Master's Thesis

Improving Unsupervised Instance detection using dino features

Ajesh Krishnan Kizhakke Menakath

Advisers: Silvio Galesso

Examiner: Prof. Dr. Thomas Brox
Prof. Dr. Matthias Teschner

Albert-Ludwigs-University Freiburg
Faculty of Engineering
Department of Computer Science
Chair of Pattern Recognition and Image Processing

August 23^{th}, 2024

Writing period

23. 02. 2023 – 23. 08. 2023

Examiner

Prof. Dr. Thomas Brox

Second Examiner

Prof. Dr. Matthias Teschner

Advisers

Silvio Galesso

Declaration

I hereby declare, that I am the sole author and composer of my thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work.
I hereby also declare, that my Thesis has not been prepared for another examination or assignment, either wholly or excerpts thereof.

Place, Date	Signature

Abstract

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Instance detection is a critical task in computer vision, with applications ranging from autonomous driving to medical imaging. Despite significant advancements, unsupervised instance detection remains a challenging problem due to the lack of annotated data. This thesis aims to improve unsupervised instance detection by analyzing the problems with the current state-of-the-art unsupervised instance detection and segmentation methods.

In the field of Computer Vision, a significant shift has occurred from traditional Convolutional Neural Networks (CNNs) to the transformative capabilities of Transformers, as exemplified by the groundbreaking Vision Transformer (ViT) paper by Dosovitskiy et al. (2020) [1]. Subsequently, the concept of utilizing deep ViT features as dense visual descriptors was introduced by Amir et al. (2021) [2], highlighting the strong semantic information these descriptors provide about instances within an image. Currently, the state-of-the-art method for unsupervised instance segmentation, CutLER, as proposed by Wang et al. (2023) [3], employs DINO features [4] as visual descriptors to effectively identify instances within images.

However, challenges such as the grouping of nearby instances, failure to identify complex background patterns, and the omission of small instances persist even in the current state-of-the-art methods. This thesis aims to investigate these failure cases, mainly based on the CutLER [3] baseline and propose methods to enhance the performance of unsupervised instance detection and segmentation tasks.

1.2 Overview of CutLER

In the field of unsupervised object detection and instance segmentation, the recent work by Xudong Wang et al. introduces Cut-and-LEaRn (CutLER) [3], a novel approach that significantly advances the state-of-the-art. CutLER leverages the capabilities of self-supervised models to identify objects without human supervision, and it enhances this capability to train a high-performance localization model without any

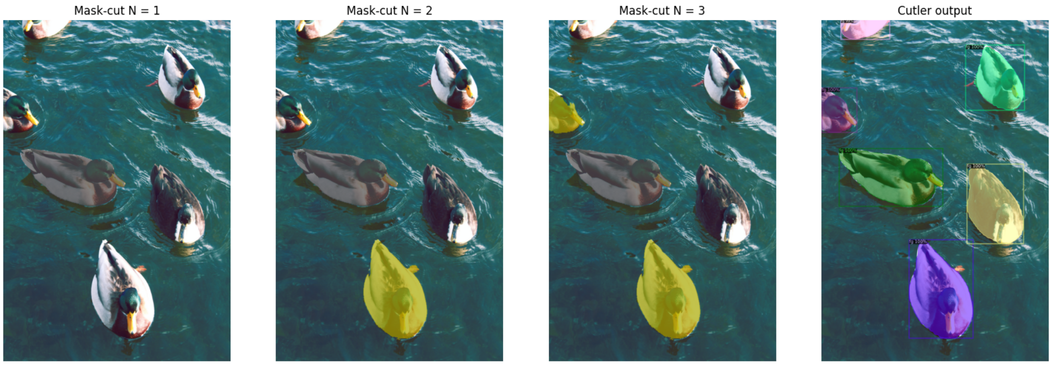

Figure 1: Comparison of maskcut and CutLER outputs. Figure illustrates the outputs of CutLER and maskcut with different N(Number of mask generated) values.

labeled data. The methodology begins with the MaskCut approach(inspired from [5]), which generates coarse masks for multiple objects within an image. Subsequently, a detector is trained on these masks using a robust loss function. Performance is further improved through a self-training process where the model is iteratively trained on its own predictions. This approach not only simplifies the training process but also proves to be compatible with various detection architectures and capable of detecting multiple objects simultaneously. Figure 1 shows original image, masks generated by Maskcut for values N=1, 2 and 3(Like in the original paper, we use N=3 in all of our experiments) and the final CutLER output.

The effectiveness of CutLER is demonstrated through extensive evaluations across diverse image domains, including video frames, paintings, sketches, and complex scenes. Notably, CutLER, utilizing a ResNet50 backbone, achieves a substantial performance increase, more than doubling the detection accuracy on 10 out of 11 benchmarks compared to the previous state-of-the-art method, FreeSOLO, which uses a ResNet101 backbone. Specifically, CutLER improves the average precision (AP50) by over 2.7 times across these benchmarks. This demonstrates CutLER's potential not only as a zero-shot unsupervised detector but also as an efficient low-shot detector, marking a significant step forward in unsupervised object detection and instance segmentation.

1.3 Contribution and Key Insights

This study focuses on the shortcomings of CutLER and the methods to minimize them. Our main contributions are:

1. **Analysis on the influence of overlapping instances**: We compare the performance of the model when trained with and without overlapping instances and conclude that training without overlapping instances results in better instance discrimination.

2. **Refining maskcut masks**: On top of self training, ee refine mask-cut masks using CutLER output to remove noisy masks and retraining to yield better performance.

1.4 Outline

- **Chapter 3:**
- **Chapter 2:**
- **Chapter 4:**
- **Chapter 5:**

2 Background

2.1 Vision Transformer

Vision Transformer(ViT) was introduced by Dosovitskiy et al. [1] to overcome the limitations of Convolutional Neural Networks(CNNs) for image recognition. The model applies Transformer architecture to image recognition tasks by treating image patches as sequences of tokens, akin to words in NLP. The highlight of the paper is reusing the transformer encoder from the revolutionary work Vaswani et al. [22] and adapting to use on images using patch tokenization and positional encoding.

As CutLER uses the features from a self-supervised ViT to generate masks, it is crucial to understand the basics architecture and working of ViT to get a complete picture of the feature generation process. Figure 2 shows the complete architecture of ViT. We are going to go into the main parts of the architecture for a better understanding of the process.

