Zero-shot Unsupervised Transfer Instance Segmentation

Gyungin Shin\textsuperscript{1,2} \quad Samuel Albanie\textsuperscript{2} \quad Weidi Xie\textsuperscript{1,3}

\textsuperscript{1}Visual Geometry Group, University of Oxford, UK
\textsuperscript{2}Cambridge Applied Machine Learning Lab, University of Cambridge, UK
\textsuperscript{3}Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, China

https://www.robots.ox.ac.uk/~vgg/research/zutis

Abstract

Segmentation is a core computer vision competency, with applications spanning a broad range of scientifically and economically valuable domains. To date, however, the prohibitive cost of annotation has limited the deployment of flexible segmentation models. In this work, we propose Zero-shot Unsupervised Transfer Instance Segmentation (ZUTIS), a framework that aims to meet this challenge. The key strengths of ZUTIS are: (i) no requirement for instance-level or pixel-level annotations; (ii) an ability of zero-shot transfer, i.e., no assumption on access to a target data distribution; (iii) a unified framework for semantic and instance segmentations with solid performance on both tasks compared to state-of-the-art unsupervised methods. While comparing to previous work, we show ZUTIS achieves a gain of 2.2 mask AP on COCO-20K and 14.5 mIoU on ImageNet-S with 919 categories for instance and semantic segmentations, respectively. Code will be made publicly available.\textsuperscript{1}

1. Introduction

In computer vision, the task of segmentation aims to group pixels within an image into coherent, meaningful regions. Accurate segmentation unlocks a host of applications such as tumour assessment in medical images [2], land cover estimation [51] for logistical planning, scene segmentation for autonomous driving [11], to name a few. The central challenge that limits the deployment of such applications is the high cost of obtaining large, accurate collections of pixel-level annotations to train appropriate segmenters. For example, it was reported that when constructing the Cityscapes dataset, it took 90 minutes to fully annotate and validate individual images [11].

To overcome this challenge, a range of unsupervised segmentation methods have been developed that forgo pixel-level supervision [10, 19, 28, 53, 56, 66]. One particularly promising line of work has focused on a setting known as unsupervised semantic segmentation with language-image pretraining (USSLIP) [48, 49], which leverages a vision-language foundation model [3] that has been pretrained on a large corpus of internet-sourced image-text pairs. USSLIP methods exhibit strong segmentation performance, category label flexibility and zero-shot transfer—the ability to perform well on a downstream task without access to images from the target distribution. However, while USSLIP methods enable semantic segmentation, no such method developed to date possesses the ability to differentiate between instances within a semantic category, a key functionality for many fine-grained applications.

In this paper, we consider a challenging task, Zero-shot Unsupervised Transfer Instance Segmentation, i.e., to segment the instances present in an image and infer its seman-
tic classes without relying on manual supervision or access to a target dataset. To tackle such challenge, we start from the recent progress in USSLIP [49], retrieving images for given concept with a pretrained visual language model (e.g., CLIP), then generating pseudo-masks for the collected images with an unsupervised saliency detector. To take one step further, we extend the prior USSLIP architectures with two critical abilities, namely, instance-level segmentation and generalisation to unseen categories. In specific, we couple a query-based Transformer [54] decoder, which generates instance mask proposals, with an image encoder, which is trained to output dense features (i.e., patch tokens) aligned with text embeddings for a set of concepts from a frozen CLIP [44] text encoder. Notably, the design of the proposed approach allows to do inference for both semantic and instance segmentations with strong performance compared to prior state-of-the-art approaches.

In summary, our contributions are three-fold: (i) We introduce a challenging task, namely, zero-shot unsupervised transfer instance segmentation, which aims to segment object instances without human supervision or access to a target data distribution; (ii) We propose a simple yet effective framework, termed ZUTIS, that goes beyond prior USSLIP approaches, and enables to concurrently perform instance segmentation in addition to semantic segmentation; (iii) We show that ZUTIS performs favourably against state-of-the-art methods on standard unsupervised segmentation benchmarks (e.g., COCO [35], ImageNet-S [15]) by a large margin in both zero-shot transfer and unsupervised domain adaptation settings.

