SELF-SUPERVISED LEARNING FOR UNSUPERVISED IMAGE CLASSIFICATION AND SUPERVISED LOCALIZATION TASKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MELIH BAYDAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

JULY 2024

Approval of the thesis:

SELF-SUPERVISED LEARNING FOR UNSUPERVISED IMAGE CLASSIFICATION AND SUPERVISED LOCALIZATION TASKS

submitted by **MELIH BAYDAR** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Assoc. Prof. Dr. Emre Akbaş
Supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Sinan Kalkan
Computer Engineering, METU _____

Assoc. Prof. Dr. Emre Akbaş
Computer Engineering, METU _____

Prof. Dr. Nazlı İkizler Cinbiş
Computer Engineering, Hacettepe University _____

Prof. Dr. Ahmet Burak Can
Computer Engineering, Hacettepe University _____

Prof. Dr. Pınar Karagöz
Computer Engineering, METU _____

Date:26.07.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Melih Baydar

Signature :

ABSTRACT

SELF-SUPERVISED LEARNING FOR UNSUPERVISED IMAGE CLASSIFICATION AND SUPERVISED LOCALIZATION TASKS

Baydar, Melih
Ph.D., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Emre Akbaş

July 2024, 86 pages

Recent self-supervised learning methods, where instance discrimination is a fundamental pretraining task for convolutional neural networks (CNNs), excel in transfer learning. While instance discrimination is effective for classification due to its image-level learning, it lacks dense representation learning, making it sub-optimal for localization tasks like object detection. In the first part of this thesis, we aim to mitigate this shortcoming of instance discrimination task by extending it to learn dense representations alongside image-level representations. By adding a segmentation branch parallel to image-level learning to predict class-agnostic masks, we enhance the location-awareness of the representations. Our approach improves performance in localization tasks, achieving up to 1.7% AP improvement on PASCAL VOC, 0.8% AP on COCO object detection, 0.8% AP on COCO instance segmentation, and 3.6% mIoU on PASCAL VOC semantic segmentation.

In recent years, Vision Transformers (ViTs) have significantly advanced deep learning models, boosting performance in traditional computer vision tasks and driving substantial progress in self-supervised learning. In the second part of this thesis, we also proposes UCLS, an unsupervised image classification framework leveraging the

improved feature representation and superior nearest neighbor performance of self-supervised ViTs. We incrementally enhance baseline methods for unsupervised image classification and further propose the use of a cluster ensembling methodology and a self-training step to optimize the utilization of multi-head classifiers. Extensive experimentation demonstrates that UCLS achieves state-of-the-art performance on ten image classification benchmarks in fully unsupervised settings, with 99.3% clustering accuracy on CIFAR10, 89% on CIFAR100, and surpassing 70% on ImageNet in an unsupervised context.

Keywords: Self-Supervised Learning, Contrastive Learning, Non-Contrastive Learning, Instance Discrimination Task, Semantic Segmentation, Object Detection, Unsupervised Image Classification, Deep Clustering

ÖZ

DENETİMSİZ GÖRÜNTÜ SINIFLANDIRMA VE DENETİMLİ YER SAPTAMA GÖREVLERİ İÇİN ÖZ-DENETİMLİ ÖĞRENME

Baydar, Melih
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Emre Akbaş

Temmuz 2024 , 86 sayfa

Son yıllarda, örnek ayrıştırma görevinin evrişimli sinir ağları (CNNler) için temel bir ön eğitim görevi olduğu öz-denetimli öğrenme yöntemleri, öğrenme aktarmasında büyük başarı elde etti. Örnek ayrıştırma görevi, görüntü düzeyinde öğrenme nedeniyle sınıflandırma için etkili olsa da, yoğun öznitelik öğreniminden yoksundur ve bu durum, nesne tespiti gibi yerelleştirme görevleri için tam uygun olmamasına yol açar. Bu tezin ilk bölümünde, örnek ayrıştırma görevinin bu eksikliğini gidermek amacıyla, onu görüntü düzeyindeki özniteliklerin yanında aynı zamanda yoğun öznitelikler öğrenmeye de teşvik etmeyi hedefliyoruz. Görüntü düzeyinde öğrenmeye paralel olarak sınıf bağımsız maskeler tahmin eden bir bölütleme dalı ekleyerek, özniteliklerin lokasyon farkındalığını artırıyoruz. Yaklaşımımız, PASCAL VOC'de %1.7 AP, COCO nesne tespitinde %0.8 AP, COCO örnek bölütlemede %0.8 AP ve PASCAL VOC anlamsal segmentasyonda %3.6 mIoU'ya kadar performans iyileşmeleri sağlıyor.

Son yıllarda, Vision Transformerlar (ViTs), geleneksel bilgisayarlı görü görevlerinde derin öğrenme modellerini önemli ölçüde ilerleterek, öz denetimli öğrenmede de büyük ilerlemelere yol açtı. Bu tezin ikinci bölümünde, iyileştirilmiş özellik temsili ve

öz-denetimli ViT'lerin üstün en yakın komşu performansından yararlanan bir denetimsiz görüntü sınıflandırma yöntemi olan UCLS'yi öneriyoruz. Denetimsiz görüntü sınıflandırma için temel yöntemleri kademeli olarak geliştiriyor ve çok başlı sınıflandırıcıların kullanımını optimize etmek amacıyla bir kümeleme toplama metodolojisinden yararlanmayı ve bir kendi kendine eğitim adımı kullanmayı öneriyoruz. Yoğun deneyler ile, UCLS'nin tamamen denetimsiz ayarlarda, CIFAR10'da %99.3, CIFAR100'de %89 ve ImageNet'te %70'in üzerinde denetimsiz performansa ulaştığını gösteriyor ve bu bağlamda en son teknolojiye uygun performans sağladığını belirtiyoruz.

Anahtar Kelimeler: Öz-Denetimli Öğrenme, Karşılaştırmalı Öğrenme, Karşılaştırmalı Olmayan Öğrenme, Örnek Ayrıştırma Görevi, Anlamsal Bölümleme, Nesne Tanıma, Denetimsiz Görüntü Sınıflandırma, Derin Öbekleme

This thesis is dedicated to my beloved family, whose unwavering love, support, and encouragement have been the foundation of all my achievements.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Emre Akbaş for his unwavering support, guidance, and encouragement throughout the course of this research. His expertise and insightful feedback have been invaluable, and I am truly grateful for the opportunity to conduct research alongside him.

I would also like to extend my thanks to the members of my thesis committee, Prof. Dr. Sinan Kalkan, and Prof. Dr. Nazlı İkizler Cinbiş, for their time, effort, and constructive suggestions. Your input has greatly enhanced the quality of this work.

I would like to thank my thesis defense jury members, Prof. Dr. Pınar Karagöz and Prof. Dr. Ahmet Burak Can, for reviewing my thesis and their valuable suggestions.

I would like to thank all my friends and colleagues who have supported me in various ways throughout this journey. Your kindness and companionship have made this experience all the more fun and rewarding.

Last but not least, I would like to have a special thanks to my family, whose love and support have been my driving force throughout this journey. To my parents and my brother, thank you for always believing in me and for the efforts you have made to help me reach this point. To my wife, Gökçe, I can not thank enough for your unconditional support, constant encouragement and all the sacrifices you have made to help me achieve my goals. I may not have completed this thesis without your committed belief in me and your endless patience. To my dearest son, you bring immense joy to our family with your presence and your wonderful sense of humor. I cherish every moment spent with you and hope that you find happiness and fulfillment in every decision you make.

The numerical calculations reported in this thesis were mainly performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

Acc	Accuracy
AP	Average Precision
ARI	Adjusted Rand Index
CNN	Convolutional Neural Network
JVCI	Journal of Visual Communication and Image Representation
k-NN / KNN	k-Nearest Neighbors
mIoU	mean Intersection over Union
NMI	Normalized Mutual Information
NNs	Nearest Neighbors
SSL	Self Supervised Learning
SOTA	State-of-the-art
ViT	Vision Transformer

CHAPTER 1

INTRODUCTION

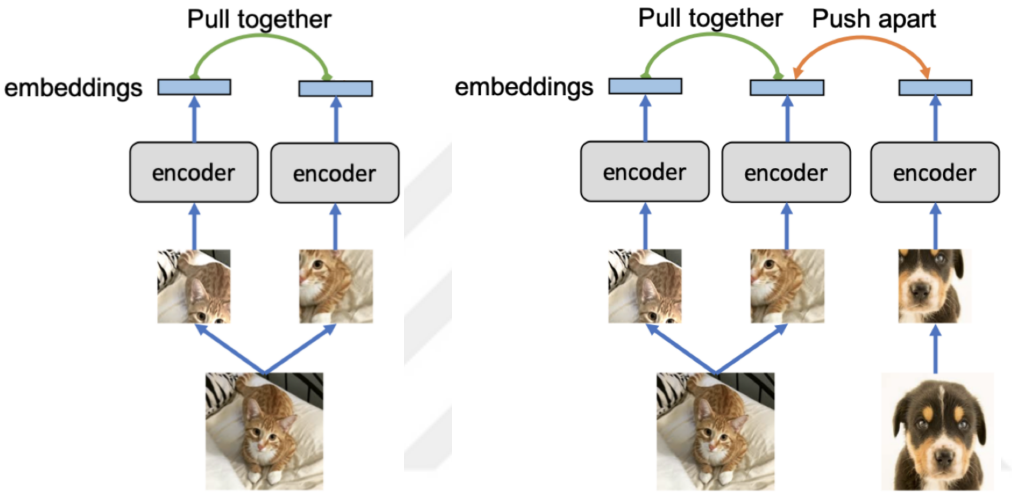
This chapter is adopted from our JVCI journal paper [6] and extended by our recent work.

Deep learning has advanced remarkably in recent years, driven by advancements in algorithms, computational power and data availability, where models are trained to learn representations from large datasets. These representations capture essential features and patterns within the data, enabling models to perform tasks with high accuracy. The prevailing paradigm to train deep models has been supervised learning for years, which relies on vast amounts of human-labeled data to train such models. However, a major limitation of supervised learning is its reliance on labeled data, which is often labor-intensive and expensive to obtain. To overcome this limitation, self-supervised learning (SSL) has emerged as a viable alternative. This approach enables models to learn representations from unlabeled data by creating learning signals through specially designed tasks, called pretext tasks, that achieves to create supervision from the data itself, thus reducing the dependency on human-labeled data.

In the field of computer vision, SSL has made significant strides in generating image representations through the use of convolutional neural networks (CNNs) and vision transformers (ViTs), effectively eliminating the dependence on human-provided labels. The learned representations can be utilized in various downstream tasks, including but not limited to image classification, object detection, image segmentation and depth estimation.

1.1 Problem Definition and Proposed Methods

The predominant pretext task to learn image representations using CNNs, known as the instance discrimination task, endeavors to map feature embeddings at the image level from two distinct views of the same image—generated through different transformations—to closely positioned points in the embedding space. This is achieved through the utilization of a similarity-based loss function, either in a contrastive [12, 13, 38, 59, 73, 84] or non-contrastive [5, 11, 14, 35, 72] manner. Figure 1.1 depicts the outline of these two learning schemes.



(a) Non-Contrastive Learning (b) Contrastive Learning

Figure 1.1: **Outline of Non-Contrastive and Contrastive Learning.** In (a) Non-Contrastive Learning, the model learns useful representations of the data by maximizing agreement between different augmented views of the same data point, without directly comparing with other data points. In (b) Contrastive Learning, the model learns by comparing different data points, pulling similar (positive) pairs together and pushing dissimilar (negative) pairs apart, thereby enhancing the discriminative power of the representations. Image is adopted from [48].

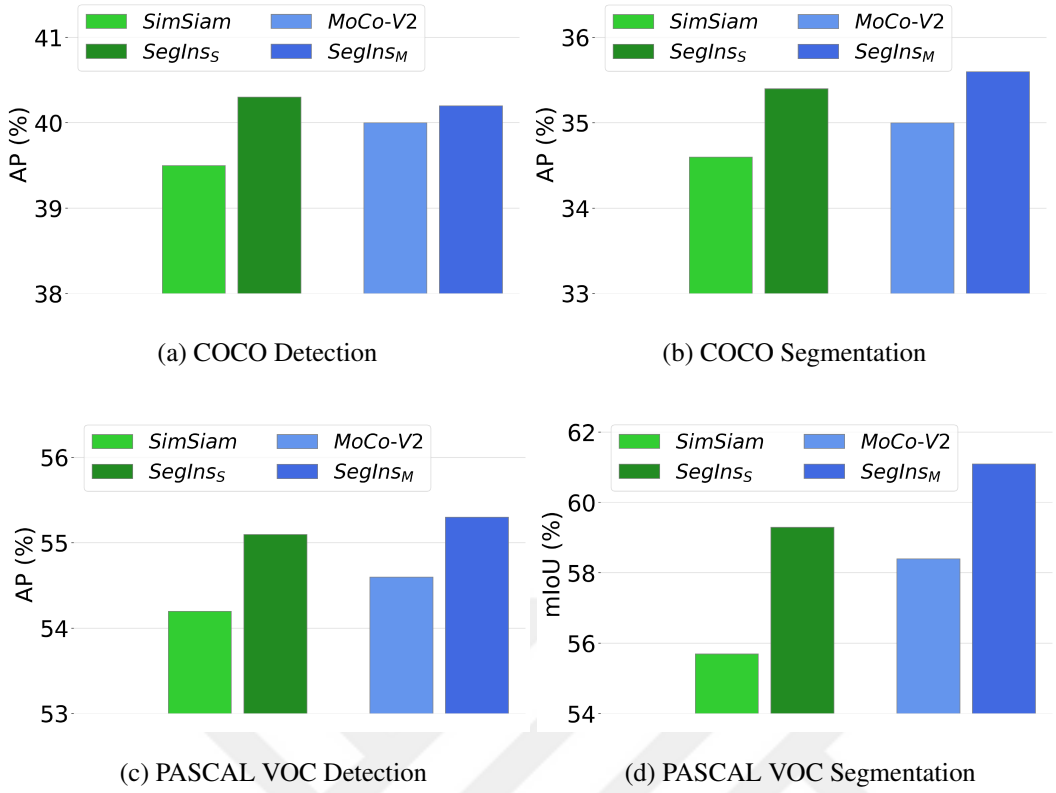
Although this methodology has narrowed the gap in representation quality between fully supervised and self-supervised learning for image classification, a disparity persists for dense prediction (localization) tasks such as object detection and semantic segmentation. This discrepancy arises predominantly from the image-level learning

of CNNs employing the instance discrimination task, wherein spatial information is lost post the global average pooling operation. Several approaches have been proposed for dense self-supervised learning of feature representations [41, 46, 75, 78, 83, 85]. However, while these approaches enhance performance in localization tasks, they often compromise image-level representation quality, resulting in degraded performance in downstream classification tasks.

In the first part of this thesis, we address the deficiency in location-aware representations in CNNs learned through the instance discrimination task. We propose to simultaneously learn dense feature representations alongside image-level representations to improve the localization capability of trained models. Unlike dense self-supervised learning methods, our approach preserves image-level representation quality. Specifically, we extend the instance discrimination learning framework with a class-agnostic segmentation task by appending a segmentation branch to the deep backbone network, working in parallel with image-level learning. The objective of this branch is to predict a binary mask that segregates foreground (object) regions from background regions. In order to provide mask supervision to the segmentation branch, we leverage ContraCAM [71], an extension of GradCAM [66], for binary mask generation within the self-supervised learning framework; however, any unsupervised segmentation method can be employed to retain the self-supervised learning paradigm. We argue that the segmentation branch provides the missing dense learning signal to the backbone, enhancing the location-awareness of the learned representations. Our method is applicable to both non-contrastive and contrastive learning methods.

For our baselines, we select SimSiam [14] and MoCo-V2 [13], building upon these methods to validate our approach in both non-contrastive and contrastive learning scenarios. We demonstrate the effectiveness of our approach by transferring the learned representations to object detection and semantic segmentation downstream tasks, achieving absolute improvements over baseline methods—up to 1.7% AP on *PASCAL VOC object detection*, 0.8% AP on *COCO object detection*, 0.8% AP on *COCO instance segmentation*, and 3.6% mIoU on *PASCAL VOC semantic segmentation*, respectively. Figure 1.2 visually represents these improvements.

In recent years, the emergence of Vision Transformers (ViTs) [26] has revolution-



(a) COCO Detection

(b) COCO Segmentation

(c) PASCAL VOC Detection

(d) PASCAL VOC Segmentation

Figure 1.2: **Transfer Learning Performance Comparison.** Comparison of transfer learning performance of self-supervised pretrained models to object detection and segmentation downstream tasks on COCO and PASCAL VOC datasets. All models are pretrained on ImageNet-100 dataset. "SegIns" denote our method, which improves the baseline performance in all four tasks.

ized the landscape of computer vision as well as language modeling. By capitalizing on the self-attention mechanism, ViTs have demonstrated superior capabilities in capturing long-range dependencies and contextual information in images, surpassing traditional CNNs in several benchmarks. The advancements with ViTs have not only boosted performance in traditional computer vision tasks but have also driven significant progress in self-supervised learning methodologies.

Vision Transformers offer several advantages over Convolutional Neural Networks in the context of self-supervised learning. First and foremost, ViTs excel in modeling global relationships within images due to their self-attention mechanism, which contrasts with the local receptive fields of CNNs. This nature of ViTs eliminate the

aforementioned drawbacks of CNNs and paves the way for both image-level learning and improved localization capabilities. DINO [11] —an approach leveraging self-distillation with no labels— leads the way with significant improvements in representation quality with the ViTs, providing superior nearest neighbor classifier (k-NN) performance and object localization ability in the self-attention modules over CNNs.

Subsequent research on self-supervised (ViTs) [91, 86, 60] has significantly enhanced the quality of feature representations, resulting in notable improvements in core computer vision tasks such as object detection, image segmentation, and image classification, all in an unsupervised manner. CutLER [76] leveraged ViT features to iteratively generate foreground object masks. These object masks were then utilized as pseudo-labels to train an unsupervised class-agnostic object detection and instance segmentation model, achieving state-of-the-art performance. U2Seg [56] advanced CutLER's approach to class-aware object detection and instance segmentation and applied a similar pseudo-labeling pipeline to tackle the unsupervised panoptic segmentation problem. TEMI [1] introduced a method by training a multi-head classifier with a novel nearest neighbor-based loss function using pretrained ViTs, setting new benchmarks in unsupervised image classification/clustering.

Our contributions in the first part of this thesis focus on enhancing self-supervised learning with Convolutional Neural Networks (CNNs). However, with the emergence of Vision Transformers in the field of self-supervised learning, we shift our focus to leveraging these features for unsupervised downstream tasks, specifically unsupervised image classification.