2.1.1 Patch Tokens and Positional Encoding

As each input is an image, unlike sequence of words or tokens in [22], the image is divided in fixed size patches(16x16 or 8x8) and each patch is treated as a token and each token is embedded into a fixed-dimensional vector using a learned embedding layer.

$$ {z^0}_i = z_i + p_i \tag{1} $$

For each token, instead of using sinusoidal position encodings [22] to retain information about the position of tokens in the sequence, a learnable position embedding is added as shown in the Eq. 1, where $p_i \in \mathbb{R}^D$ is the learnable position embedding for patch i.

$$ Z = [z_{class}; {z^0}_1; {z^0}_2; ...; {z^0}_N] \tag{2} $$

Apart from [22], ViT [1] introduces a special classification token z_{class} which is

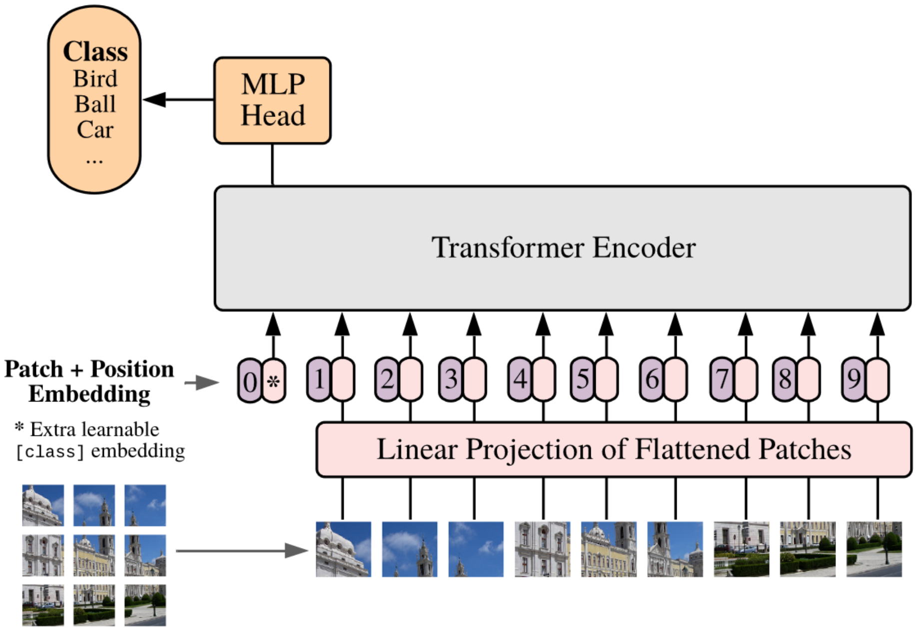

Figure 2: ViT Architecture. ViT architecture from [1].

prepended to the sequence of patch embeddings. This token aggregates information from all patches and is used for the final classification task. The final encoding look like Eq. 2.

2.1.2 Transformer Encoder

The sequence of patch embeddings, augmented with positional information, is processed by the Transformer encoder. The encoder consists of multiple layers, each comprising Multi-Head Self-Attention (MSA) and Multi-Layer Perceptrons (MLPs), with Layer Normalization (LN) and residual connections. A weighted average [23] of individual attention outputs constitute the final output. Figure 3 illustrates the architecture of transformer encoder. We briefly look into each part.

Multi-Head Self-Attention (MSA)

Self-attention allows the model to weigh the importance of different patches relative to each other.

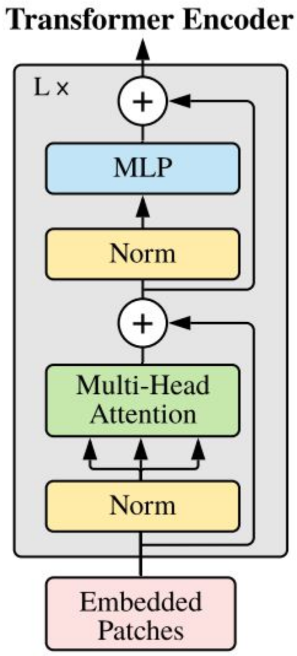

Figure 3: Transformer Encoder Architecture. Illustration of the Transformer encoder architecture in ViT [1].

$$\text{Queries } Q = zW_i^Q \tag{3a}$$
$$\text{Keys } K = zW_i^K \tag{3b}$$
$$\text{Values } V = zW_i^V \tag{3c}$$

Given that d_k is the dimensionality of the key, query, and value vectors and $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{D \times d_k}$ are learnable weight matrices, query, key, and value are computed as given in Eq. 3.

For each attention head i,

$$head_i = \text{Attention}\ (Q_i, K_i, V_i) = \text{softmax}\ \left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \tag{4}$$

The outputs from all heads are concatenated and linearly transformed. Given $W^O \in \mathbb{R}^{h - d_k \times D}$:

$$MSA(z) = \text{Concat}\ (head_i, head_2 ..., head_h) W^O \tag{5}$$

Layer Normalization and Residual Connections

Each layer in the Transformer encoder includes Layer Normalization (LN) and residual (skip) connections

$$z' = \mathrm{MSA}(\mathrm{LN}(z)) + z \tag{6}$$

$$z'' = \mathrm{MLP}(\mathrm{LN}(z')) + z' \tag{7}$$

The Multi-Layer Perceptron (MLP) usually consists of two linear transformations with a GELU non-linearity in between. Assuming W_1 ans W_2 are learnable weight matrices:

$$\mathrm{MLP}(x) = W_2(\mathrm{GELU}(W_1 x)) \tag{8}$$

Output Layer

The final output of the classification token is passed through a linear layer to produce the classification logits. Given C is the number of classes and $W_{class} \in \mathbb{R}^{C \times D}$:

$$\mathrm{logits} = W_{class} \cdot z''_{class} \tag{9}$$

The linear layer projects the final representation of the classification token into the space of class labels.

2.2 DINO

The self-supervised model DINO, introduced by Caron, Mathilde, et al. [4], achieves remarkable performance that rivals many state-of-the-art Convolutional Networks (CNN) trained with supervision. DINO stands out for its ability to extract features that reveal clear information about semantic segmentation and scene layout within images. This capability distinguishes DINO from supervised Vision Transformers (ViTs) and ConvNets, underscoring its potential for sophisticated computer vision tasks without relying on annotated data.

As we will be using DINO features for producing the pseudo masks in CutLER [3], we need a basic understanding of DINO architecture and training.

