Image and Vision Computing 150 (2024) 105206

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

DynaSeg: A deep dynamic fusion method for unsupervised image segmentation incorporating feature similarity and spatial continuity

Boujemaa Guermazi [a,*,1], Riadh Ksantini [b], Naimul Khan [c]

[a] Electrical, and Computer Engineering, Toronto Metropolitan University, 350 Victoria Street, Toronto M5B 2K3, Ontario, Canada
[b] Computer Science, University of Bahrain, 1017 Road 5418, Zallaq, 1054, Sakhir, Kingdom of Bahrain
[c] Electrical, Computer, and Biomedical Engineering, Toronto Metropolitan University, 350 Victoria Street, Toronto M5B 2K3, Ontario, Canada

ARTICLE INFO

Keywords:
Unsupervised learning
Image segmentation

ABSTRACT

Our work tackles the fundamental challenge of image segmentation in computer vision, which is crucial for diverse applications. While supervised methods demonstrate proficiency, their reliance on extensive pixel-level annotations limits scalability. We introduce DynaSeg, an innovative unsupervised image segmentation approach that overcomes the challenge of balancing feature similarity and spatial continuity without relying on extensive hyperparameter tuning. Unlike traditional methods, DynaSeg employs a dynamic weighting scheme that automates parameter tuning, adapts flexibly to image characteristics, and facilitates easy integration with other segmentation networks. By incorporating a Silhouette Score Phase, DynaSeg prevents undersegmentation failures where the number of predicted clusters might converge to one. DynaSeg uses CNN-based and pre-trained ResNet feature extraction, making it computationally efficient and more straightforward than other complex models. Experimental results showcase state-of-the-art performance, achieving a 12.2% and 14.12% mIOU improvement over current unsupervised segmentation approaches on COCO-All and COCO-Stuff datasets, respectively. We provide qualitative and quantitative results on five benchmark datasets, demonstrating the efficacy of the proposed approach. Code available at \url{https://github.com/RyersonMultimediaLab/DynaSeg}

1. Introduction

In recent years, computer vision has taken remarkable strides, driven by the availability of large-scale image datasets as well as the development of advanced machine learning algorithms. Within various real-time applications [1–4], image segmentation stands out as a pivotal computer vision task, playing a crucial role in interpreting visual content at a granular level. Unlike the straightforward task of image classification, which allocates a category label to the entire image [5], segmentation involves assigning category labels to individual pixels, effectively outlining the image into meaningful segments or regions. This nuanced approach not only enhances the interpretation of complex scenes but also contributes to a deeper understanding of visual content.

The segmentation task can be approached from different perspectives, leading to three main types of image segmentation. Semantic segmentation [6] identifies uncountable and shapeless regions, often referred to as 'stuff,' such as grass, sky, or road, based on similar textures or materials. Instance segmentation [7] focuses on pinpointing and segmenting individual instances of countable objects, such as people, animals, and tools, treating them as distinct 'things.' Lastly, panoptic segmentation [8] unifies the two distinct concepts used to segment images, assigning a semantic label to each pixel in an image and giving the pixel a unique identifier if it is an object instance. The semantic nuances that differentiate these tasks have led to the creation of methods that employ specialized architectures. However, despite the advancements in segmentation methodologies, significant challenges persist. Variability in object appearance, complex interactions between objects, variations in object scale, and noisy or ambiguous edges all pose obstacles to accurate segmentation. Objects can look different due to lighting, color, and texture changes, making consistent identification difficult. Complex interactions, such as overlapping people in a crowd or intertwined tree branches and power lines, further complicate segmentation. Additionally, objects appear in various sizes depending on their distance from the camera, requiring algorithms to handle different scales effectively. Noisy or ambiguous edges, caused by low resolution or motion blur, add another layer of difficulty. Developing robust and

* Corresponding author.
E-mail addresses: bguermazi@torontomu.ca (B. Guermazi), rksantini@uob.edu.bh (R. Ksantini), n77khan@torontomu.ca (N. Khan).
[1] Code will be made available upon manuscript acceptance.

https://doi.org/10.1016/j.imavis.2024.105206
Received 31 May 2024; Received in revised form 22 July 2024; Accepted 7 August 2024
Available online 10 August 2024

adaptable segmentation methods that can address these complexities is crucial for improving the accuracy and reliability of image segmentation in real-world applications. Therefore, while task-specific architectures have significantly advanced segmentation objectives, there is a pressing need for methods that offer more flexibility to generalize across different segmentation tasks, ultimately enhancing the performance of segmentation algorithms in challenging real-world scenarios.

Classical pioneering segmentation techniques such as active contour models (ACM) [9], k-means [10], and graph-based segmentation (GS) [11,12] impose global and local data and geometry constraints on the masks. As a result, these techniques are sensitive to initialization and require heuristics such as point resampling, making them unsuitable for modern applications. Currently, the state of the art in segmentation is dominated by deep learning. The Mask R-CNN framework [13] has provided advancement in semantic and instance image segmentation. However, a major drawback is a substantial requirement for hand-labeled data, limiting the widespread applicability across diverse domains. This challenge becomes particularly pronounced in the context of pixel-wise classification, where the cost of annotation per image is prohibitively expensive.

Unsupervised image segmentation is a possible solution to automatically segment an image into semantically similar regions where the system can find objects, precise boundaries and materials that even the annotation system may not unveil properly. The task has been studied as a clustering problem in the recent literature [14–16], with promising results. Differentiable feature clustering, a state-of-the-art CNN-based algorithm proposed by [17], simultaneously optimizes pixel labels and feature representations through a combination of *feature similarity* and *spatial continuity* constraints [17]. Feature similarity corresponds to the constraint that pixels in the same cluster should be similar to each other. Spatial continuity refers to the constraint that pixels in the same cluster should be next to each other (continuous). However, to achieve the desired segmentation result, [17] applies a manual parameter tuning to find the optimal balancing weight μ, which fails to achieve a good balance between the two constraints, as mentioned above, depending on the degree of detail in the image and the dataset.