Unsupervised image classification involves grouping unlabeled images into clusters without prior knowledge of their semantic labels. In the second part of this thesis, we address this problem and introduce UCLS, an unsupervised image classification framework that achieves state-of-the-art performance across ten image classification benchmarks. Our primary motivations are the untapped potential of enhanced k-nearest neighbor performance in the latest self-supervised Vision Transformers, and the under-utilization of multi-head classifiers in unsupervised classification.

As a first phase, we select TEMI [1] as our baseline method and propose an improved unsupervised multi-head classifier training to better utilize both the nearest neighbor

of images and the capacity of pretrained models. First, we conduct a hyperparameter search to achieve a slower and improved learning. Next, we propose a per-image adaptive nearest neighbor selection using a distance threshold to encompass a broader support set for each image. Additionally, we enhance the quality of teacher outputs to provide a better learning signal to the student network and propose a modification to the loss function based on empirical observations. In the second phase, we leverage the differences in predictions from various heads of the multi-head classifiers and propose using a cluster ensembling method to consolidate all predictions into combined class predictions for the images. In the third and final phase, we employ a self-training approach to train a classifier on pseudo-labels generated by the cluster ensembling phase, resulting in a classifier ready for inference. UCLS achieves a fully unsupervised image classification accuracy of 88.98% on CIFAR100 and 71.5% on ImageNet datasets. To the best of our knowledge, we are the first to pass 70% accuracy on ImageNet dataset for the fully unsupervised image classification task.

1.2 Contributions and Novelties

Our contributions can be summarized as follows:

- We introduce SegIns, an enhancement to the instance discrimination pretext task that improves the localization capability of learned representations while preserving image-level representation quality in convolutional neural networks.
- We investigate the potential trade-off between image-level and dense representation learning, demonstrating that when combined effectively, both aspects can boost each other's performance.
- SegIns enhances transfer learning performance of models trained the instance discrimination task on object detection (+1.7% AP), instance segmentation (+0.8% AP), and semantic segmentation (+3.6% mIoU) downstream tasks while maintaining performance on the image classification task under a linear evaluation protocol.
- We introduce UCLS, an unsupervised image classification framework to achieve state-of-the-art performance on several benchmarks, surpassing 70% accuracy

on ImageNet, 99% accuracy on CIFAR-10 and 88% accuracy on CIFAR-100 datasets for the first time in fully unsupervised settings.

- We propose an improved unsupervised classification loss which that better leverages the nearest neighbors capabilities of recent self supervised Vision Transformers (ViTs) which can achieve performance comparable to supervised image classification on several benchmarks.
- We investigate the contribution of each component in our unsupervised image classification framework through extensive ablation analysis.

1.3 The Outline of the Thesis

This thesis is organized as follows: In Chapter 2, we review previous works related to our methods, focusing on self-supervised learning with Convolutional Neural Networks (CNNs) and unsupervised image classification using self-supervised Vision Transformers (ViTs). In Chapter 3, we describe our proposed method, SegIns, and its improvements to the instance discrimination task with CNNs for better localization learning. In Chapter 4, we demonstrate the effectiveness of our proposed method through transfer learning experiments on object detection and semantic/instance segmentation tasks, and provide ablation studies to better understand the impact of different components. In Chapter 5, we introduce our proposed method in unsupervised image classification, including the necessary background information to understand our work better, and give a detailed description of each of our contributions. In Chapter 6, We present step-by-step improvements of our contributions through ablation studies and compare our approach to other methods in unsupervised image classification to demonstrate its effectiveness. Finally, in Chapter 7, we conclude this thesis with an overview of our contributions.

CHAPTER 2

RELATED WORK

This chapter is adopted from our JVCI journal paper [6] and extended for our recent work.

2.1 Self Supervised Learning with CNNs

2.1.1 Pretext Tasks

The effectiveness of self supervised learning has started to excel with the advent of artificially designed pretext tasks. Doersch et al. [25] split an image into patches and task the model to guess the relative position of a patch to another patch while preventing trivial shortcuts by applying location jittering and color projection/dropping to the patches. Noroozi et al. [57] designed a jigsaw puzzle task where the model guesses the correct ordering of shuffled patches of an image. Zhang et al. [89] propose to predict colors of a given grayscale image in the CIE *Lab* color space in a classification setting. Gidaris et al. [33] rotate images in one of 0, 90, 180 and 270 degrees and propose a classification task where model predicts the class of the rotational degree for the given image. Noroozi et al. [58] learn to count objects in non-overlapping patches of an image versus the whole image to generate semantic representations.

DeepCluster [9] introduces a method that alternates between clustering image features and using the cluster assignments as pseudo-labels for training the network. DeepCluster-v2 [10] improves DeepCluster by introducing an explicit comparisons to k-means centroids, leading to enhanced stability and performance.

Exemplar-CNN [27] leverages surrogate classes created through image transforma-

tion on a single ("exemplar") image and considers each image as a class in the learning algorithm. This method is also very similar to the idea of instance discrimination task where augmentations of the same image are utilized in the learning. Next, we elaborate more on the discrimination task.

2.1.2 Instance Discrimination Task

Instance discrimination task [82] has emerged as a pivotal pretext task for self-supervised representation learning. MoCo [38] achieved notable success through contrastive learning, utilizing a memory bank to store negative samples and employing an exponential moving average (EMA) key encoder to maintain the memory bank embeddings up to date. SimCLR [12] eliminates the memory bank, relying on a large batch size to facilitate learning from negative samples within the mini-batch. BYOL [35] simplifies contrastive learning with a non-contrastive framework, introducing a predictor network to one branch of the flow to break symmetry, while retaining the EMA encoder on the other branch. SimSiam [14] further streamlines BYOL by removing the EMA encoder, ensuring identical feature extractors, and incorporating a stop-gradient operation on one branch to prevent representation collapse. DINO [11] introduces centering and sharpening mechanisms, and VicReg [5] incorporates variance, invariance, and covariance terms into the loss function to prevent collapse. Despite their successes, these methods universally employ a global average pooling operation at the end of feature extractors, resulting in the loss of spatial information and degradation of the localization capabilities of learned representations. Our objective is to extend the instance discrimination task, whether contrastive or non-contrastive, to enhance the localization capability of representations while preserving image-level representation quality.

2.2 Segmentation in Self-Supervised Learning

Segmentation maps have been widely integrated into self-supervised learning methods. CAST [67] incorporates saliency maps in the cropping procedure, constraining random crops to object regions, and computes cosine similarity between saliency

maps and the attention map of the penultimate backbone layer to enhance the internal attention mechanism of the model. DetCon [41] utilizes unsupervised segmentation maps to identify regions belonging to the same objects, performing contrastive learning on local regions in the dense feature maps. Mo et al. [54] introduces a self-supervised class activation map generation method to extract foreground objects and paste them onto different backgrounds, aiming to decouple foreground-background representations. Ki et al. [46] employs rotation augmentation along with a top-k attention pooling operation to generate attention maps, applying contrastive learning on rotated attention maps. Our approach distinguishes itself by directly predicting a class-agnostic segmentation mask in a supervised framework, where self-supervised methods provide the supervision, utilizing this semantic segmentation branch as an auxiliary task to enhance the localization capability of the instance discrimination task.

2.3 Unsupervised Image Classification using Self-Supervised ViTs

2.3.1 Self-Supervised Vision Transformers

Vision Transformers propose a very strong baseline in many computer vision tasks including image classification [26, 21], object detection [8, 50, 88] and segmentation [15, 45] problems. This performance boost by ViTs also propagated to self-supervised learning [11, 91, 86, 3] due to its improved feature representation quality provided by the self-attention mechanism.

DINO [11] explored the capabilities of ViTs with non-contrastive learning through centering and sharpening mechanism, and showed that even though training is performed with image-level objectives, ViTs also successfully learn object boundaries in the attention layers, which later motivated advancements in unsupervised localization tasks such as object discovery [69, 77, 79], object detection [76] and image segmentation [36, 68, 56].

BEiT [4] and iBot [91] applied BERT [24] pretraining on ViTs, referred to as *Masked Image Modeling* (MIM), where BEiT predicts the visual tokens of the original image

as the learning task, while iBOT applies self-distillation on `[CLS]` tokens and patch tokens. MAE [37] use an encoder-decoder architecture to predict pixels by masking a big portion of the images. In a concurrent work, SimMIM [86] similarly propose to predict pixel values instead of patch embeddings using an ℓ_1 loss while experimenting with various masking strategies. MSN [3] propose to apply a random mask on one augmentation of an image while leaving the other augmentation unmasked, and solve a clustering problem on the two representations using cluster prototypes.

DINOv2 [60] propose to curate a large-scale dataset through a self-supervised retrieval system, gathering 142M images in LVD-142M dataset. They introduce several design improvements over the iBOT method including a distillation step and obtain robust self-supervised visual features, achieving state-of-the-art k-NN and linear evaluation results on several image classification and video classification benchmark datasets.

We leverage the robust and high-quality visual representations of the DINOv2 method and propose improvements for unsupervised image classification by making better use of nearest neighbors and multi-head classifiers.

2.3.2 Unsupervised Image Classification/Clustering

Unsupervised image classification, also referred to as deep image clustering, has garnered significant attention as it aims to group images without relying on any human annotations. DeepCluster [9] learn feature representations through alternating between classification and k-means clustering phases while using the cluster assignments as pseudo-labels for the classification phase. SeLa [87] induce an equipartition problem of the data and show that the resulting label assignment problem is the same as the *optimal transport problem*, and solves this problem with fast Sinkhorn-Knopp Algorithm rather than using the k-means clustering phase. DeepClusterV2 [10] solved the correspondence problem of DeepCluster and SeLa methods between different cluster assignments by using a codebook and using comparison between features and these codebooks instead of predicting cluster assignments. ProPos [43] propose to do contrastive learning on cluster prototypes while also aligning the representations of the neighbors to learn well-separated clusters.

SCAN [74] proposed a two stage deep clustering method where they first learn a multi-head deep clustering model using nearest neighbor of images from extracted features, and then apply self-training on the pseudo-labels obtained by the model trained in the first stage. CoKe [63] changes the assumption of having equal-sized clusters and only set a lower-bound on the minimum size of the clusters for a more flexible distribution. SeCu [62] propose a stable cluster discrimination task through mixing the soft labels of different views of an image to optimize the consistency among different views, and applying an instance weighting mechanism that assign higher weights to the harder samples. MIM-Refiner [2] observe that middle blocks of the encoder models trained with masked image modeling have high representation quality, and propose a nearest neighbor alignment task where an instance discrimination head is attached to the intermediate blocks of the encoders instead of only the last block, and applies contrastive learning in the on nearest neighbors. TURTLE [31] use the representation spaces of vision foundation models and propose an unsupervised transfer task to discover the underlying human labeling behind a dataset.

Finally, TEMI [1] proposes an unsupervised multi-head classifier training based on nearest neighbors of samples in a dataset. TEMI uses a student-teacher network, built on top of pretrained self-supervised vision models, which learns to cluster images into semantic groups. TEMI uses the classifier head with the lowest training loss during inference. In this work, we introduce several components that improve the performance of TEMI to a new state-of-the-art on fully unsupervised image classification/clustering problem. We further propose to use the under-utilized classifier heads in an ensembling framework to improve the clustering quality of the best classifier head.

2.3.3 Nearest Neighbors in Self-Supervised Learning

Nearest neighbors are widely used in many self-supervised learning studies. NNCLR [28] leverages nearest neighbors to extend beyond single-instance positives, creating a support set for each image to generate positive pairs, thereby enhancing representations. SNCLR [32] introduces a soft measurement to mine nearest neighbors through cross-attention scores, termed soft nearest neighbors, to support samples using adap-

tive weights. SCAN [74] presents a self-supervised learning task that clusters images, operating on the assumption that nearest neighbors are likely to share the same semantic class. TEMI [1] employs self-supervised Vision Transformers (ViTs) as backbone models to train unsupervised classifiers using nearest neighbors, applying a pointwise mutual information loss between the embeddings of nearest neighbors, similar to the SCAN method. Building on this, we propose an adaptive nearest neighbor mining method to enhance the support set provided by the TEMI method, significantly improving unsupervised image classification accuracy.

CHAPTER 3

SEGINS: MODELS AND METHOD

This chapter is adopted from our JVCI journal paper [6].

We introduce *Segmentation-Enhanced Instance Discrimination (SegIns)* self-supervised learning framework, an extension of the instance discrimination task, to better learn the lacking localization information in the learned representations while preserving the image-level representation quality. The outline of our method is depicted in Fig. 3.1.

As a pseudo-first stage, a model (i.e. the encoder backbone and the projector/predictor in the instance discrimination branch of the figure) is trained using the instance discrimination pretext task to obtain image-level representations. This stage is same with previously proposed methods such as SimSiam [14], BYOL [35], MoCo-V2 [13] or SimCLR [12]; thus, this stage can be skipped for existing methods by re-using the available self-supervised pretrained models. Learned representations are then used to generate class-agnostic segmentation masks to later feed to our learning pipeline as dense supervisory signals. To this end, we use ContraCAM [54] to generate pseudo-masks, while any unsupervised segmentation method can be used to retain the self-supervised learning. In the main stage, we extend the instance discrimination pretext task with a class-agnostic semantic segmentation branch to simultaneously learn dense features along with the image-level representations, which injects localization information to the learned representations.

For completeness, we first provide background information on instance discrimination task and pseudo-mask generation with ContraCAM, and then elaborate more on our proposed method.

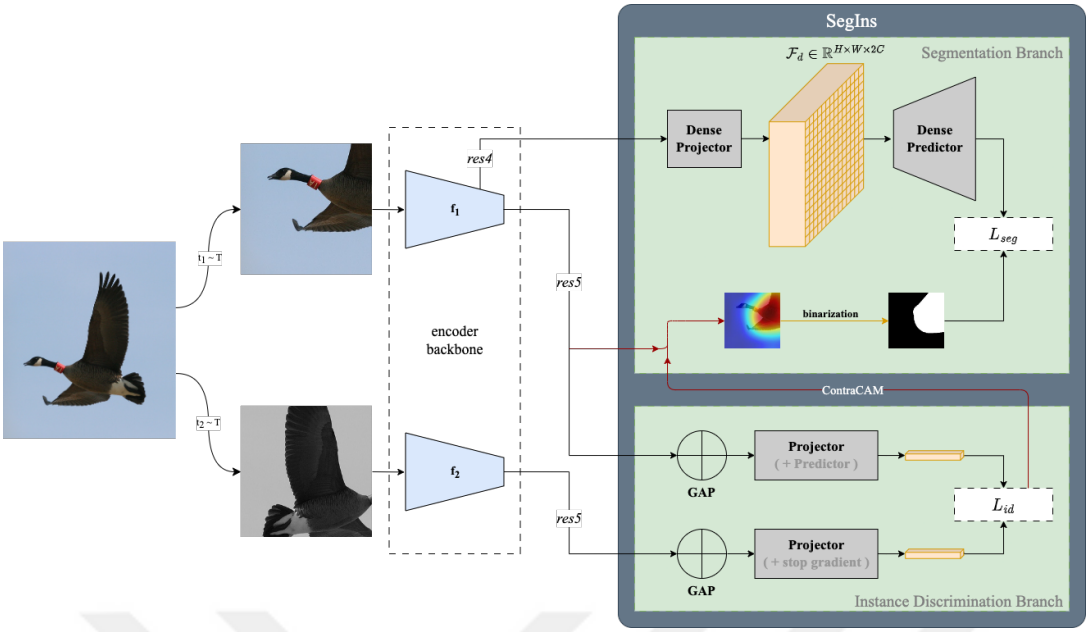


Figure 3.1: **Overview of our Proposed Pretext Task, SegIns.** From an image, two views are sampled with transformations t_1 and t_2 and fed to the encoder backbones f_1 and f_2 to extract dense feature maps. *Instance Discrimination Branch* consists of a standard image-level instance discrimination method such as MoCo [38] or SimSiam [14], where image-level embeddings are generated using a projector network (and then a predictor network for SimSiam). *Segmentation Branch* extends this standard approach by integrating a dense learning flow, which learns a class-agnostic segmentation of the objects from pseudo-binary-segmentation-masks produced by an unsupervised segmentation or saliency based method (e.g. ContraCAM [71]). We show the pseudo-segmentation-mask generation flow as an online process for completeness; however, we use pre-computed segmentation masks during training). Overall, representations learned by SegIns improves localization downstream task performance when transferred to object detection and segmentation while also preserving the original image-level categorization performance.

3.1 Background

3.1.1 Instance Discrimination Task

In the instance discrimination [82] pretext task; first, two views (called the positive pair) of the same image are generated using two sets of random augmentations. During learning, image-level embeddings z_1^+ and z_2^+ that are extracted from the positive pairs through two branches of deep feature extractors f_1 and f_2 are pulled together in the embedding space. In addition, in the contrastive learning setting, embeddings of views generated from different images z^- (called the negative pairs) are pushed apart. In this work, we use special cases of instance discrimination task, namely SimSiam [14] and MoCo-V2 [13] as our baselines.

In SimSiam, the encoder consists of a deep convolutional neural network followed by a 3-layer projector MLP, where parameters are shared between the two branches f_1 and f_2; whereas, a 2-layer predictor MLP h breaks the symmetry on one side, and the stop-gradient operation breaks it on the other in order to avoid representation collapse. Similarity between image-level embeddings are computed with:

$$\mathcal{D}(p_1^+, z_2^+) = -\frac{p_1^+}{\|p_1^+\|_2} \cdot \frac{z_2^+}{\|z_2^+\|_2}, \tag{3.1}$$

where $p_1^+ = h(z_1^+)$ and the $\|.\|_2$ is ℓ_2-norm. Following (3.1), instance discrimination loss is computed with a symmetric loss function as:

$$\mathcal{L}_{id} = \frac{1}{2}\mathcal{D}(p_1^+, z_2^+) + \frac{1}{2}\mathcal{D}(p_2^+, z_1^+). \tag{3.2}$$

In MoCo-V2, the encoder consists of a deep convolutional neural network followed by a 2-layer projector MLP while there is no predictor network. The encoders f_1 and f_2 are referred to as query and key encoders where key encoder is an exponential moving average of the query encoder. Instance discrimination loss is computed with a contrastive loss function, InfoNCE [59], as:

$$\mathcal{L}_{id} = -\log \frac{\exp(z_1^+ \cdot z_2^+ / \tau)}{\sum_{i=0}^{K} \exp(z_1^+ \cdot z_i^- / \tau)}, \tag{3.3}$$

where K is the number of negative samples, τ is a temperature hyper-parameter and $\cdot$ denotes dot product.

Pretraining with instance discrimination pretext task leads to image representations that performs well when transferred to image classification downstream task since both the pretext task and the downstream task utilizes image-level representations; however, it performs worse than dense (or pixel-level) self-supervised learning methods when transferred to object detection / semantic segmentation downstream tasks as the localization information is lost in CNNs during the computation of the image-level embeddings with the use of global average pooling layer.

