be an input sample and K the number of clusters. Accordingly, the KL-based objective function is computed as follows:

$$L_{KL} = \sum_{x} \sum_{k=1}^{K} p_k(x) log\left(\frac{p_k(x)}{q_k(x)}\right) \quad (1)$$

where $p_k(x)$ represents the target assignment distribution and $q_k(x)$ denotes the Softmax output distribution. However, its performance is strongly contingent upon the quality and

temporal stability of the pseudo-labels, which can exhibit substantial noise, particularly during the initial phases of training. The research in [20] relied on the extraction of a density core for each cluster, which was employed instead of centroids in order to enhance pseudo-label quality during the optimization process. In [21], the researchers introduced Semantic Clustering by Adopting Nearest Neighbours (SCAN), which adopts a self-supervision method to train the ResNet model. Their main contribution consists in employing an intelligent filter to distinguish between reliable and unreliable pseudo-labels. In particular, a reliability-based weighting technique was used to guide the clustering algorithm. The Swapping Assignments between Views (SwAV) framework proposed in [22] relies on a dual-branch neural network architecture that combines two feature extractors and a prototype layer. Moreover, a swap stage between the branch predictions to maintain clustering consistency was introduced. Consistently, the network was progressively updated through minimizing the loss computed using the predictions of both branches. Furthermore, the Sinkhorn step [23] was considered to keep the prototypes balanced. The authors in [24] introduced the Stable Cluster Discrimination (SeCu) method to ensure the stability of models trained on limited batches, thereby enabling the development of high-performance predictive models. Specifically, a clustering layer was appended from two views, and the assignments were obtained directly through a Softmax clustering layer instead of using a conventional clustering algorithm. The key improvement is that the centroids were updated in small batches based on the gradient from positive samples only. Additionally, these samples were weighted based on their hardness and their constraints regarding cluster assignments. The authors of [25] focused on examining and enhancing the adversarial robustness of deep multi-view clustering methods. First, they employed a GAN to produce adversarial perturbations that disrupt consistency and completion. Next, they applied adversarial training to fit the clustering model, using a combination. Finally, an Attack Mitigator was integrated to lessen the impact of adversarial perturbations and enhance the robustness of the clustering process.

The authors in [10] remarked that data augmentation affects the performance of deep clustering methods. They also found that the deep clustering [4] framework along with a self-distillation strategy represents an alternative regularization approach to data augmentation. They employed ResNet50 [19] as a feature extractor and modified its architecture to adapt it to the proposed self-distillation strategy. For the implementation of the student–teacher network, each layer was extended by adding auxiliary classifier heads as students and a single classifier head as a teacher. One should note that the distillation was conducted at two levels: feature representation and prediction. In addition, the objective function employed for feature distillation facilitates the transfer of implicit knowledge from the teacher's feature representations to those of the student, thereby guiding the learning of the student's feature maps. In parallel, for prediction distillation, the teacher's output is used to supervise the training of the student's prediction heads. The overall objective function was combined with the main classifier loss, the auxiliary head loss, and the self-distillation losses. To evaluate the model, the authors applied a top classifier protocol, in which the parameters of the feature extractor were frozen, and a linear classifier was appended on top of the model and consequently trained. The method introduced in [26] relies on a deep clustering framework based on ResNet-50 with extended classification heads. In addition, it adopts the same optimization objective and self-distillation strategy in [10]. The researchers also used a contrastive learning mechanism during the pre-training stage where two augmented views of the same image are generated. Then, both views are treated as a positive pair, where the other samples represent the negative one. More recently, ref. [27] introduced a Redundancy-Aware External Signal Learning framework that leverages an information bottleneck to refine external semantic features by filtering

out redundant content. To mitigate semantic inconsistencies between visual and external representations, the framework utilizes multiple clustering heads along with cross-head distillation.

In [28], the researchers observed that overconfident pseudo-labels constitute a significant issue in many multi-model settings, particularly in multi-view deep clustering. In fact, overconfident pseudo-labels can introduce errors into the prediction stage and, consequently, affect the model optimization during training. To address this issue and mitigate the overconfidence associated with pseudo-labels, the authors proposed an approach that integrates multi-view deep clustering with distillation. The multi-view deep clustering extraction shared a semantic representation across views and generated hard pseudo-labels by employing the K-means clustering algorithm. The distillation strategy offered an alternative approach by providing soft probabilistic outputs as guidance. Additionally, a teacher network is responsible for providing soft probabilistic output distributions, which the student network uses as a reference during training. Through distillation, the student network enhances its predictive capabilities. Another dual-network method is explored in [29], which finds a gap in existing deep clustering research, namely the assumption that the importance and difficulty of clustering all samples are equal; in reality, the difficulty of clustering differs. A self-distillation mechanism was proposed to identify the most challenging positive and negative samples. During training, the samples are processed by adaptively reweighting them within a self-distillation framework. More recently, the researchers extended the association of deep clustering with Knowledge Distillation (KD) to address multitask learning [30]. Such a framework incorporates a teacher network and multiple student networks, and also employs both forward and backward Knowledge Distillation. For forward KD, the prior knowledge, encoded in the student networks, guides the teacher network in learning new clustering structures. On the other hand, during the backward KD phase, a newly introduced student network is trained to memorize the clustering assignments produced by the teacher. In [31], the authors presented a deep clustering paradigm with a pairwise relational distillation approach that involves extracting features using CNNs. Within this framework, a teacher head and a student head are employed to model sample relationships. In particular, the teacher head is intended to compute the pairwise similarity across samples. Subsequently, the student head is designed to mimic the relations provided by the teacher head, with the aim of enhancing the quality of the cluster assignments.

The works presented in [32,33] demonstrate the effectiveness of Knowledge Distillation in conjunction with online clustering; however, their primary focus is on speech representation learning rather than on image-based deep clustering. The authors in [32] enhanced the effectiveness of an online clustering methodology by jointly leveraging self-distillation and masked language modelling, by improving the representation and capturing complex speech patterns. They began by feeding the input audio into a teacher network to extract contextualized embeddings and subsequently applied online clustering to these embeddings to derive a machine-discovered phonetic inventory. In the final stage, a student network was trained on the discretized representation to optimize the objective function. Another study [33] integrated online clustering with Knowledge Distillation to improve the quality and discriminative power of speaker feature representations. A teacher network continually refined pseudo-labels by performing clustering and distilling the resulting information to the student network. Importantly, the approach incorporated specialized filtering procedures designed to detect noise and identify potentially unreliable labels.

Despite the recent advances in deep clustering, the optimization of neural networks based on pseudo-labels derived from the clustering algorithms still faces several challenges. In fact, the pseudo-labels can be noisy and unstable, specifically in early epochs. Moreover,

CNN layers in the feature learning stage may face obstacles such as early layers having limited semantic abstraction. Studies on supporting deep clustering through KD information remain relatively scarce. For instance, the research in [10] is based on transferring a logit distribution from the teacher head to the student heads. However, this approach is prone to clustering instability. Conversely, dual-stream methodologies, such as SwAV introduced in [22], employ the concurrent processing of two distinct views of the same scene, which entails higher memory consumption and consequently increases the computational cost of training.

3. Proposed Teacher–Student-Based Deep Clustering Approach

In this section, we introduce a novel Teacher–Student-based Deep Clustering (TSDC) approach that incorporates distillation information without requiring an extra teacher network. TSDC is distinct from prior approaches not just because it merges two loss functions, but because it conducts knowledge transfer at two internal points within a single backbone: across channels within the same layer and between layers at different depths, linking deeper and shallower representations. Specifically, we define intra-distillation where low-response channels are motivated to mimic the high-response channels as a teacher. In addition, inter-distillation contributes to transferring semantic information from the deepest semantic representations to early layers. This allows for adjusting parameters of the network using deep clustering, intra-distillation, and inter-distillation loss functions. Thus, the proposed TSDC approach can be perceived as a deep clustering framework grounded in a KD architecture which leverages more informative feature representations to exploit information from deeper network layers to supervise and refine the representations learned by the shallower layers. Figure 1 depicts the overall workflow of the proposed method, comprising feature representation, pseudo-labels generated using the K-means clustering algorithm, and subsequent network optimization guided by appropriately designed loss functions. One should mention that the learnable parameters are updated based on the proposed deep clustering, intra-distillation, and inter-distillation loss functions.