In this work, we present a novel dynamically weighted loss scheme, offering flexibility in updating the parameters and automatically tuning the balancing weight μ. Our approach conditions the value of μ based on the number of predicted clusters and the iteration number. At each iteration, we dynamically prioritize one of the constraints, resulting in a well-balanced optimization. Key contributions include[2]:

- **Dynamically Weighted Loss:** Introducing a flexible and adaptive loss function that dynamically adjusts the balancing weight μ during training, ensuring optimal balance between feature similarity and spatial continuity. This makes the model adaptable to different datasets without requiring manual changes to parameters, such as the balancing weight, and facilitates easy implementation of the proposed method on other segmentation networks.
- **Silhouette Score Phase:** Incorporating a silhouette score-based phase to guide the optimization process, ensuring improved cluster quality. Unlike traditional methods that rely on fixed thresholds to determine segmentation termination and may converge to a single cluster, failing to capture the diversity and complexity of real-world images, our approach dynamically assesses cluster quality using silhouette scores. This prevents under-segmentation failures, where the segmentation process prematurely stops, leading to inaccurate results. By dynamically evaluating cluster quality, our method ensures that the segmentation process continues until meaningful clusters are formed, without the need for predefined thresholds. This not only improves segmentation accuracy but also reduces the burden of fine-tuning additional hyperparameters.
- **ResNet with FPN Integration:** Leveraging the power of ResNet combined with the Feature Pyramid Network (FPN) for enhanced feature extraction in the unsupervised segmentation task. The FPN enhances the representation map, making it semantically strong and improving the overall segmentation performance. This integration is particularly beneficial for segmenting objects at different scales and resolutions, as FPN enables multi-scale processing and feature integration across different levels of abstraction. Additionally, by incorporating ResNet, we capitalize on its inherent strengths, similar to those of CNNs, while leveraging residual information to mitigate issues like vanishing gradients. This combination amplifies the network's capacity to capture intricate details and semantic information critical for accurate segmentation.

Experimental results across five benchmark datasets demonstrate that our method, DynaSeg, achieves superior quantitative metrics and qualitative segmentation results. The dynamically weighted loss facilitates a more effective balance between feature similarity and spatial continuity, leading to enhanced segmentation performance.

2. Related work

Recent advancements in machine learning have significantly impacted various vision applications, particularly image segmentation. The introduction of convolutional neural networks (CNNs) and the availability of large annotated datasets have revolutionized image segmentation, enabling these models to surpass traditional algorithms such as active contour models (ACM) [9], k-means clustering [10], and graph-based segmentation (GS) [11]. Notable CNN-based techniques, including Mask R-CNN [13], U-Net [18], SegNet [19], and DeepLab [20], offer precise segmentation capabilities. However, these models are heavily reliant on large annotated datasets, which not only escalate annotation costs but also introduce challenges related to missing labels and data quality variations [21–23]. These limitations have driven the development of unsupervised segmentation models. Among these approaches, clustering-based and CNN-based methods have gained prominence, each offering unique strategies to address these challenges. Our model, DynaSeg, exemplifies this trend by providing scalable and cost-efficient solutions without the need for explicit annotations.

2.1. Clustering-based methods

Foundational unsupervised segmentation methods like K-means [10], Mean Shift [24], and graph-based segmentation (GS) [11] cluster pixels based on feature similarity but often lack spatial coherence. These methods treat each pixel independently, resulting in fragmented segmentations, especially in images with complex structures. In contrast, our model integrates feature similarity and spatial continuity, ensuring that similar and contiguous pixels form the same cluster. Unlike static methods requiring a predetermined number of clusters [10,24], DynaSeg adjusts the number of clusters during training, enhancing flexibility and robustness.

A notable approach is Deep Image Clustering (DIC) [15], which transforms features and clusters them using deep sub-networks but relies on superpixels with predetermined boundaries. Similarly, Invariant Information Clustering (IIC) [16] and its variants [25] use mutual-information-maximization but require uniform cluster distributions, making them effective with balanced datasets [22,23]. Their efficacy can be influenced by data augmentation strategies [26–28]. Unlike methods that separate feature extraction and clustering [15,16], DynaSeg employs Joint Optimization where the backpropagation of clustering losses directly influences the CNN. This integration ensures a cohesive segmentation process, continuously refining the model and overcoming the limitations of predetermined boundaries and data imbalance.

[2] A preliminary version of this work was published in [40].

2.2. Deep learning-based methods

Deep learning-based methods have gained significant attention in unsupervised image segmentation, showcasing various innovative approaches. Generative Adversarial Networks (GANs) [29–32] have emerged as powerful tools for creating segmented images by separating foreground and background. For example, Melas-Kyriazi [29] leverages pre-trained GANs like BigBiGAN [30] to extract salient object segmentation from the latent space. Despite their innovation, these methods often require some form of class supervision, limiting their purely unsupervised nature. Labels4Free [31] extends Style-GAN2 [32] with a segmentation branch for unsupervised foreground object segmentation, typically focusing on binary clustering. However, these approaches tend to rely on discriminative losses, such as binary cross-entropy loss, for training the discriminator network, which may not directly optimize segmentation quality, resulting in suboptimal outcomes. In contrast, our model adopts a hybrid approach for multi-class segmentation by integrating a clustering strategy with CNN-based feature extraction.