3.1.2 ContraCAM: Self-Supervised Mask Generation

ContraCAM [54] is a class activation map (CAM) generation method that is tailored to self-supervised learning and can highlight the salient regions of the image that are likely to contain objects. To this end, ContraCAM replaces the usage of cross entropy loss by GradCAM [66] with the instance discrimination loss, and discards negative signals coming from the gradients to further enhance the generated saliency maps which were found to hinder object localization.

While class activation maps have originally been proposed to explain the outputs of supervised models, we employ the generated maps as class-agnostic pseudo-masks to provide supervision to our segmentation branch. We apply thresholding on the resulting ContraCAM outputs at 0.5 to obtain the binary pseudo-masks.

Although any unsupervised semantic segmentation method can be used for the mask generation to retain the self-supervised learning paradigm, ContraCAM prevails with two advantages: (i) it makes our method self-sufficient as it enables using the baseline self-supervised method (e.g. SimSiam, MoCo-V2) to generate pseudo-masks, and (ii) it accommodates a simpler pipeline by not depending on more complex unsupervised mask generation methods.

3.2 SegIns: Segmentation-Enhanced Instance Discrimination Task

We propose *SegIns*, an extension to the instance discrimination task, which simultaneously learns dense feature representations alongside the already occurrent image-level representations.

As for the image-level learning, SegIns uses the pipeline of the adopted baseline method, and further extends it with a class-agnostic segmentation branch for dense representation learning. Specifically, given the feature maps $\mathcal{F}_b \in \mathbb{R}^{H\times W\times C}$ extracted by the encoder backbone; we opt for a simple segmentation branch design which consists of a projection layer with two consecutive 1x1 convolutional layers (followed by batch normalization and ReLU) that outputs dense feature vectors $\mathcal{F}_d \in \mathbb{R}^{H\times W\times 2C}$, and a prediction layer with a 2x2 deconvolutional layer with a stride of 2 and a 1x1 convolutional layer that maps the output of the projection layer to a class-agnostic segmentation map $\mathcal{M}_p \in \mathbb{R}^{2H\times 2W\times 1}$. Finally, we use dice loss [71] to compute the segmentation loss $\mathcal{L}_{seg}$ between the predicted mask $\mathcal{M}_p$ and the pseudo-mask $\mathcal{M}_g$ with the equation:

$$\mathcal{L}_{seg} = 1 - 2\frac{\sum_{i\in I}\mathcal{M}_p^i\mathcal{M}_g^i + 1}{\sum_{i\in I}\mathcal{M}_p^i + \sum_{i\in I}\mathcal{M}_g^i + 1}, \tag{3.4}$$

where $i \in I$ is the index of each pixel; and $\mathcal{M}_p^i$ and $\mathcal{M}_g^i$ are the values of pixels in predicted masks and pseudo-masks in the batch, respectively. 1 is added in numerator and denominator for numerical stability. We compute the final segmentation loss on masks that are generated for both of the augmented views and take the average.

Overall, during training, we define the SegIns loss as:

$$\mathcal{L}_{SegIns} = \mathcal{L}_{id} + \lambda\mathcal{L}_{seg}, \tag{3.5}$$

where the two losses are balanced by λ which is set to 1 for SimSiam, and 2 for MoCo-V2 experiments.

CHAPTER 4

SEGINS EXPERIMENTS

This chapter is adopted from our JVCI journal paper [6].

4.1 Datasets

We use ImageNet-100, a subset of the ImageNet [22] dataset, to train and evaluate our models due to resource restrictions. ImageNet-100 consists of $\sim$125k images which is sufficiently large to perform experiments of statistical significance, as stated in [84].[1] As for the localization transfer learning experiments, we use PASCAL VOC [29] for object detection and semantic segmentation, and MS COCO [52] for object detection, instance segmentation and semantic segmentation downstream tasks.

4.2 Implementation Details

To show that our approach is applicable to both non-contrastive and contrastive learning, we conduct experiments on both MoCo-V2 [13] and SimSiam [14] as the baseline methods; denoting SegIns versions as SegIns$_M$ and SegIns$_S$, respectively. We compute the class-agnostic pseudo-masks using ContraCAM [54] after the baseline instance-discriminator model is trained and before starting the training of SegIns. We use these masks throughout the training. We utilize ResNet50 [40] as the backbone network, and use outputs from the 4-th stage (res4) to feed to the segmentation branch. We employed the default training hyper-parameters and augmentations defined by the

[1] We were not able to use the full ImageNet dataset due to our limited GPU resources.

baseline methods[2], and apply the same geometric augmentations (i.e. random crop, random flip) to the pre-computed pseudo-masks. In addition, we resize the augmented pseudo-masks to a spatial shape of 28x28 and compute the segmentation loss at this resolution. All the models are trained for 500 epochs.

Multi-crop setting. We also employ the multi-crop augmentation setting [10] through the experiments. More specifically, we generate 6 views instead of 2 during the augmentations where 2 views are of shape 224 x 224, called the *global* crops, and 4 views are of shape 96 x 96, called the *local* crops. We use a random scale in the interval [0.4, 1.0] for the global crops, and [0.05, 0.4] for the local crops in the random resized crop augmentation. All other augmentations are kept the same. In this setting, we compute the instance discrimination loss by averaging the loss between global-global and global-local crop duos, and segmentation loss only on the global crops.

4.3 Evaluation Protocols

4.3.1 ImageNet-100 classification

We use the common linear evaluation protocol on ImageNet-100 validation set to evaluate the classification performance of the self-supervised pretrained models, where representations are kept frozen and only a linear classifier is trained on top of the pretrained backbone model. Following [38], we train the linear classifier with a batch size of 256 and a learning rate of 30 for 200 epochs.

4.3.2 PASCAL VOC object detection

We use the pretrained models to initialize the feature extractor of Faster R-CNN [64] with a C4 backbone, and use the settings described in [38] as the training hyperparameters. We fine-tune all layers on `trainval07+12` set for 24k iterations, evaluate on `test2007` set and report the bounding box AP. We use detectron2 [81] as

[2] Due to resource restrictions, we could not afford hyperparameter search. A proper search would likely improve our results.

the detection code base.

4.3.3 PASCAL VOC semantic segmentation

We employ a linear segmentation protocol where a linear layer is trained on top of the backbone network which we initialize with the pretrained models. We use an image size of 512 x 512. The linear layer operates on a feature map with dimensions (2048, 32, 32) and outputs a map with dimensions (num_classes, 32, 32) which is then upsampled to (num_classes, 512, 512) to match the original image dimensions. We train on `train_aug2012` set for 40k iterations with a batch-size of 16 using both frozen and fine-tuned representations. We use the SGD optimizer with a learning rate of 0.01. We evaluate the final model on `val2012` set and report the segmentation mIoU. We use the mmsegmentation toolbox [18] as the linear segmentation code base.

4.3.4 COCO object detection and segmentation

We use the pretrained models to initialize the feature extractor of Mask R-CNN [39] with a C4 backbone and fine-tune all layers on `train2017` set and evaluate on `val2017` set. We again use the settings described in [38] as training hyper-parameters, while training the models with the 2x schedule. We use detectron2 [81] as the detection and segmentation code base.

4.4 Experimental Results

4.4.1 Linear evaluation

We first compare the linear evaluation results between the baseline methods and their *SegIns* extensions on ImageNet-100 dataset. Table 4.1 shows that the classification capabilities of the models don't degrade overall, even slightly improve with the addition of the segmentation branch.

Table 4.1: **Linear Evaluation.** Comparison of linear evaluation performances of baseline methods and their respective SegIns extensions. Overall, SegIns maintains and even slightly improves the classification accuracy with the addition of the segmentation branch.

Method	multi-crop	Top-1	Top-5
SimSiam		81.1	95.7
SegIns$_S$		80.6	95.4
SimSiam	✓	85.4	97.1
SegIns$_S$	✓	85.6	96.7
MoCo-V2		79.8	94.4
SegIns$_M$		80.1	94.8
MoCo-V2	✓	81.1	94.8
SegIns$_M$	✓	82.0	95.3

4.4.2 Affinity of ImageNet and ImageNet-100 pretraining

We first train models using SimSiam and MoCo-V2 methods on ImageNet-100 dataset and apply transfer learning to localization downstream tasks to build up intuition on baseline performance differences caused by the pretraining datasets. Table 4.2 compares linear evaluation Top-1 accuracy on corresponding training datasets, and transfer learning performances on PASCAL VOC object detection and COCO object detection and instance segmentation tasks. ImageNet-100 models are trained for 500 epochs which corresponds to $\sim 25\%$ seen images compared to the 200 epoch ImageNet training. Linear evaluation performance is higher for ImageNet-100 due to the fewer number of classes in the dataset. ImageNet pretrained models perform better than the ImageNet-100 counterparts when transferred to object detection and segmentation downstream tasks. There is a high correlation of transfer learning performance between ImageNet-100 and ImageNet pretraining; thus, we can consider that the ratio of performance difference between two pretraining datasets depicted in Table 4.2 is maintained throughout the rest of the experiments as well.

Table 4.2: **Pretraining Dataset Correspondence.** Performance correspondences of SimSiam and MoCo-V2 methods on ImageNet-100 and ImageNet pretraining datasets. *Linear evaluation:* Models are evaluated on their respective training datasets; *VOC 07 + 12 detection:* Faster R-CNN C4-backbone fine-tuned on `trainval07+12` set for 24k iterations and evaluated on `test2007` set; *COCO detection and COCO instance segmentation:* Mask R-CNN C4-backbone fine-tuned on `train2017` with 1x schedule and evaluated on `val2017` set. ImageNet results are adopted from the SimSiam [14] paper.

			Linear Evaluation	VOC 07 + 12 Detection			COCO Detection			COCO Instance Segmentation		
Method	Dataset	epoch	Top-1	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP^{mask}_{50}	AP^{mask}	AP^{mask}_{75}
SimSiam	ImageNet	200	67.5	82.0	56.4	62.8	57.5	37.9	40.9	56.0	34.4	36.7
SimSiam	ImageNet-100	500	79.8	80.1	53.8	59.1	56.6	37.3	40.1	53.4	32.7	34.9
MoCo-V2	ImageNet	200	70.0	82.3	57.0	63.3	58.8	39.2	42.5	55.5	34.3	36.6
MoCo-V2	ImageNet-100	500	81.1	80.0	53.7	58.6	56.7	37.4	40.1	53.7	32.9	35.0

4.4.3 PASCAL VOC object detection

Table 4.3 compares the transfer learning results of baseline methods and the proposed method. SegIns$_S$ provides 0.7% and 0.9% AP improvements over the baseline SimSiam on two-crops and multi-crop augmentations respectively. Similarly, SegIns$_M$ improves the MoCo-V2 performance by 1.7% and 0.6% AP for two-crops and multi-crop versions. Furthermore, our SegIns$_S$ and SegIns$_M$, despite using a smaller pretraining dataset (ImageNet-100 – only 10% of full ImageNet), consistently outperform the transfer learning performance of ImageNet fully supervised pretrained models by up to 1.4% AP.

4.4.4 COCO object detection and segmentation

We transfer pretrained models to the more complex COCO dataset on object detection and instance segmentation tasks. Table 4.4 presents the 2x schedule fine-tuning results with both SimSiam and MoCo-V2 baselines and their SegIns complements using the Mask R-CNN C4-backbone. While SegIns$_S$ and SegIns$_M$ stays stable for two-crop augmented versions, multi-crop performance improves by 0.8% and 0.6%

Table 4.3: **Object detection fine-tuned on PASCAL VOC.** SegIns extensions of both SimSiam and MoCo-V2, all pretrained on ImageNet-100 dataset, provides improvements in the localization fine-tuning performance by up to 1.2% AP. We show improvements over 0.5 AP with bold fonts.

Method	multi-crop	AP_{50}	AP	AP_{75}
IN Supervised	-	81.8	54.0	59.2
SimSiam		79.9	53.4	59.1
SegIns$_S$		79.6	**54.1**	**59.7**
SimSiam	✓	80.3	54.2	59.4
SegIns$_S$	✓	**80.9**	**55.1**	**60.5**
MoCo-V2		80.0	53.7	58.6
SegIns$_M$		**80.9**	**55.4**	**61.1**
MoCo-V2	✓	80.4	54.6	60.0
SegIns$_M$	✓	**81.0**	**55.4**	**61.3**

AP on object detection and 0.8% and 0.1% AP on instance segmentation. In addition, SegIns$_S$ and SegIns$_M$ ImageNet-100 pretrained multi-crop versions also improve the ImageNet-supervised pretrained model with only 10% images by up to 0.6% AP. Fig. 4.1 shows the qualitative comparisons between multi-crop MoCo-v2, our SegIns$_M$ counterpart and human annotations on COCO `val2017`.

4.4.5 PASCAL VOC linear segmentation

We transfer the pretrained models to the PASCAL VOC semantic segmentation task by either freezing the backbone weights and only training the linear segmentation layer, or fine-tuning the whole model including the backbone. Table 4.5 reports that SegIns consistently improves the semantic segmentation performance for both SimSiam and MoCo-V2 baselines on frozen and fine-tuned settings. Particularly, SegIns$_S$ improves the baseline mIoU by up to 3.6% on fine-tuned and 1.0% on frozen representations; while SegIns$_M$ provides 2.7% and 1.4% improvements over the baseline

Table 4.4: **Object detection and instance segmentation fine-tuned on COCO.** While SegIns$_S$ and SegIns$_M$ stays stable for the two-crop augmentation performance, multi-crop augmentation performances are increased by 0.8% and 0.4% AP on object detection and 0.8% and 0.1% AP on instance segmentation. We show improvements over 0.5 AP with bold fonts.

		COCO Detection			COCO Instance Segmentation		
Method	multi-crop	AP^{box}_{50}	AP^{box}	AP^{box}_{75}	AP^{mask}_{50}	AP^{mask}	AP^{mask}_{75}
random init.	-	54.6	35.6	38.2	51.5	31.4	33.5
IN Supervised	-	59.9	40.0	43.1	56.5	34.7	36.9
SimSiam		59.2	39.8	43.3	56.1	34.8	37.1
SegIns$_S$		59.2	39.8	42.8	56.0	34.7	37.0
SimSiam	✓	59.0	39.6	43.0	55.7	34.5	36.7
SegIns$_S$	✓	**60.3**	**40.4**	**44.0**	**56.7**	**35.3**	**37.7**
MoCo-V2		59.7	39.9	43.2	56.2	34.8	37.5
SegIns$_M$		59.3	40.0	43.5	56.2	34.9	37.3
MoCo-V2	✓	59.5	40.0	43.3	56.3	35.0	37.3
SegIns$_M$	✓	**60.1**	**40.6**	**43.9**	56.7	35.2	**37.7**

for fine-tuned and frozen representations, respectively.

4.5 Analysis

4.5.1 Image-level vs dense representations

Here, we investigate the potential trade-off between the image-level representation learning (through instance discrimination task) and dense representation learning (through class-agnostic segmentation task). To this end, we experiment with varying values of λ (in Eq. (3.5)) which adjusts the relative importance of the segmentation loss in our setting. Table 4.6 shows that even though low or high values of λ degrades the classification performance in favor of better dense representations, a more balanced

Figure 4.1: **Qualitative comparisons between transfer learning performance of MoCo-V2 and SegIns$_M$ pretraining on COCO.** SegIns$_M$ over MoCo-V2 baseline: (i) can better find the extent of the objects, (ii) helps suppressing wrong detections, (iii) improves segmentation mask quality. Best viewed when zoomed in.

image-level and dense learning boosts the transfer learning performance of both classification and localization downstream tasks. Concretely, SegIns performs best with a λ value of 2 where linear classification performance slightly improves while detection fine-tuning and linear segmentation on Pascal VOC dataset both improves significantly.

4.5.2 Segmentation loss function

We analyze the effect of segmentation loss function on SegIns performance, and replace dice loss with binary cross-entropy (BCE) loss for SegIns$_S$ multi-crop setting. Table 4.7 shows that dice loss performs slightly better than the binary cross-entropy loss. We argue that this is caused by the random crop augmentations which tend to generate imbalanced segmentation masks for which dice loss is known to handle better than cross-entropy loss.

Table 4.5: **Linear semantic segmentation on PASCAL VOC.**. While we perform experiments on both frozen and fine-tuned representations on PASCAL VOC semantic segmentation task; SegIns provides solid improvements over the SimSiam and MoCo-V2 baselines, improving baselines up to 3.6% mIoU over SimSiam, and up to 2.7% mIoU over the MoCo-V2. We report the best of 3 runs for each experiment, and show improvements over 0.5 AP with bold fonts.

		Linear Seg. (mIoU)	
Method	multi-crop	Frozen	Fine-Tuned
SimSiam		28.6	57.4
SegIns$_S$		**29.2**	**59.1**
SimSiam	✓	32.9	55.7
SegIns$_S$	✓	**33.9**	**59.3**
MoCo-V2		34.0	59.8
SegIns$_M$		**34.9**	**60.6**
MoCo-V2	✓	34.5	58.4
SegIns$_M$	✓	**35.9**	**61.1**

4.5.3 Effect of unsupervised segmentation outputs

In this experiment, we investigate the effect of quality of the segmentation outputs that are used as the pseudo-labels for the segmentation branch. We utilized the TokenCut [79] unsupervised segmentation model to generate higher quality segmentation maps. While ContraCAM segmentation outputs are mostly blobs around the center of the objects, TokenCut provides more fine-grained segmentation outputs, as shown in Figure 4.2.