The primary factor in key optimizing the proposed network is the quality of the pseudo-labels, which in turn depends on the network's capacity to learn and represent a discriminative feature space. Therefore, we first embed all samples, denoted as $z_i = f_\theta(x_i)$, and store the resulting representations in a memory bank M. During the clustering stage, the K-means algorithm runs to derive the pseudo-labels, $\hat{y}$, and the corresponding cluster centroids. The clustering loss utilizes a cross-entropy function computed for each mini-batch to encourage the network outputs to be harmonious with the corresponding pseudo-labels as follows:

$$L_{cluster} = \frac{1}{|B|} \sum_{x_i \in B} L_{CE}\left(p_i, \hat{y}_i\right) \tag{2}$$

where z_i represents the embedding, and p_i denotes the output of the prototype layer produced by ResNet-50. Finally, after backpropagation, the memory bank entry associated with each individual sample is updated accordingly. One should note that a stop-gradient step is applied to the current embedding during updating the memory bank, thereby preventing any backpropagation updates to the model parameters. The proposed deep clustering algorithm is delineated in Algorithm 1.

Algorithm 1 Deep Clustering

Require: Dataset D, model f_θ, number of clusters K

Initialize memory bank **M** for all samples in D
for epoch = 1 to E **do**
  $\hat{y}, \hat{C} \leftarrow$ K-means(**M**, K)
  for all mini-batch $B = \{x_i\}$ **do**
    $(z_i, p_i) \leftarrow f_\theta(x_i)$
    $\mathcal{L}_{\text{cluster}} \leftarrow \ell_{\text{CE}}(p_i, \hat{y}_i)$
    Update θ using $\nabla_\theta \mathcal{L}_{\text{cluster}}$
    $\mathbf{M}[x_i] \leftarrow \text{STOPGRAD}(z_i)$
  end for
end for

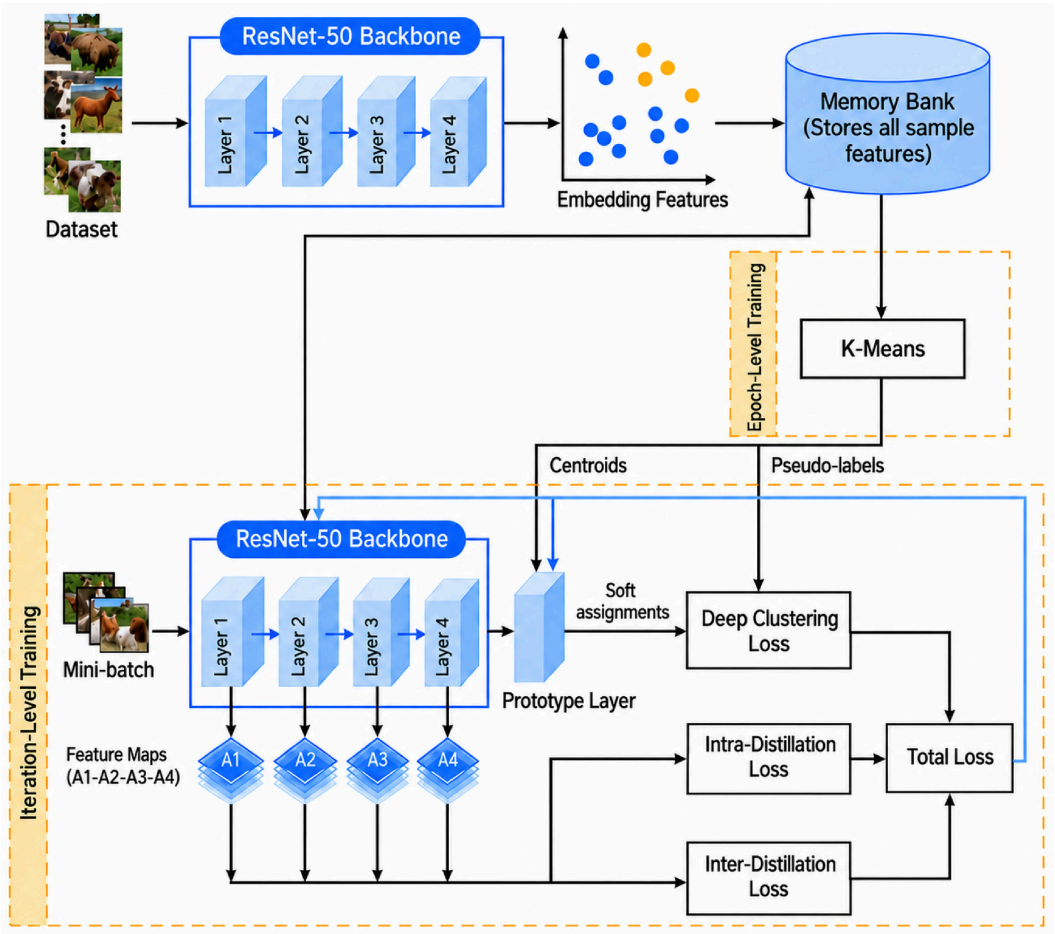


Figure 1. Architecture overview of the proposed TSDC approach.

In addition, we design an unsupervised intra-distillation loss, which is jointly optimized to guide the pseudo-label based learning process. The central concept involves employing high-response channels as a teacher signal to supervise and guide the learning of low-response channels, which are treated as a student representation. We first sort the channels within the same layer according to their respective L2-norm values. In reality, the L2 norm is applied solely for ranking the channel responses. A higher norm value indicates a higher response magnitude; therefore, the associated channel is chosen as the teacher signal. Alternatively, a lower norm value indicates a lower response magnitude, and the associated channel is represented as a student channel that learns from the teacher channel. For each selected layer l, let $F_l \in \mathbb{R}^{B \times C_l \times H_l \times W_l}$ denote its feature map, where B, C_l, H_l, and

W_l correspond to the batch size, number of channels, height, and width, respectively. For each channel c, we calculate an L2-based response score, defined as the squared L2 norm of its activation map, normalized over the batch and spatial dimensions. The channels are then sorted in descending order based on these response scores. Let $M_l = \lfloor C_l/2 \rfloor$. The top, M_l channels in this sorted list constitute the high-response teacher tensor $F_l^{[T]}$, while the subsequent, M_l channels form the low-response student tensor $F_l^{[S]}$. As a result, both tensors share the same shape $B \times M_l \times H_l \times W_l$, as depicted in Figure 2.

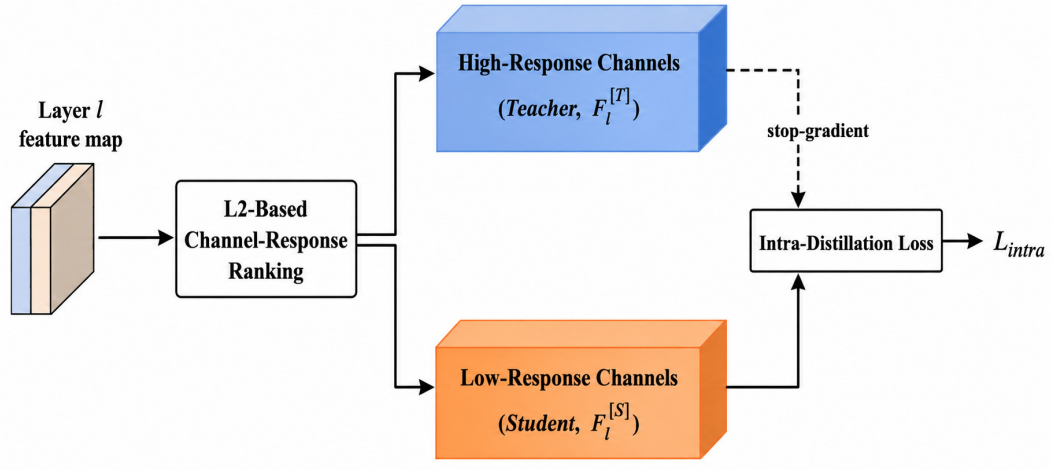


Figure 2. The intra-distillation loss is computed by channel-response ranking.

The teacher and student channels are matched based on their respective relative positions in the two sorted halves. In particular, the channel at index k in the student tensor is paired with the channel at index k in the teacher tensor. Before computing the intra-distillation loss, the teacher tensor is detached from the gradient flow. The proposed loss is then formally defined as follows:

$$r_{l,c} = \frac{1}{BH_lW_l} \| F_l[:, c, :, :] \|_2^2, \tag{3a}$$

$$r_{l,\pi_l(1)} \geq r_{l,\pi_l(2)} \geq \cdots \geq r_{l,\pi_l(C_l)}, \; M_l = \left\lfloor \frac{C_l}{2} \right\rfloor, \tag{3b}$$

$$F_l^{[T]} = F_l[:, \pi_l, (1 : M_l), ::], \tag{3c}$$

$$F_l^{[S]} = F_l[:, \pi_l, (M_l + 1 : 2M_l), ::], \tag{3d}$$

$$\mathcal{L}_{intra} = \frac{1}{|\mathcal{I}|} \sum_{l \in \mathcal{I}} MSE\left(F_l^{[S]}, sg\left(F_l^{[T]}\right)\right), \tag{3e}$$

where $r_{l,c}$ represents the L2-based response score of channel c in layer l, and π_l represents the permutation of channel indices obtained by sorting these response scores in descending order. The set $\mathcal{I}$ denotes the layers chosen for intra-distillation, and sg(.) corresponds to the stop-gradient operation. The MSE loss aligns student and teacher channels based on their relative positions within their respective sorted halves. This rank-based matching yields a straightforward, deterministic, and parameter-free correspondence grounded in relative activation strength. It does not require that the matched channels encode the same semantic features; instead, it serves as a heuristic to pass guidance from high- to low-response channels, avoiding the need for an extra learnable matching component. Consequently, a semantic match between the two channels in each pair cannot be assumed.