Beyond generative models, contrastive learning [33,34] has gained traction for its ability to learn representations by contrasting similar and dissimilar pairs of image patches. PICIE [33] introduces a Siamese perspective to unsupervised segmentation, emphasizing equivariance learning and consistent clustering assignments across different image views. DenseSiam [34] focuses on learning dense representations through contrastive learning and clustering-based techniques. While these methods effectively capture images' inherent structures and patterns, contributing significantly to segmentation performance, they have notable limitations. Specifically, these approaches introduce additional computational operations, struggle with unknown transformations, and are sensitive to noisy or low-quality data, often resulting in biased representations or suboptimal clustering results. Our model addresses these limitations and offers a more robust solution. DynaSeg employs a spatial continuity loss that acts as a high-pass filter, enhancing segmentation accuracy by preserving high-frequency details and attenuating low-frequency components. This discourages smooth regions from being segmented into multiple clusters, promoting homogeneity within each cluster, making the segmentation more coherent and reducing noise within clusters.

Wnet [35] combines U-net [18] structures into auto-encoders for segmentation, followed by a post-processing phase for refinement. While effective, this method is computationally expensive and requires extensive hyperparameter tuning. Infomax [14] presents an alternative where an input image is partitioned into superpixels. The Region-Wise-Embedding (RWE) extracts a feature embedding for each superpixel region, and Mutual-Information-Maximization is employed with adversarial training. Despite its efficiency, it faces challenges like superpixel reliance and predetermined boundaries. In contrast, our proposed hybrid model offers a streamlined solution by eliminating the need for post-processing. Our model generates compact clusters directly from the feature space by dynamically optimizing the clustering loss and spatial continuity loss. This integrated approach enables our model to extract rich and informative features, facilitating accurate and robust segmentation, even in challenging scenarios. DynaSeg offers enhanced flexibility and adaptability by removing the reliance on predefined boundaries or superpixels [36], ensuring more natural and visually appealing segmentation outcomes.

Additionally, recent advancements in CNN architectures, such as Vision Transformers (ViT) [37], have shown promise in various computer vision tasks, including image segmentation. However, ViTs come with their own set of challenges, such as high computational complexity, data inefficiency, training instability, and a propensity for overfitting, which can limit their accessibility and applicability in specific contexts. In contrast, our hybrid model offers a more efficient and scalable solution by dynamically optimizing the clustering loss and leveraging a CNN-based network or a pre-trained ResNet with FPN to extract high-level features from input images. This approach ensures robustness and adaptability across different datasets.

2.3. Dynamic weight adjustment

Dynamic loss functions have been explored in several contexts within machine learning. For example, Dynamic Autoencoders (DynAE) [38] employ adaptive loss functions to balance reconstruction and clustering objectives dynamically during training, enhancing clustering performance by adapting to the data's changing characteristics. Similarly, DFNet [39] utilizes dynamic loss weights to manage class imbalance and refine segmentation accuracy in semantic segmentation tasks. By calculating weights based on the pixel count of each class within a batch, DFNet mitigates the imbalance and improves performance, particularly for minority classes.

The application of dynamic loss functions in image segmentation, particularly in unsupervised settings, has received relatively less attention. Most existing segmentation methods rely on fixed weights, such as Differentiable Feature Clustering [17], which optimizes pixel labels and feature representations using a combination of feature similarity and spatial continuity constraints. However, achieving the desired segmentation result with this approach often requires manual parameter tuning to find the optimal balancing weight (μ) between feature similarity and spatial continuity. This manual tuning process can be cumbersome and may fail to balance the two constraints well, especially when dealing with images of varying levels of detail and datasets with different characteristics. Moreover, relying solely on fixed weight factors for feature similarity and spatial continuity constraints can lead to the number of predicted clusters converging to one, making it challenging to adapt to different datasets and requiring extensive fine-tuning. Additionally, the lack of adaptability and the need for manual parameter adjustments make integrating this approach with other networks difficult. In contrast, DynaSeg introduces a unique dynamic loss function specifically designed for unsupervised image segmentation. This model employs a Dynamic Weighting Scheme that dynamically adjusts the weighting parameter (μ) during iterations. By incorporating a silhouette score-based phase to guide the optimization process, our model prevents under-segmentation failures and ensures robust performance across diverse datasets. This dynamic approach not only simplifies the integration of our model with other networks but also enhances its adaptability and scalability, making it well-suited for various segmentation tasks and datasets. One of the key benefits of DynaSeg is its ability to autonomously balance competing objectives, thereby enhancing segmentation quality without the need for manual hyperparameter tuning.

3. Methodology

This section details the proposed DynaSeg method, including the architecture design, dynamic weighting scheme, silhouette score integration, and training procedure. Each component is meticulously described to ensure reproducibility and clarity.

3.1. Model architecture

Our unsupervised image segmentation model, DynaSeg, is designed to effectively capture complex patterns and structures within images. Building upon the architecture introduced in our preliminary work [40], we present further enhancements tailored to adapt to varying image content. The architecture comprises three main components: a Feature Extractor Network, a Dynamic Weighting Scheme, and a Clustering Mechanism.