2.2.1 Knowledge distillation

Knowledge distillation plays a crucial role in training a student model to mimic the behavior and representations learned by a larger teacher model, both of which are

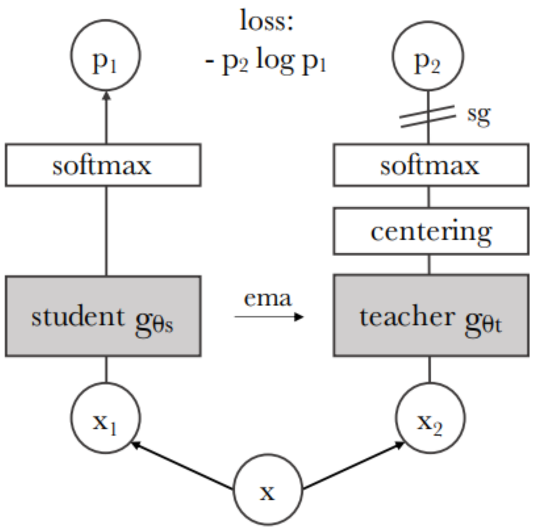

Figure 4: Architecture of DINO Illustration provided in [4].

ViTs.

Initially, the teacher model is typically a ViT that is pre-trained on a large dataset using self-supervised learning techniques. The teacher captures rich, generalized features from the data. The student model is a smaller ViT that aims to replicate the teacher's performance but with fewer parameters, making it computationally lighter and potentially faster during inference.

Momentum Encoder for Teacher

Instead of using the teacher model directly, DINO employs a momentum encoder mechanism for stability and improved generalization. This means that the parameters of the teacher model are updated using a moving average of the student model's parameters, rather than directly during training.

$$\theta_t \leftarrow m\theta_t + (1 - m)\theta_s \tag{10}$$

The teacher model's parameters are updated using a momentum update rule as given in Eq. 10. Where θ_t are the parameters of the teacher model, θ_s are the parameters of the student model, and $m$is a momentum parameter (typically close to 1) that controls the rate of updating.

2.2.2 Training Process

DINO uses different augmentations of the same image to create multiple views. These augmented views are passed through both the teacher and student models. Outputs from both models are projected into a lower-dimensional space using projection heads. Outputs from both models are projected into a lower-dimensional space using projection heads. The optimization objective is to minimize the cross-entropy loss between the predicted probability distributions of the teacher and student models. Assume $P_t(x)$ and $P_s(x)$ represent the probability distributions predicted by the teacher and student models, respectively. The training process is illustrated in 4

$$\min_{\theta_s} \mathcal{H}(P_t(x), P_s(x)) \tag{11}$$

The cross-entropy loss is computed between the softened distributions of the teacher and student models across all augmented views as given in Eq. 11.

2.3 CutLER

CutLER [3] introduces a novel approach to address the challenges of object detection and instance segmentation in an unsupervised learning framework. By integrating Copy-Paste [24] augmentation and a contrastive learning framework, the method not only circumvents the need for labeled data but also achieves state-of-the-art results in object detection and instance segmentation. The complete process is illustrated in Fig. 5.

2.3.1 Maskcut

Maskcut considers image segmentation problem as a graph partitioning task [25]. The inspiration of Maskcut comes from TokenCut [5], which constructs a fully connected undirected graph by representing each patch as a node.

$$(D - W)x = \lambda Dx \tag{12}$$

Edges connect every pair of nodes, with weights W_{ij} reflecting the similarity between the connected nodes and reduces the cost of dividing the graph into two sub-graphs, or a bipartition, by solving a generalized eigenvalue problem as given in Eq. 12. In Tokencut [5], the authors determine the similarity weight W_{ij} in NCut based on the similarity of patches in the DINO feature space. Following recent

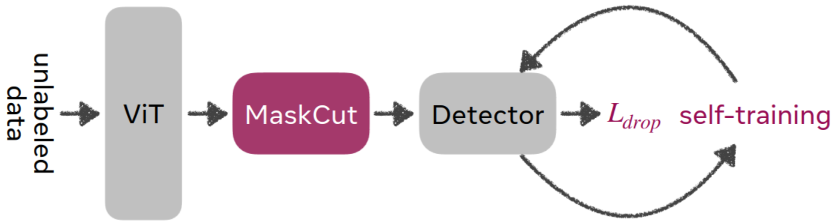

Figure 5: CutLER overview The flow consist of MaskCut for extracting coarse masks from the features of a self-supervised ViT. Following this, a detector utilizing a loss dropping strategy designed to be resilient against objects that MaskCut may overlook is used. Additionally, the model undergoes further enhancement through multiple rounds of self-training. Illustration taken from [3]

.

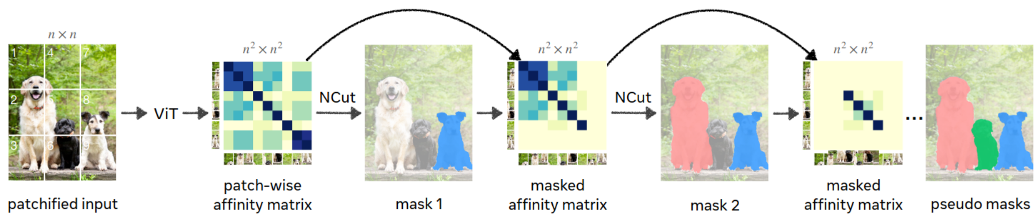

Figure 6: Maskcut works on the patch-wise similarity matrix for the image using a self-supervised DINO [4] model feature. N=3 defines the number of times NCut [25] is repeated on the background. In this case, 3 instances will be discovered per image in each step. Illustration taken from [3]

.

methods [26, 27, 28], they specifically employ the cosine similarity of 'key' features from the final attention layer of the DINO-pretrained model, represented as:

$$W_{ij} = \frac{K_i \cdot K_j}{\|K_i\|_2 \|K_j\|_2} \tag{13}$$

where K_i denotes the 'key' feature of patch i. They then solve Eq. 12 to find the second smallest eigenvector x. The main drawback of the approach is only using the smallest eigenvector resulting in finding only one instance in the image. Maskcut overcomes this drawback and finds more instances by iteratively applying the same process in the background N times as given in Fig. 6. The figure shows the flow of Maskcut algorithm for N=3(Defines the number of times NCut [25] is repeated. In this case, 3 instances will be discovered per image) Building on the work of [28, 29], a patch-wise similarity matrix for the image using features from a self-supervised DINO model [4] is created. Normalized Cuts [25] is applied to this matrix to obtain a single foreground object mask for the image. Subsequently, this foreground mask is used to mask out the affinity matrix values and repeat the process. This iterative approach enables MaskCut to identify multiple object masks within a single image.

Maskcut uses two conditions to improve the performance. 1. An object centric prior [30] is used to filter out backgrounds. ie, if the foreground contains more than 2 out of 4 corners, foreground and background are switched. 2. From the intuition that foreground patches are more prominent than background ones [29, 31], we assert that foreground mask should contain the patch corresponding to the maximum absolute value in the second smallest eigenvector. If condition 1 is not satisfied and current foreground contains two corners, background and foreground are switched.