For completeness, when the number of channels is odd, the single remaining unpaired channel is excluded from the intra-distillation loss.

Algorithm 2 provides an overview of the full workflow, including channel ranking, tensor construction, rank-based pairing, and loss computation.

Algorithm 2 Intra-distillation

Require: Feature maps $\{F_l\}_{l \in \mathcal{I}}$ from the selected layers

$\mathcal{L}_{\text{intra}} \leftarrow 0$
for each selected layer $l \in \mathcal{I}$ **do**
  Let $F_l \in \mathbb{R}^{B \times C_l \times H_l \times W_l}$
  for $c = 1$ to C_l **do**
    $r_{l,c} \leftarrow \|F_l[:, c, :, :]\|_2^2 / (B \cdot H_l \cdot W_l)$
  end for
  $\pi_l \leftarrow \text{argsort}_{\text{desc}}(\{r_{l,c}\}_{c=1}^{C_l})$
  $M_l \leftarrow \lfloor C_l / 2 \rfloor$
  $F_l^{[T]} \leftarrow F_l[:, \pi_l(1 : M_l), :, :]$
  $F_l^{[S]} \leftarrow F_l[:, \pi_l(M_l + 1 : 2M_l), :, :]$
  Pair the student and teacher channels according to their corresponding ranks
  $\mathcal{L}_{\text{intra}}{}^{(l)} \leftarrow \text{MSE}(F_l^{[S]}, \text{sg}(F_l^{[T]}))$
  $\mathcal{L}_{\text{intra}} \leftarrow \mathcal{L}_{\text{intra}} + \mathcal{L}_{\text{intra}}{}^{(l)}$
end for
$\mathcal{L}_{\text{intra}} \leftarrow \mathcal{L}_{\text{intra}} / |\mathcal{I}|$
return $\mathcal{L}_{\text{intra}}$

Furthermore, we propose an unsupervised inter-distillation framework that enables the extraction of the most semantically informative high-level representations from deeper layers and their subsequent transfer to shallower layers. In particular, an alignment strategy is applied to the feature maps prior to distillation in order to ensure that they are compatible in terms of dimensionality. Specifically, given a collection of feature maps organized into network stages in an increasing order, we utilize adaptive average pooling as the alignment strategy, denoted as ALIGNPAIR$(F_{\ell i}, F_{\ell i+1})$, which is applied to the feature maps $(F_{\ell i}, F_{\ell i+1})$, associated with a given pair of layers and its outputs are (A_i, A_{i+1}). This solves the mismatches risks in the spatial resolutions. Their channel dimensions are then aligned by keeping, from each feature map, only the initial subset of channels that matches the smaller channel size. As a result, the student and teacher feature maps become identical in both spatial and channel dimensions, enabling the computation of the MSE loss. In our approach, a teacher signal is extracted from the deeper representation A_{i+1} and is activated in a stop-gradient mode. In other words, the teacher provides guidance to the student and does not participate in backpropagation processes. We refer specifically to the configuration in which only the shallower (student) layer updates its parameters, as explained in Figure 3.

The main idea is that a student layer ℓ_i mimics a teacher layer ℓ_{i+1}. Consistently, an inter-distillation loss, namely L_{inter} , is expressed as follows:

$$L_{inter} = \frac{1}{|\,\mathrm{L}\,| - 1} \sum_{i=1}^{|\mathrm{L}|-1} MSE(A_i,\ sg(A_{i+1})) \tag{4}$$

where $|\mathrm{L}|$ represents the number of selected layers.

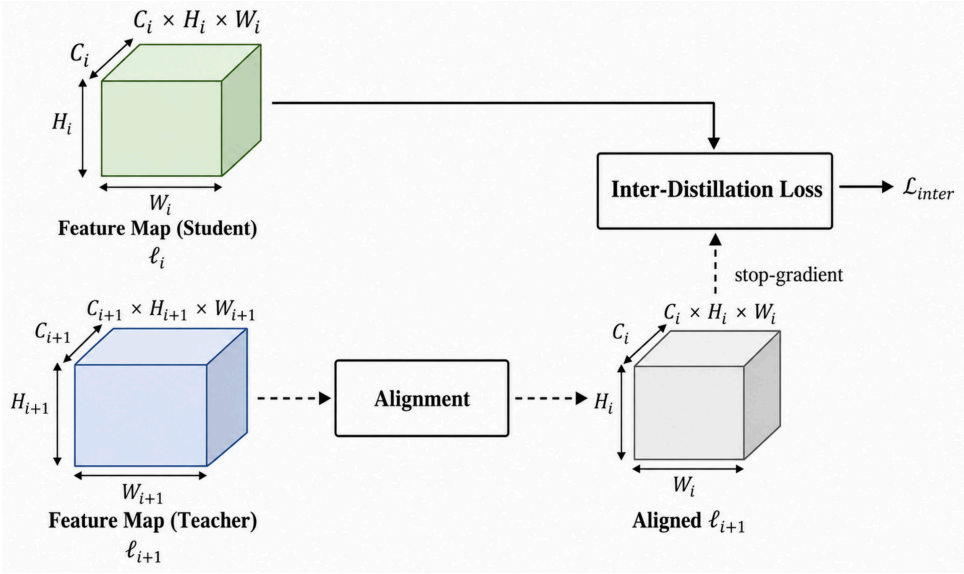


Figure 3. Inter-distillation alignment between the student layer ℓ_i and the teacher layer ℓ_{i+1}. The dotted arrows indicate the aligned teacher pathway with stop-gradient.

In the final stage, the mean-squared error, $\mathrm{MSE}(A_i, sg(A_{i+1}))$, is computed for all adjacent pairs, after which these values are summed and subsequently averaged. The computation of the inter-distillation loss function is detailed in Algorithm 3.

Algorithm 3 Inter-distillation

Require: Feature maps from selected layers $\{F_l\}_{l\in\mathcal{L}}$, where $\mathcal{L} = \{l_1, \ldots, l_{|\mathcal{L}|}\}$ is ordered from shallow to deep

$\mathcal{L}_{\text{inter}} \leftarrow 0$
for $i = 1$ to $|\mathcal{L}| - 1$ **do**
    $(A_i, A_{i+1}) \leftarrow \text{ALIGNPAIR}(F_{l_i}, F_{l_{i+1}})$
    $\mathcal{L}_{\text{inter}} \leftarrow \mathcal{L}_{\text{inter}} + \text{MSE}(A_i, \text{sg}(A_{i+1}))$
end for
$\mathcal{L}_{\text{inter}} \leftarrow \mathcal{L}_{\text{inter}} / (|\mathcal{L}| - 1)$
return $\mathcal{L}_{\text{inter}}$

The proposed TSDC method aims to improve both the quality and discriminative capability of learned feature representations by employing clustering-based self-supervision. Accordingly, we shape the intra and inter-distillation losses to support the main deep clustering loss. The training procedure of the model comprises four principal stages: First, a memory bank is constructed for all samples as per the obtained feature representations. Subsequently, at each training epoch, the K-means algorithm is applied to update the cluster centroids and to assign pseudo-labels to the samples within each batch. To optimize the proposed network, the pseudo-labels are used to improve the subsequent representation. The total loss is formulated as follows:

$$L_{total} = L_{cluster} + \lambda_{intra} L_{intra} + \lambda_{inter} L_{inter} \tag{5}$$

where λ_{intra} denotes the weighting factor associated with the intra-distillation loss, and λ_{inter} denotes that associated with the inter-distillation loss. Finally, the memory bank is updated with the most recent detached embeddings, thereby preserving an up-to-date and consistent representation of the underlying feature space. The main steps of the TSDC algorithm are detailed in Algorithm 4.