As shown in Fig. 1, the Feature Extractor Network produces a p-dimensional feature map r, which is then classified into q classes using a linear classifier layer. A batch normalization function is applied to obtain a normalized map r'. Lastly, the argmax function assigns each pixel a cluster label c_n based on the maximum value in r'. Each pixel is assigned the corresponding cluster label c_n, identical to allocating each

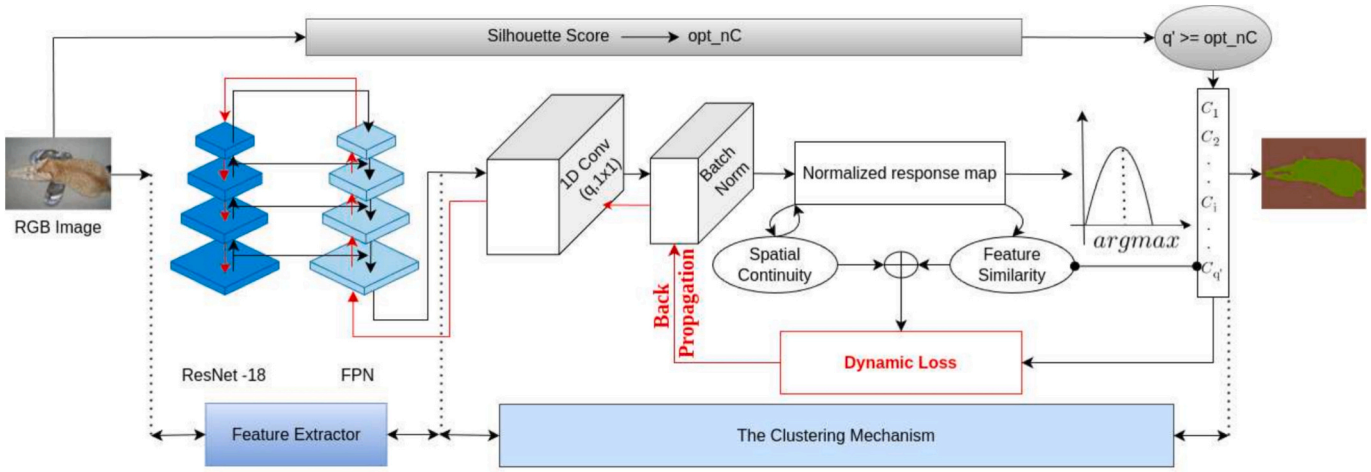

Fig. 1. Overview of the DynaSeg Framework: The Feature Extractor Network generates a feature map, classified into q clusters via a linear classifier and batch normalization, resulting in a normalized response map. Cluster labels c_i are assigned to each pixel using the argmax function. The number of clusters q' is dynamically updated based on feature similarity and spatial continuity. Loss L is computed during backpropagation, with parameters updated using SGD. This process iterates T times to refine cluster labels c_i, achieving final segmentation. The Silhouette Score sets *opt_nC* as the threshold for q' to prevent under-segmentation. Black arrows indicate the feedforward path, while red arrows represent backpropagation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

pixel to the closest point among the q' representative points.

The loss L (defined later) is calculated during backward propagation, and the convolutional filters and classifier parameters are updated using stochastic gradient descent. This process is iterated T times to achieve the final prediction of cluster labels c_n. The segmentation is handled in an unsupervised manner, with the number of clusters q' dynamically updated based on feature similarity and spatial continuity.

The components of DynaSeg work synergistically: the feature extractor captures complex details, the dynamic weighting scheme adapts the model's focus during training, and the clustering mechanism assigns meaningful labels to pixels. The integration of these components is crucial for achieving state-of-the-art unsupervised image segmentation.

3.2. Feature extractor network

The Feature Extractor Network, functioning as the primary processing stage, is tasked with extracting high-level features from input images. This pivotal role lays the foundation for subsequent stages by unraveling intricate details and capturing nuanced patterns within the input images. In our extended model, we explore two alternatives for feature extraction: a CNN-based Network comprising M convolutional components and a pre-trained ResNet [41] with a Feature Pyramid Network (FPN) decoder. This exploration allows us to compare and leverage the strengths of each approach for improved image segmentation, ensuring robustness and efficiency in feature extraction.

3.2.1. CNN-based network

Our CNN-based network consists of M convolutional components, each integrating 2D convolution, ReLU activation, and batch normalization. This architecture forms a robust feature extraction pipeline, producing a p-dimensional feature map x_n. The network's simplicity ensures computational efficiency, making it suitable for practical applications without the need for extensive computational resources.

Pooling layers often reduce the spatial resolution of feature maps, leading to the loss of fine-grained details crucial for accurate image segmentation. By avoiding pooling, our network retains the original resolution of the feature maps, preserving critical spatial information. This design choice enhances the network's ability to produce high-quality, semantically meaningful segments without sacrificing computational efficiency. Additionally, the CNN-based approach's robustness in handling high-dimensional data makes it particularly effective for various image segmentation tasks.

3.2.2. Pre-trained ResNet with Feature Pyramid Network (FPN) decoder

We incorporate a pre-trained ResNet-18 model with a Feature Pyramid Network (FPN) decoder to enhance our network's feature extraction capabilities. While pyramid networks have been utilized in segmentation tasks, such as in APCNet [42] and FPN for Land Segmentation [43], our methodology incorporates several novel elements that distinguish DynaSeg from these existing methods. Unlike traditional pyramid networks [42] that construct multi-scale contextual representations using global-guided local affinity, DynaSeg extends this by dynamically adjusting the segmentation process, effectively balancing feature similarity and spatial continuity to produce more nuanced and context-aware segmentation outputs. This dynamic adjustment addresses the limitations of static parameter settings in conventional pyramid networks. Additionally, our approach uniquely integrates a pre-trained ResNet with an FPN decoder specifically tailored for unsupervised segmentation, enhancing feature extraction through robust multi-resolution processing. Optimized for unsupervised learning, distinguishing it from other FPN methods [42,43].