2.3.2 Droploss

A standard detection loss penalizes predicted regions r_i that do not overlap with the 'ground truth'. Since the ground truth masks from MaskCut may miss some instances, the standard loss does not allow the detector to identify new instances not labeled in the ground truth. To address this, the author proposes ignoring the loss for predicted regions r_i with minimal overlap with the ground truth.

$$L_{\text{drop}}(r_i) = \mathbb{1}(\text{IoU}_i^{\max} > \tau^{\text{IoU}}) L_{\text{vanilla}}(r_i) \tag{14}$$

Specifically, during training, the loss is dropped for any predicted region r_i that has a maximum overlap of τ^{IoU} with any ground truth instance as given in Eq. 14 where $\text{IoU}_i^{\max}$ denotes the maximum IoU with all ground truth for r_i, and L_{vanilla}

Figure 7: Illustration of Copy-Paste augmentation from [24]

.

refers to the standard loss function for detectors. L_{drop} avoids penalizing the model for detecting objects missed in the 'ground truth', thus encouraging the exploration of different image regions.

2.3.3 Copy-Paste Augmentation

Copy-Paste Augmentation [24] is a data augmentation technique used to enhance the diversity of training datasets by artificially creating new training samples. This method involves copying objects from one image and pasting them onto another, thereby generating new training examples with diverse object placements and backgrounds as illustrated in Fig. 7. The pasted objects can be resized, rotated, or otherwise manipulated to fit the new context. Lastly, the ground-truth annotations are also adjusted accordingly

In Cutler, instead of using the standard copy-paste augmentation, where masks are rescaled with a factor between 0.8 and 1.25, masks are randomly downsampled with a scalar uniformly sampled between 0.3 and 1.0. This approach enables Cutler to recognize even smaller instances in the image effectively.

2.3.4 Training

The training process is divided into two stages: initial training followed by self-training, as depicted in Fig.5. CutLER is agnostic regarding the choice of detector, allowing the use of any preferred detector. However, based on the experiments detailed in the paper, Cascade Mask R-CNN[34] yields better results compared to

Mask R-CNN [33].

First, pseudo-ground truth masks are generated using MaskCut for all images in Imagenet [32] training set. The detector with a ResNet-50 [35] backbone is trained using these pseudo-ground truth masks for 160K iterations with Copy-Paste augmentations and DropLoss.

To further improve the model performance, several self-training loops are also carried out. The CutLER mask predictions of each image with confidence score > 0.7 generated using the final model of last training phase and corresponding MaskCut masks which doesn't overlap more than 50% with the predicted masks together forms the pseudo-ground truth masks for the first round of self-training. The same process is repeated for the following self-training rounds except that instead of MaskCut masks, pseudo-ground truth masks of the previous stage is used to compare with predicted masks. Each self training round is consist of 80K rounds and does not use DropLoss as we obtain comparatively good quality masks in the first round it self. The detailed information about the training are explained later in Section 5.2.

2.3.5 Mask refinement in CutLER

Before each self-training loop, 30 masks per image are generated for the entire Imagenet dataset using the latest trained model. Of these, masks are filtered based on the confidence score. In the paper, the masks with confidence score greater than $0.7 - 0.05 * i$ on the i th iteration are kept and the rest are rejected. These filtered CutLER masks are compared with the corresponding Maskcut masks(in the first self-training round) or the pseudo-ground truth masks from the last round for each image. If the IoU between CutLER mask and Maskcut mask is less than 0.5, the corresponding Maskcut mask is added along with CutLER masks and this constitutes the pseudo-ground truth for the self-training loop.

The intuition is to retain masks from previous pseudo-ground truths that do not significantly overlap (i.e., overlap less than 0.5) with the current predictions. This strategy allows CutLER to explore new regions of the image that have not been thoroughly examined in previous iterations. However, a challenge with this approach is that it may perpetuate the inclusion of noisy masks in the ground truth during each self-training loop. Our approach seeks to address this issue by implementing a more refined method for removing the noisy background masks from the MaskCut masks and to improve the quality of the masks iteratively.

3 Related Work

3.1 Self-supervised feature learning

Self-supervised feature learning is a crucial process that identifies patterns within extensive unlabeled datasets without the need for human-annotated labels. Plenty of research has been done in this field in the recent years. Several methods have been proposed, each with unique mechanisms and varying levels of success.

3.1.1 Contrastive learning

Contrastive learning has gained significant attention for its effectiveness in self-supervised feature learning. One of the seminal works in this area is SimCLR [6], It employs a simple yet robust framework that leverages data augmentations to create positive pairs from the same image and negative pairs from different images. The model uses a contrastive loss to distinguish between these pairs, learning robust representations in the process. On the other hand, MoCo (Momentum Contrast) [7] introduces a dynamic dictionary with a momentum encoder. This approach allows the model to maintain a queue of negative samples, effectively reducing memory requirements and improving scalability. Nevertheless, it still requires a substantial number of negative samples to function optimally and necessitates careful tuning of the momentum parameter to balance stability and learning efficiency.

3.1.2 Clustering-based feature learning

Clustering-based feature learning approaches automatically uncover the natural groupings of data within the latent representation space. This clustering process helps in understanding the inherent structure of the data by grouping similar data points together based on learned features. Agglomerative Clustering with Self-supervision [8] can capture multi-scale structures and found to be effective for diverse datasets. But found to be computationally expensive and needs careful tuning of the self-supervised task. SwAV [9] combines clustering with contrastive learning by swapping assignments between different augmented views of the image.

This method is efficient in terms of computational resources and achieves state-of-the-art performance on several benchmarks. But it is sensitive to the choice of hyperparameters.

3.1.3 Distillation-based methods

Distillation-based methods have also shown considerable promise in self-supervised learning. BYOL (Bootstrap Your Own Latent) [10] introduces a teacher-student network where the student learns to predict the teacher's representations. Remarkably, BYOL achieves this without using negative samples, simplifying the training process and reducing computational demands. However, it is sensitive to the choice of data augmentations and network architecture, and there is a potential risk of model collapse if not properly tuned. DINO [4], extends the self-distillation approach to Vision Transformers [1]. DINO captures global image representations effectively without relying on negative samples. It shows strong performance on object detection and segmentation tasks, showcasing the potential of transformers in self-supervised learning.

Unlike these unsupervised representation learning efforts, our research revolves around CutLER [3], which focuses on automatically identifying natural pixel groupings and detecting instances within each image.