Algorithm 4 Teacher–Student Deep Clustering based on Knowledge Distillation (TSDC)

Require: Dataset D, model f_θ, selected layers $\mathcal{L}$, number of clusters K, weights λ_{intra}, λ_{inter}

Initialize memory bank $\mathbf{M}$ for all samples in D
for epoch = 1 to E **do**
  $\hat{y}, \hat{C} \leftarrow$ K-means($\mathbf{M}$, K)
  for all mini-batch $B = \{x_i\}$ **do**
    $(z_i, p_i), \{F_l\}_{l \in \mathcal{L}}) \leftarrow f_\theta(x_i)$
    $\mathcal{L}_{\text{cluster}} \leftarrow \ell_{\text{CE}}(p_i, \hat{y}_i)$
    $\mathcal{L}_{\text{intra}} \leftarrow$ Intra-distillation($\{F_l\}_{l \in \mathcal{L}}$)
    $\mathcal{L}_{\text{inter}} \leftarrow$ Inter-distillation($\{F_l\}_{l \in \mathcal{L}}$)
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{cluster}} + \lambda_{\text{intra}} \mathcal{L}_{\text{intra}} + \lambda_{\text{inter}} \mathcal{L}_{\text{inter}}$
    Update θ using $\nabla_\theta \mathcal{L}$
    $\mathbf{M}[x_i] \leftarrow$ STOPGRAD(z_i)
  end for
end for
return trained model f_θ

4. Experiments

This section delineates the experimental pipeline employed to assess the proposed method. First, we introduce the dataset in detail. Next, we describe the experimental configuration and the evaluation protocol. Finally, we conduct a diverse set of experiments and provide a comprehensive analysis of the results obtained.

4.1. Datasets

To assess the effectiveness of the proposed TSDC model, we consider a diverse collection of datasets. Namely, MNIST [34], CIFAR-10 [35], CIFAR-100 [35] and TissueMNIST [36] were used in these experiments. Relevant statistics on these datasets are shown in Table 1.

Table 1. Overview of the datasets used in the experiments.

Dataset	Total Instances	Train	Validation	Test	Image Size	Channels	Classes
MNIST	70,000	48,000	12,000	10,000	28 × 28	1/converted to 3	10
CIFAR-10	60,000	40,000	10,000	10,000	32 × 32	3	10
CIFAR-100	60,000	40,000	10,000	10,000	32 × 32	3	100
TissueMNIST	236,386	165,466	23,640	47,280	28 × 28	1/converted to 3	8

MNIST dataset [34] provides 70,000 grayscale handwritten digit images. That is, it covers 10 classes from zero to nine. The resolution of each sample is 28 × 28 pixels. Some sample instances from the MNIST dataset are presented in Figure 4a. It should be noted that, in this study, the MNIST images are converted to three-channel format and resized to conform to the input dimensionality requirements of the ResNet architecture. On the other hand, CIFAR-10 dataset [35] includes 10 categories covering some animals and transportation images. It has 60,000 RGB images of dimensions 32 × 32. Figure 4b exhibits examples from the CIFAR-10 dataset.

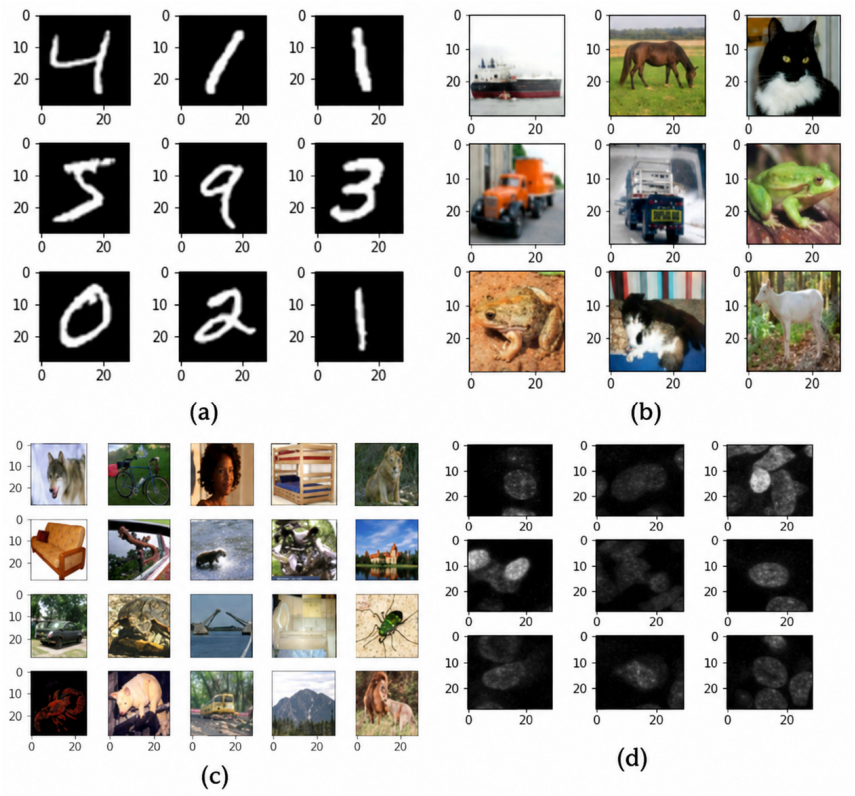


Figure 4. Example images from benchmark datasets: (**a**) MNIST [34], (**b**) CIFAR-10 [35], (**c**) CIFAR-100 [35], and (**d**) TissueMNIST [36].

Similarly, CIFAR-100 [35] comprises 60,000 RGB images, distributed over 100 classes. Each image has dimensions 32 × 32. In Figure 4c, we present some examples from CIFAR-100 dataset. Finally, TissueMNIST dataset [36] covers 236,386 images of human kidney cortex cells. It is resized to 28 × 28 and categorized into 8 tissue classes. The data are transformed into a three-channel representation to ensure compatibility with the ResNet backbone. The sample from the dataset is presented in Figure 4d.

One should note that for MNIST, CIFAR-10, and CIFAR-100 datasets, we adopted the original training and test partitions. Subsequently, we further divided the original training set into distinct training and validation subsets using an 80/20 split. On the other hand, for TissueMNIST dataset, we used the training, validation, and test partitions as released by the dataset provider.

4.2. Experimental Setup

This research experiment was run on NVIDIA GeForce RTX 5080 (16 GB) GPU (NVIDIA Corporation, Santa Clara, CA, USA). PyTorch (version 2.9.1+cu128) was used to implement our proposed method. Besides, ResNet-50 [19] was utilized as a feature extractor, followed by a projection head and a prototype layer. Detailed information on the layer configuration, the number of blocks, and the corresponding output channels is reported in Table 2.

All models were trained for 100 epochs using Stochastic Gradient Descent (SGD) [37] to optimize the network parameters. Across all datasets, the model was trained using a batch size of 64, an initial learning rate of 0.1, and a final learning rate of 0.001. The number of prototypes was set to 5 for MNIST and CIFAR-10. On the other hand, we set it to 50 and 4 for CIFAR-100 and TissueMNIST, respectively. Note that the intra-distillation was applied to Layers 2 and 4, whereas the inter-distillation was imposed across Layers 2, 3, and 4. Moreover, the intra-distillation and inter-distillation loss weights were initialized to

0.0008 and 0.00025, respectively, and were kept fixed during training for the first 60 epochs in order to allow the distillation signals to effectively guide the representation learning process. Subsequently, we used a linear decay strategy to progressively attenuate their influence during the later stages of training. Unless specified otherwise, all experiments were performed with the random seed set to 123. Besides, the proposed approach was evaluated using a linear classification protocol [4]. Because TSDC uses clustering as an intermediate form of self-supervision, the assessment centers on how good the learned feature representations are. In line with DeepCluster [4] and SwAV [22], the backbone network is kept fixed after unsupervised training, and a linear classifier is then trained on top of the resulting features. Specifically, we froze the convolutional layers after training, and we optimized a linear classifier on top of the resulting feature extractor. Accordingly, the best model was selected based on its validation performance. One should note that we used five evaluation metrics, namely, top-1 accuracy (Acc@1), top-5 accuracy (Acc@5), mean per-class accuracy, macro-averaged F1-score (Macro-F1), and weighted F1-score (Weighted-F1) to measure the attained performance [38]. For a fair comparison, DeepCluster, DAC-SD, and TSDC were all assessed under identical conditions: the same ResNet-50 backbone, SwAV-based initialization, dataset splits, training configuration, and linear evaluation protocol. All baseline numbers presented in this work stem from our own reimplementations conducted within this unified experimental setup.

Table 2. ResNet-50 backbone architecture, projection head, and prototype layer.