We pretrained models for the SimSiam two-crops and multi-crops augmentation baselines using the TokenCut segmentation maps as pseudo-masks. Table 4.8 shows the fine-tuning performance of pretrained models transferred to PASCAL VOC object detection task. We observed that even though TokenCut generates fine-grained segmentaiton masks, it didn't provide improved localization capabilities over the Con-

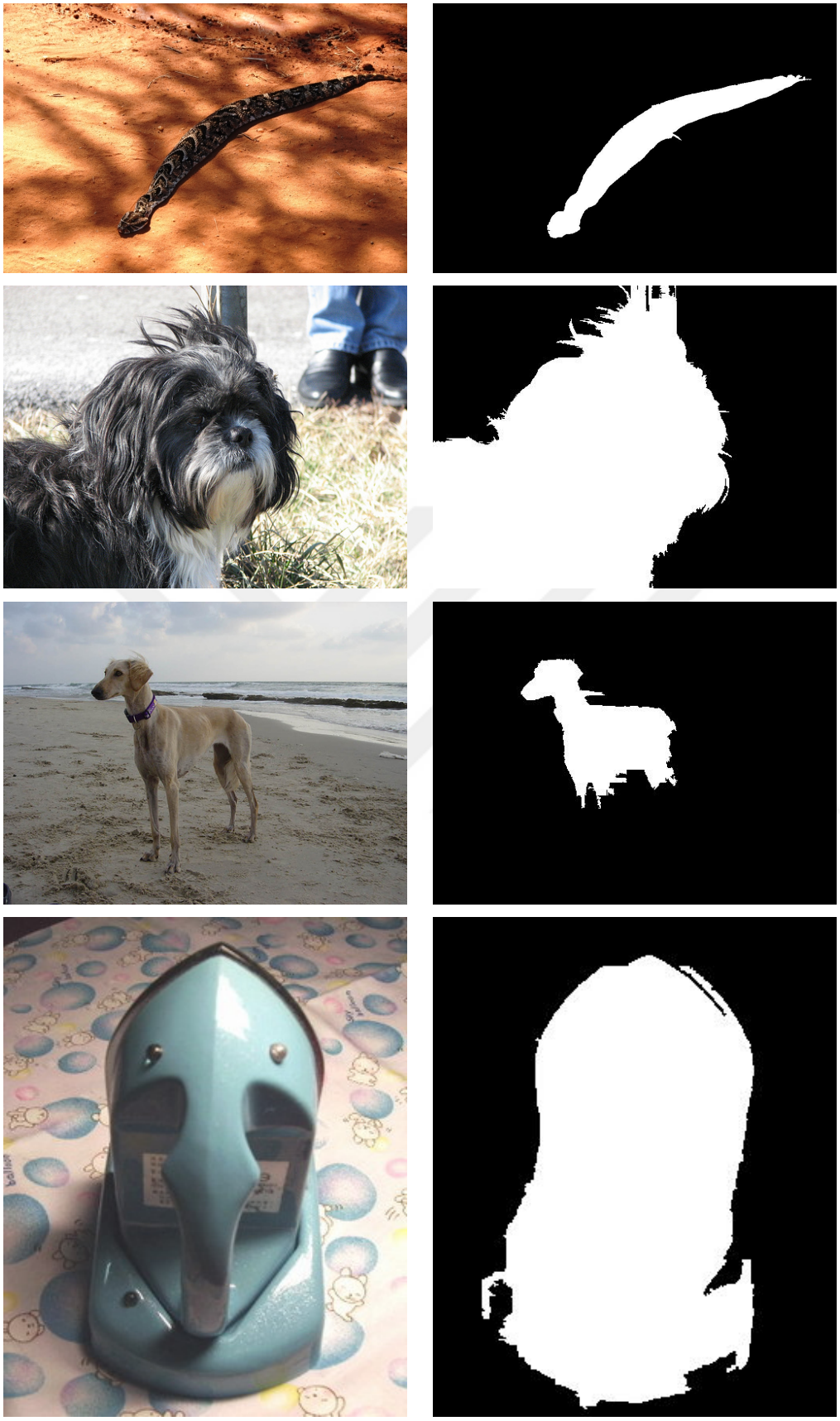


Figure 4.2: Binary segmentation mask outputs by TokenCut [79] on samples from IN-100 dataset.

Table 4.6: **Image-level vs dense representations.** Comparison of the effect of different λ values, which controls the effect of image-level representation learning (instance discrimination loss $\mathcal{L}_{id}$) and dense representation learning (segmentation loss $\mathcal{L}_{seg}$) with SegIns$_M$, two-crops setting on ImageNet-100 dataset. $\lambda = 0$ corresponds to vanilla MoCo-V2. We show the best results with bold fonts.

	IN-100 Linear Evaluation		VOC 07 + 12 Detection			VOC 12 Linear Seg. (mIoU)	
λ	Top-1	Top-5	AP_{50}	AP	AP_{75}	Frozen	Fine-Tuned
0	79.8	94.4	80.0	53.7	58.6	34.0	59.8
0.5	79.0	94.1	80.7	54.4	59.6	34.8	59.5
1.0	**80.1**	94.3	80.4	55.0	60.8	**35.2**	60.3
2.0	**80.1**	**94.8**	**80.9**	**55.4**	**61.1**	34.9	**60.6**
5.0	79.0	93.2	80.7	54.8	61.0	33.7	60.5

Table 4.7: **Segmentation loss function.** Comparison of the effect of dice loss and binary cross-entropy loss as the class-agnostic segmentation loss functions to the fine-tuning performance of SegIns$_S$ on PASCAL VOC object detection task.

Loss Function	AP_{50}	AP	AP_{75}
BCE	80.7	54.6	60.3
Dice	80.9	55.1	60.5

traCAM outputs, and even deteriorated the performance for the two-crop setting. We hypothesize that this can be due to (i) the simplicity of our segmentation branch design to be able to learn fine-grained object masks.

Table 4.8: **Object detection transfer learning on PASCAL VOC dataset with different pseudo-mask generation methods.** SegIns extensions of SimSiam, pretrained on the ImageNet-100 dataset, show improvements in transfer learning for localization tasks when using ContraCAM pseudo-masks. However, the SegIns extension of SimSiam with TokenCut pseudo-masks does not outperform the ContraCAM counterpart and even results in a drop in performance.

Method	multi-crop	Pseudo-Mask	AP_{50}	AP	AP_{75}
ImageNet Supervised	-	-	81.8	54.0	59.2
SimSiam		-	79.9	53.4	59.1
SegIns$_S$		ContraCAM [54]	79.6	54.1	59.7
SegIns$_S$		TokenCut [79]	77.8	52.1	57.1
SimSiam	✓	-	80.3	54.2	59.4
SegIns$_S$	✓	ContraCAM [54]	80.9	55.1	60.5
SegIns$_S$	✓	TokenCut [79]	80.6	55.0	60.9

CHAPTER 5

UNSUPERVISED IMAGE CLASSIFICATION WITH CLUSTER ENSEMBLES

In this chapter, we propose *UCLS*, an *U*nsupervised image *CL*a*S*sification framework consisting of an unsupervised multi-head image classification/clustering phase, a cluster ensembling phase and a self-training phase. We incrementally build upon TEMI [1] —a multi-head image classification method that leverages the nearest neighbors of unlabeled images— and achieve state-of-the-art performance on several image classification benchmarks in unsupervised image classification problem. Figure 5.1 illustrates an overview of our proposed method.

In the following, we begin by covering foundational information on the elements of our approach, including a summary of the TEMI [1] and an overview of the Cluster Ensambles [70] methods. Subsequently, we detail our incremental enhancements to TEMI method aimed at achieving state-of-the-art performance in unsupervised image classification.

5.1 Background

5.1.1 TEMI: *T*eacher *E*nsemble-Weighted Pointwise *M*utual *I*nformation

TEMI [1] is an unsupervised image classification/clustering method based on the nearest neighbors of images. The core idea is that nearest neighbors —computed in the feature space of self-supervised pretrained models— are image pairs that share the same semantic label with a high probability. Consequently, the mutual information between these image pairs is also likely to be high. Building on this observation,

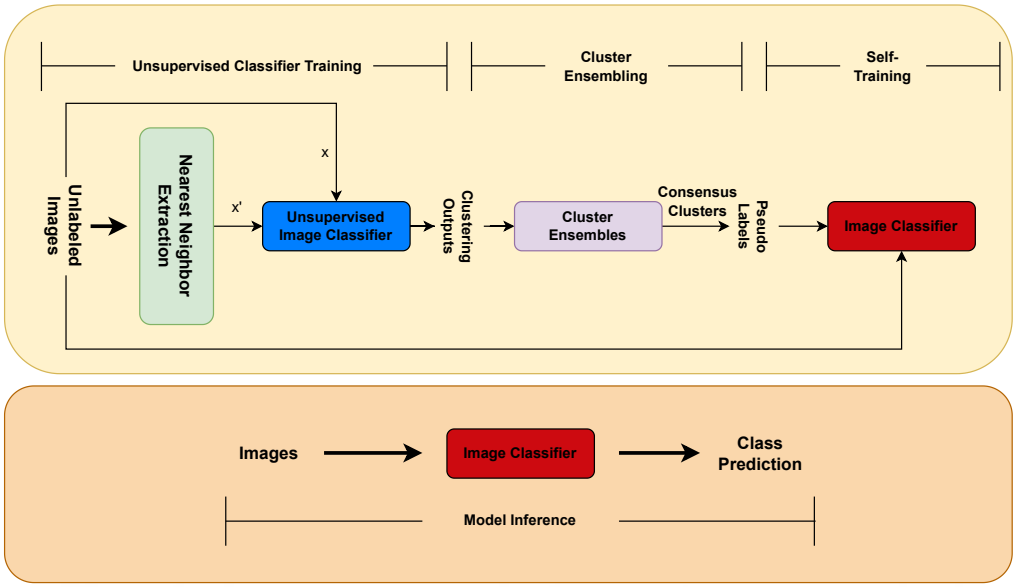


Figure 5.1: **Overview of the Training and Inference Pipeline for our Proposed UCLS Framework.** We propose UCLS, a fully unsupervised image classification/-clustering method. Our framework consists of three stages. In *Stage-1*, an unsupervised multi-head classifier is trained on top of frozen pretrained features by leveraging the nearest neighbors of images. Empirically, we observed that this model alone achieves state-of-the-art unsupervised classification results on image classification benchmark datasets. In *Stage-2*, clustering output of each classifier head is passed to the cluster ensembling method to generate consensus clusters, which improves the clustering quality of the multi-head classifier. In *Stage-3*, an image classification model is trained on the pseudo-labels from the previous stage. This image classifier serves as an unsupervised classification model, assigns images to semantic groups. By combining these stages, UCLS achieves significant improvements in unsupervised image classification/clustering.

given an image x and its nearest neighbor set S_x, TEMI proposes training a parameterized probabilistic classifier, $q(c|x)$, which distributes samples x among classes $c \in C$ by maximizing the pointwise mutual information score $pmi(x, x')$ between x and its neighbor $x' \in S_x$, defined by

$$\mathrm{pmi}(x, x') = \log \sum_{c=1}^{C} \frac{q(c|x)q(c|x')}{q(c)} \tag{5.1}$$

with class occupancy $q(c) = \mathbb{E}_{x\sim p(x)}[q(c|x)]$.

Starting with a pretrained feature extractor, $g(\cdot)$, which maps each image x to a feature vector $g(x)$; a nearest neighbor set S_x is first computed for x using cosine similarity between $g(x)$ and feature vectors of other images in the dataset D. Next, a self-distillation clustering framework is introduced, incorporating a multi-head student network $h_s(\cdot)$ and an exponential moving average (EMA) multi-head teacher network $h_t(\cdot)$, both consisting of three feed-forward layers. Given image pairs x and x', clustering heads compute $q_s(c|x)$ and $q_t(c|x')$ using a temperature scaled softmax function with the formula for student head

$$q_s(c|x) = \frac{\exp(h_s(g(x))_c/\tau)}{\sum_c' \exp(h_s(g(x))_c'/\tau)}, \tag{5.2}$$

where τ is the temperature parameter. While we omit the head index for simplicity, each head computes the probability that a sample belongs to a cluster within its respective clustering space. Using the cluster probabilities, pointwise mutual information is approximated by

$$\widetilde{\mathrm{pmi}}(x, x') := log \sum_{c=1}^{C} \frac{q_s(c|x)q_t(c|x')}{\widetilde{q}_t(c)}, \tag{5.3}$$

and estimate the class probabilities $q(c)$ by en EMA over batches using the outputs of the teacher head

$$\widetilde{q}_t(c) \leftarrow m\widetilde{q}_t(c) + (1-m)\frac{1}{B}\sum_{i=1}^{B} q_t(c|x'), \tag{5.4}$$

where B is the batch size, and m is the momentum parameter. On top of the pointwise mutual information (pmi) estimation, class utilization is also balanced through an exponent in the pmi to prevent over-confident predictions by the network using

$$\widetilde{\text{pmi}}(x, x') := log \sum_{c=1}^{C} \frac{(q_s(c|x)q_t(c|x'))^{\beta}}{\widetilde{q_t}(c)}, \tag{5.5}$$

where $\beta = 0.6$ in experiments. Using a symmetrized pointwise mutual information, the loss function becomes

$$\mathcal{L}(x, x') = -\frac{1}{2}(\widetilde{\text{pmi}}(x, x') + \widetilde{\text{pmi}}(x', x)). \tag{5.6}$$

In addition to this symmetrized loss function, an instance weighting term based on the predictions of the multi-head classifiers is proposed to assign a higher weight to the true positive pairs while assigning a lower weight to possible false positives. This weight term is aimed at decreasing the effect of the noise in the nearest neighbors, thus increase the performance with

$$w(x, x') = \frac{1}{H} \sum_{i=1}^{H} w_i(x, x'), \tag{5.7}$$

where $i \in \{1, ..., H\}$ denotes the index of the head, and individual head weights $w_i(x, x')$ is computed by

$$w_i(x, x') = \sum_{c=1}^{C} q_t^i(c|x) q_t^i(c|x'). \tag{5.8}$$

Following the weighting term and the symmetrized loss function, the final TEMI loss function is defined by

$$\mathcal{L}_{\text{TEMI}}(x, x') := w(x, x')\mathcal{L}(x, x'). \tag{5.9}$$

5.1.2 Cluster Ensembles

Strehl et al. [70] propose the problem of *cluster ensemble*, which involves combining multiple partitionings of a set of objects without accessing the original features. This

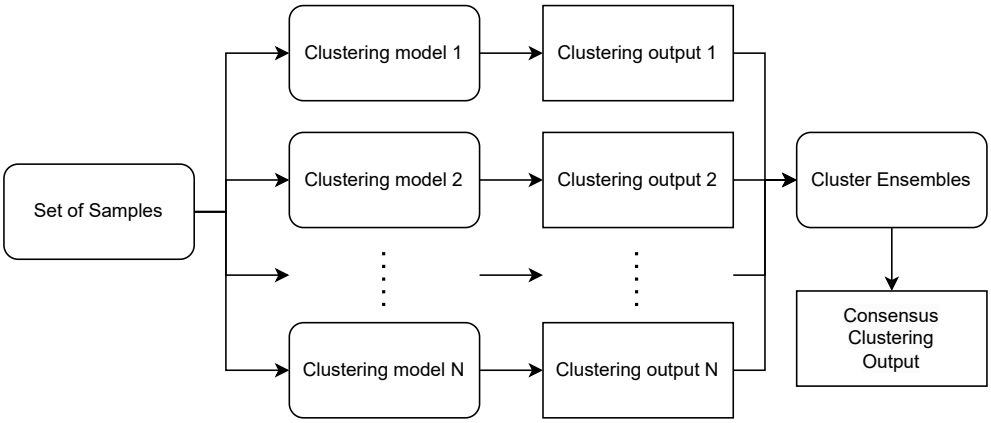


Figure 5.2: **Overview of the Cluster Ensemble problem.** Based on several clustering methods, a set of samples are grouped into different clusterings, which serve as the input for cluster ensemble methods. Using consensus functions, these clustering outputs are combined into a single consensus clustering output.

problem is more challenging than classifier ensembles because it also requires solving the label correspondences between different clusterings. Figure 5.2 depics the outline of the cluster ensemble problem.

In order to solve the cluster ensemble problem, Strehl et al. [70] propose three graph partitioning based consensus functions where each base clustering output is transformed into a hyper-graph representation to achieve a consensus clustering output.

Cluster-based Similarity Partitioning Algorithm (CSPA). In CSPA, it is observed that the clustering output emphasizes a relationship between sample in the same cluster, thus a pairwise similarity measure can be utilized to recluster the samples, yielding a combined clustering.

HyperGraph Partitioning Algorithm (HGPA). In HPGA, the cluster ensemble problem is considered as a partitioning problem of a hyper-graph and a constrained minimum cut objective is used to solve this partitioning problem, where Hyper-edges represent clusters.

Meta-CLustering Algorithm (MCLA). MCLA groups and collapses related hyper-edges, where each cluster is a hyper-edge, and assign each sample to the collapsed hyper-edge (cluster) in which it participates most strongly.

While the details of solving hyper-graph partitioning problem are beyond the scope of this thesis, we provided a summary of each method from Strehl et al. [70] to build a heuristic on how the problem of combining multiple clusterings into a consensus clustering can be solved. For more details, we refer readers to the Strehl et al. [70].

5.2 UCLS: Unsupervised Image Classification with Cluster Ensembles

We present *UCLS*, a state-of-the-art unsupervised image classification framework that exploits the variety of clustering outputs across different heads in a multi-head image classification/clustering model through a cluster ensembling approach. Our proposed framework begins with an enhanced multi-head image classification model, built on top of a self-supervised pretrained backbone model, which learns to classify unlabeled images to a predefined number of classes by utilizing nearest neighbors information.

We begin our contributions by incrementally improving TEMI [1] method with several updates and additional components which we will cover in detail later, specifically: (*i*) a better set of hyperparameters to optimize the distribution of learning throughout the training process, (*ii*) switching the pretrained backbone to DinoV2 [60] for improved feature quality and better nearest neighbor performance, (*iii*) addition of a batch normalization layer after the backbone to stabilize the training process, (*iv*) teacher output normalization with Sinkhorn-Knopp algorithm, (*v*) adaptive nearest neighbors selection by distance thresholding to better leverage the k-NN capabilities of ViTs, (*vi*) enhanced feature representations with the utilization of the last attention block of ViTs, and (*vii*) an improvement to the TEMI loss function for improved performance.

Expanding on this foundation, we propose leveraging the diversity of clustering outputs across the heads of the multi-head classifier. In the TEMI method, inference is performed on the classification head with the lowest training loss. However, we have observed that this approach can lead to suboptimal performance, as each head in the

multi-head classifier forms different clusters during inference, and the head with the lowest loss is not necessarily the best performing one. Essentially, the model trains n classifier heads, and each head can be regarded as a distinct model. Ensembling is a well-established approach to combine different outputs on a problem. However, directly ensembling the results with a hard voting or soft voting is not viable due to the lack of correspondence between the clustering outputs of different heads. In other words, while a classification head may assign class index 1 to an image, another head might assign class index of 5, with both predictions being correct within their respective clustering space. Therefore, we introduce the concept of using a cluster ensembling method, outlined as *Cluster Ensembling* step in Figure 5.1, which solves this correspondence problem.

As a concluding step in our framework, we employ self-training to train a model based on the predictions generated by cluster ensembles. To elaborate, we perform inference on the training set using the multi-head classification model trained in the first phase. We then apply cluster ensembling to the outputs from multiple heads in the second phase and utilize the resulting cluster ensembles outputs as pseudo-labels for training a classification model. The resulting model in this third phase acts as an "unsupervised" classification model as we do not use any label annotations during the pipeline.

Next, we provide detailed information on each of the updates and additional components on the TEMI method.

5.2.1 Hyperparameter Optimization

Based on our empirical studies, we noticed that the performance of the unsupervised classifier reaches a plateau early in training and remains largely unchanged throughout most of the process. In response to this trend, we extend the number of warmup epochs and reduce the base learning rate to facilitate slower training, resulting in enhanced final performance. Finally, we increase the batch size to decrease the training time. We provide a detailed analysis on hyperparameter changes in Section 6.4.