2. Related work

Our work relates to diverse themes in the literature including zero-shot semantic/instance segmentation, unsupervised semantic segmentation (with and without language-image pretraining), unsupervised object segmentation, class-agnostic unsupervised instance segmentation, universal architectures, and open-vocabulary segmentation.

Zero-shot semantic/instance segmentation with language pre-training. Zero-shot semantic/instance segmentation aims to generalise to unseen categories after training for seen categories with ground-truth annotations. The dominant approach exploits the relationships between category label embeddings produced by a language model (e.g., word2vec [38] or GloVe [43]) [4, 18, 23, 31, 34, 42, 63, 68, 69] to facilitate generalisation. More recently, there has been growing interest in leveraging the joint image-text embedding space produced by a pretrained vision-language model (e.g., CLIP) to enable dense predictions [12, 33, 37, 45, 64]. In a similar vein, we build our approach on a pretrained vision-language model to enable generalisation to novel categories, but with two key differences. First, we do not assume access to a target data distribution, a setting termed zero-shot transfer in [44]. Second, we do not use any manual annotations during training. Note that the “annotation free” variant of MaskCLIP [70] enables semantic segmentation in a similar regime in which neither access to the target distribution nor ground-truth annotations are available. We compare our method to MaskCLIP on semantic segmentation tasks in Sec. 4.

Unsupervised semantic segmentation. A rich line of work has considered the problem of unsupervised semantic segmentation, creatively constructing learning signals from proxy tasks [10, 19, 28, 41, 52, 56, 66]. One practical challenge associated with these approaches is their reliance on a matching stage to enable deployment (typically performed with Hungarian matching [32] on pixel-level segmentation annotations) that establishes correspondences between segments and category names. By contrast, ZUTIS requires no access to pixel-level supervision during either training or inference. Furthermore, unlike the above, ZUTIS is capable of instance-level predictions as well as semantic segmentation. We note one exception: MaskDistill [53] also reports on unsupervised instance segmentation in addition to semantic segmentation (also using Hungarian matching to assign categories to predictions). In Sec. 4, we compare ZUTIS with MaskDistill on unsupervised instance segmentation.

Unsupervised semantic segmentation with language-image pretraining (USSLIP). To achieve independence from pixel-level annotations during both training and inference, a recent line of work targeting unsupervised semantic segmentation proposes to leverage a vision-language model (e.g., CLIP [44]) to assign names to categories [48, 49]. To do so, images are curated from an unlabelled image collection using the retrieval abilities of the vision-language model, and then segmented via co-segmentation [48] or salient object detection [49]. However, while these methods avoid pixel-level annotations, they are either fragile (i.e., co-segmentation used in [48]) or rigid (i.e., a new segmenter needs to be retrained from scratch for each new category [49]). Moreover, no USSLIP method to date supports instance segmentation. ZUTIS builds on this line of work, but addresses its limited functionality by enabling instance segmentation, and improves both robustness and flexibility.

Unsupervised object segmentation. Unsupervised object segmentation, also referred to as saliency detection, aims to train a detector to segment prominent object regions in images without human supervision. Traditionally, handcrafted methods have been proposed utilising low-level cues such as centre prior [29], contrast prior [27], and boundary prior [62]. A more recent line of research uses objectness properties emerging from self-supervised features extracted from modern vision architectures [47, 50, 61] (i.e., DINO [6]). In this work, we adopt SelfMask [47] to generate object masks for images that are used as pseudo-masks.
for our training.

**Class-agnostic unsupervised instance segmentation.** Recently, FreeSOLO [59] proposed a self-supervised framework for the class-agnostic instance segmentation task. For this, coarse object masks are first generated by using the object localisation property of self-supervised features (e.g., DenseCL [60]), then a class-agnostic object detector is trained with the initial masks via a self-training scheme [59]. Concurrent work, CutLER [58], follows the similar framework, but with better initial masks produced by proposed MaskCut. Unlike the above, we focus on the conventional class-aware instance segmentation. The classification of each instance mask is made possible as ZUTIS is built on the recent progress in unsupervised semantic segmentation with language-image pretraining (i.e., ReCo [48]).