3.2 Unsupervised object detection and instance segmentation

If we consider the recent methods for unsupervised object detection semantic segmentation, most of them leverage on self-supervised Vision Transformer(ViT) [1] features. In DINO [4] it is observed that the underlying semantic segmentation of images can be extracted using the saliency maps from the ViT.

The quality of this segmentation is superior to the existing methods if the image contains only is one instance. The superiority of DINO features to separate foreground and background has been affirmed by later works [11]. Building on this observation, both LOST [12] and TokenCut [5] utilize DINO features to segment a single salient object from each image. These methods capitalize on the strength of DINO to construct a graph from the features of image patches. Unlike TokenCut and DINO, which can only detect one instance, LOST is capable of finding multiple instances within an image. But it can't be used as a pre-trained model for down stream tasks. But CutLER [3] not only can detect multiple instances, the model can be further used as a pretrained model for label-efficient and fully-supervised learning.

FreeSOLO [13] and the follow up work Exemplar-FreeSOLO [14] (with its addition of a randomly drawn pool of exemplars used in a contrastive learning loss) generates coarse segmentation masks with low guality and refines it further through self training similar to CutLER. But the poor quality of the coarse maps is a major draw back of this method, where as CutLER masks made by the MaskCut [3, 5] algorithm are usually better in quality and quantity than the initial masks used by MaskDistill [15] and [13]. Even though Maskdistill produces similar quality masks compared to MaskCut, as it only produces one class agnostic mask per image and MaskCut produces N fixed number of masks per image to use as pseudo labels, MaskCut weighs over Maskdistill in quantity.

As CutLER dominates in most cases, including producing better pseudo ground truth masks, ability to detect multiple instances, compatibility with various detection architectures, usable as pretrained model for supervised detection, our work would mostly focus on studying and improving the performance of CutLER.

3.3 Semi-supervised object detection and instance segmentation

Semi-supervised learning leverages both unlabeled data and a fraction of labeled data to improve the performance of models. In the context of object detection and semantic segmentation, several recent works have explored various techniques to enhance these tasks.

Most of the recent works in this domain includes a student-teacher model as in Adaptive Teacher [16], Unbiased Teacher [17] and Soft Teacher [18]. These methods slightly differ in the type of augmentations(strong, weak and hybrid augmentation) used and the method of action on the pseudo labels. For instance, Unbiased teacher addresses bias in pseudo-labeling by using a teacher-student model. The teacher generates pseudo-labels which are then used to train the student. The process iterates with the student eventually replacing the teacher. Where as the Soft teacher leverages both hard and soft pseudo-labels. The method uses a teacher model to generate soft labels (probabilistic outputs) for unlabeled data, which are used to train a student model.

Unlike relying on augmentations like most recent semi-supervised works, we tried to extract instances using keypoint correspondences taking inspiration from the Superglue [19] paper. But the main challenge was instead of working with images taken two different perspectives, we are dealing with entirely different pair of images

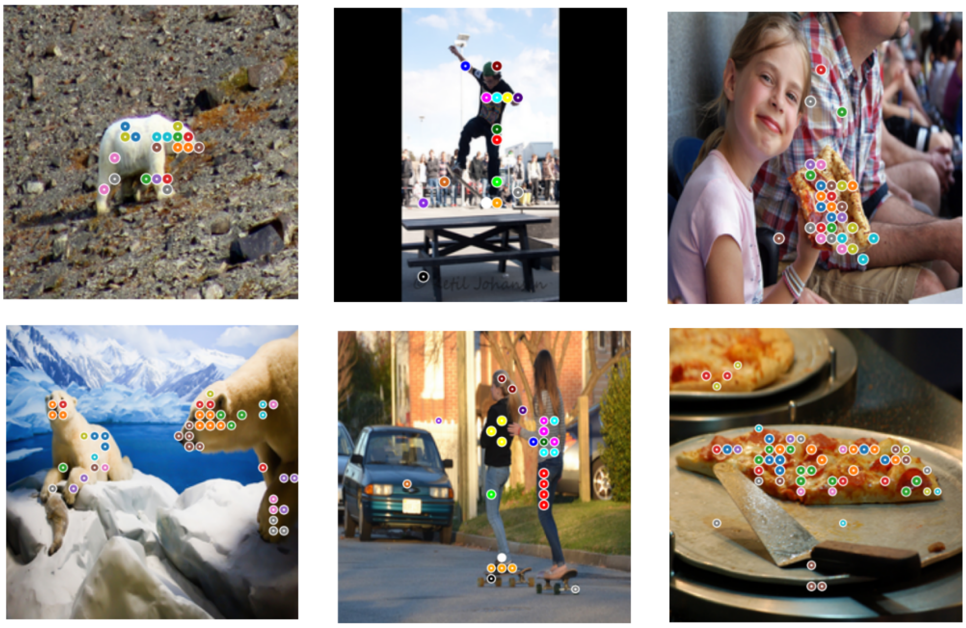

Figure 8: Keypoint correspondecences using relaxed best buddies. Prototype images at the top and query images at the bottom. Same colored points represent similar features

with similar instances.

Figure 8 shows the keypoint correspondences between the prototype and query images. The correspondence is calculated by applying the relaxed best buddy algorithm on the descriptors corresponding to the foreground part of the prototype and query images, which is selected by applying a threshold on the saliency map. We relaxed the original best buddy algorithm [20] to extract more correspondence points enough to form a graph to perform graph-cut similar to [5, 19]. We explored our options to geometrically separate the instances using the semantic information in the correspondences. But as we were unable to find a promising approach or inspiration from our research. For instance, Integer programming for multidimensional assignment problem [21] has a strict initial graph specification which restrict us to adapt it to our problem.

By considering the limitations of implementing a semi-supervised method using keypoint correspondences, we focus mostly on improving the unsupervised instance detection and segmentation by diving into the current state-of-the-art method CutLER [3]; exploring it's limitations and ways to improve them.

4 Approach

In the field of unsupervised instance detection and segmentation, CutLER [3] gives a strong performance by exploiting a object-centric prior by training on ImageNet [32], as most images contain a single object in the center of the frame. Due to its strong instance discrimination abilities, CutLER is the current state-of-the-art method for this task.

In this chapter, we are exploring the limitation of CutLER, looking deeper into the special cases where CutLER fails such as overlapping instances, complex backgrounds etc. We also analyze the change in performance when the model is trained without overlapping instances, as a main reason for CutLER's superior performance is it's object-centric prior [11]. Using the gathered information from the analysis, we introduce a hypothesis to refine Maskcut masks using CutLER outputs to train the model from scratch to obtain a better evaluation score across a variety of datasets.