- **ResNet Adaptation:** We modify the ResNet-18 architecture by removing the Global Average Pooling (GAP) and Fully Connected (FC) layers, typically used for classification tasks. Instead, we replace the final fully connected layer with a convolutional layer to produce spatially preserved feature maps suitable for segmentation tasks. ResNet's use of residual connections effectively mitigates the vanishing gradient problem, allowing for deeper network training and more accurate feature extraction.
- **FPN Decoder:** We integrate a Feature Pyramid Network (FPN) decoder instead of employing a standard decoder. The FPN utilizes lateral connections to fuse features from different resolutions, enhancing the representation of multi-scale features. By applying the FPN on top of the high-level features from the ResNet (specifically the Conv5_x layer), we generate a p-dimensional feature map x_n. This approach retains high-resolution features that are semantically rich and detailed.

This strategic combination leverages ResNet-18's depth and robust feature extraction capabilities along with FPN's multi-scale feature

integration. As a result, our model can capture and utilize details at various granularity levels, significantly improving image segmentation performance. The residual connections in ResNet-18 help in mitigating the vanishing gradient problem, allowing for deeper and more accurate feature extraction. This complements the FPN's ability to integrate features at multiple scales, providing a comprehensive representation that enhances segmentation accuracy.

The dual approach of using both a CNN-based network and a pre-trained ResNet with FPN decoder allows us to compare and leverage the strengths of each method. The CNN-based network offers robustness and computational efficiency, making it suitable for scenarios with limited computational resources. In contrast, the pre-trained ResNet with FPN decoder provides richer and more diverse feature representations, which are crucial for capturing fine-grained details in complex images. By incorporating both approaches, we achieve a balanced and highly effective feature extraction mechanism that enhances the overall performance of our segmentation framework.

3.3. Dynamic weighting scheme

The Dynamic Weighting Scheme is a novel contribution of our method, designed to dynamically adjust the weighting parameter (μ) during training iterations. This adaptive approach offers several key advantages:

- **Flexibility and Robustness:** Traditional methods often rely on fixed weighting parameters, which may not be optimal for all datasets or image complexities. Our dynamic weighting scheme continuously adapts μ, ensuring an optimal balance between feature similarity and spatial continuity throughout training. This reduces the need for extensive manual parameter tuning, enhancing the model's robustness and usability across diverse applications.
- **Improved Performance:** By adjusting μ dynamically, the model can emphasize feature similarity or spatial continuity as needed during different training stages. This adaptability leads to improved segmentation performance, as the model fine-tunes itself to the specific characteristics of the data being processed.

The dynamic adjustment of μ addresses specific challenges such as the difficulty in maintaining a consistent balance between feature similarity and spatial continuity and the sensitivity of traditional models to fixed parameter settings. By continuously evaluating and updating μ, our scheme ensures that the model remains adaptable and effective across different training states.

3.3.1. Loss function

Our methodology incorporates specific loss functions to strike a balance between feature similarity and spatial continuity—two pivotal criteria for distinguishing pixel clusters. The feature similarity loss, denoted as L_{sim} (Eq. (1)), plays a crucial role in ensuring that pixels with similar features receive identical labels. This loss quantifies the cross-entropy between the normalized response map r'_n and the corresponding cluster labels c_n. Minimizing this loss facilitates the extraction of precise attributes, significantly contributing to segmentation accuracy.

$$L_{\text{sim}}\left(r'_n, c_n\right) = -\sum_{i=1}^{N}\sum_{j=1}^{q}\delta(j-c_i)\ln r'_{i,j}, \tag{1}$$

where r'_n: normalized response; c_n: cluster labels; $\delta(t)$ is the Kronecker delta function defined as:

$$\delta(t) = \begin{cases} 1 & \text{if } t = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Conversely, the spatial continuity loss L_{con} acts as a high-pass filter, ensuring that spatially continuous pixels receive the same label. Defined by Eq. (6), L_{con} as the Manhattan Distance $L1$ Norm of horizontal and vertical differences in the response map r'_n. This mitigates the deficiencies caused by superpixels [36], efficiently removing excess labels due to complex patterns or textures and ensuring spatially continuous pixels share the same label. This loss also promotes homogeneity within each cluster, enhancing segmentation coherence and reducing noise within clusters.

$$\Delta h_{i,j} = \left|r'_n(i,j) - r'_n(i,j+1)\right| \tag{2}$$

$$\Delta v_{i,j} = \left|r'_n(i,j) - r'_n(i+1,j)\right| \tag{3}$$

$$L1_h\left(r'_n\right) = \sum_{i=1}^{H}\sum_{j=1}^{W-1}\left|\Delta h_{i,j}\right| \tag{4}$$

$$L1_v\left(r'_n\right) = \sum_{i=1}^{H-1}\sum_{j=1}^{W}\left|\Delta v_{i,j}\right| \tag{5}$$

$$L_{\text{con}}\left(r'_n\right) = L1_h\left(r'_n\right) + L1_v\left(r'_n\right) \tag{6}$$

Eqs. (2) and (3) quantify the horizontal $\left(\Delta h_{i,j}\right)$ and vertical differences $\left(\Delta v_{i,j}\right)$ between adjacent pixels in the response map r'_n, effectively capturing changes along the horizontal and vertical axes. The subsequent computations, as defined by Eqs. (4) and (5), aggregate the absolute horizontal differences and vertical differences, giving rise to measures $L1_h\left(r'_n\right)$ and $L1_v\left(r'_n\right)$. These measures represent the cumulative horizontal and vertical changes in the response map, respectively.

The culmination of these individual components is expressed in Eq. (6), where $L_{\text{con}}\left(r'_n\right)$ represents the total spatial continuity loss. This holistic measure is derived by summing both horizontal and vertical components. In essence, Eq. (6) succinctly captures the pixel-wise Manhattan distances along both axes, culminating in the $L1$ Norm. This comprehensive measure effectively encapsulates spatial continuity by considering both horizontal and vertical proximity, contributing collectively to the overarching continuity criterion.