**Universal architectures.** Recently, universal architectures that deliver multiple object detection/segmentation tasks in a unified manner have gained considerable attention [5, 8, 9, 65]. Similarly, we propose an architecture that can tackle both semantic and instance segmentations with a single architecture with two key differences: (i) ZUTIS is flexible in terms of categories to segment as we use a text encoder as a classifier; (ii) unlike the above which need to train a model from scratch for a different task, ZUTIS requires only a single training for semantic and instance segmentations.

**Open-vocabulary segmentation.** Increasing the number of object categories to be segmented has been explored by utilising class-incremental few-shot learning [25], captions [17], grounded text descriptions [30], as well as annotation transfer [24, 26] and pairwise class balance regularisation [22]. Similarly, we seek to scale the number of classes to be segmented, but without human supervision.

### 3. Method

In this section, we start by introducing the considered problem scenario, namely, zero-shot unsupervised transfer instance segmentation in Sec. 3.1, and describe the core building blocks of our proposed approach in Sec. 3.2, followed by the architecture details for addressing unsupervised semantic and instance segmentation tasks with pretrained language-image models in Sec. 3.3.

#### 3.1. Problem scenario

We consider the problem of zero-shot unsupervised transfer instance segmentation, which aims to jointly segment objects present in an image and predicts their semantic categories in both unsupervised and zero-shot transfer manner. The unsupervised property of the task prohibits any reliance on manual supervision for instance segmentation, while the zero-shot transfer property assumes that a segmentor has no access to a target data distribution (e.g., a training split of an evaluation benchmark). Note that, such properties pose significant differences from the existing zero-shot instance segmentation, which leverages human supervision (e.g., pixel-level annotations) in a training split (of an evaluation benchmark) for a certain group of classes (e.g., seen categories) during training.

To tackle this challenge, we propose a simple yet effective framework in which we first predict class-agnostic object masks (mask proposal), then classify each mask (mask classification) based on the pixelwise classification obtained in a joint image-text space. Formally, we seek to train a segmenter \( \Phi_{\text{seg}} \), consisting of an image encoder \( \Phi_{\text{enc}}^{\text{im}} \), an image decoder \( \Phi_{\text{dec}}^{\text{im}} \), and a text encoder \( \Phi_T \). The segmenter ingests an image \( x \in \mathbb{R}^{3 \times H \times W} \), a set of concepts/object categories \( (C) \), and outputs a set of masks for semantic segmentation (SS) and instance segmentation (IS):

\[
\Phi_{\text{seg}}(x, C) = \begin{cases} \Phi_{\text{T}}(C) W \Phi_{\text{enc}}^{\text{im}}(x) \in \{0, 1\}^{C \times H \times W} & \text{for SS}, \\ \Phi_{\text{dec}}^{\text{im}} \circ \Phi_{\text{enc}}^{\text{im}}(x) \in \{0, 1\}^{n \times H \times W} & \text{for IS}. \end{cases}
\]

where \( W \) is a matrix projecting image features into a text embedding space and \( n \) denotes a pre-defined number of object mask predictions. Note that, at this stage, the object masks from the image decoder are class-agnostic. To decide a class of the mask proposals, each mask is assigned a category via a dot-product between its average image embedding (from the image encoder) and text embeddings followed by a softmax (detailed in Sec. 3.3). It is worth noting that the design of our framework allows the model to tackle both instance and semantic segmentations concurrently—we show performance of our model on both tasks in Sec. 4.

In the following sections, we introduce the key components for our framework in an unsupervised and zero-shot transfer fashion: generating pseudo-labels for unlabelled images with existing pretrained foundation models, and an efficient transformer-based architecture for simultaneous semantic and instance segmentation.

#### 3.2. Pseudo-label training

To train a segmentor without relying on manual labels, we adopt pseudo-label training as in [48, 49]. Here, we briefly describe our pseudo-mask generation process, composed of archive construction, unsupervised saliency detection, and copy-paste augmentation used to generate synthetic images containing multiple objects.