Stage	Layer/Block	No. of Blocks	Kernel/Block Type	Stride	Output Channels
Stem	Conv1 + BN + ReLU	1	7×7 convolution	2	64
Stem	MaxPool	1	3×3 max pooling	2	64
L1 (conv2_x)	Bottleneck $\times$ 3	3	$[1 \times 1, 3 \times 3, 1 \times 1]$	1	256
L2 (conv3_x)	Bottleneck $\times$ 4	4	$[1 \times 1, 3 \times 3, 1 \times 1]$	2	512
L3 (conv4_x)	Bottleneck $\times$ 6	6	$[1 \times 1, 3 \times 3, 1 \times 1]$	2	1024
L4 (conv5_x)	Bottleneck $\times$ 3	3	$[1 \times 1, 3 \times 3, 1 \times 1]$	2	2048
Head	Global Average Pool	1	Adaptive average pooling	—	2048
Projection head	MLP (2 layers)	2	Linear + ReLU + Linear	—	128
Prototype layer	Prototypes (linear)	1	Linear	—	K

4.3. Results and Analysis

In the following, we outline the experiments conducted to investigate the influence of the loss weighting coefficients, data augmentation strategies, and alternative backbone architectures. In addition, TSDC model was compared with DeepCluster [4] and the combination of deep clustering and Knowledge Distillation (KD) [10].

Prior to detailing the individual experiment procedures, we first examined the learning dynamics of the proposed method through the evolution of the total loss across the training epochs. As can be seen in Figure 5, the loss decreases rapidly during the early epoch of training, indicating that the model efficiently captures salient structural information induced by the clustering loss and the distillation losses. Following this initial decrease, the loss shows a gradual, smooth, and stable decline, without noticeable oscillations. During the final training epochs, the loss curve displays a plateau, indicating minimal further reduction in the loss value. Moreover, the progressive stabilization of the loss function suggests that the learned representations attain increasing consistency and robustness over the course of training.

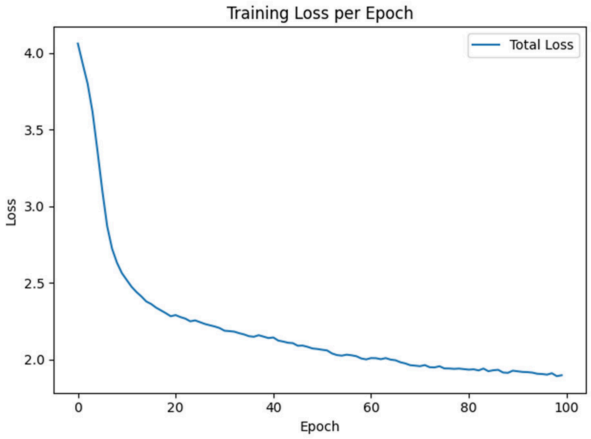


Figure 5. Convergence of training loss Across epochs.

To compare the effectiveness of using only a clustering loss versus combining clustering loss with distillation losses, in order to improve the quality of the learned feature representations. We evaluate these two models on variety of datasets with different levels of visual complexity. The results shown in Table 3 come from two separate runs with random seeds 123 and 456, and are expressed as mean ± standard deviation.

Table 3. Results obtained using the clustering loss only and using the proposed TSDC using multiple datasets.

Dataset	Method	Acc@1	Acc@5	Mean Per-Class Accuracy	Macro-F1	Weighted-F1
MNIST	Clustering Loss only	97.77 ± 0.16	99.96 ± 0.02	97.75 ± 0.15	0.97 ± 0.001	0.97 ± 0.001
MNIST	TSDC	98.20 ± 0.18	99.98 ± 0.02	98.17 ± 0.18	0.98 ± 0.0003	0.98 ± 0.0005
CIFAR-10	Clustering Loss only	77.15 ± 0.94	98.94 ± 0.23	77.15 ± 0.94	0.76 ± 0.012	0.76 ± 0.012
CIFAR-10	TSDC	77.91 ± 0.70	98.93 ± 0.08	77.91 ± 0.70	0.77 ± 0.0037	0.77 ± 0.0037
CIFAR-100	Clustering Loss only	54.77 ±0.07	82.89 ± 0.26	54.77 ± 0.07	0.54 ± 0.0032	0.54 ± 0.0032
CIFAR-100	TSDC	57.58 ± 1.85	84.70 ± 0.51	57.58 ± 1.85	0.57 ± 0.02	0.57± 0.02
TissueMNIST	Clustering Loss only	60.19 ± 0.15	97.36 ± 0.21	47.01 ± 0.02	0.495 ± 0.021	0.595 ± 0.021
TissueMNIST	TSDC	60.71 ± 0.49	97.13 ± 0.01	46.63 ± 0.03	0.495 ± 0.021	0.610 ± 0.028

In Table 3, we compare the performance achieved using the clustering loss only with that of the proposed TSDC across four benchmark datasets, namely MNIST [34], CIFAR-10 [35], CIFAR-100 [35] and TissueMNIST [36]. As it can be seen, the results across all datasets indicate that TSDC consistently yields improvements in terms of Acc@1. In particular, it increased the Acc@1 from 97.77 ± 0.16 to 98.20 ± 0.18% for MNIST dataset. Consistently, TSDC increased the Acc@1 from 77.15 ± 0.94 to 77.91 ± 0.70% for CIFAR-10 benchmark and additionally achieved higher mean per-class accuracy, macro-F1 score, and weighted-F1 score. The most pronounced performance enhancement was observed on CIFAR-100, where the proposed approach enhanced Acc@1 and Acc@5 from 54.77 ± 0.07% to 57.58 ± 1.85%, and from 82.89± 0.26 to 84.70± 0.51%, respectively.

These findings indicate that the TSDC method improved further the discriminative power of the learned feature representation. One should also note that for TissueMNIST, TSDC yielded a marginal improvement in terms of Acc@1 (from 60.19 ± 0.15% to 60.71 ± 0.49%). This observation indicates that, while the proposed method maintains

competitive performance, its improvement on TissueMNIST is comparatively modest compared to the gains recorded on the natural image datasets. In summary, TSDC consistently outperforms the model trained solely using the clustering loss, with performance gains particularly pronounced on more complex datasets, such as CIFAR-100.

Furthermore, a grid search over a range of hyperparameter values was performed to determine the optimal trade-off between the intra-distillation and the inter-distillation losses. Specifically, the intra-distillation weighting coefficient, λ_{intra}, was set to 0.008, 0.0008 and 0.00016, whereas the inter-distillation weighting coefficient, λ_{inter}, was set to 0.0025, 0.00025 and 0.0005. We evaluated the model's accuracy when the intra-distillation loss or the inter-distillation loss is omitted in order to quantify the individual contribution of each component of the proposed loss function.

Table 4 reports the top classifier accuracy of all considered weighting configurations. Overall, the results demonstrate the influence of the parameters λ_{intra} and λ_{inter} on both Acc@1 and Acc@5 performance metrics. As one can see, some joint optimizations of both loss functions yielded superior performance compared to employing either component only. The highest top-1 accuracy of 58.98 was achieved using a λ_{intra} = 0.0008 and λ_{inter} = 0.00025, whereas the highest top-5 accuracy of 85.44 was obtained using λ_{intra} = 0.00016 and λ_{inter} = 0.0025. These findings indicate that moderate weight is more effective than excessive degrees of weight loss.

Table 4. Comparative analysis of model performance under varying intra- and inter-distillation weight configurations.

$\lambda_{intra} \backslash \lambda_{inter}$	Metric	0.0025	0.00025	0.0005	0.0
0.008	Acc@1	56.64	56.92	54.90	56.39
	Acc@5	83.78	84.20	82.85	83.65
0.0008	Acc@1	54.80	58.98	58.09	54.39
	Acc@5	82.66	85.06	85.39	82.61
0.00016	Acc@1	58.40	55.18	57.10	55.96
	Acc@5	85.44	83.41	83.98	83.79
0.0	Acc@1	54.91	55.34	56.10	54.72
	Acc@5	82.59	83.55	83.37	82.70

To further investigate how sensitive the proposed framework is to the distillation weights, we separately tested a range of values for λ_{intra} and λ_{inter}. For each evaluation, the weight of the other distillation component was set to zero, allowing us to isolate and measure its individual impact on performance.

In Table 5, in the intra-distillation setting, the highest Top-1 accuracy of 56.60% is obtained when λ_{intra} is set to 0.004, suggesting that a moderate weight yields the most effective guidance. For inter-distillation, the highest accuracy of 56.89% is obtained at $\lambda_{\text{inter}} = 0.0001$, suggesting that a smaller weight is more suitable for preserving stable feature learning.

Table 5. Independent assessment of intra-distillation and inter-distillation weight settings on CIFAR-100.