4.1 Limitations of CutLER

Even though CutLER is the state-of-the-art model for unsupervised instance detection and segmentation, it still has several drawbacks. We go through some of them in this section.

4.1.1 Overlapping instances

Identifying instances using an unsupervised instance detection or segmentation method presents significant challenges, especially when instances are closely positioned or overlapping in an image. In such scenarios, the algorithm must discern subtle differences in texture, color, and shape without the benefit of labeled training data. Overlapping objects often blend together, making it difficult for the model to accurately segment and differentiate them as distinct entities. This lack of explicit supervision complicates the model's ability to learn and generalize the spatial relationships and boundaries between objects. Moreover, unsupervised methods typically rely on clustering and feature extraction techniques, which may not be robust enough to handle the complexities of overlapping instances, leading to errors in segmentation and misidentification.

The problem is not exclusive to CutLER, but most of the existing methods [11, 31, 37] also address this issue. Solving this problem requires, for instance, the learning of an instance-variant representation, which is a challenging task.

4.1.2 Noisy pseudo-ground truths

Noisy pseudo-ground truths generated by MaskCut can significantly hinder the performance of CutLER. These pseudo-ground truths often contain inaccuracies due to imperfect initial segmentation, which can arise from factors like complex backgrounds, occlusions, and variations in object appearance. Such noise can mislead the model, causing it to learn incorrect features and boundaries, ultimately degrading the quality of instance detection and segmentation. The presence of noisy masks can lead to overfitting on incorrect patterns or failure to generalize properly across different instances. To mitigate these issues, techniques like iterative refinement, robust loss functions, and the incorporation of consistency constraints have been proposed. Tang et al. [38] and Wang et al. [39] explore these approaches, highlighting the importance of addressing noise in pseudo-ground truths to enhance the robustness of unsupervised instance detection and segmentation methods.

In CutLER, a self-training loop is implemented to iteratively refine the pseudo ground truth masks. Even though it improves the performance of the model till 3 self training loops, it can further improved by making some changes in the mask filtration process, which will be explained in the next section.

4.2 Impact of Overlapping Instances

When instances are closely positioned or overlapping in an image, it often makes the model difficult to accurately segment and differentiate them as distinct entities [40]. But as CutLER mainly benefits from it's object-centric prior from training on Imagenet [11], we introduce a hypothesis that CutLER when trained with images without overlapping instances of Imagenet might perform better than training with all images.

For this, we make use of ground truth bounding box annotations provided by Imagenet and remove the images with an overlap(IoU) of $\tau > 10\%$ and $\tau > 25\%$ are removed from the training set. The model's performance is compared with model trained using all image. For completeness, we also train a model using images with overlapping instances only.

Through this approach, we expect to observe an improvement by using less training data. But using this method in unsupervised fashion is rather challenging. Due to the grouping of nearby instances, the process of filtering images with overlapping instance is extremely challenging.

4.3 Mask Refinement

Generating initial pseudo-ground truth masks using a pre-trained model or some heuristic methods may contain errors or inaccuracies. Hence, iteratively refining the pseudo-ground truth masks(self- training) is essential for improving the performance of the model. Iterative refinement helps in progressively reducing this noise, leading to cleaner and more reliable labels [41]. Popular refinement methods incorporate strategies like thresholding, where only high-confidence predictions are used for retraining, or use ensemble methods to combine predictions from multiple models for more reliable masks. CutLER also uses this method to filer the masks.

4.3.1 Proposed mask refinement

Inspired by our analysis in Section 4.2, which emphasizes quality over quantity, we introduce an improved approach for mask refinement. Noting that the current mask refinement method in CutLER tends to include noisy masks in its pseudo ground truths, we propose to enhance the process by removing ambiguous masks from the ground truth instead of retaining them. This adjustment aims to improve the overall quality and reliability of the pseudo ground truths, leading to better model performance.

Instead of adding masks from Maskcut with IoU < 0.5 to the pseudo-ground truth, we refine the CutLER predictions by removing masks that have IoU < 0.5. This approach effectively eliminates potentially noisy masks from the pseudo-ground truth, ensuring higher quality and more accurate mask predictions.

Even though the method might improve the precision, as we are limiting the range of exploration by removing more masks, we expect the recall to decrease by a small factor. However, our experiments indicate that this change is negligibly small. Detailed results and analysis can be found in the Experiments section.

5 Experiments and Results

In this chapter, we present a comprehensive analysis of the experiments conducted to compare our proposed method with the baseline approach, CutLER. We thoroughly evaluate both models across a diverse set of datasets to assess their performance. Additionally, we delve into the impact of training images containing overlapping instances, providing detailed quantitative results to illustrate how these images affect model's performance.

5.1 Datasets

For a fair comparison, we use the same datasets as the baseline for both training and evaluation. All models are trained on the ImageNet dataset and evaluated on a diverse set of benchmark datasets, including COCO, Pascal VOC, and KITTI. This ensures more consistent and comprehensive assessment of performance across different types of datasets.

5.1.1 ImageNet

The ImageNet dataset is a large-scale visual database designed for use in visual object recognition research. Developed by researchers at Princeton and Stanford, it contains more than 10,000,000 labeled images depicting 10,000+ object categories. Each image in the dataset is hand-labeled by humans, making it a valuable resource for training and benchmarking deep learning models in computer vision.

We generate MaskCut annotations for all images on the subset of ImageNet containing the 1000 categories and 1.3 million images(ImageNet-1K), which serve as the pseudo-ground truth for our experiments. Both the baseline method (CutLER) and our proposed method are trained on the ImageNet dataset. However, in the proposed method, a fraction of images are excluded during the mask-refinement process(Images with no annotaions are removed).

5.1.2 COCO

The COCO (Common Objects in Context) dataset [42] is a widely-used benchmark in the field of computer vision, designed to spur advancements in object detection, segmentation, and captioning. It contains over 200,000 images with more than 80 object categories, annotated with precise bounding boxes, segmentation masks, and context-related captions.

We use the validation set of the COCO 2017 split, which contains 5,000 images, for evaluating the models. Both bounding box coordinates and segmentation annotations are utilized as ground truths for evaluation.

5.1.3 PASCAL VOC

The PASCAL VOC 2012 dataset is a widely recognized benchmark in visual object recognition, comprising 11,530 images across 20 categories with comprehensive annotations for object detection, classification, and segmentation tasks. For evaluation, we use both the training and test images from the PASCAL VOC dataset and detailed segmentation annotations as ground truths.

5.1.4 KITTI

The KITTI dataset [43] is a prominent benchmark for evaluating performance in autonomous driving and computer vision tasks, including object detection, tracking, and scene flow. It features high-resolution images captured from a stereo camera setup mounted on a moving vehicle, encompassing a variety of urban and rural driving scenarios.