**Archive construction.** Given an image encoder \( \phi_I \) and a text encoder \( \phi_T \) from a pretrained vision-language model (e.g., CLIP), we first build archives of images for a set of categories \( C \) by curating images for each concept from an unlabelled image dataset \( U \) (called an index dataset). Formally, we extract a set of normalised image embeddings \( \mathcal{F}_I \) as follows:

\[
\mathcal{F}_I = \{ \phi_I(x_i) \in \mathbb{R}^d, i = 1, \ldots, N \}
\]
where \( x_i \in \mathbb{R}^{3 \times H \times W} \) and \( N \) denotes the total number of images in \( \mathcal{U} \). Similarly, we extract a set of normalised text embeddings \( \phi_T(c) \in \mathbb{R}^d \) for a name of each category \( c \in \mathcal{C} \) from the text encoder. Then we select \( k \) images with highest similarities between image and text embeddings to form an archive for a category \( c \):

\[
\mathcal{U}_c = \{ x_i \in \mathcal{U} | i \in \text{argtop}_k[\mathcal{F}_I \phi_T(c)] \} \quad (3)
\]

where \( \text{argtop}_k \) returns indices of \( k \) largest values.

**Unsupervised saliency detection.** Given the image archives for the categories of interest, we generate category-agnostic saliency masks \( S_i \in \{0, 1\}^{H \times W} \) by feeding each image \( x_i \) into an unsupervised saliency detector (e.g., Self-Mask [47]). We then assign the corresponding category name (from archive construction) and an instance id to the inferred saliency mask, which allows for training a segmenter for semantic and instance segmentations as described in Sec. 3.3.

**Synthetic image generation with copy-paste augmentation.** To train a segmenter which can segment multiple objects within an image, we follow [49] and use copy-paste augmentation [16] to synthesise an image with multiple objects. A pseudo-mask is created accordingly by copy-pasting the binary pseudo-masks of the images used for the synthetic image, with a unique instance id and a category label allocated to each mask.

### 3.3. Architecture

To tackle both semantic and instance segmentation tasks while preserving zero-shot ability of a pretrained vision-language model (VLM), we propose a simple framework termed, ZUTIS, which operates on features from image and text encoders of VLM (shown in Fig. 2).

**Semantic segmentation.** Given an image encoder \( \psi_I \) and a text encoder \( \psi_T \) from a pretrained VLM, we extract dense features \( \psi_I(x_i) \in \mathbb{R}^{e_v \times h \times w} \) (e.g., patch tokens from a vision transformer) for an image \( x_i \) from the image encoder where \( e_v, h, \) and \( w \) denote the dimensionality of a visual embedding space, height and width of the features, respectively. The dense features are projected into a text embedding space by a projection matrix \( W \in \mathbb{R}^{e_t \times e_v} \), where \( e_t \) is a dimension of the text space. With text embeddings \( \psi_T(C) \in \mathbb{R}^{|\mathcal{C}| \times e_t} \) from the text encoder for a set of categories, we compute logits via dot-product between the projected image features and text features which are followed by a softmax function:

\[
P_i = \text{softmax}(\psi_T(C)\bar{\psi}_T(x_i), \text{dim}=0) \quad (4)
\]

where \( \bar{\psi}_T(.) \) and \( P_i \) denote \( W\psi_T(.) \) and the probability map, respectively. Then a cross-entropy loss \( \mathcal{L}_{ce} \) is used to minimise differences between the prediction and the corresponding pseudo-mask generated in Sec. 3.2. To inherit the zero-shot ability of pretrained VLM, we only optimise the parameters of the image encoder, leaving the text encoder frozen. Note that this approach is related to MaskCLIP, but with a key difference: we update the parameters in image encoder, while MaskCLIP keeps the parameters fixed, and instead uses value features from the last self-attention layer of the image encoder to produce a semantic prediction. We
compare our method to MaskCLIP in Sec. 4.3.

**Instance segmentation.** Here, we first produce object mask proposals using a query-based transformer decoder. In detail, given dense image features $\psi_T(x_i)$ before projection to the textual space, we pass the features to a feed-forward network (FFN) with a hidden layer (e.g., an MLP with three layers) whose output features are used as values $V \in \mathbb{R}^{d \times h \times w}$ for the transformer decoder. Given $n_q$ object queries $Q \in \mathbb{R}^{n_q \times d}$ and $V$, the decoder outputs query vectors that are fed into another FFN before producing mask proposals $M \in \mathbb{R}^{n_q \times h \times w}$ via a dot-product between the resulting $Q$ and $V$. Then, we update the model with a bipartite matching loss $[5, 8, 9] \mathcal{L}_{\text{mask}}$ between the proposals and the pseudo-masks for the image. We find that it is essential to stop gradients from the transformer decoder flowing to the image encoder, otherwise the model fails to converge (see Sec. 4.2). For $\mathcal{L}_{\text{mask}}$, we use a mixture of dice coefficient $[39]$ and binary cross-entropy losses $\mathcal{L}_{\text{mask}} = \mathcal{L}_{\text{dice}} + \mathcal{L}_{\text{bce}}$ with equal weights following [8].