5.2.2 Switching to DinoV2 Features

DinoV2 [60] leverages advanced self-supervised learning techniques on a large curated dataset that enhance the representation learning process, resulting in more accurate and contextually meaningful embeddings. This improvement translates directly into better nearest neighbor performance, where DinoV2 can more effectively retrieve semantically similar instances or data points compared to Dino [11] and MSN [3] features. Thus, we utilize DinoV2 models for our main embedding representation.

5.2.3 Batch Normalization Layer Following the Backbone

Similar to the findings in TEMI [1] paper, we also noted that backbone models generate unnormalized features, leading to unstable training. While TEMI paper addressed this by standardizing feature embeddings using precomputed mean and standard deviation values, we observed a drop of performance in the case of DinoV2 featuers. Therefore, we introduce a batch normalization layer after the backbone model to ensure a more stable training process.

5.2.4 Sinkhorn-Knopp Centering on Teacher Outputs

We adopt the approach outlined in Ruan et al. [65] and follow [60] to incorporate Sinkhorn-Knopp normalization onto the outputs of the teacher model within the multi-head classifier. This adjustment promotes a more uniform utilization of outputs, aligning with the classes in the multi-head classification framework. Similar to DinoV2, we run the Sinkhorn-Knopp algorithm for 3 iterations.

5.2.5 Adaptive Nearest Neighbors Selection by Distance Thresholding

In order to better understand the impact of nearest neighbor quality to the clustering head performance, we conducted experiments using ground truth nearest neighbors. The ground truth for each image is defined as the set of images from the same semantic class according to the human annotations. As depicted in Table 6.5, when

the nearest neighbor quality is optimal, all performance metrics exhibit significant improvements, approaching results comparable to linear probing. Furthermore, we validate that DinoV2's features exhibit high nearest neighbor quality through accuracy experiments, where we calculate the accuracy of nearest neighbors based on correct semantic class matches among nearest neighbor pairs. Figure 6.3 illustrates the nearest neighbor matching accuracy of DinoV2 ViT-L model features across different distance threshold values. Assessing DinoV2's performance in k-NN classification, we revisited the nearest neighbor selection process and introduced a distance thresholding approach. We keep the hard nearest neighbor thresholding by *neighbor count* to fix the minimum number of NNs per image, and further improve the NN set if possible by including the image samples with a higher cosine distance than the provided *distance threshold*.

5.2.6 Feature Enhancement with Last Attention Blocks

Inspired by DinoV2's linear evaluation protocol, which incorporates the last attention blocks with the final CLS token for improved linear probing results, we adopt a similar strategy to enhance our backbone features. Specifically, before the nearest neighbor mining step, we augment the feature dimension by integrating the last attention block. This enhancement extends to the training phase of classification heads, leveraging these enriched features. As a result, not only do we observe enhanced nearest neighbor quality across various datasets, thereby indirectly bolstering clustering head performance during training, but we also experience heightened clustering efficacy due to operating within a more enriched representation space.

5.2.7 Improving Loss Function with Cross Entropy Loss

In our empirical study involving ground truth nearest neighbors, we observed that the performance of the TEMI loss on these nearest neighbors does not match the level achieved by linear probing despite sharing the same frozen backbone features across both experiments. Linear probing involves training a classifier on frozen feature representations using human annotations. We partly attribute this discrepancy to the

limitations of the TEMI loss rather than solely to the usage of nearest neighbors. To address this issue, we propose to add a cross-entropy (CE) loss term to the TEMI loss. We first assign a predicted label $\hat{c}$ to the x' by the argmax operation on the teacher model's output using

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}}\, q_t(c|x'). \tag{5.10}$$

We later use this label as the pseudo-label for the sample x and compute the CE loss between the student network's output $q_s(c|x)$ and the one hot pseudo-label vector $\vec{\hat{c}}$, defined by

$$\mathcal{L}_{CE}((q_s(c|x), \vec{\hat{c}}) = -\sum_{i=1}^{C} \vec{\hat{c}} \log(\hat{y}). \tag{5.11}$$

With the addition of the cross-entropy loss term, the intermediate loss in Equation (5.1.1) becomes

$$\mathcal{L}(x, x') = -\widetilde{\text{pmi}}(x, x') + \lambda \mathcal{L}_{CE}((q_s(c|x), \vec{\hat{c}}) \tag{5.12}$$

where λ is the weighting term for the cross-entropy loss. We empirically observed that a symmetric loss doesn't provide improvements to the final performance under these settings, thus computed the loss only one-way. To ensure effectiveness, we gradually increase the weight of the CE loss using a cosine schedule on λ throughout training, as initial teacher probabilities are unreliable and improve as training progresses. We use $\lambda = 0.5$ throughout our experiments.

Mitigating Errors with Multiple Neighbors Smoothing. To mitigate potential errors in nearest neighbor selection and in class prediction by the classifier model, which can distort the effectiveness of the TEMI and CE loss, we adopted a strategy involving multiple nearest neighbors. We sample m neighbors per image x and compute the mean of the teacher model's outputs across these neighbors, defined by

$$q_t^{mean}(x, x') = \frac{1}{m} \sum_{i=1}^{m} q_t(x, x'_i) \tag{5.13}$$

and use $q_t^{mean}(x, x')$ instead of $q_t(x, x')$ in the loss computation. This approach helps smooth out inaccuracies that may arise from individual neighbor predictions, thereby providing a more reliable and stable basis for training. By leveraging the collective insights from multiple neighbors, our method enhances the robustness of the loss application within our framework, contributing to slightly improved overall performance and training stability.

CHAPTER 6

UCLS EXPERIMENTS

6.1 Datasets

We train and evaluate our proposed framework on 6 different image classification benchmark datasets, namely CIFAR10, CIFAR20, CIFAR100 [47], STL10 [17], Food101 [7] and ImageNet [23]. CIFAR10, CIFAR20 and CIFAR 100 datasets consist of 50k training images and 10k validation images of size 32x32. CIFAR20 dataset has the same training and validation set with the CIFAR100 dataset, while the 100 classes in the CIFAR100 dataset are mapped to 20 superclasses in CIFAR20. STL10 dataset consists of 5k training images and 8k validation images with size 96x96. Food101 dataset consists of 750 training images and 250 test images per class with 101 classes, and ImageNet dataset have 1,281,167 training images and 50k validation images of varying sizes. We also use subsets of ImageNet dataset for our experiments. Tiny-ImageNet [16] consist of 100k training images and 10k validation images that are resized to 64x64. Other variants —ImageNet-50, ImageNet-100 and ImageNet-200 [74]— have 64,274, 128,545 and 256,558 training images and 2,500, 5,000 and 10,000 validation images, respectively. Table 6.1 shows an overview of these datasets.

6.2 Implementation Details

Similar to TEMI [1], for a fair comparison, we assume to know the number of ground truth classes/clusters in the datasets during training. We resize all images to the size of 224x224 and use an AdamW optimizer [53] and a weight decay of 10^{-4}. We experiment with a longer training regime and select training the classifier heads for

Table 6.1: **An overview of image classification benchmark datasets.** We perform our experiments on image classification benchmarks for the second part of this thesis. We used the train set for training our models and validation set to report unsupervised classification performance metrics.

Dataset	Train Images	Val Images	Number of Classes
CIFAR10	50,000	5,000	10
CIFAR20	50,000	5,000	20
CIFAR100	50,000	5,000	100
STL10	5,000	8,000	10
Food101	75,750	25,250	101
ImageNet-50	64,274	2,500	50
ImageNet-100	128,545	5,000	100
ImageNet-200	256,558	10,000	200
Tiny-ImageNet	100,000	10,000	200
ImageNet	1,281,167	5,000	1,000

400 epochs instead of 200. However, we found that longer training provides marginal improvements without consistency. We use 800 epochs for STL10 dataset given the fewer number of images in the training set. We use a batch size of 256 on variants of CIFAR datasets and 1024 on other datasets as we found works better. Following TEMI, we set the number of classifiers heads $H = 50$ and $\beta = 0.55$ for CIFAR20 and $\beta = 0.6$ for the rest of the datasets.

Similar to the findings in TEMI [1], we found that data augmentations don't improve the performance, thus we precompute the feature representations and reuse them for a faster training and evaluation. We conduct our experiments on a single A100 GPU.

During the cluster ensembling experiments, we use the public implementation[1] of Cluster Ensembles [70] method.

For the self-training experiments, we follow DINOv2 [60] evaluation protocol to search for the optimal hyperparameters, where we experiment with different learn-

[1] https://github.com/GGiecold-zz/Cluster_Ensembles

ing rates, how many output layers we use, and whether or not to concatenate the average-pooled patch token features with the class token in one training. We train the linear layers for 12500 iterations using an SGD optimizer and report the accuracy of the best performing setting.

6.3 Evaluation Protocols

We evaluate our models on the validation set of the datasets and use the Hungarian matching algorithm to map the predicted clusters to the ground truth labels. We report our results using three main evaluation metrics: clustering accuracy, normalized mutual information (NMI) and adjusted rand index (ARI).

Clustering Accuracy The Clustering Accuracy (Acc) is the percentage of correct predictions, based on the hungarian matching labels, over the number of images in the validation set. The resulting accuracy score ranges from 0 to 1, where 1 indicates perfect clustering where every data point is correctly grouped, and lower values indicate lesser degrees of alignment with the true labels.

Normalized Mutual Information. The Normalized Mutual Information (NMI) quantifies the amount of information shared between the predicted clusters and the true labels, ranges between 0 and 1. The normalization ensures that the score is not biased by the number of clusters or the size of the dataset. NMI is calculated using the mutual information of the clustering and the true labels, divided by the average of the entropy of the clustering and the entropy of the true labels, ensuring a balanced evaluation. An NMI score of 0 indicates that the clustering results and true labels are completely independent, while a score of 1 indicates perfect correlation, meaning the clustering results exactly match the true labels.

Adjusted Rand Index. The Adjusted Rand Index (ARI) is a measure of the similarity between two data clusterings, adjusted for chance. It evaluates how well the clustering algorithm has performed by comparing the predicted clusters to the true

labels, taking into account the possibility of random clustering. The ARI ranges from -0.5 to 1, where 1 indicates perfect agreement between the clustering and the true labels, 0 indicates random clustering, and negative values indicate less agreement than expected by chance. It corrects the Rand Index by accounting for the fact that some agreement between clusters is expected by chance, making it a more reliable and unbiased metric for evaluating clustering performance across different datasets and clustering algorithms

6.4 Experimental Results

In this section, we present our results on incremental enhancement experiments for the unsupervised multi-head classification, cluster ensembling experiments using the clustering outputs of multi-head classifiers, and finally self-training experiments on the cluster ensembling outputs for the complete *UCLS* framework.

6.4.1 Unsupervised Classification Enhancement Experiments

First, we present our incremental enhancements to unsupervised multi-head classification. We begin with TEMI [1] as the foundation for our proposed UCLS framework. Our baseline experiments are conducted using the ImageNet-100 dataset with the DINO [11] ViT-Base backbone, building upon our reproduced results. Following TEMI, we trained the classifier for 200 epochs, including 20 epochs of linear warm-up, and used a learning rate of 10^{-4}. We achieved a baseline clustering accuracy of 79.36%.

6.4.1.1 Hyperparameter optimization

During our initial experiments on the baseline method, we observed that training performance quickly elevates and remains largely unchanged throughout the training. Based on this observation, we hypothesized that the learning process could be more evenly distributed throughout the entire training period by increasing the number of warm-up epochs and lowering the learning rate. We also increased the batch size to

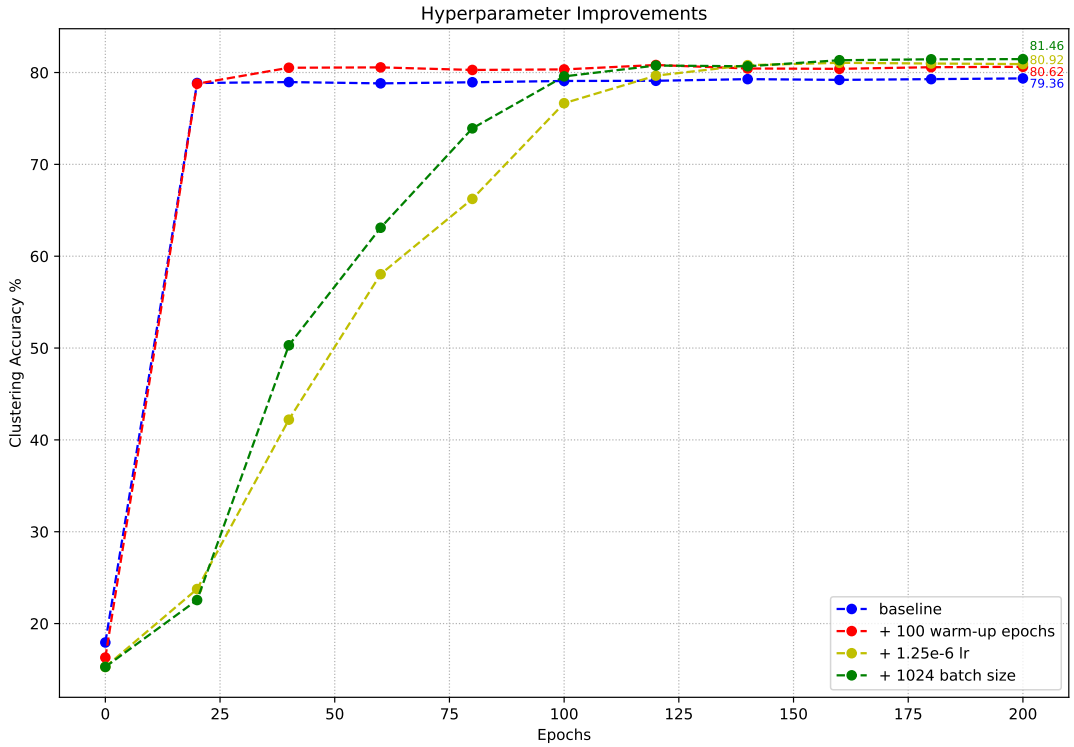


Figure 6.1: **Hyperparameter Optimization.** Clustering accuracy change during the incremental updates on the baseline training hyperparameters of the TEMI method using DINO ViT-Base [11] backbone on ImageNet-100 dataset. We gain an absolute of 2.1% clustering accuracy improvement through hyperparameter updates.

speed up the training.

To implement these changes, we increased the number of warm-up epochs from 20 to 100 and reduced the learning rate from 10^{-4} to 1.25×10^{-6}. Additionally, we increased the batch size from 512 to 1024, which not only decreased the training time but also provided slight improvements in final performance. Figure 6.1 illustrates the changes in clustering accuracy during the incremental hyperparameter updates. This step provided an absolute 2.1% clustering accuracy improvement over the baseline method.

Table 6.2: **SSL Method Comparison.** k-NN and linear probing performance comparison of the SSL methods with strong baselines on ImageNet dataset. Results are taken from DINOv2 [60] paper.

Method	Arch	INet-1k k-NN	INet-1k linear
DINO	ViT-S/8	78.6	79.2
MSN	ViT-L/7	79.2	80.7
DINOv2	ViT-L/14	83.5	86.3
DINOv2+Reg	ViT-L/14	**83.8**	**86.7**

6.4.1.2 Switching to DINOv2 Features

As a second step towards improving unsupervised classification, we enhanced the feature representations by switching to the DINOv2 [60] models. DINOv2 offers more robust feature representations, particularly in terms of nearest neighbor quality. Table 6.2 compares the k-nearest neighbor and linear probing performance of four self-supervised learning methods, namely DINO [11], MSN [3] and DINOv2 [60] and DINOv2 with registers [20]. Since higher k-NN performance directly correlates with better feature representations and improved nearest neighbor accuracy, adopting the DINOv2 backbone provides multiple advantages. We leveraged the latest advancements in ViTs and employed DINOv2 with registers [20], which offers slightly improved k-NN performance and enhanced attention maps. This update to the backbone model pushed the unsupervised classification/clustering accuracy to a new state-of-the-art of 87.7% on Imagenet-100 dataset. In the rest of the experiments, we refer to DINOv2 with registers as DINOv2 for bravity.

6.4.1.3 Batch Normalization

We empirically observed that the feature normalization using the precomputed mean and standard deviation in the TEMI [1] paper provides sub-optimal performance with the DINOv2 feature embeddings. Therefore, we proposed adding a batch normalization layer [44] after the backbone network. Table 6.3 presents the improvements pro-

vided by the batch normalization layer to the performance different backbone models. We gain an absolute accuracy of 1.76% on DINO ViT-B/16, 2.20% on MSN ViT-L/16 and 0.92% on DINOv2 ViT-L/14 features, pushing the unsupervised classification/-clustering accuracy to a 88.6% with the DINOv2 features. The addition of the batch normalization layer further stabilizes the training for the rest of our experiments.

Table 6.3: **Batch Normalization.** Improvements provided by the batch normalization layer on various backbone models on ImageNet-100 dataset. Baseline refers to the TEMI settings with hyperparameter optimizations.

Methods	NMI(%)	ACC(%)	ARI(%)
baseline w/ DINO ViT-B/16	88.66	81.46	72.87
+ BN	89.41	83.22	75.03
baseline w/ MSN ViT-L/16	89.22	82.18	73.93
+ BN	90.20	84.38	76.85
baseline w/ DINOv2 ViT-L/14	93.14	87.66	82.29
+ BN	**93.22**	**88.58**	**82.79**

6.4.1.4 Sinkhorn-Knopp Centering

We adopted the teacher network output normalization used by SwAV [10], Ruan et al. [65] and DINOv2 [60], utilizing the Sinkhorn-Knopp (SK) centering. This normalization significantly enhanced clustering performance across the classifier heads. Table 6.4 shows the changes in clustering metrics for both the head with the lowest loss and the average performance across all clustering heads. The addition of the Sinkhorn-Knopp centering further improved the unsupervised classification/clustering accuracy performance to 90.9%.