The unsupervised segmentation loss function is represented by Eq. (7):

$$L = L_{sim}\left(\left\{r'_n, c_n\right\}\right) + \mu L_{con}\left(\left\{r'_n\right\}\right), \tag{7}$$

Here, μ represents the weight for balancing.

While the combination of feature similarity (L_{sim}) and spatial continuity (L_{con}) losses in the unified function L yields reasonably accurate unsupervised segmentation results, the parameter μ plays a crucial role and can lead to varying outcomes. The sensitivity to μ is illustrated in Fig. 2, underscoring the challenge of selecting an appropriate value. As can be seen, for $\mu = 50$ and $\mu = 100$, the segmentation is coarse, resulting in sky, buildings, and coastal regions. However, the image is further segmented with $\mu = 1$ and $\mu = 5$, where buildings are further segmented into glass buildings, concrete buildings, and different floors. Although the authors in [17] argue that the value of μ is proportional to the coarseness of segmentation, We see that the results are not consistent, e.g. the segmentation for $\mu = 50$ appears coarser than $\mu = 100$. This inconsistency poses a practical problem. The high sensitivity to the parameter means that extensive tuning is required for each dataset to obtain semantically meaningful results.

The competitive nature of both L_{sim} (Feature Similarity Loss) and L_{con} (Spatial Continuity Loss) in our methodology serves to strike a balance between two crucial aspects of unsupervised segmentation:

- **Feature Similarity (Clustering):** L_{sim} encourages pixels with similar features to be grouped into the same cluster. This is vital for capturing the intrinsic similarities among pixels and achieving meaningful clusters representing distinct objects or regions in the image.

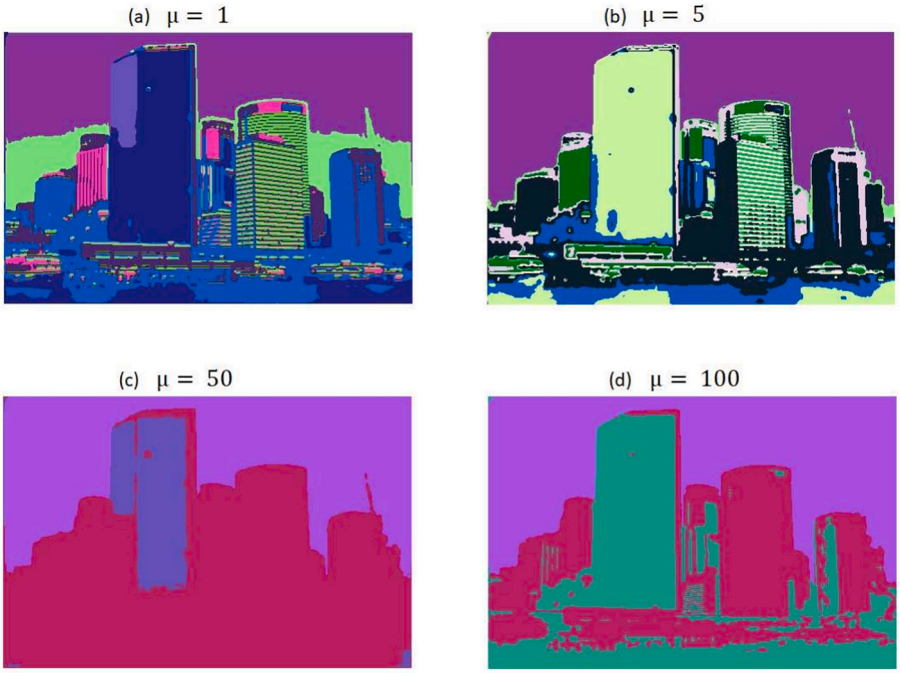

Fig. 2. Results for different μ values on a sample image from the BSD500 dataset.

- **Spatial Continuity (Smoothness):** Inversely, L_{con} ensures spatially continuous pixels receive the same label. This promotes spatial coherence in the segmentation results and contributes to the smoothness of the segmented regions.

The competitiveness arises from the fact that these two objectives may sometimes conflict. For instance, promoting spatial continuity may result in smoother but less accurate segmentation if not balanced properly with the need to capture fine-grained feature similarities. Conversely, focusing too much on feature similarities may lead to fragmented segmentation without considering the spatial arrangement of pixels. By making L_{sim} and L_{con} competitive, we implicitly guide the model to find an optimal trade-off between capturing feature similarities and maintaining spatial coherence. This competitive interaction helps the unsupervised segmentation model produce results that are both accurate in terms of content and visually coherent.

In our innovative approach to unsupervised image segmentation, we introduce a dynamic weighting scheme that addresses these inherent challenges. Recognizing the significance of adaptability during training, our method incorporates a novel dynamic adjustment for the weighting parameter μ. This adjustment responds to the evolving number of predicted clusters and ongoing training iterations, resulting in a flexible and responsive balancing weight.