During inference, we assign each mask proposal $m_i \in M$ a category whose text embedding shares the highest similarity with an average image embedding of the mask. For this, we first binarise $m_i$ with a threshold $t$ and compute the average image embedding $\overline{\psi}_T(x_i, m_i; t)$:

$$\overline{\psi}(x_i, m_i; t) = \text{mean}(\psi_T(x_i)|m_i > t]) \in \mathbb{R}^e. \quad (5)$$

Then, we assign the mask that category with highest similarity to the average image embedding:

$$\arg \max_{c \in \mathcal{C}} [\psi_T(C)\overline{\psi}_T(x_i, m_i; t)], \quad (6)$$

Note that both text and average image embeddings are L2-normalised before dot-product. In addition, we compute a confidence score $s_i \in [0, 1]$ for each mask proposal, defined as the average value of the mask region multiplied by the maximum class probability for the mask (similarly to [8]). Lastly, to reduce false positives occurring from redundant predictions for a single object, we apply non-maximum suppression (NMS) to the proposals before outputting final instance predictions. We show the effect of NMS in Sec. 4.2.

**Discussion.** The key differences of ZUTIS from prior work for unsupervised semantic segmentation with language-image pretraining are two-fold: (i) rather than using a fixed n-way classifier, we use a pretrained, frozen text encoder as a classifier, and optimise an image encoder to output dense features aligned with the textual features from the text encoder, a design choice that allows the model to be open-vocabulary; (ii) we enable instance segmentation by training a query-based transformer decoder by bootstrapping the results from saliency detection via copy-paste augmentation.

## 4. Experiments

In this section, we first describe the details of our experiments including datasets, network architecture, training and inference details, and evaluation metrics. Next, we ablate components of our method such as the use of stop-gradient and non-maximum suppression, and report the performance of the model on both semantic and instance segmentation.

### 4.1. Implementation details

**Datasets.** We evaluate our model on COCO2017 [35] val split, PASCAL VOC2012 [14], CoCA [67], and ImageNet-S [15] test split for semantic segmentation and COCO-20K [55] for instance segmentation following [53]. To demonstrate our model’s zero-shot ability to new concepts, we additionally consider the CUB-200-2011 [57] test split. In detail, VOC2012 trainval split has 2,913 images with 21 categories including a background. COCO2017 val and CoCA are composed of 5,000 and 1,295 images with 80 object categories and a background class. ImageNet-S test consists of 27,423 images with 919 object classes which are a subset of ImageNet1K [46] classes. COCO-20K comprises 19,817 images from the COCO2014 train split with the same 80 object classes as COCO2017. CUB-200-2011 test is composed of 5,794 images with 200 fine-grained bird breeds.

Note that, in this paper, we primarily consider the zero-shot transfer setting, in which the model has no access to training data sharing a data distribution with a downstream benchmark. Thus, throughout our experiments, we use images retrieved from ImageNet1K (1.2M images) and PASS [1] (1.4M images) by the ViT-L/14@336px CLIP model to form an index image dataset except an experiment in the unsupervised domain adaptation setting for the ImageNet-S benchmark where we only retrieve ImageNet1K images. For prompt engineering, we average text embeddings from 85 templates to obtain a textual feature for a category following [48,49,70]. In all cases, we fix the number of images for an archive as 500 (i.e. 500 images for a category) as in [49].

**Architecture.** We use transformer-based CLIP models for the image encoder (e.g., ViT-B/16) and text encoder. We use 6 transformer layers for the transformer decoder and three-layer MLP for the FFN. We feed patch tokens from the last layer of the image encoder to the decoder after bilinearly upsampling them by a factor of 2 to enable predictions at a higher resolution.