Although the KITTI dataset offers rich annotations, including 3D object labels and depth information, our evaluation focuses solely on bounding boxes. Since the dataset does not provide segmentation annotations, we utilize only the bounding box data to evaluate 7521 images from KITTI's trainval split.

5.1.5 Comic and Watercolor

In addition to real-world image datasets, we also incorporate art datasets, such as Comic and Watercolor [44], to evaluate the model's generalization capabilities across diverse visual styles. Since these datasets lack segmentation annotations, we use only the bounding box data for evaluation, as in our approach for the KITTI dataset.

5.2 Implementation Details

Our implementation largely follows the baseline approach; however, it is important to note a key difference in our setup. While in the baseline paper experiments use a batch size of 16, we utilize batch sizes of 4 and 8 due to resource constraints. To ensure a fair comparison, we also train the baseline model from scratch using these same batch sizes of 4 and 8.

5.2.1 Training data

Only the images from ImageNet dataset(1.3 Million images) are used for the training(including self-training). We do not use any supervised pretrained models or labels for training baseline or the proposed method. However, the bounding box annotations are used to analyze the impact of images with overlapping instances in section <REF:>

5.2.2 MaskCut

We apply MaskCut with N=3, generating three masks per image through repeated N-Cut operations, on images resized to 480×480 pixels to create pseudo-ground truths. The value of N is optimal at 3 for generating best quality masks for ImageNet dataset [3]. The patch-wise affinity matrix generated from the key descriptors of the ViT-B/8 DINO model is used to perform the N-Cut operation. Additionally, we employ Conditional Random Fields (CRF) to refine the masks and extract their bounding boxes.

5.2.3 Detector

Although CutLER is designed to be agnostic to the choice of object detector, we chose to use Cascade Mask R-CNN for all our experiments. This decision is based on the baseline paper's findings, which demonstrated that Cascade Mask R-CNN outperforms Mask R-CNN. We train the detector on ImageNet with MaskCut pseudo masks and bounding boxes for 160K iterations with a batch size of 8.

The copy-paste augmentation is also used during the training process to improve robustness of object detection and segmentation models by exposing them to a wider range of scenarios and object contexts. In order to detect small objects, instead of vanilla copy-paste augmentation, masks are randomly downsampled with a scalar uniformly sampled between 0.3 and 1.0.

We optimize the Detector using SGD for 160K iterations with a learning rate of 0.005, weight decay of 5×10^{-5} and a momentum of 0.9. Training follows a learning rate schedule which decreases it by 5 after 80K iterations.

5.2.4 Self Training

In each stage, along with CutLER mask predictions with confidence score > 0.7 generated using the model from previous stage, Maskcut masks which have IoU < 0.5 with the CutLER prediction masks together make the pseudo ground truth masks for that stage. The detector is then optimized using SGD with a learning rate of 0.01 over 80,000 iterations. We do not employ DropLoss during these self-training phases.

5.2.5 Resources

Generating MaskCut annotations for all images in ImageNet is supposed to most time consuming part. But we used the pre-generated MaskCut annotations to save time.

Initial training on ImageNet with batch size 8 spans over 160K iterations on four NVIDIA rtx-2080 gpus takes around 1 day 18 hours and self-training of 80K iteration takes around 21 hours. The Training using filtered MaskCut masks generated by our method takes 4 hours less (1 day 14 hrs) as around 130K images are dropped in the mask filtration step for not having any pseudo-ground truth masks.

5.3 Experiments

5.3.1 Exploring Impact of Overlapping Instances

5.3.2 Proposed Method

5.4 Result of bs-8 CutLER training

[3]

5.4.1 Choice of batch size

In the baseline paper, experiments were conducted using a batch size of 16. Due to resource constraints, we performed our experiments with batch sizes of 4 and 8. As shown in Table <Table Number>, we observe a slight improvement in performance

	COCO		KITTI		VOC		Comic		Watercolor	
	AP	AP50	AP	AP50	AP	AP50	AP	AP50	AP	AP50
Train										
Baseline	11.17	20.12	4.79	10.27	19.98	36.26	**11.80**	**28.39**	**14.67**	**35.60**
Ours	**11.47**	**20.81**	**6.28**	**13.48**	**20.24**	**36.56**	10.81	26.50	14.00	35.27
Self-train-r1										
Baseline	11.70	21.15	6.60	14.19	19.56	36.81	**11.05**	**27.53**	12.92	33.45
Ours	**11.94**	**21.65**	**8.21**	**18.49**	**20.24**	**37.95**	10.80	27.40	**15.37**	**37.74**
Self-train-r2										
Baseline	11.02	20.32	6.73	15.02	18.06	35.09	9.90	25.36	13.59	34.31
Ours	11.02	20.32	6.73	15.02	18.06	35.09	9.90	25.36	13.59	34.31

Table 1: AP and AP50 for Training and Self-Training (Batch size 8)

		COCO		KITTI		VOC		Comic		Watercolor	
		AP	AP50	AP	AP50	AP	AP50	AP	AP50	AP	AP5
Train	Baseline	11.17	20.12	4.79	10.27	19.98	36.26	**11.80**	**28.39**	**14.67**	**35.6**
	Ours	**11.47**	**20.81**	**6.28**	**13.48**	**20.24**	**36.56**	10.81	26.50	14.00	35.2
Self-train-r1	Baseline	11.70	21.15	6.60	14.19	19.56	36.81	**11.05**	**27.53**	12.92	33.4
	Ours	**11.94**	**21.65**	**8.21**	**18.49**	**20.24**	**37.95**	10.80	27.40	**15.37**	**37.7**
Self-train-r2	Baseline	11.02	20.32	6.73	15.02	18.06	35.09	9.90	25.36	13.59	34.3
	Ours	11.02	20.32	6.73	15.02	18.06	35.09	9.90	25.36	13.59	34.3

Table 2: AP and AP50 for Training and Self-Training (Batch size 8)

train	COCO	KITTI	VOC	Comic	Watercolor
Baseline	11.17,20.12	4.79,10.27	19.98,36.26	11.80,28.39	14.67,35.6
Ours	**11.47,20.81**	**6.28,13.48**	**20.24,36.56**	10.81,26.50	14.00,35.27

Table 3: AP and AP50

self-train-r1	COCO	KITTI	VOC	Comic	Watercolor
Baseline	11.7,21.15	6.60,14.19	19.56,36.81	11.05,27.53	12.92,33.45
Ours	**11.94,21.65**	**8.21,18.49**	**20.24,37.95**	10.80,27.4	**15.37,37.74**