λ_{intra}	0.008	0.004	0.002	0.0008	0.0004	0.00016
Intra-only Acc@1	56.39	56.60	55.21	54.39	56.36	55.96
λ_{inter}	0.0025	0.0015	0.001	0.0005	0.00025	0.0001
Inter-only Top-1 Acc@1	54.91	56.40	55.86	56.10	55.34	56.89

The objective of the following experiment is to determine the optimal combination of the stages that most effectively transfer knowledge from the teacher layers to the student layers, thereby enhancing the overall accuracy of the model. In our experimental study, we systematically assessed a range of configurations involving interactions between adjacent layers in order to quantify their impact on the inter-distillation loss. The explored configurations span from a minimal setting that incorporates only two layers to a maximal setting that utilizes the entire set of available layers.

Table 6 specifies the directionality of knowledge transfer in each inter-distillation configuration, wherein the deeper layer functions as the teacher and the immediately preceding, shallower layer functions as the student. As can be seen in Figure 6, the recorded accuracy improved when employing an inter-distillation based on deeper feature representations rather than relying solely on shallower features. The lowest performance was observed when the knowledge distillation was performed between Layer1 and Layer2, yielding 53.13 and 81.70 as Acc@1 and Acc@5, respectively. This result suggests that supervision based solely on early-stage feature representations yields limited benefit. By introducing an additional distillation stage between Layer2 and Layer3, the performance increased to 54.92 and 82.85, for Acc@1 and Acc@5, respectively. This implies the presence of intermediate semantic representations, which in turn provide informative signals that guide the learning process. One should also note that the optimal performance was achieved by incorporating Layer2, Layer3, and Layer4, which resulted in the highest Acc@1 and Acc@5 with of 57.24 and 84.6, respectively. For the configuration employing the combination of Layer1, Layer2, Layer3, and Layer4, the performance remained below the results obtained using best-performing setup, even though this configuration incorporates a deeper layer.

Table 6. Configurations of distillation between the teacher and student layers.

Layer Configuration	Student Layer(s)	Teacher Layer(s)	Knowledge Transfer Direction
Layer1–Layer2	Layer1	Layer2	Layer2 → Layer1
Layer1–Layer2–Layer3	Layer1, Layer2	Layer2, Layer3	Layer2 → Layer1; Layer3 → Layer2
Layer1–Layer2–Layer3–Layer4	Layer1, Layer2, Layer3	Layer2, Layer3, Layer4	Layer2 → Layer1; Layer3 → Layer2; Layer4 → Layer3
Layer2–Layer3	Layer2	Layer3	Layer3 → Layer2
Layer2–Layer3–Layer4	Layer2, Layer3	Layer3, Layer4	Layer3 → Layer2; Layer4 → Layer3
Layer3–Layer4	Layer3	Layer4	Layer4 → Layer3

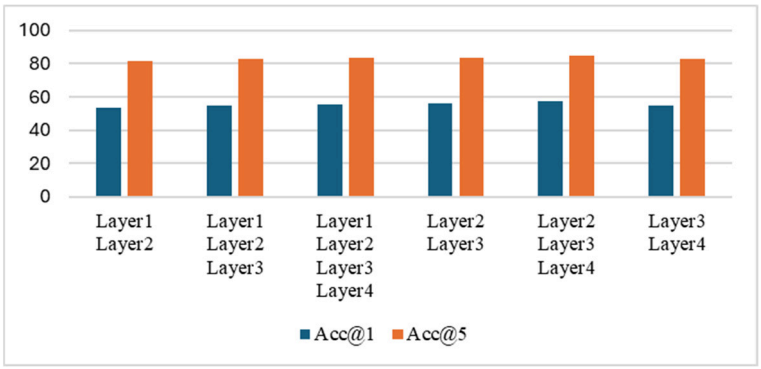


Figure 6. Performances achieved using various layer configurations on the inter-distillation loss.

To systematically examine different configurations of the intra-distillation loss, we consider both single-layer and multi-layer formulations that encompass shallow, interme-

diate, and deep network layers. The objective is to measure the impact of these layers on the overall performance when it contributes to early layers, deeper layers, or combinations of layers.

The results presented in Figure 7 indicate a marked impact on accuracy when intra-distillation is applied under different configurations. The combination of Layer2 and Layer4 yields the highest performance, achieving an Acc@1 of 58.89 and an Acc@5 of 85.06. In addition, one can deduce that applying intra-distillation on an intermediate layer and a deeper layer yields a most effective guidance for the representation learning. Another configuration exhibiting high performance is the combination of Layer1, Layer2, Layer3, and Layer4, which is followed in effectiveness by the combination of Layer1 through Layer3. Overall, the findings prove that selecting complementary yields greater effectiveness than uniformly applying intra-distillation across the entire set of layers.

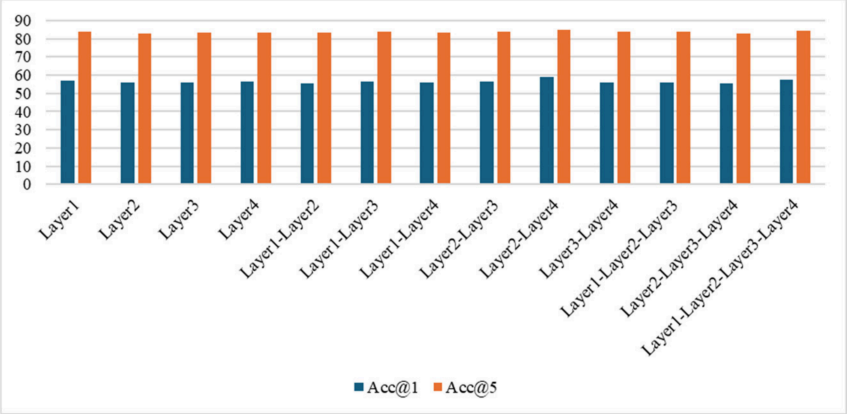


Figure 7. The effect of various layer configurations on the intra-distillation loss.

In the following, we investigate the effects of crop diversity and crop size on the accuracy and feature representations. In particular, we evaluated multiple configurations by varying the number of crops and the minimum crop scale, while keeping the maximum crop scale fixed at 1.

As it can be seen in Table 7, the best configuration corresponds to setting 1, which achieved an Acc@1 of 58.89 and an Acc@5 of 85.06. In addition, configuring the minimum scale to 0.14 in settings 1 and 6 yielded a superior performance compared with configuring the minimum scale to 0.50 in Settings 2 and 7. When comparing settings 1 and 2, as well as settings 5 and 7, there are noticeable Acc@1 improvements of approximately 10.1 and 4.85 units, respectively. Overall, the experimental results show that increasing the number of crops does not necessarily lead to an accuracy improvement.

Table 7. Performance achieved using multi-crop configurations across different crop scale settings.

Setting	No. of Crops	Min Scale	Max Scale	Acc@1	Acc@5
1	1	0.14	1.0	58.89	85.06
2	1	0.50	1.0	48.79	77.68
3	2	0.30	1.0	49.87	78.74
4	2	0.08	1.0	55.98	83.12
5	2	0.50	1.0	52.80	81.02
6	3	0.14	1.0	56.20	83.89
7	3	0.50	1.0	47.95	77.07

The objective of this experiment is to quantitatively characterize the relationship between the number of prototypes and the quality of the learned representations. We

determine the prototype set sizes of 25, 50, 70, 100, and 150 by means of a grid search procedure. In addition, we apply the same procedure to the deep clustering loss.

Table 8 showcases the influences of the number of prototypes on the performance of the model. As it can be seen, the highest accuracy of the proposed TSDC attained 58.89 and 85.06 as Acc@1 and Acc@5, respectively, using 50 prototypes. The second-best performance was obtained when the number of prototypes was set to 100, achieving 58.05 and 84.82as Acc@1 and Acc@5, respectively whereas lower performance was observed for configurations with 25, 70, and 150 prototypes. For the deep clustering settings, the best performance was achieved using a prototypes number set to 100, yielding 58.12 as Acc@1 of and 85.12 as Acc@5. In a comparative analysis between the deep clustering model and the TSDC model, TSDC exhibits a superior performance when the number of prototypes was set to 50. These results demonstrate that TSDC derives the greatest benefit from intermediate prototype configuration. Conversely, for large number of prototypes, a risk of over-partitioning the feature space emerges, and the reliability of the resulting pseudo-labels decreases considerably.

Table 8. Performance results of different prototype values in the TSDC model and Deep clustering model.