**Training.** We compute the final loss $L$ for the model as $L = L_{ce} + \lambda_{\text{mask}}L_{\text{mask}}$ with $\lambda_{\text{mask}}$ set to 1.0. We optimise our model with the AdamW optimiser [36], with an initial learning rate of 5e-5 and a weight decay of 0.05. For the image encoder, we use a smaller learning rate of 5e-6. We train for 20K iterations with the Poly learning rate scheduler [7], except when training for 919 categories of objects.
Evaluation metrics. To measure our model’s performance, we use the standard metrics such as mean intersection-over-union (mIoU) for semantic segmentation and COCO-style mask average precision (AP\textsuperscript{mk}) for instance segmentation.

4.2. Ablation study

Here, we study the influence of the components in our method, including the choice of encoder architecture, stop-gradient and NMS. For experiments in the ablation study, we report the results on the VOC2012 trainval split.

Effect of encoder architecture. In Fig. 3, we show the performance of our model with different transformer-based CLIP image encoders such as ViT-B/32, ViT-B/16, and ViT-L/14 [13].\textsuperscript{2} We observe that at the cost of computation measured in throughput, heavier models consistently outperform lighter models in both mIoU (left) and AP\textsuperscript{mk} (right). While ViT-L/14 performs best, we report results with either ViT-B/32 or ViT-B/16 in the following experiments to limit differences in performance due to model size (ResNet50 [21] is typically used by previous unsupervised methods). Note that ViT-B/32 and ViT-B/16 are the lightest CLIP models compatible with our framework.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
\textbf{stop-grad} & \textbf{NMS} & \textbf{AP\textsuperscript{mk}} & \textbf{AP\textsuperscript{mk}_{50}} & \textbf{AP\textsuperscript{mk}_{75}} \\
\hline
\checkmark & \checkmark & 0.3 & 0.4 & 0.3 \\
\checkmark & \checkmark & 4.4 & 8.7 & 4.2 \\
\checkmark & \checkmark & 13.7 & 30.9 & 11.1 \\
\hline
\end{tabular}
\caption{Applying a stop-grad operation between the image encoder and decoder allows the encoder features to keep semantic representations.}
\end{table}

Effect on stop-gradient and non-maximum suppression. As described in Sec. 3.3, we prevent gradients from back-propagating to the encoder parameters when optimising the transformer decoder to generate mask proposals. We observe in Tab. 1 that this is crucial, otherwise the optimisation does not converge to a reasonable solution. For our model trained with stop-gradient, applying NMS to its mask proposals brings a noticeable gain in performance from 4.4 to 13.7 AP\textsuperscript{mk}. This is because the model tends to predict redundant mask proposals for a single object, increasing false positives during evaluation. We therefore employ both stop-gradient and NMS throughout the experiments.

Analysis on a stop-gradient operation. To further investigate why applying a stop-grad operation is essential in our framework, we hypothesise that if stop-grad is not applied, gradients from the mask loss for instance segmentation could potentially dominate the visual feature learning, thus harming the visual-language alignment in pretrained VLM. To verify this, we evaluate the models trained w/ and w/o the stop-grad w.r.t. class-agnostic and -aware AP\textsuperscript{mk}. As shown in Tab. 2, while the model trained w/o stop-grad performs poorly on class-aware AP\textsuperscript{mk}, it performs reasonably on class-agnostic AP\textsuperscript{mk}, indicating that the resulting features are capable of segmenting objects but not suitable for classification.

4.3. Main results

Here, we compare ZUTIS to existing unsupervised instance segmentation and semantic segmentation approaches. While we mainly focus on the zero-shot transfer setting in which the model has no access to training data for the target downstream task, we also report results for semantic segmentation in the unsupervised domain adaptation setting to draw comparison with existing approaches, where the target data distribution is exposed to the model.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
\textbf{stop-grad} & \textbf{class-agnostic} & \textbf{class-aware} \\
\hline
\checkmark & \checkmark & 8.4 & 19.9 & 6.4 & 1.0 & 1.9 & 0.9 \\
\checkmark & \checkmark & 9.9 & 24.1 & 7.4 & 13.7 & 30.9 & 11.1 \\
\hline
\end{tabular}
\caption{Applying a stop-grad operation between the image encoder and decoder allows the encoder features to keep semantic representations.}
\end{table}

\textsuperscript{2}While there are also ResNet-based CLIP encoders, we found them not suitable for instance segmentation as they directly output features in the joint image-text space via an attention pooling [44].
initiation and architecture for a backbone (i.e., ReCo initialises its backbone with supervised Styliised-ImageNet training and NamedMask with DINO), thus a direct comparison is not possible. Relative to MaskCLIP (which is comparable), our model shows improvements of 12.2 and 12.5 mIoU on COCO and CoCA, respectively.