This dynamic weighting scheme involves changing the weighting parameter's value during training. We observe that prioritizing feature similarity in the earlier training iterations and gradually shifting focus to spatial continuity (or vice versa) enhances adaptability. The proposed approach introduces a dynamic loss function with a continuous variable μ, where the weight μ dynamically adjusts based on the number of predicted clusters and iterations. We present and compare two versions of this innovative dynamic weighting scheme:

- ***Feature Similarity Focus (FSF)***: Initiate the training process by prioritizing the identification of continuous regions, gradually transitioning to a heightened focus on feature similarity. In this case, the performed trials lead to a linear function of the number of clusters (q') for the new dynamic balancing weight $\mu = (q'/\alpha)$ as shown in eq. (8). We tried other versions that vary exponentially with the value of q'; However, such functions resulted in a rapid change in the value of μ, which was not conducive to the balance we sought to achieve between the two constraints.

$$L_{FSF} = L_{sim}\left(\{r'_n, c_n\}\right) + (q'/\alpha)L_{con}\left(\{r'_n\}\right) \tag{8}$$

- ***Spatial Continuity Focus (SCF)***: Initiate training with a focus on feature similarity criteria, gradually transitioning to a stronger emphasis on spatial continuity. In this scenario, the proposed dynamic weight, denoted as μ, takes the form of the multiplicative inverse of the number of clusters, defined as $\mu = (\alpha/q')$ (as shown in Eq. (9)). While exploring alternative formulations, including exponential functions, we found that an exponential decay led to an overly rapid change in the value of μ, making it less effective in achieving the desired balance.

$$L_{SCF} = L_{sim}\left(\{r'_n, c_n\}\right) + (\alpha/q')L_{con}\left(\{r'_n\}\right) \tag{9}$$

3.3.2. Silhouette score phase

We introduce a critical component known as the Silhouette Score Phase to solve an issue in the existing approaches. The clustering model in [17,40] only rely on pixel similarity and spatial continuity criteria, assigning the same label to pixels with similar and spatially continuous features. This can lead to a simple solution with q' = 1, causing under-segmentation. By evaluating the compactness and separation of clusters, the Silhouette Score acts as a corrective measure to prevent such under-segmentation issues, ensuring the model achieves a balanced and meaningful segmentation outcome.

The Silhouette Score is a metric widely used in unsupervised learning tasks, particularly clustering, to quantify the goodness of a clustering technique. In the context of our unsupervised image segmentation model, the Silhouette Score Phase operates as follows: at the first iteration of processing the raw image, the Silhouette Score is calculated based on the initial cluster labels. This score serves as a valuable criterion to determine the optimal number of clusters for the given image. By leveraging the Silhouette Score, we introduce a self-regulating mechanism that prevents the model from converging to a single cluster or excessively splitting into numerous small clusters.

3.4. The clustering mechanism

A linear classifier generates a response map $r_n = W_c x_n$, initiating the clustering process. This response map undergoes a normalization step, leading to r'_n with zero mean and unit variance. Employing intra-axis normalization before applying the argmax classification for assigning cluster labels, transforms the original responses r_n into r'_n and introduces a preference for a larger q', enhancing the model's adaptability to varying image content. Consequently, the cluster label c_n for each pixel is determined by selecting the dimension with the maximum value in r'_n, a process referred to as the argmax classification. This intuitive classification rule aligns with the overarching goal of clustering feature vectors into q' clusters.

This clustering mechanism assigns each pixel to the closest representative point among the q' points, strategically placed at an infinite distance on the respective axes in the q'-dimensional space. It is noteworthy that C_i can be $0/$, allowing the number of unique cluster labels to flexibly range from 1 to q'.

In summary, the Dynamic Weighting Scheme in DynaSeg significantly enhances segmentation accuracy and flexibility by integrating several innovative approaches. The dynamically weighted loss function allows the model to adapt to different datasets without manual parameter adjustments, simplifying implementation across various segmentation networks. Adaptive clustering adjusts the number of clusters dynamically during training, enhancing the model's flexibility and robustness to accommodate the varying complexities of different datasets. Joint optimization seamlessly integrates feature extraction and clustering, with the backpropagation of clustering losses directly influencing the CNN, overcoming limitations associated with predetermined boundaries and data imbalance. The spatial continuity loss further improves segmentation accuracy by preserving high-frequency details and promoting homogeneity within each cluster, resulting in more coherent segmentations with reduced noise. DynaSeg's design eliminates the need for predefined boundaries or superpixels, providing enhanced flexibility and adaptability, making the segmentation process robust to input data variations and effective across a wide range of scenarios. Overall, these integrated features lead to more accurate, coherent, and adaptable segmentation results.

4. Experimental results

The objective of our experiments is to showcase the effectiveness of our proposed dynamic weighting scheme for achieving semantically meaningful image segmentation. We conducted extensive evaluations on multiple datasets, comparing against state-of-the-art methods.

4.1. Experiment setup

In our experiments, we introduce two versions of the feature extractor: a CNN-based extractor and a pre-trained ResNet-18 [41] with Feature Pyramid Network (FPN). For the CNN-based extractor, we set the number of components in the feature extraction phase, denoted as M, to 3. Conversely, for the ResNet18 with FPN, we modify the output channel to match the number of classes in the dataset (e.g., 27 on COCO stuff-thing dataset).

For consistency across all tests, we set the dimensions of the feature space, p, and the cluster space, q, to be equal, with both values set to 100. We employ a learning rate with a base lr = 0.1 and SGD optimizer, where the weight decay is 0.0001, and the SGD momentum optimizer is set to 0.9. The best α for SCF and FSF clustering were experimentally determined from {25, 45, **50**, 55, 60, 75,100, 200} and {2, 10, **15**, 25,50, 100}, respectively. We report results for $\alpha = 15$ for FSF clustering; $\alpha = 50$ for SCF clustering.

The mean Intersection Over Union (*mIOU*) is reported for the benchmark datasets. It's important to note that ground truth is utilized solely during the assessment phase and plays no role in the training process.

4.1.1. COCO-stuff

In accordance with the methodology outlined in [16,33,34], we evaluate the performance of our model using the COCO-Stuff dataset [44]. This dataset is distinguished for its extensive collection of scene-centric images, featuring 80 thing and 91 stuff categories. The model is evaluated on curated subsets [16,33] of the COCO val2017 split, consisting of 2175 images, Following the preprocessing procedure detailed in [33], we amalgamate classes to establish 27 categories, comprising 15 stuff and 12 things. It's noteworthy that, unlike earlier studies focusing solely on stuff categories, our evaluation encompasses both things and stuff categories.