Number of Prototypes	TSDC Acc@1	TSDC Acc@5	Deep Clustering Acc@1	Deep Clustering Acc@5
25	56.77	84.14	54.82	82.35
50	58.89	85.06	54.72	82.50
70	55.13	83.32	56.11	83.48
100	58.05	84.82	58.12	85.12
150	56.20	83.90	55.24	83.07

DeepCluster was chosen as the main baseline since it provides the core clustering framework on which TSDC is built, while DAC-SD was included as a closely related, single-backbone approach that integrates deep clustering with self-distillation. Furthermore, we compared TSDC with the Domain-Agnostic Clustering with Self-Distillation (DAC-SD) [10] approach. Specifically, as per the setup detailed in [10], we employed a ResNet50 backbone and extended it with additional output heads that are jointly optimized using all corresponding loss functions (cross-entropy loss with pseudo-labels, KL-divergence distillation loss, and feature hint loss).

The results reported in Table 9 indicate that TSDC consistently and substantially outperforms DAC-SD [10] across all considered datasets and performance measures. In particular, for CIFAR-10 dataset, Acc@1 increased from 65.87% to 78.40% while Acc@5 attained 98.98%. This confirms TSDC ability to learn more discriminative feature representations for natural image datasets, in which intra-class variability and visual complexity. Besides, the most substantial performance improvement was observed on CIFAR-100, where TSDC increased Acc@1 and Acc@5 from 35.58% to 58.89%, and from 67.80% to 85.06%, respectively. This substantial performance gain demonstrates that TSDC distillation losses are essential for learning more discriminative and robust feature representations.

The objective of the following experiment is to investigate the impact of different ResNet-50 backbones on TSDC performance. Specifically, we used four ResNet-50 configurations that share identical network depth and architectural design, while systematically increasing the number of channels in each layer. Table 10 specifies the configuration adopted for each considered model, including the number of channels in each layer, as well as the total number of learnable parameters. It is important to note that all models were initialized using the SwAV pretrained weights obtained from 400 epochs of training [39].

Table 9. Comparison with DAC-SD [10] across multiple datasets.

Dataset	Method	Acc@1	Acc@5	Mean Per-Class Accuracy	Macro-F1	Weighted-F1
MNIST	DAC-SD [10]	95.94	99.89	95.90	0.96	0.96
MNIST	TSDC	98.33	99.96	98.30	0.98	0.98
CIFAR-10	DAC-SD [10]	65.87	96.72	65.87	0.66	0.66
CIFAR-10	TSDC	78.40	98.98	78.40	0.78	0.783
CIFAR-100	DAC-SD [10]	35.58	67.80	35.58	0.34	0.34
CIFAR-100	TSDC	58.89	85.06	58.89	0.587	0.587

Table 10. ResNet-50 model variants used TSDC backbones.

Model	Conv1	Layer1 Output	Layer2 Output	Layer3 Output	Layer4 Output	No. of Parameters (Millions)
ResNet-50	64	256	512	1024	2048	23.51
ResNet-50w2	128	512	1024	2048	4096	93.91
ResNet-50w4	256	1024	2048	4096	8192	375.38
ResNet-50w5	320	1280	2560	5120	10,240	586.45

Figure 8 makes it clear; as the number of channels increases, TSDC accuracy rises. Both accuracy metrics, Acc@1 and Acc@5, exhibit a steady increase as the model width increases. As it can be seen, the ResNet-50w5 configuration yielded the highest accuracy, with 66.09 and 89.76 for Acc@1 and Acc@5, in comparison to the baseline ResNet-50, which attained only 52.92 as Acc@1 and 81.58 as Acc@5. We also observed an improvement of approximately 13.17% and 8.18% in Acc@1 and Acc@5, respectively. This indicates that increasing the number of channels in the backbone yields more discriminative feature representations under such experimental settings.

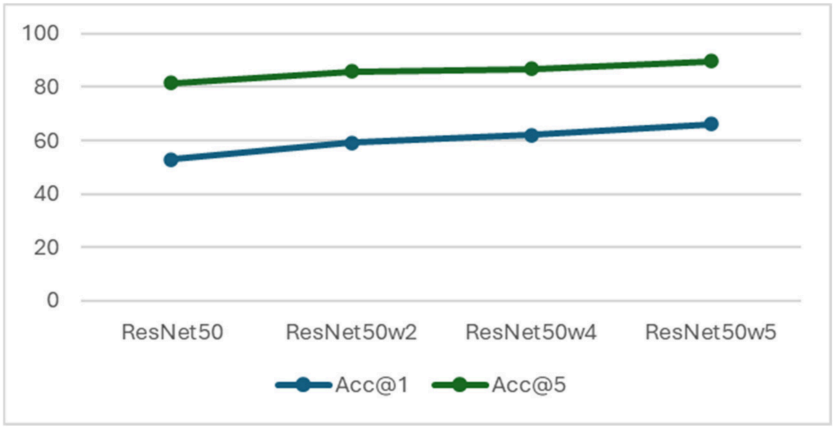


Figure 8. Relationship between classification accuracy and the ResNet-50 architectural.

5. Discussion

The experimental results indicate that incorporating both intra- and inter-distillation into the deep clustering model leads to enhanced feature representations. Compared with the clustering loss, the proposed TSDC loss yields performance improvements across all evaluated datasets, with particularly pronounced gains on CIFAR-10 and CIFAR-100. This provides evidence that the proposed distillation objectives yield effective regularization signals that are complementary to the pseudo-label-based clustering loss.

The performance enhancement is particularly pronounced on the CIFAR-100 benchmark, where the application of TSDC increases Acc@1 from 54.77 $\pm$ 0.07 to 57.58 $\pm$ 1.85 and Acc@5 from 82.89 $\pm$ 0.26 to 84.70 $\pm$ 0.51. By transferring information from high- to low-response channels and enabling shallower layers to exploit the richer semantic representations learned by deeper layers, TSDC facilitates extracting more informative feature representations. Furthermore, the comparative analysis with DAC-SD provides additional evidence supporting the efficacy of the proposed approach. TSDC demonstrated superior performance compared to DAC-SD across all evaluated datasets, with an especially pronounced improvement observed on CIFAR-100.

However, the performance gains observed on TissueMNIST were relatively modest compared with those achieved on the natural image datasets. The reason for that is the visual characteristics of biomedical images; for example, grayscale texture patterns may call for more specialized preprocessing techniques or augmentation methods tailored to that specific domain.

Future research may further develop this framework by integrating adaptive loss-weighting strategies, and comprehensive evaluations on domain-specific image datasets.

6. Conclusions

Most deep clustering approaches still face challenges in learning sufficiently expressive feature representations. This has motivated the adoption of Knowledge Distillation (KD) as a promising alternative that typically trains a student model to reproduce internal representations and predictions of a larger and more complex teacher model. This research introduces a novel KD-based Teacher–Student-based Deep Clustering (TSDC) approach that does not require an additional teacher network; instead, it leverages the inherent structure of the network itself. Specifically, the most high-response channels are exploited to provide guidance to the low-response channels. Accordingly, the more representative layers serve as a teacher for other network layers. Specifically, the intra-distillation encourages low-response channels to approximate the representational behaviour of the high-response ones, whereas the inter-distillation facilitates the transfer of semantic information from the deepest, high-level feature representations to the earlier network layers. Moreover, TSDC reformalizes the deep clustering problem through the design of a new objective function that includes both intra- and inter-distillation terms. This allows for optimizing the model parameters based on losses from deep clustering and intra- and inter-distillation.

The proposed approach was assessed using benchmark datasets, namely, MNIST [34], CIFAR-10 [35], CIFAR-100 [35] and TissueMNIST [36]. TSDC demonstrated superior performance compared to the clustering loss only on MNIST, CIFAR-10 and CIFAR-100, with the most pronounced performance gain observed at CIFAR-100. This showcased TSDC ability to learn more expressive representations. We also conducted supplementary experiments aimed at identifying the optimal layer configurations for each type of distillation loss. Experimental findings show that TSDC consistently increases Acc@1 across all evaluated datasets relative to the clustering loss only. The most pronounced improvements are observed on CIFAR-10 and CIFAR-100, where TSDC enhances all evaluation metrics, raising Acc@1 from 77.15 $\pm$ 0.94% to 77.91 $\pm$ 0.70% on CIFAR-10 and from 54.77 $\pm$ 0.07 to 57.58 $\pm$ 1.85 on CIFAR-100. These results suggest that the proposed intra- and inter-distillation objectives are particularly effective for learning more discriminative feature representations on visually complex datasets.

Nevertheless, this study has several limitations. First, the L2-norm-based channel ranking strategy is heuristic in nature and captures response magnitude rather than the semantic importance of each channel. Second, the proposed rank-based pairing scheme does not explicitly ensure that the matched teacher and student channels are semantically

aligned. Third, we consider only a limited set of carefully controlled baseline methods. Finally, the performance gain on TissueMNIST is relatively small compared with the improvements achieved on the natural-image datasets. Future work will explore adaptive or learnable channel-matching mechanisms, more extensive controlled comparisons, and additional evaluations on domain-specific image datasets.