In Tab. 5, we evaluate our method in the unsupervised domain adaptation setting, where the model is trained with images retrieved from the ImageNet1K train split, and evaluated on ImageNet-S. Compared to the state-of-the-art unsupervised method (i.e. NamedMask), our approach achieves a gain of 4.6 mIoU with ViT-B/32 and 14.5 mIoU with ViT-B/16 at the expense of lower throughput.

**Generalisation to new categories.** Since we optimise our image encoder to produce visual embeddings aligned to the text embedding from the frozen text encoder, we expect the resulting model to be capable of segmenting objects of novel concepts which are unseen during training. To verify this, we consider two scenarios: (i) a high-level to low-level category transfer, i.e., the model is evaluated on categories that it did not encounter during training but only their super-set category; (ii) transfer to unseen categories semantically far from those it has seen during training.

For the first scenario, we evaluate our model, trained for 80 categories in COCO including ‘bird’, on the test split of CUB-200-2011 benchmark which has 200 fine-grained bird categories. Here, given a high-level category (i.e. “bird”) or a low-level category for an image (i.e. image-specific fine-grained bird categories), we encode the category with the text encoder and proceed with segmentation as usual. Then we compare the segmentation result with the groundtruth mask. It is worth mentioning that the performance is measured in IoU rather than mIoU, as we do not expect the model to distinguish between the fine-grained categories. This is because the image archive for “bird” is likely to contain images of different bird breeds, which encourages the model to learn the invariance between birds. However, we expect the model to also identify the “bird” regions given a specific bird breed as a target. As shown in Tab. 6, when given low-level categories (bird breeds) as input, our model can perform equally well as given a high-level category (“bird”). Note that the semantically closest category to the fine-grained categories among the 80 classes in COCO is “bird” and that only 16 out of 200 fine-grained bird categories contain “bird” as a part of its name (e.g. “Anna Hummingbird”). This implies that the model is equipped with knowledge about fine-grained bird species.

For the second scenario, we split 65 seen and 15 unseen classes in the COCO dataset and evaluate our model on the unseen classes following prior work on zero-shot instance segmentation [69]. For this, we train our model with image archives constructed only for the seen categories. As shown in Tab. 7, when compared to a baseline unsupervised

<table>
<thead>
<tr>
<th>model</th>
<th>backbone</th>
<th>AP\text{\textsuperscript{50}}</th>
<th>AP\text{\textsuperscript{75}}</th>
</tr>
</thead>
<tbody>
<tr>
<td>unsupervised methods w/o language-image pretraining</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DINO [6]</td>
<td>ViT-S/16</td>
<td>0.7</td>
<td>2.0</td>
</tr>
<tr>
<td>LOST [50]</td>
<td>ViT-S/16</td>
<td>1.2</td>
<td>3.3</td>
</tr>
<tr>
<td>MaskDistill [53]</td>
<td>ViT-S/16</td>
<td>1.7</td>
<td>4.1</td>
</tr>
<tr>
<td>MaskDistill [53]</td>
<td>RN50-C4</td>
<td>3.5</td>
<td>7.7</td>
</tr>
<tr>
<td>unsupervised method w/ language-image pretraining</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaskCLIP [70]</td>
<td>ViT-B/32</td>
<td>0.3</td>
<td>0.8</td>
</tr>
<tr>
<td>ZUTIS (Ours)</td>
<td>ViT-B/32</td>
<td>3.4</td>
<td>8.0</td>
</tr>
<tr>
<td>\textsuperscript{1}MaskCLIP [70]</td>
<td>ViT-B/16</td>
<td>1.3</td>
<td>5.4</td>
</tr>
<tr>
<td>ZUTIS (Ours)</td>
<td>ViT-B/16</td>
<td>5.7</td>
<td>11.0</td>
</tr>
</tbody>
</table>

Table 3. Comparison to previous unsupervised instance segmentation methods on COCO-20K. \textsuperscript{1}Initialised with supervised Stylised-ImageNet pretraining [40]. \textsuperscript{2}Initialised with DINO [6].