Additionally, Sinkhorn-Knopp centering improved the distribution of samples to clusters, particularly in the training set. Figure 6.2 presents a comparison of sample distribution among clusters using the DINOv2 features with and without Sinkhorn-Knopp centering. While performance on the validation set remains similar between the two

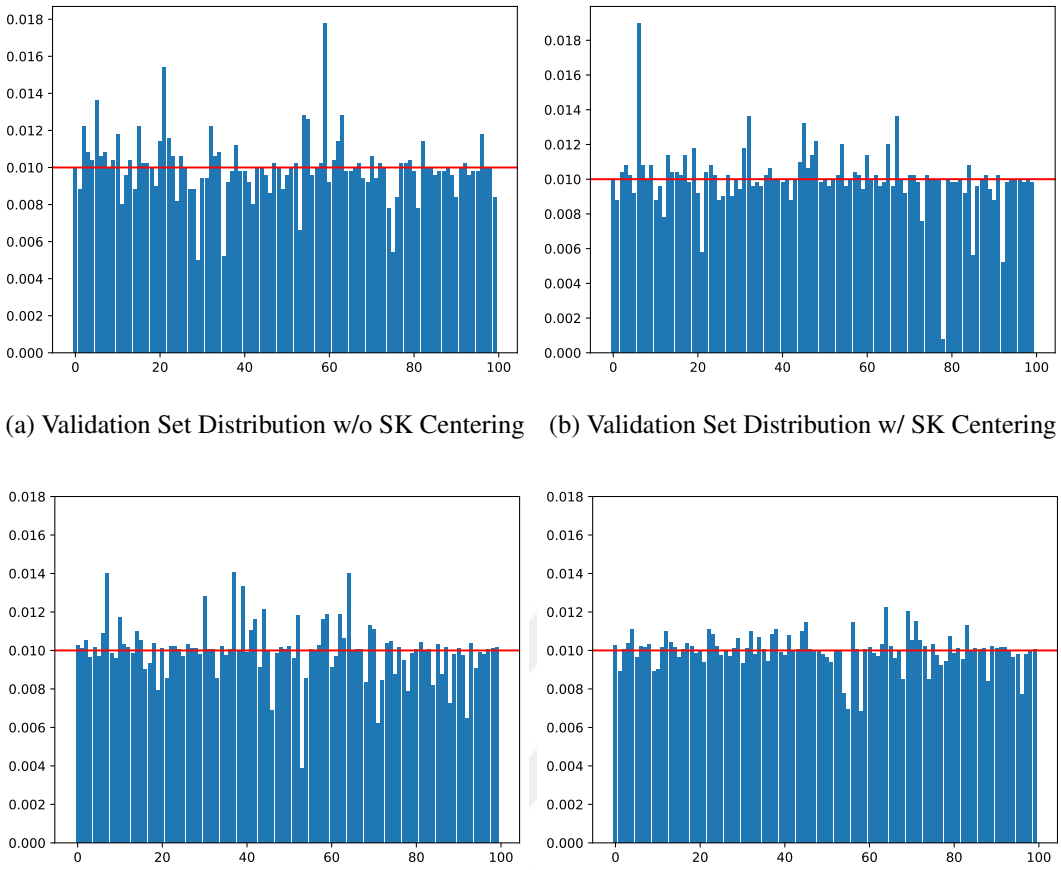
Table 6.4: **Sinkhorn-Knopp Centering.** Unsupervised image classification improvements on ImageNet-100 dataset provided by the Sinkhorn-Knopp centering on various backbone models. The *Best Head* results pertain to the classifier head with the lowest training loss, while the *Overall* results present the mean and standard deviation across all 50 classification heads. The best results are boldfaced. Baseline results refer to TEMI settings with hyperparameter optimization and batch normalization.

	Best Head			Overall		
Methods	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)
baseline w/ DINO ViT-B/16	89.41	83.22	75.03	88.43±0.44	78.74±1.61	71.31±1.30
+ SK	89.31	83.04	74.69	89.29±0.29	83.20±0.90	74.84±0.84
baseline w/ MSN ViT-L/16	90.20	84.38	76.85	88.83±0.39	79.69±1.55	72.68±1.27
+ SK	90.08	85.30	76.81	89.49±0.27	83.36±0.95	75.44±0.82
baseline w/ DINOv2 ViT-L/14	93.22	88.58	82.79	92.17±0.48	84.49±1.68	79.22±1.61
+ SK	**94.10**	**90.90**	**85.56**	**93.02±0.48**	**87.81±1.32**	**82.47±1.43**

outputs, the distribution of training set samples improves significantly. Following TEMI [1], we compute the KL-divergence between the predictions and the uniform distribution for both the validation and training sets. We find that KL divergence is notably lower with SK centering on the training set, 1.1×10^{-2} compared to 3×10^{-3}, while remaining at similar levels on the validation set, 1.4×10^{-2} and 1.8×10^{-2} respectively. It should be noted that a KL divergence of 0 would indicate that the predictions perfectly match a uniform distribution.

6.4.1.5 Adaptive Nearest Neighbors Selection with Distance Thresholding

Upper Bound Analysis with Ground Truth Nearest Neighbors. After achieving significant improvements through several enhancements to the baseline TEMI [1] method, we explored the upper bound for multi-head classifier training on the nearest neighbor set, as mentioned in Section 5.2.5. This analysis aims to highlight the importance of the quality of the nearest neighbor set during unsupervised training. To this end, we utilized the ground truth nearest neighbors set for each sample, determined by the samples sharing the same ground truth semantic label. We trained a



(a) Validation Set Distribution w/o SK Centering

(b) Validation Set Distribution w/ SK Centering

(c) Training Set Distribution w/o SK Centering

(d) Training Set Distribution w/ SK Centering

Figure 6.2: **Cluster Utilization w/ and w/o SK Centering.** We illustrate the sample distribution among classes/clusters with and without the Sinkhorn-Knopp centering on ImageNet-100 dataset. While the horizontal red line represents the ideal histogram in the case of uniform distribution, the usage of the SK centering leads to a better utilization of the clusters.

new model incorporating all the enhancements made thus far, including hyperparameter optimizations, batch normalization, and Sinkhorn-Knopp centering on teacher outputs. Table 6.5 demonstrates that using the ground truth NNs, clustering accuracy approaches the linear probing performance, suggesting that better utilization of nearest neighbors should enhance clustering quality. It is important to note that even though we use the ground truth NN set for this experiment, no semantic labels are fed into the training process, whereas linear probing uses 100% of the labels with a cross-entropy loss.

Table 6.5: **Upper Bound Analysis on Nearest Neighbors Quality.** Clustering accuracy improves with the ground truth nearest neighbor usage during the training of the multi-head classification model. Experiments are performed with DINOv2 ViT-L/14 model on ImageNet-100 dataset.

Method	NMI(%)	ACC(%)	ARI(%)
Top-50 NNs	94.10	90.90	85.56
Ground Truth NNs	96.58	95.46	92.32
Linear Probing	96.67	96.32	92.98

DINOv2 Nearest Neighbor Accuracy Analysis. Encouraged by the aforementioned upper-bound performance analysis towards the utilization of nearest neighbors; we further dig into the nearest neighbor quality analysis using DINOv2 features on various image classification benchmark datasets. Mainly, we apply different distance thresholds to filter the nearest neighbors set for each image, and compute the accuracy of the nearest neighbor sets based on correct semantic class matches among nearest neighbor pairs. We observe that the quality of the nearest neighbor sets can raise around the ground truth level for some datasets with correct distance thresholds, as illustrated in Figure 6.3.

Adaptive Nearest Neighbors Selection with Distance Threshold. Based on the upper bound performance and nearest neighbor accuracy analysis, we modified the nearest neighbor selection process and achieved further improvements by introducing an adaptive distance-based nearest neighbor selection method. This approach aims to better leverage the superior nearest neighbor performance of the DINOv2 features. While TEMI [1] selects the top-50 nearest neighbors for each image in the ImageNet-100 dataset, we propose using this set as the minimal NN set and expanding it by applying a distance threshold, including every neighbor that meets this criterion.

We conducted initial experiments using a distance threshold of 0.5, both with and without Sinkhorn-Knopp (SK) centering. This strategy significantly increases the average number of nearest neighbors per image, from 50 to 651.4, on the ImageNet-100

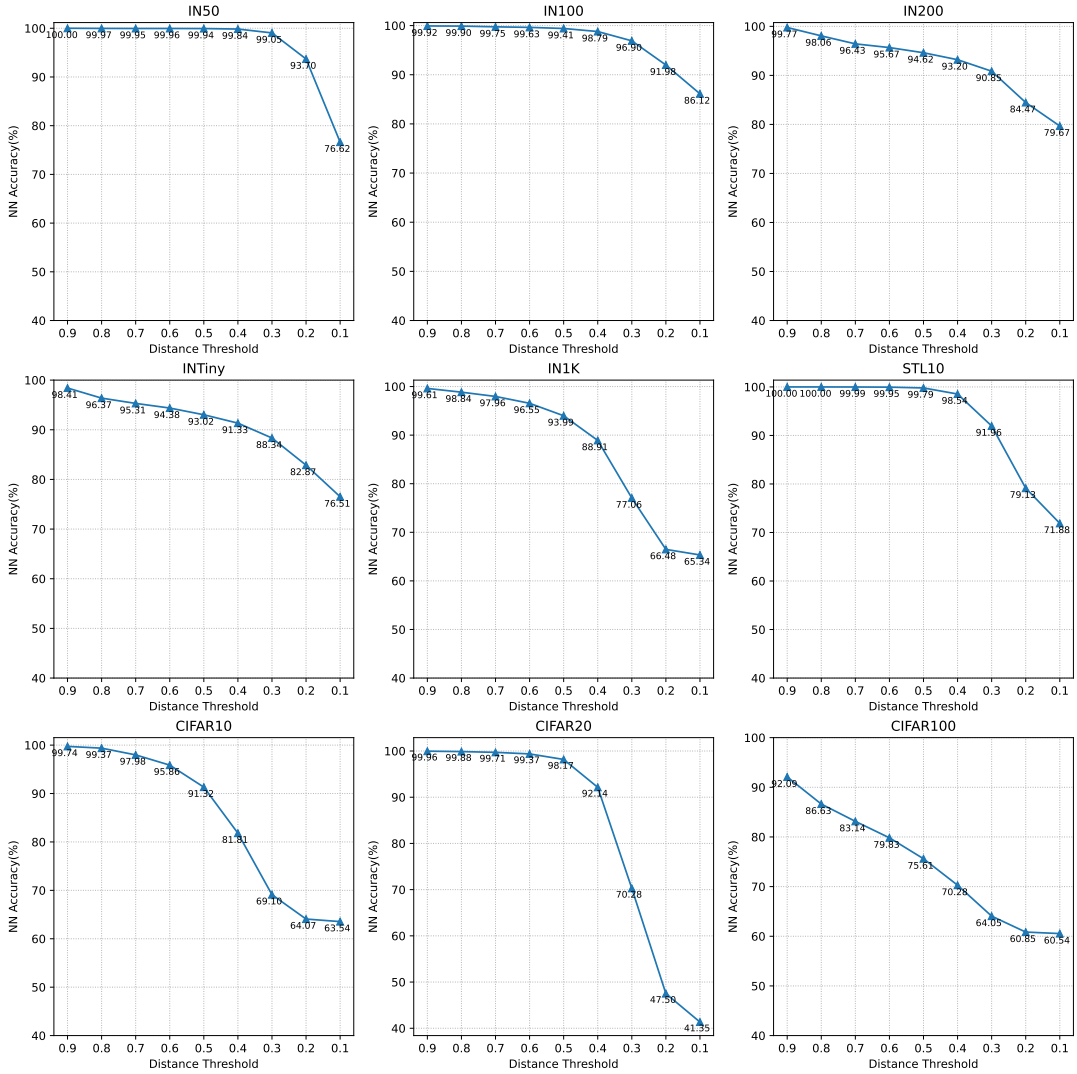


Figure 6.3: **Nearest Neighbor Accuracy Analysis on Various Datasets with DINOv2 ViT-L/14.** Nearest neighbor accuracy remains very high across all datasets when the distance threshold is set to a higher value. Although the NN accuracy decreases with lower distance threshold values, it remains sufficiently high to achieve performance improvements when used in the nearest neighbor selection process. Best viewed when zoomed in.

dataset. We observed that distance thresholding without the SK centering provides a similar improvement to SK centering and further boosts performance when used together. Table 6.6 shows that adaptive NN thresholding results in an absolute 1.8% clustering accuracy improvement over the batch normalization enhancements and a 0.2% improvement over SK centering enhancements. Additionally, adaptive NN se-

Table 6.6: **Adaptive Nearest Neighbor Selection.** Unsupervised image classification improvements provided by the adaptive nearest neighbor selection strategy on ImageNet-100 dataset using various settings. The *Best Head* results pertain to the classifier head with the lowest training loss, while the *Overall* results present the mean and standard deviation across all 50 classification heads. Baseline refers to TEMI settings with hyperparameter optimization and batch normalization.

	Best Head			Overall		
Methods	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)
baseline w/ DINOv2 ViT-L/14	93.22	88.58	82.79	92.17±0.48	84.49±1.68	79.22±1.61
+ SK	94.10	90.90	85.56	93.02±0.48	87.81±1.32	82.47±1.43
+ Adaptive NN	94.24	90.34	85.75	93.27±0.45	86.20±1.72	81.91±1.56
+ SK + Adaptive NN	**94.41**	**91.10**	**86.46**	**93.91±0.33**	**89.64±0.99**	**84.88±1.04**

lection enhances performance across all classification heads, providing an absolute 1.8% mean clustering accuracy improvement. Overall, the nearest neighbor selection strategy further improved unsupervised classification/clustering accuracy performance to 91.1%. We further analyze the effect of different distance thresholds in Section 6.4.4.1.

6.4.1.6 Feature Enhancement with Attention Blocks

Motivated by the linear probing experiments of DINOv2 [60], we incorporated the features from the last attention block into the feature set used in our experiments. This adjustment doubled the feature embedding size from 1024 to 2048, resulting in a slight improvement in the k-NN performance of the backbone features, which likely indicates an enhanced feature representation space. Table 6.7 shows the improvements in final performance, where the baseline employs all the enhancements made thus far. Overall, the enhanced feature representation space further improved unsupervised classification/clustering accuracy to 92.1%, with an absolute 1.0% increment.

Table 6.7: **Feature Space Enhancement.** Unsupervised image classification improvements provided by the feature space enhancement through attention layers on ImageNet-100 dataset. The *Best Head* results pertain to the classifier head with the lowest training loss, while the *Overall* results present the mean and standard deviation across all 50 classification heads. Feature enhancement improves the mean clustering accuracy across clustering heads, surpassing the 90% mark. Baseline refers to TEMI settings with hyperparameter optimization, batch normalization, adaptive nearest neighbor selection and sinkhorn-knopp centering.

	Best Head			Overall		
Methods	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)
baseline w/ DINOv2 ViT-L/14	94.41	91.10	86.46	93.91±0.33	89.64±0.99	84.88±1.04
+ Feat	**94.92**	**92.10**	**88.02**	**94.38±0.30**	**90.09±0.95**	**85.86±0.98**

6.4.1.7 Improved Loss Function with Cross-Entropy Loss

Based on the ground truth nearest neighbor experiments in Table 6.5, we observed that unsupervised classification performance, which uses the ground truth NN sets, can not match with the linear probing results, which directly utilizes the semantic labels during training. As a remedy, we propose to add a cross-entropy (CE) loss term to the TEMI loss as explained in Section 5.2.7 to see whether the performance discrepancy is caused by the usage of the CE loss. We train the multi-head classifier using the latest model enhancements, and further use the CE loss term in the loss objective. We also find it beneficial to train the classifier for a longer period, thus we increase the training epochs from 200 to 400. Table 6.8 presents the incremental improvements provided by the longer training and the new loss term. Specifically, we achieve an unsupervised classification/clustering accuracy of 92.9%, with an absolute 0.8% increment. The usage of CE loss term improved the performance even though there are two sources of error which can inject noise in the learning signal; *(i)* classification errors made by the model can provide wrong cluster index to the CE loss, and *(ii)* errors caused by the nearest neighbor selection can select samples with non-matching semantic labels which can similarly provide wrong cluster index to the CE loss.

Table 6.8: **Cross-Entropy Loss Term with Longer Training.** Unsupervised image classification improvements provided by the cross-entropy loss term and longer training on ImageNet-100 dataset. The *Best Head* results pertain to the classifier head with the lowest training loss, while the *Overall* results present the mean and standard deviation across all 50 classification heads. Baseline refers to TEMI settings with hyperparameter optimization, batch normalization, adaptive nearest neighbor selection, sinkhorn-knopp centering and features space enhancement.

	Best Head			Overall		
Methods	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)
baseline w/ DINOv2 ViT-L/14	94.92	92.10	88.02	94.38±0.30	90.09±0.95	85.86±0.98
+ 400 epochs	94.66	92.00	87.53	94.22±0.31	90.34±0.99	85.79±1.01
+ TEMI w/ CE	**95.33**	**92.92**	**88.74**	**95.04±0.19**	**91.71±0.86**	**87.64±0.79**

6.4.1.8 Comparison with the Baseline

First, we present the overall enhancements in Table 6.9 to provide a holistic view of the performance improvements. We trained a model using the baseline TEMI [1] settings with DINOv2 [60] features, achieving a clustering accuracy of 79.4%. We excluded the hyperparameter optimization and longer training from this set of experiments to better observe the training stability that our components provide. The proposed enhancements increased the unsupervised classification/clustering accuracy to 93.2%, representing an 13.7% absolute and 17.3% relative improvement. This result sets a new state-of-the-art for unsupervised classification/clustering performance on the ImageNet-100 dataset.

Next, we demonstrate the effectiveness of our proposed enhancements by comparing our trained multi-head classifiers with the baseline TEMI method on eight image classification benchmark datasets in Table 6.10. Our approach achieves new state-of-the-art results across all benchmark datasets for fully unsupervised image classification/clustering problems, attaining a clustering accuracy of 99.3% on CIFAR10, 87.5% on CIFAR100, and 70.2% on ImageNet datasets.

Table 6.9: **Ablation Results on the Enhancements.** We optimize the unsupervised multi-head image clustering performance with several enhancements. These enhancements later provide a notable boost of performance with the usage of cluster ensambling method in the next set of experiments. An adaptive distance threshold of 0.3 is used to select nearest neighbors. Total performance gains provided by the cumulative enhancements on ImageNet-100 dataset are presented in the bottom row.

Methods	NMI(%)	ACC(%)	ARI(%)
Baseline (DINOv2)	88.75	79.44	71.50
+ Feat.	89.58 ↑ 0.83	79.82 ↑ 0.38	72.66 ↑ 1.16
+ BN	90.10 ↑ 0.52	81.86 ↑ 2.04	74.44 ↑ 1.78
+ Adaptive NN (0.3)	94.83 ↑ 4.73	91.00 ↑ 9.14	87.02 ↑ 12.58
+ SK	95.03 ↑ 0.20	92.60 ↑ 1.60	88.23 ↑ 1.21
+ TEMI w/ CE.	**95.16** ↑ 0.13	**93.18** ↑ 0.58	**88.80** ↑ 0.57
Δ	**+6.41**	**13.74**	**+17.30**

Training Details. Building upon the improvements gained through the proposed enhancements in Table 6.9, we further tweak some hyperparameters for different datasets, which empirically provided better results.

Teacher temperature. We use a teacher temperature of 0.07 for CIFAR datasets, and 0.1 for all other datasets.