Author Contributions: Conceptualization, R.A., M.M.B.I. and O.B.; methodology, R.A., M.M.B.I. and O.B.; software, R.A.; validation, R.A., M.M.B.I. and O.B.; formal analysis, R.A.; investigation, R.A., M.M.B.I. and O.B.; resources, R.A., M.M.B.I. and O.B.; data curation, R.A.; writing—original draft preparation, R.A.; writing—review and editing, R.A., M.M.B.I. and O.B.; visualization, R.A.; supervision, M.M.B.I. and O.B.; project administration, M.M.B.I. and O.B.; funding acquisition, R.A., M.M.B.I. and O.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research project is supported by a grant from King Abdelaziz City for Science and Technology (KACST), represented by the Science Technology and Innovation Unit at King Saud University (STU-KSU), Riyadh, Saudi Arabia (Grant No.STU-32).

Data Availability Statement: The datasets utilized in this research are publicly available. MNIST is available at [http://yann.lecun.com/exdb/mnist/] (accessed on 7 January 2026). CIFAR-10 and CIFAR-100 are available at [https://www.cs.toronto.edu/~kriz/cifar.html] (accessed on 7 January 2026). TissueMNIST is available through the MedMNIST collection at [https://medmnist.com/] (accessed on 7 January 2026). The datasets were prepared using Python packages (the torchvision and medmnist).

Acknowledgments: The authors express their sincere gratitude to King Abdelaziz City for Science and Technology (KACST), represented by the Science Technology and Innovation Unit at King Saud University (STU-KSU) for their valuable support.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Acc@1	Top-1 accuracy
Acc@5	Top-5 accuracy
AI	Artificial Intelligence
BN	Batch Normalization
CE	Cross-Entropy
CNN	Convolutional Neural Network
DAC-SD	Domain-Agnostic Clustering with Self-Distillation
DBC	Discriminatively Boosted Clustering
DEC	Deep Embedded Clustering
F1	F1-score
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
KD	Knowledge Distillation
KL	Kullback-Leibler
Macro-F1	Macro-averaged F1-score
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean-Squared Error
ReLU	Rectified Linear Unit
ResNet	Residual Network
RGB	Red, Green, and Blue
SCAN	Semantic Clustering by Adopting Nearest Neighbours

SeCu	Stable Cluster Discrimination
SGD	Stochastic Gradient Descent
SwAV	Swapping Assignments between Views
TSDC	Teacher–Student-based Deep Clustering
VAE	Variational Autoencoder
Weighted-F1	Weighted F1-score

References

1. Mitchell, T.M. *Machine Learning*; McGraw-Hill: New York, NY, USA, 1997.
2. Xu, D.; Tian, Y. A comprehensive survey of clustering algorithms. *Ann. Data Sci.* **2015**, *2*, 165–193. [CrossRef]
3. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2323. [CrossRef]
4. Caron, M.; Bojanowski, P.; Joulin, A.; Douze, M. Deep clustering for unsupervised learning of visual features. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 139–156. [CrossRef]
5. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 818–833.
6. Li, L. Self-regulated feature learning via teacher-free feature distillation. In *Computer Vision—ECCV 2022*; Springer: Cham, Switzerland, 2022; pp. 347–363. [CrossRef]
7. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531. [CrossRef]
8. Zhang, L.; Song, J.; Gao, A.; Chen, J.; Bao, C.; Ma, K. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3712–3721. [CrossRef]
9. Mobahi, H.; Farajtabar, M.; Bartlett, P.L. Self-distillation amplifies regularization in Hilbert space. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 3351–3361.
10. Adnan, M.; Ioannou, Y.A.; Tsai, C.-Y.; Taylor, G.W. Domain-agnostic clustering with self-distillation. *arXiv* **2021**, arXiv:2111.12170. [CrossRef]
11. Ren, Y.; Pu, J.; Yang, Z.; Xu, J.; Li, G.; Pu, X.; Yu, P.S.; He, L. Deep clustering: A comprehensive survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *36*, 5858–5878. [CrossRef] [PubMed]
12. Min, E.; Guo, X.; Liu, Q.; Zhang, G.; Cui, J.; Long, J. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access* **2018**, *6*, 39501–39514. [CrossRef]
13. Zhou, S.; Xu, H.; Zheng, Z.; Chen, J.; Li, Z.; Bu, J.; Wu, J.; Wang, X.; Zhu, W.; Ester, M. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. *arXiv* **2022**, arXiv:2206.07579. [CrossRef]
14. Coates, A.; Ng, A.Y. Learning feature representations with K-means. In *Neural Networks: Tricks of the Trade*, 2nd ed.; Montavon, G., Orr, G.B., Müller, K.-R., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 561–580.
15. Xie, J.; Girshick, R.; Farhadi, A. Unsupervised deep embedding for clustering analysis. In Proceedings of the 33rd International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016; pp. 740–749.
16. Li, F.; Qiao, H.; Zhang, B. Discriminatively boosted image clustering with fully convolutional auto-encoders. *Pattern Recognit.* **2018**, *83*, 161–173. [CrossRef]
17. Murtagh, F.; Contreras, P. Algorithms for hierarchical clustering: An overview. *WIREs Data Min. Knowl. Discov.* **2012**, *2*, 86–97. [CrossRef]
18. Yang, J.; Parikh, D.; Batra, D. Joint unsupervised learning of deep representations and image clusters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 5147–5156. [CrossRef]
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778. [CrossRef]
20. Zheng, H.; Zhang, R.; Wang, H. Deep image clustering based on curriculum learning and density information. In Proceedings of the 2024 International Conference on Multimedia Retrieval, Phuket, Thailand, 10–14 June 2024; pp. 330–338.
21. Van Gansbeke, W.; Vandenhende, S.; Georgoulis, S.; Proesmans, M.; Van Gool, L. SCAN: Learning to classify images without labels. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 268–285. [CrossRef]
22. Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9912–9924.
23. Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2292–2300.

24. Qian, Q. Stable cluster discrimination for deep clustering. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 16645–16654.
25. Huang, H.; Zhou, G.; Zheng, Y.; Qiu, Y.; Wang, A.; Zhao, Q. Adversarially robust deep multi-view clustering: A novel attack and defense framework. In Proceedings of the 41st International Conference on Machine Learning (ICML), Vienna, Austria, 21–27 July 2024; Volume 235, pp. 20526–20558.
26. Chen, W.C.; Chu, W.T. SSSD: Self-supervised self distillation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–7 January 2023; pp. 2769–2776. [CrossRef]
27. Lin, Z.; Zhou, G.; Huang, H.; Huang, Z.; Zhao, Q.; Xie, S. Redundancy-aware external signal learning for image clustering. *IEEE Trans. Syst. Man Cybern. Syst.* **2026**, *56*, 4984–4997. [CrossRef]
28. Wang, J.; Xu, Z.; Wang, X.; Li, T. Toward generalized multistage clustering: Multiview self-distillation. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *36*, 10311–10324. [CrossRef] [PubMed]
29. Zhang, H.; Huang, D. Deep clustering with diffused sampling and hardness-aware self-distillation. *arXiv* **2024**, arXiv:2401.14038. [CrossRef]
30. Sadeghi, M.; Wang, Z.; Armanfard, N. Forward-backward knowledge distillation for continual clustering. *arXiv* **2024**, arXiv:2405.19234. [CrossRef]
31. Dong, Y.; Miller, K.; Lei, Q.; Ward, R. Cluster-aware semi-supervised learning: Relational knowledge distillation provably learns clustering. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 40799–40831. [CrossRef]
32. Liu, A.H.; Chang, H.-J.; Auli, M.; Hsu, W.-N.; Glass, J. DinoSR: Self-distillation and online clustering for self-supervised speech representation learning. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 58346–58362. [CrossRef]
33. Cai, D.; Cai, Z.; Li, M. Self-supervised reflective learning through self-distillation and online clustering for speaker representation learning. *arXiv* **2024**, arXiv:2401.01473. [CrossRef]
34. Deng, L. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
35. Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*; Technical Report; University of Toronto: Toronto, ON, Canada, 2009.
36. Yang, J.; Shi, R.; Wei, D.; Liu, Z.; Zhao, L.; Ke, B.; Pfister, H.; Ni, B. MedMNIST v2—A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Sci. Data* **2023**, *10*, 41. [CrossRef] [PubMed]
37. Robbins, H.; Monro, S. A stochastic approximation method. *Ann. Math. Stat.* **1951**, *22*, 400–407. [CrossRef]
38. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
39. Facebook Research. SwAV: PyTorch Implementation of SwAV and DeepCluster-v2. Available online: https://github.com/facebookresearch/swav (accessed on 7 January 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.