Table 4: AP and AP50

self-train-r2	COCO	KITTI	VOC	Comic	Watercolor
Baseline	11.02,20.32	6.73,15.02	18.06,35.09	9.90,25.36	13.59,34.31
Ours	-	-	-	-	-

Table 5: AP and AP50

Eval dataset	Imagenet-all	Imagenet-wo-ol	Imagenet-maskcut-filter
Coco Eval	10.35, 19.15	11.52, 20.67	11.44, 21.31
Coco Eval w/o overlapping inst.	23.45, 38.11	24.77, 39.46	24.64, 40.64
Coco Eval only overlapping inst.	7.25, 15.11	8.52, 16.8	8.32, 17.28

Table 6: AP and AP50 of evaluation(box) on COCO Eval datasets on models trained on imagenet for 90K iterations

Eval dataset	Imagenet-all	Imagenet-wo-ol	Imagenet-maskcut-filter
Coco Eval	7.87, 16.05	8.80, 17.47	9.1, 18.15
Coco Eval w/o overlapping inst.	19.19, 35.19	20.13, 36.36	21.15, 38.31
Coco Eval only overlapping inst.	5.12, 11.55	6.18, 13.27	9.78, 18.92

Table 7: AP and AP50 of evaluation(segm) on COCO Eval datasets on models trained on imagenet for 90K iterations

with a higher batch size. Our experiments also indicate that batch size influences the relative improvement achieved through self-training.

For batch sizes of 4 and 8, we observed that in self-training round 2, the performance of both the baseline and our method decreased instead of improving. This contrasts with the results in the baseline paper with batch size 16, where performance continued to improve until round 2.

6 Conclusion and Future Work

7 Acknowledgments

I extend my sincere gratitude to Dr. Thomas Brox for his invaluable contribution as the examiner of this research thesis. I also wish to thank Dr. Matthias Teschner for graciously agreeing to be my second examiner and for showing a keen interest in my work.

My heartfelt thanks go to my supervisor, Silvio Galesso, whose unwavering support and mentorship have been the cornerstone of this research. Silvio's continuous encouragement, constructive feedback, and expert guidance have been invaluable throughout the entire research process.

I would also like to acknowledge the support of my friends, whose encouragement and understanding have been a source of motivation. Additionally, I am grateful to ChatGPT for supporting me with the difficult writing process by covering my underwhelming English language skills.

The completion of this thesis has been a significant personal achievement, and I am genuinely appreciative of the contributions and support from these individuals and resources. Their influence has been instrumental in reaching this milestone.

Bibliography

[1] 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.

[2] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel, "Deep vit features as dense visual descriptors," *arXiv preprint arXiv:2112.05814*, vol. 2, no. 3, p. 4, 2021.

[3] X. Wang, R. Girdhar, S. X. Yu, and I. Misra, "Cut and learn for unsupervised object detection and instance segmentation," 2023.

[4] 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*, pp. 9650–9660, 2021.

[5] 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," *arXiv preprint arXiv:2209.00383*, 2022.

[6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020.

[7] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2020.

[8] Y. M. Asano, C. Rupprecht, and A. Vedaldi, "Self-labelling via simultaneous clustering and representation learning," 2020.

[9] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," 2021.

[10] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised learning," 2020.

[11] P. Engstler, L. Melas-Kyriazi, C. Rupprecht, and I. Laina, "Understanding self-supervised features for learning unsupervised instance segmentation," 2023.

[12] 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," 2021.

[13] X. Wang, Z. Yu, S. D. Mello, J. Kautz, A. Anandkumar, C. Shen, and J. M. Alvarez, "Freesolo: Learning to segment objects without annotations," 2022.

[14] T. Ishtiak, Q. En, and Y. Guo, "Exemplar-freesolo: Enhancing unsupervised instance segmentation with exemplars," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15424–15433, June 2023.

[15] W. V. Gansbeke, S. Vandenhende, and L. V. Gool, "Discovering object masks with transformers for unsupervised semantic segmentation," 2022.

[16] Y.-J. Li, X. Dai, C.-Y. Ma, Y.-C. Liu, K. Chen, B. Wu, Z. He, K. Kitani, and P. Vajda, "Cross-domain adaptive teacher for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7581–7590, June 2022.

[17] Y.-C. Liu, C.-Y. Ma, Z. He, C.-W. Kuo, K. Chen, P. Zhang, B. Wu, Z. Kira, and P. Vajda, "Unbiased teacher for semi-supervised object detection," 2021.

[18] M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, and Z. Liu, "End-to-end semi-supervised object detection with soft teacher," 2021.

[19] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," 2020.

[20] K. Aberman, J. Liao, M. Shi, D. Lischinski, B. Chen, and D. Cohen-Or, "Neural best-buddies: sparse cross-domain correspondence," *ACM Transactions on Graphics*, vol. 37, p. 1–14, July 2018.

[21] J. L. Walteros, C. Vogiatzis, E. L. Pasiliao, and P. M. Pardalos, "Integer programming models for the multidimensional assignment problem with star costs," *European Journal of Operational Research*, vol. 235, no. 3, pp. 553–568, 2014.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[23] L. Weng, "The transformer family," *lilianweng.github.io*, Apr 2020.

[24] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," 2021.

[25] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[26] 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," 2021.

[27] W. V. Gansbeke, S. Vandenhende, and L. V. Gool, "Discovering object masks with transformers for unsupervised semantic segmentation," 2022.

[28] 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," 2023.

[29] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," 2021.

[30] S. Maji, N. K. Vishnoi, and J. Malik, "Biased normalized cuts," in *CVPR 2011*, pp. 2057–2064, 2011.

[31] H. V. Vo, P. Pérez, and J. Ponce, "Toward unsupervised, multi-object discovery in large-scale image collections," in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 779–795, Springer International Publishing, 2020.

[32] 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*, pp. 248–255, Ieee, 2009.

[33] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2018.

[34] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," 2019.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[36] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

[37] Y. Wang, X. Shen, S. X. Hu, Y. Yuan, J. L. Crowley, and D. Vaufreydaz, "Self-supervised transformers for unsupervised object discovery using normalized cut," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14543–14553, June 2022.

[38] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers, "Normalized cut loss for weakly-supervised cnn segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[39] A. Ziegler and Y. M. Asano, "Self-supervised learning of object parts for semantic segmentation," 2022.

[40] S. Kara, H. Ammar, F. Chabot, and Q.-C. Pham, "Image segmentation-based unsupervised multiple objects discovery," 2022.

[41] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," 2020.

[42] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015.

[43] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[44] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa, "Cross-domain weakly-supervised object detection through progressive domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.