<table>
<thead>
<tr>
<th>model</th>
<th>arch.</th>
<th>COCO</th>
<th>CoCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>initialised with different encoder features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReCo\textsuperscript{1} [48]</td>
<td>DeiT-S/16 &amp; RN50x16</td>
<td>23.8</td>
<td>28.8</td>
</tr>
<tr>
<td>NamedMask\textsuperscript{1} [49]</td>
<td>RN50 &amp; DLv3+</td>
<td>28.4</td>
<td>27.3</td>
</tr>
<tr>
<td>initialised with CLIP encoder features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaskCLIP [70]</td>
<td>ViT-B/16</td>
<td>20.6</td>
<td>20.2</td>
</tr>
<tr>
<td>ZUTIS (Ours)</td>
<td>ViT-B/16</td>
<td>32.8</td>
<td>32.7</td>
</tr>
</tbody>
</table>

Table 4. Comparison to previous unsupervised semantic segmentation methods leveraging image-language pretraining on COCO and CoCA in terms of mIoU (%). \textsuperscript{1}Initialised with supervised Stylised-ImageNet pretraining [40]. \textsuperscript{2}Initialised with DINO [6].

**Unsupervised instance segmentation.** In Tab. 3, we evaluate our model on unsupervised instance segmentation on the COCO-20K dataset. As a baseline for our method, we evaluate MaskCLIP for instance segmentation by treating its semantic segmentation masks for an image as mask proposals. When comparing to the state-of-the-art approach [53], our model shows comparable (with ViT-B/32) or better performance than thresholding. For MaskCLIP, we simply provide a text embedding for “background” along with other object category embeddings which we find more effective than thresholding.

In Tab. 4, we evaluate our model in the zero-shot transfer setting on the COCO \text{val} and CoCA benchmarks and compare to unsupervised methods. Note that ReCo and NamedMask have different settings from ours in terms of initialisation and architecture for a backbone (i.e., ReCo initialises its backbone with supervised Styliised-ImageNet training and NamedMask with DINO), thus a direct comparison is not possible. Relative to MaskCLIP (which is comparable), our model shows improvements of 12.2 and 12.5 mIoU on COCO and CoCA, respectively.
Table 5. Comparison to existing unsupervised methods on the ImageNet-S benchmark with 919 object categories in the unsupervised domain adaptation setting. We also show mIoU in diverse object sizes from small (S), medium-small (MS), medium-large (ML), and large (L). †Encoder initialised with supervised Stylized-ImageNet pretraining. ‡Encoder initialised with unsupervised pretraining (i.e., DINO).

Table 6. High-level to low-level zero-shot transfer on the CUB-200-2011 benchmark. When given a finegrained bird breed, ZUTIS can segment the corresponding bird regions as good as when it is given a high-level category “bird.”

Table 7. Zero-shot unsupervised instance segmentation for 15 unseen categories on COCO-20K.

5. Broader impact

The goal of this work is to propose a practical framework for instance and semantic segmentation. As such, we hope that our work facilitates many useful applications of segmentation (medical image analysis, fault detection in manufacturing, security monitoring etc.). However, automatic segmentation represents a dual-use technology and is therefore subject to misuse (unlawful surveillance, for example). We also note that we build ZUTIS on top of foundation models like CLIP [44]. These models are known to reflect biases present in large, minimally curated internet corpora and thus our model is likely to inherit these biases also. Consequently, any practical deployment of ZUTIS will require assessment (and potentially also mitigation) of the risks posed by such biases.

6. Conclusion

In this work, we introduced ZUTIS, the first framework for joint instance segmentation and semantic segmentation in a zero-shot transfer setting that requires no pixel-level or instance-level annotation. We employ a query-based transformer architecture for instance segmentation and train it on pseudo-labels generated from applying an unsupervised saliency detector to images retrieved by CLIP. Through careful experiments, we demonstrated the effectiveness of ZUTIS across both instance segmentation and semantic segmentation tasks. In future work, we intend to explore the application of ZUTIS to other modalities such as video.
References


[70] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In ECCV, 2022. 2, 5, 7, 8