Adaptive Nearest Neighbor Threshold. Through additional experiments, we found that a nearest neighbor distance threshold of 0.3 works better across all datasets. Therefore, we set the default distance threshold to 0.3. We provide a thorough analysis on distance threshold in Section 6.4.4.1.

Backbone Model. While we generally use a DINOv2 ViT-L/14 + Registers model on all benchmark datasets, we found that this model causes a performance drop on the STL10 dataset. Hence, we switch to the DINOv2 ViT-B/14 model for the STL10 dataset.

Batch Size. We use a batch size of 256 for the CIFAR datasets, which provides im-

Table 6.10: **Overall Improvements Over the TEMI Baseline.** We report the mean and standard deviation of 5 independent runs with different seeds as our results and adopted the results of the TEMI [1] method from the corresponding paper. Our enhancements provide significant improvements over the baseline, detailed in the last column as the delta between the mean accuracy of TEMI and our method.

	TEMI Baseline				Our Results				Δ
Datasets	Backbone	NMI(%)	ACC(%)	ARI(%)	Backbone	NMI(%)	ACC(%)	ARI(%)	ACC(%)
STL10	DINO (ViT-B/16)	96.5±0.13	98.5±0.04	96.8±0.04	DINOv2 (ViT-B/14)	98.90±0.06	99.57±0.04	99.05±0.08	+1.1
CIFAR10	DINO (ViT-B/16)	88.6±0.05	94.5±0.03	88.5±0.08	DINOv2 (ViT-L/14)	97.93±0.07	99.27±0.04	98.38±0.08	+4.8
CIFAR20	DINO (ViT-B/16)	65.4±0.45	63.2±0.38	48.9±0.21	DINOv2 (ViT-L/14)	74.09±0.83	67.71±1.03	55.83±1.09	+4.5
CIFAR100	DINO (ViT-B/16)	76.9±0.45	67.1±1.30	53.3±1.02	DINOv2 (ViT-L/14)	90.60±0.05	87.49±0.24	80.08±0.21	+20.4
ImageNet-50	MSN (ViT-L/16)	88.14±0.55	84.87±1.16	76.46±1.17	DINOv2 (ViT-L/14)	96.81±0.11	97.06±0.10	94.13±0.18	+12.1
ImageNet-100	MSN (ViT-L/16)	88.53±0.56	82.86±0.73	74.08±1.20	DINOv2 (ViT-L/14)	95.44±0.09	93.42±0.26	89.34±0.33	+10.5
ImageNet-200	MSN (ViT-L/16)	86.65±0.32	77.96±0.71	66.70±0.71	DINOv2 (ViT-L/14)	93.54±0.09	88.64±0.26	82.58±0.29	+10.6
ImageNet	MSN (ViT-L/16)	82.5	61.56±0.28	48.4	DINOv2 (ViT-L/14)	87.73±0.03	70.23±0.16	59.45±0.14	+8.6

proved results under the same learning rate and warm-up epochs. For the rest of the benchmark datasets, we use a default batch size of 1024.

6.4.2 Cluster Ensembles and Self-Training Experiments

As the second and third components of our UCLS framework, we present the use of cluster ensembling [70] and self-training on the outputs of the multi-head classifier. Previously, in the TEMI method [1], only the teacher head with the lowest loss was utilized during inference on the test set, while the rest of the heads contributed solely to the training process through the sample weighting approach in the loss function. However, as empirically demonstrated in Section 6.4.1, other classifier heads also exhibit strong clustering performance when used during inference.

Table 6.11: **Cluster Ensembling Results on Different Accuracy Levels.** Cluster ensembling consistently provide improvements across different accuracy spectrum. While the improvement is substantial for models with lower accuracy, the gap between the baseline and the ensembling narrows for higher accuracy models. With the addition of each component, ensembling performance improves since all the clustering heads perform better along with the best head. †corresponds to ensembling on the validation set clustering outputs, while ‡corresponds to ensembling on the training set clustering outputs first and then using self-training on the pseudo-labels obtained from the training set for inference.

Methods	NMI(%)	ACC(%)	ARI(%)
Baseline (DINOv2)	88.75	79.44	71.50
+ Ensemble †	92.95 ↑ 4.20	87.80 ↑ 8.36	82.47 ↑ 10.97
+ Ensemble + Self-Training ‡	93.17 ↑ 4.42	88.34 ↑ 8.90	82.30 ↑ 10.80
+ Feat.	89.58	79.82	72.66
+ Ensemble †	94.24 ↑ 4.66	91.10 ↑ 11.28	85.72 ↑ 13.06
+ Ensemble + Self-Training ‡	93.90 ↑ 4.32	90.54 ↑ 10.72	84.74 ↑ 12.08
+ BN	90.10	81.86	74.44
+ Ensemble †	94.65 ↑ 4.55	92.06 ↑ 10.20	87.28 ↑ 12.84
+ Ensemble + Self-Training ‡	94.41 ↑ 4.31	90.84 ↑ 8.98	85.34 ↑ 10.90
+ Adaptive NN (0.3)	94.83	91.00	87.02
+ Ensemble †	95.18 ↑ 0.35	92.78 ↑ 1.78	88.36 ↑ 1.34
+ Ensemble + Self-Training ‡	94.99 ↑ 0.16	92.40 ↑ 1.40	87.92 ↑ 0.90
+ SK	95.03	92.60	88.23
+ Ensemble †	95.07 ↑ 0.04	92.72 ↑ 0.12	88.34 ↑ 0.11
+ Ensemble + Self-Training ‡	95.06 ↑ 0.03	92.54 ↓ 0.06	88.19 ↓ 0.04
+ TEMI w/ CE.	95.16	93.18	88.80
+ Ensemble †	95.18 ↑ 0.02	93.20 ↑ 0.02	88.84 ↑ 0.04
+ Ensemble + Self-Training ‡	95.20 ↑ 0.04	93.16 ↓ 0.02	88.80 =

To observe the performance changes introduced by the cluster ensembling and self-training steps at different accuracy levels, we apply ensembling on the outputs of multi-head classifiers trained during the incremental enhancement experiments in Section 6.4.1. We evaluate cluster ensembling results under two different scenarios.

First, we assume that the test images are available in advance, allowing us to combine the clustering outputs from each classifier head on the test set. Second, we assume that the test images are not available beforehand, making it impossible to ensemble clustering outputs of the test set directly. In this scenario, we perform inference on the training set samples and use the cluster indexes assigned to each sample as pseudo-labels to apply self-training on this new set. The resulting model is then used as the classification model to predict the classes in the validation set. We perform ensembling on the outputs of all the classifier heads in both scenarios.

Cluster ensembling results under these two scenarios are presented in Table 6.11. We observe that while the improvement is substantial for models with lower accuracy, the gap between the baseline and the ensembling narrows for higher accuracy models. We hypothesize that this observation is due to: *(i)* the error types between different classifier heads becoming more similar as the enhancements improve all the heads and bring them closer in performance, and *(ii)* the performance reaching a saturation point as it improves with the enhancements.

6.4.3 Comparison with the State-of-the-art

We demonstrate the effectiveness of our proposed framework for fully unsupervised image classification/clustering by comparing it to the state-of-the-art methods. Using the training details provided in Section 6.4.1.8, we report the mean and standard deviation of the multi-head classifier outputs, as well as the metrics for the best performing model, and ensembling results for the two aforementioned scenarios.

First, we show our results on widely studied CIFAR10, CIFAR20 and STL10 datasets for the unsupervised image classification/clustering. As shown in Table 6.12, UCLS achieves state-of-the-art results, with clustering accuracies of 99.3% on CIFAR10, 74.2% on CIFAR20, and 99.6% on STL10 datasets.

Table 6.12: **Comparison with the State-of-the-art on Small-Scale Datasets.** We report the mean and standard deviation of 5 independent runs with different seeds as our stage-1 results. We also report the best results from stage-1. †corresponds to ensembling on the validation set clustering outputs, while ‡corresponds to ensembling on the training set clustering outputs first and then using self-training on the pseudo-labels obtained from the training set for inference. * indicates methods that utilize validation split during training. UCLS provide significant improvements over the previous methods and achieve state-of-the-art results in unsupervised image classification/clustering problem on CIFAR10, CIFAR20 and STL10 datasets. We use DINOv2 ViT-Large model [20] for the CIFAR10 and CIFAR20 datasets and DINOv2 ViT-Base [20] model for the STL10 dataset. Best results are boldfaced.

Datasets	CIFAR10			CIFAR20			STL10		
Methods	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)
DCCM [80]	49.6	62.3	40.8	28.5	32.7	17.3	37.6	48.2	26.2
DeepCluster [9]	-	37.4	-	-	18.9	-	-	33.4	-
PICA [42]	59.1	69.6	51.2	31	33.7	17.1	61.1	71.3	53.1
GCC [90]	76.4	85.6	72.8	47.2	47.2	30.5	68.4	78.8	63.1
NNM [19]	74.8	84.3	70.9	48.4	47.7	31.6	69.4	80.8	65
PCL [49]	80.2	87.4	76.6	52.8	52.6	36.3	71.8	41.0	67.0
SCAN [74]	79.7	88.3	77.2	48.6	50.7	33.3	69.8	80.9	64.6
SCAN + RUC [61]	-	90.1	-	-	54.5	-	-	86.6	-
SPICE [55]	86.5	92.6	85.2	56.7	53.8	38.7	87.2	93.8	87.0
ProPos* [43]	88.6	94.3	88.4	60.6	61.4	45.1	75.8	86.7	73.7
TCL [51]	81.9	88.7	78.0	-	-	-	79.9	86.8	75.7
TSP [92]	88.0	94.0	87.5	61.4	55.6	43.3	95.8	97.9	95.6
CoKe [63]	76.6	85.7	73.2	49.1	49.7	33.5	-	-	-
SeCu [62]	86.1	93.0	85.7	55.1	55.2	39.7	73.3	83.6	69.3
TEMI DINO ViT-B/16 [1]	88.6±0.05	94.5±0.03	88.5±0.08	65.4±0.45	63.2±0.38	48.9±0.21	96.5±0.13	98.5±0.04	96.8±0.09
TEMI MSN ViT-L/16 [1]	82.9±0.16	90.0±0.14	80.7±0.22	59.8±0.04	57.8±0.42	42.5±0.08	93.6±1.10	96.7±0.89	93.0±1.74
HUME [30]	-	88.4	77.6	-	55.5	37.7	-	90.8	81.2
TURTLE DINOv2 ViT-g/14 [31]	-	99.3	-	-	-	-	-	72.3	-
UCLS stage-1 (Ours)	97.93±0.07	99.27±0.04	98.38±0.08	74.09±0.83	67.71±1.03	55.83±1.09	98.90±0.06	99.57±0.04	99.05±0.08
UCLS stage-1 / Best (Ours)	98.03	**99.32**	**98.50**	75.38	69.26	57.50	99.00	**99.64**	**99.20**
UCLS Ensemble (Ours)†	98.03	99.31	98.48	76.61	73.67	59.67	98.98	99.60	99.12
UCLS Ensemble + Self-Training (Ours) ‡	**98.23**	99.20	97.72	**76.49**	**74.23**	**60.09**	**99.06**	99.58	98.88

Next, we present our results on CIFAR100 dataset, ImageNet dataset and its subsets, namely ImageNet-50, ImageNet-100, ImageNet-200, Tiny-ImageNet and ImageNet-1000. Tables 6.13 and 6.14 shows state-of-the-art results on all ImageNet versions and CIFAR100, achieving a 70.3% on ImageNet and being the first work to surpass 70% mark on ImageNet dataset in fully unsupervised image classification problem. While excluding the scenario-1, where results of validation set are ensembled, UCLS gains absolute increases of % on CIFAR100, % on ImageNet-50, % on ImageNet-100, % on ImageNet-200, % on Tiny-ImageNet, and % on ImageNet-1000.

Table 6.13: **Comparison with the State-of-the-art on ImageNet Subsets.** We report the mean and standard deviation of 5 independent runs with different seeds as our stage-1 results. We also report the best results from stage-1. †corresponds to ensembling on the validation set clustering outputs, while ‡corresponds to ensembling on the training set clustering outputs first and then using self-training on the pseudo-labels obtained from the training set for inference. UCLS provide significant improvements over the previous methods on ImageNet subsets and achieve state-of-the-art results in unsupervised image classification/clustering problem on ImageNet-50, ImageNet-100 and ImageNet-200 datasets with 12.1%, 10.5% and 10.6% absolute improvements, respectively. We use DINOv2 ViT-Large model [20] for our experiments. Best results are boldfaced.

Datasets	ImageNet-50			ImageNet-100			ImageNet-200		
Methods	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)
SCAN [74]	82.2	76.8	66.1	80.8	68.9	57.6	77.2	58.1	47.0
ProPos [43]	82.8	-	69.1	83.5	-	63.5	80.6	-	53.8
TEMI MSN ViT-L/16 [1]	88.14±0.55	84.87±1.16	76.46±1.17	88.53±0.56	82.86±0.73	74.08±1.20	86.65±0.32	77.96±0.71	66.70±0.71
UCLS stage-1 (Ours)	96.81±0.11	97.06±0.10	94.13±0.18	95.44±0.09	93.42±0.26	89.34±0.33	93.54±0.09	88.64±0.26	82.58±0.29
UCLS stage-1 / Best (Ours)	**96.95**	**97.16**	**94.33**	95.54	93.62	89.67	93.37	88.79	82.40
UCLS Ensemble (Ours)†	96.87	97.12	94.24	95.56	93.68	89.71	93.36	88.67	82.29
UCLS Ensemble + Self-Training (Ours) ‡	96.44	96.72	93.48	**95.55**	**93.66**	**89.67**	**93.43**	**89.00**	**82.58**

Finally, achieve a new state-of-the-art for fully unsupervised settings on Food101 dataset with a % clustering accuracy improvement as shown in Table 6.15. Overall, UCLS achieves state-of-the-art on ten image classification benchmark datasets in fully unsupervised settings.

6.4.4 Ablation Studies

We further investigate the effect of different components to the final performance of the multi-head classifier models. Specifically, we conduct experiments on the effects of adaptive nearest neighbor selection with different distance thresholds, the effect of number of classifier heads, and the effect of our enhancements under different loss function to the final performance. We train all the models including the components proposed to improve multi-head classifiers, while using 200 epochs.

Table 6.14: **Comparison with the State-of-the-art on CIFAR100, Tiny-ImageNet and ImageNet.** We report the mean and standard deviation of 5 independent runs with different seeds as our stage-1 results. We also report the best results from stage-1. †corresponds to ensembling on the validation set clustering outputs, while ‡corresponds to ensembling on the training set clustering outputs first and then using self-training on the pseudo-labels obtained from the training set for inference. UCLS sets new state-of-the-art results in unsupervised image classification/clustering problem on all datasets with 1.0%, 54.1% and 1.2% absolute improvements for CIFAR100, Tiny-ImageNet and ImageNet, respectively. We use DINOv2 ViT-Large model [20] for all experiments. Best results are boldfaced. § results are taken from SeCu [62] paper.

Datasets	CIFAR100			Tiny-ImageNet			ImageNet		
Methods	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)
DCCM [80]	-	-	-	22.4	10.8	3.8	-	-	-
SCAN [74]	-	-	-	-	-	-	72.0	39.9	27.5
ProPos [43]	-	-	-	40.5	25.6	14.3	-	-	-
SPICE [55]	-	-	-	44.9	30.5	16.1	-	-	-
TCL [51]	52.9	53.1	35.7	-	-	-	-	-	-
TSP [92]	61.4±1.4	55.6±2.5	43.3±1.8	-	-	-	-	-	-
CoKe§ [63]	-	-	-	-	-	-	76.2	47.6	35.6
SeCu [62]	65.2	51.3	37.1	-	-	-	79.4	53.5	41.9
TEMI DINO ViT-B/16 [1]	76.9±0.45	67.1±1.30	53.3±1.02	-	-	-	82.5	61.56±0.28	48.4
MIM-Refiner [2]	-	-	-	-	-	-	86.3	67.4	40.5
TURTLE DINOv2 ViT-g/14 [31]	-	87.1	-	-	-	-	-	69.1	-
UCLS stage-1 (Ours)	90.60±0.05	87.49±0.24	80.08±0.21	89.93±0.10	84.34±0.21	74.55±0.32	87.73±0.03	70.23±0.16	59.45±0.14
UCLS stage-1 / Best (Ours)	90.62	87.76	**80.32**	**90.03**	**84.62**	**75.00**	**87.77**	70.34	**59.67**
UCLS Ensemble (Ours)†	90.97	88.40	80.75	90.11	84.93	75.18	87.84	70.92	60.05
UCLS Ensemble + Self-Training (Ours) ‡	**90.65**	**88.14**	80.28	89.37	84.35	73.89	87.55	**70.36**	58.64

6.4.4.1 Adaptive Distance Threshold

We investigate the effects of an adaptive distance threshold by training multi-head classifiers on the ImageNet-100 dataset using 10 different equally spaced thresholds, ranging from 0.1 to 1.0, where a threshold of 1.0 corresponds to using only the top-50 nearest neighbors. Table 6.16 demonstrates that a very low threshold of 0.1 significantly outperforms using only the top-50 neighbors, highlighting the effectiveness of the DINOv2 features. Overall mean and standard deviation values show that a distance threshold of 0.3 maximizes the performance across all classifier heads compared to other thresholds. Consequently, we set a distance threshold of 0.3 as the default for our UCLS framework.

Table 6.15: **Comparison with the State-of-the-art on Food101 Dataset.** We report the mean and standard deviation of 5 independent runs with different seeds as our stage-1 results. We also report the best results from stage-1. †corresponds to ensembling on the validation set clustering outputs, while ‡corresponds to ensembling on the training set clustering outputs first and then using self-training on the pseudo-labels obtained from the training set for inference. UCLS provide an absolute 2.7% accuracy improvement over TURTLE [31] method on fully unsupervised settings. We use DINOv2 ViT-Large model [20] for our experiments. Best results are boldfaced.

Dataset	Food101		
Methods	NMI(%)	ACC(%)	ARI(%)
TURTLE DINOv2 ViT-g/14 [31]	-	78.9	-
UCLS stage-1 (Ours)	80.54±0.33	85.67±0.10	71.03±0.21
UCLS stage-1 / Best (Ours)	85.82	80.89	71.26
UCLS Ensemble (Ours)†	85.76	81.01	71.39
UCLS Ensemble + Self-Training (Ours) ‡	86.32	81.60	72.29

6.4.4.2 Number of Classifier Heads

Multi-head classifiers are utilized during the computation of training loss by assigning a low weight to training sample that are likely noisy nearest neighbors. We also use all the classifier heads during the cluster ensembling step to improve the performance of the unsupervised classifiers. To evaluate the effects of number of classifier heads on the multi-head classifier performance, we train several classifier models with number of heads ranging from 10 to 80. As shown in Table 6.17, interestingly, classifier with 10 heads performs best on single-head performance, while it drops behind for the ensembling results. Ensembling benefits from the increased number of heads as there are more possibilities for different types of errors which is known to improve ensembling performance [34]. However, we also observed 10-head classifiers train ∼5x faster than the 50-head classifiers, providing a strong alternative in limited resources.

Table 6.16: **Ablation Study on Adaptive Nearest Neighbor Selection Distance Threshold.** We analyze the effects of adaptive nearest neighbor selection through distance threshold on the performance of multi-head classifiers. The *Best Head* results correspond to the classifier head with the lowest training loss, while the *Overall* results present the mean and standard deviation across all 50 classification heads. We also provide the average number of neighbors per image that are utilized during training, and the nearest neighbor accuracy at the distance threshold over the ImageNet-100 dataset. We use DINOv2-L/14 features for all the experiments. While the classifier with the 0.1 distance threshold provide the single best-head performance, the threshold of 0.3 better propagates the improvements to all the classifier heads.

	Best Head			Overall			Nearest Neighbor Stats	
Dist. Thr.	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)	Average NN Count	NN ACC(%)
1.0	93.68	87.26	83.22	93.47±0.41	87.15±1.39	82.51±1.33	50.0	-
0.9	94.26	89.06	84.98	93.47±0.49	87.37±1.52	82.67±1.61	71.9	99.77
0.8	93.97	89.00	84.42	93.78±0.39	88.21±1.31	83.60±1.32	216.5	98.06
0.7	94.69	90.56	86.28	94.01±0.43	88.52±1.34	84.16±1.39	374.9	96.43
0.6	94.65	91.20	86.49	94.29±0.36	89.22±1.28	85.05±1.23	511.3	95.67
0.5	95.48	92.62	89.04	94.63±0.32	90.16±1.10	86.12±1.13	643.7	94.62
0.4	95.37	92.20	88.57	95.09±0.22	91.18±0.80	87.40±0.77	782.6	93.20
0.3	95.39	92.34	88.52	95.22±0.20	91.83±0.81	87.92±0.76	940.7	90.85
0.2	95.36	92.30	88.49	95.12±0.15	91.34±0.75	87.41±0.64	1158.7	84.47
0.1	95.06	92.58	88.11	94.74±0.14	90.69±0.75	86.48±0.66	1284.6	79.67

6.4.4.3 Multiple Neighbors Smoothing

Even though nearest neighbor training provides a robust grouping of images for training an unsupervised image classifier, errors in model predictions can degrade performance. This is due to incorrect learning signals in the predicted class distributions and the pseudo-labels provided to the cross-entropy loss term introduced in Section 5.2.7.

To mitigate these errors, we propose smoothing the outputs from nearest neighbors by sampling multiple neighbors per image and averaging their predicted probability distributions. To observe the effects of our smoothing approach, we sample four neighbors per image during training and use the average probability distribution as the teacher output in Equation (5.1.1). We conduct experiments on CIFAR20, CIFAR100,

Table 6.17: **Ablation Study on the Number of Classifier Heads.** We analyze the effect of number of classifier heads to the unsupervised image classification performance. The *Best Head* results correspond to the classifier head with the lowest training loss, while the *Overall* results present the mean and standard deviation across all classification heads. We also provide the cluster ensembling results on the validation set as it is a good approximation of the self-training results based on training set pseudo-labels. We use the ImageNet-100 dataset and DINOv2-L/14 features for the experiments.

	Best Head			Overall			Val. Ensembles		
Classifier Heads	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)
10	95.37	92.54	88.64	95.07±0.14	91.57±0.57	87.56±0.54	95.23	92.42	88.38
20	95.27	92.74	88.54	95.07±0.18	91.59±0.67	87.54±0.65	95.26	92.54	88.35
30	95.22	91.88	88.09	95.16±0.19	91.69±0.75	87.74±0.74	95.50	93.40	89.38
40	95.41	92.14	88.50	95.20±0.21	91.67±0.80	87.80±0.75	95.41	92.74	88.69
50	95.39	92.34	88.52	95.22±0.20	91.83±0.81	87.92±0.76	95.40	93.32	89.19
60	95.48	92.20	88.62	95.17±0.22	91.63±0.82	87.75±0.80	95.63	93.70	89.82
70	95.42	92.08	88.49	95.21±0.20	91.76±0.80	87.87±0.78	95.50	93.42	89.38
80	95.28	92.04	88.12	95.21±0.18	91.77±0.79	87.88±0.71	95.61	93.60	89.72

ImageNet-100, and ImageNet datasets to evaluate the impact of multiple neighbor smoothing.

Table 6.18 shows that multiple-neighbors smoothing provides improvements to the final performance of each dataset, providing absolute accuracy gains of 2.25% on CIFAR20, 0.33% on CIFAR100 and 1.12% on ImageNet; pushing the ImageNet clustering accuracy to 71.46%; while decreasing the performance on ImageNet-100 dataset with an absolute 0.14% accuracy loss.

Multiple neighbors smoothing also moderately increases the training time for all the models due to the increased computation requirements.

6.4.4.4 Effect of Enhancements under Different Loss Functions

We conduct additional experiments using various loss functions to determine whether our improvements are independent of the specific baseline method. To this end, we

Table 6.18: **Effects of Multiple Nearest Neighbors Smoothing.** We report the performance metric comparison of the best classifier heads of models trained with and without multiple nearest neighbors smoothing. Corresponding models share the same random number generation seed and are trained with all the proposed enhancement components.

	w/o Smoothing			w/ Smoothing		
Datasets	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)
CIFAR20	74.72	68.38	56.61	75.46	70.63	58.44
CIFAR100	90.62	87.76	80.32	90.35	88.09	79.87
ImageNet-100	95.33	92.92	88.73	95.13	92.78	88.59
ImageNet	87.77	70.34	59.67	87.66	71.46	59.98

replace the TEMI loss function [1] with the SCAN loss function [74] and incorporate our enhancements into the baseline settings. As shown in Table 6.19, our enhancements consistently improve baseline performance on ImageNet-100 dataset regardless of the loss function applied, achieving comparable results across both configurations despite differences in baseline accuracy. We use a entropy regularization hyperparameter $\lambda = 4$ for the SCAN loss.

Table 6.19: **Effects of Enhancements Under Different Loss Functions.** We report the performance metric comparison of the best classifier heads of models trained with different loss functions, namely TEMI and SCAN losses.

	w/ TEMI Loss			w/ SCAN Loss		
Methods	NMI(%)	ACC(%)	ARI(%)	NMI(%)	ACC(%)	ARI(%)
Baseline (DINOv2)	88.75	79.44	71.50	90.77	84.64	77.89
+ Enhancements	95.16	93.18	88.80	95.06	93.00	88.62

CHAPTER 7

CONCLUSION

This chapter is adopted from our JVCI journal paper [6] and extended for our recent work.

In this thesis, we addressed two main aspects of self-supervised learning; training a backbone model by proposing a self-supervised learning pretext task, and the usage of self-supervised pretrained models on downstream tasks.

As for the initial problem we discussed, we proposed a straightforward yet impactful extension to the instance discrimination pretext task, aiming to address the inherent limitation in localization representations during the image-level learning of CNNs, particularly when spatial information is lost due to the global average pooling operation. To this end, we attached a segmentation branch to the feature extractor before the global pooling operation to work on dense features, and work in parallel with the instance discrimination task to inject additional dense representations through a supplementary segmentation loss. We provide the necessary supervision to the segmentation branch using a self-supervised pseudo-mask generation method. We show that our extension improved the localization capability of both contrastive and non-contrastive learning baselines on PASCAL VOC object detection and semantic segmentation, and COCO object detection and instance segmentation tasks, while also preserving or slightly improving the classification performance using the linear evaluation protocol.

It is also important to acknowledge certain limitations and possible future work associated with our approach. In our experiments, we observed that training SegIns took approximately 30% longer than the baseline methods. This increase in train-

ing time may be attributed to the additional computational load introduced by the class-agnostic segmentation task. Although our method delivers improved localization capabilities, the longer training duration could pose practical constraints in resource-intensive scenarios. As we look ahead, future investigations will focus on exploring techniques to optimize and streamline the training process, making SegIns more efficient without compromising its benefits. Furthermore, we aim to explore more complex segmentation branch designs and replace the class-agnostic segmentation branch with a class-aware one in a self-supervised manner to further improve the supplied localization information and also serve for a wider range of use-cases.

In the second part of this thesis, in order to address the utilization of self-supervised pretrained models on downstream tasks, we introduced an enhanced unsupervised image classification framework by proposing several improvements to multi-head classifier approach. We achieve state-of-the-art results in unsupervised image classification problem on ten image classification benchmarks with big margins. To the best of our knowledge, we are the first to break the 70% barrier on ImageNet dataset in the fully unsupervised image classification task. We also propose using a cluster ensembling method to better leverage the multiple heads in the classifiers and further enhance our results through a self-training step. We analyze each of the components in our framework in detail through incremental analysis and ablation studies.

There are certain limitations to our proposed unsupervised classification framework. While our stage-1 improvements already achieve state-of-the-art results, the cluster ensembling stage, which provides additional performance improvements, can make the framework cumbersome and not directly applicable to other domains such as unsupervised object detection and instance segmentation, which include both classification and localization branches that require end-to-end training. We evaluate potential improvements to simplify the framework and create an end-to-end unsupervised classification pipeline, aiming to reduce the extra effort required during ensembling and self-training. One consideration is using a codebook during training to align cluster assignments between multiple heads of the classifier. This could enable the use of classifier ensembling methods, such as hard voting or soft voting, and eliminate the need for a separate cluster ensembling method. Additionally, there is a gap between the upper bound (based on ground truth nearest neighbor experiments) and the cur-

rent results of the UCLS framework. Better utilization of the nearest neighbor set with possible false positive filtering heuristics can improve the final performance.

REFERENCES

[1] N. Adaloglou, F. Michels, H. Kalisch, and M. Kollmann. Exploring the limits of deep image clustering using pretrained models. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*. BMVA, 2023.

[2] B. Alkin, L. Miklautz, S. Hochreiter, and J. Brandstetter. Mim-refiner: A contrastive learning boost from intermediate pre-trained representations. *arXiv preprint arXiv:2402.10093*, 2024.

[3] M. Assran, M. Caron, I. Misra, P. Bojanowski, F. Bordes, P. Vincent, A. Joulin, M. Rabbat, and N. Ballas. Masked siamese networks for label-efficient learning. In *European Conference on Computer Vision*, pages 456–473. Springer, 2022.

[4] H. Bao, L. Dong, S. Piao, and F. Wei. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

[5] A. Bardes, J. Ponce, and Y. LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[6] M. Baydar and E. Akbas. Segins: A simple extension to instance discrimination task for better localization learning. *Journal of Visual Communication and Image Representation*, 100:104122, 2024.

[7] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014.

[8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[9] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[10] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[11] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[13] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[14] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[15] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.

[16] P. Chrabaszcz, I. Loshchilov, and F. Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

[17] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[18] M. Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020.

[19] Z. Dang, C. Deng, X. Yang, K. Wei, and H. Huang. Nearest neighbor matching for deep clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13693–13702, 2021.

[20] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

[21] M. Dehghani, J. Djolonga, B. Mustafa, P. Padlewski, J. Heek, J. Gilmer, A. P. Steiner, M. Caron, R. Geirhos, I. Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[25] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is

worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[27] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, 27, 2014.

[28] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021.

[29] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[30] A. Gadetsky and M. Brbic. The pursuit of human labeling: a new perspective on unsupervised learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[31] A. Gadetsky, Y. Jiang, and M. Brbić. Let go of your labels with unsupervised transfer. In *International Conference on Machine Learning*, 2024.

[32] C. GE, J. Wang, Z. Tong, S. Chen, Y. Song, and P. Luo. Soft neighbors are positive supporters in contrastive visual representation learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[33] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[34] R. Gontijo-Lopes, Y. Dauphin, and E. D. Cubuk. No one representation to rule them all: Overlapping features of training methods. *arXiv preprint arXiv:2110.12899*, 2021.

[35] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[36] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, and W. T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *International Conference on Learning Representations*, 2022.

[37] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[38] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[39] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[40] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[41] O. J. Hénaff, S. Koppula, J.-B. Alayrac, A. Van den Oord, O. Vinyals, and J. Carreira. Efficient visual pretraining with contrastive detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10086–10096, 2021.

[42] J. Huang, S. Gong, and X. Zhu. Deep semantic clustering by partition confidence maximisation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8849–8858, 2020.

[43] Z. Huang, J. Chen, J. Zhang, and H. Shan. Learning representation for clustering via prototype scattering and positive sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7509–7524, 2022.

[44] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[45] J. Jain, J. Li, M. T. Chiu, A. Hassani, N. Orlov, and H. Shi. Oneformer: One transformer to rule universal image segmentation. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2989–2998, 2023.

[46] M. Ki, Y. Uh, J. Choe, and H. Byun. Contrastive attention maps for self-supervised co-localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2803–2812, 2021.

[47] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[48] J. Li. Prototypical contrastive learning: Pushing the frontiers of unsupervised learning. https://blog.salesforceairesearch.com/prototypical-contrastive-learning-pushing-the-frontiers-of-unsupervised-learning/.

[49] J. Li, P. Zhou, C. Xiong, and S. C. Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.

[50] Y. Li, H. Mao, R. Girshick, and K. He. Exploring plain vision transformer backbones for object detection. In *European conference on computer vision*, pages 280–296. Springer, 2022.

[51] Y. Li, M. Yang, D. Peng, T. Li, J. Huang, and X. Peng. Twin contrastive learning for online clustering. *International Journal of Computer Vision*, 130(9):2205–2221, 2022.

[52] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[53] I. Loshchilov, F. Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5, 2017.

[54] S. Mo, H. Kang, K. Sohn, C.-L. Li, and J. Shin. Object-aware contrastive learning for debiased scene representation. *Advances in Neural Information Processing Systems*, 34:12251–12264, 2021.

[55] C. Niu, H. Shan, and G. Wang. Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*, 31:7264–7278, 2022.

[56] D. Niu, X. Wang, X. Han, L. Lian, R. Herzig, and T. Darrell. Unsupervised universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22744–22754, 2024.

[57] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[58] M. Noroozi, H. Pirsiavash, and P. Favaro. Representation learning by learning to count. In *Proceedings of the IEEE international conference on computer vision*, pages 5898–5906, 2017.

[59] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[60] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[61] S. Park, S. Han, S. Kim, D. Kim, S. Park, S. Hong, and M. Cha. Improving unsupervised image clustering with robust learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12278–12287, 2021.

[62] Q. Qian. Stable cluster discrimination for deep clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16645–16654, 2023.

[63] Q. Qian, Y. Xu, J. Hu, H. Li, and R. Jin. Unsupervised visual representation learning by online constrained k-means. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16640–16649, 2022.

[64] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[65] Y. Ruan, S. Singh, W. Morningstar, A. A. Alemi, S. Ioffe, I. Fischer, and J. V. Dillon. Weighted ensemble self-supervised learning. *arXiv preprint arXiv:2211.09981*, 2022.

[66] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[67] R. R. Selvaraju, K. Desai, J. Johnson, and N. Naik. Casting your model: Learning to localize improves self-supervised representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11058–11067, 2021.

[68] H. S. Seong, W. Moon, S. Lee, and J.-P. Heo. Leveraging hidden positives for unsupervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19540–19549, 2023.

[69] O. Siméoni, G. Puy, H. V. Vo, S. Roburin, S. Gidaris, A. Bursuc, P. Pérez, R. Marlet, and J. Ponce. Localizing objects with self-supervised transformers and no labels. In *Proceedings of the British Machine Vision Conference (BMVC)*, November 2021.

[70] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

[71] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 240–248. Springer, 2017.

[72] Y. Tian, X. Chen, and S. Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278. PMLR, 2021.

[73] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer, 2020.

[74] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer, 2020.

[75] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, and L. Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10052–10062, 2021.

[76] X. Wang, R. Girdhar, S. X. Yu, and I. Misra. Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3124–3134, 2023.

[77] X. Wang, Z. Yu, S. De Mello, J. Kautz, A. Anandkumar, C. Shen, and J. M. Alvarez. Freesolo: Learning to segment objects without annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14176–14186, June 2022.

[78] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021.

[79] Y. Wang, X. Shen, Y. Yuan, Y. Du, M. Li, S. X. Hu, J. L. Crowley, and D. Vaufreydaz. Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *IEEE transactions on pattern analysis and machine intelligence*, 2023.

[80] J. Wu, K. Long, F. Wang, C. Qian, C. Li, Z. Lin, and H. Zha. Deep comprehensive correlation mining for image clustering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8150–8159, 2019.

[81] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[82] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[83] T. Xiao, C. J. Reed, X. Wang, K. Keutzer, and T. Darrell. Region similarity representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10539–10548, 2021.

[84] T. Xiao, X. Wang, A. A. Efros, and T. Darrell. What should not be contrastive in contrastive learning. In *International Conference on Learning Representations*, 2021.

[85] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16684–16693, 2021.

[86] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9653–9663, 2022.

[87] A. YM., R. C., and V. A. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2020.

[88] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022.

[89] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 649–666. Springer, 2016.

[90] H. Zhong, J. Wu, C. Chen, J. Huang, M. Deng, L. Nie, Z. Lin, and X.-S. Hua. Graph contrastive clustering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9224–9233, 2021.

[91] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong. ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations (ICLR)*, 2022.

[92] X. Zhou and N. L. Zhang. Deep clustering with features from self-supervised pretraining. *arXiv preprint arXiv:2207.13364*, 2022.

CURRICULUM VITAE

Name Surname: Melih Baydar

SUMMARY

I am a Computer Vision Engineer at Pensa Systems as of July 2024. My research interest lies in Computer Vision and I am working on self-supervised representation learning for classification and localization tasks. Other fundamental problems I have worked on include Optical Character Recognition (OCR), Image Classification and Human Pose Estimation.

EDUCATION

- MSc in Computer Engineering, Bilkent University, 2014 - 2017
- BSc in Computer Engineering, Hacettepe University, 2009 - 2014

WORK EXPERIENCE

- Computer Vision Engineer, Pensa Systems, 2021 - Present
- Computer Vision Engineer, Huawei Technologies, 2018 - 2021
- Research and Teaching Assistant, Bilkent University, 2014 - 2017

PUBLICATIONS

- Baydar, Melih, and Emre Akbas. "SegIns: A simple extension to instance discrimination task for better localization learning." Journal of Visual Communication and Image Representation 100 (2024): 104122.
- Örenbaş, Halit, Anıl Oymagil, and Melih Baydar. "Efficient Scene Text Detection in Images with Network Pruning and Knowledge Distillation." 2021 29th Signal Processing and Communications Applications Conference (SIU). IEEE, 2021.
- Baydar, Melih. Enhancing Feature Selection with Contextual Relatedness Filtering Using Wikipedia. MS thesis. Bilkent Universitesi (Turkey), 2017.