4.1.2. BSD500

Additionally, we utilize the Berkley Segmentation Dataset BSD500 [45] and PASCAL Visual Object Classes 2012 [46] for both quantitative and qualitative evaluation of the segmentation results. The BSD500 dataset comprises 500 color and grayscale natural images. Following the experimental setup in [17], we use the 200 color images from the BSD500 test set to evaluate all models.

4.1.3. Pascal VOC2012

For PASCAL VOC2012 [46], we treat each segment as an individual entity, disregarding object classification. The VOC2012 dataset is expansive, containing 17,124 images, of which 2913 have semantic segmentation annotations. We use the 2913 semantic segmentation images for evaluating our method. In addition, we utilize select images from the Icoseg [47] and Pixabay [48] datasets to present qualitative results.

4.2. Evaluation

To assess the performance of the model trained without labels, a correspondence between the model's label space and the ground truth categories must be established. Initially, the model predicts on each image within the validation set. Subsequently, the confusion matrix is computed between the predicted labels and the ground truth classes. Employing linear assignment, we establish a one-to-one mapping between the predicted labels and ground truth classes, with the confusion matrix serving as the assignment cost. The mean Intersection over Union (mIoU) is then calculated over all classes based on this mapping [33,34]. For a more comprehensive analysis of the model's behavior, we present the mean IoU values for stuff and things classes.

Given the multiple ground truth types in BSD500, we adopt three mIOU counting strategies for assessment: "BSD500 All" considers all ground truth files, "BSD500 Fine" focuses on the ground truth file with the most significant number of segments, and "BSD500 Coarse" considers the ground truth file with the smallest number of segments. We define "BSD500 Mean" as the average of these three measurements.

4.3. Quantitative results

The performance of various state-of-the-art methods in unsupervised semantic segmentation is comprehensively evaluated on multiple datasets.

4.3.1. COCO-all dataset

Table 2 presents the results on the COCO-All dataset, comparing mean Intersection over Union (mIoU) scores. Our proposed DynaSeg models, particularly the Feature Similarity Focus (FSF) and Spatial Continuity Focus (SCF) versions, exhibit significantly improved performance compared to existing methods. Notably, DynaSeg - SCF with ResNet-18 and FPN Extractor achieves the highest mIoU of 30.52, surpassing benchmarks like DenseSiam and Picie by substantial margins

(14.12 and 16.08, respectively). This establishes our approach as the new state-of-the-art in unsupervised semantic segmentation.

4.3.2. COCO-stuff dataset

Table 3 summarizes the results for the COCO-stuff dataset, considering different backbones and architectures. SCF with CNN-Based achieves an mIoU Stuff of 42.41 and pAcc of 76.7, demonstrating its effectiveness. However, FSF with ResNet-18 FPN emerges as the new state-of-the-art, surpassing all methods with the highest mIoU Stuff of 54.10 and pAcc of 81.1. These results are particularly noteworthy because they surpass the performance of other models, including those utilizing Vision Transformer (ViT) architectures. For example, methods like STEGO [49] and DINO + CAUSE-TR [50], which are built on ViT-B/16 and ViT-B/8 backbones respectively, do not achieve the same level of performance as our DynaSeg models. Specifically, DINO + CAUSE-TR, which previously set a high benchmark with an mIoU Stuff of 41.9, is significantly outperformed by our DynaSeg - FSF model.

The exceptional performance of the ResNet-18 FPN integration in our model highlights its ability to enhance feature extraction through multi-resolution processing. This capability allows our model to capture details at different scales more effectively, which is crucial for achieving superior segmentation performance, especially on datasets with diverse object scales and complexities. This underscores the advantage of our approach in leveraging the strengths of both ResNet and FPN, even against advanced transformer-based models.

4.3.3. BSD500 and PASCAL VOC2012 datasets

The quantitative results on the BSD500 and PASCAL VOC2012 datasets are detailed in Table 1. DynaSeg's Spatial Continuity Focus (SCF) achieves a remarkable mIOU of 0.396 on PASCAL VOC2012, the highest among all methods compared, demonstrating its effectiveness in managing complex segmentation challenges. This performance underscores DynaSeg's adept handling of spatial details and continuity, which are crucial for achieving high-quality segmentation results. DynaSeg's Feature Similarity Focus (FSF) not only consistently outperforms other methods across the entire BSD500 dataset but also specifically surpasses the Graph-based Segmentation method on the BSD500 Fine dataset, showcasing its robustness and effectiveness in maintaining feature homogeneity across diverse segmentation tasks. This highlights FSF's distinct advantage over both traditional approaches like Graph-based Segmentation and recent innovations such as Differentiable Double Clustering with Edge-Aware Superpixel Fitting (DDCESF) [51], and other state-of-the-art techniques. In addition to quantitative evaluations, we provide a demo [52] of incremental segmentation using DynaSeg on a sample image from the Pascal VOC2012 dataset. This demo, available in the supplemental video, effectively showcases the practical applications of DynaSeg to real-world images, illustrating its capabilities beyond traditional and contemporary unsupervised segmentation methods.

In summary, the proposed FSF and SCF methods exhibit superior performance across different datasets and evaluation metrics. The achieved results reinforce their effectiveness, establishing them as prominent approaches in unsupervised semantic segmentation.

Table 1
Comparison of *mIOU* for unsupervised segmentation on BSD500 and PASCAL VOC2012. Best scores are in bold.

Method	dataset				
	BSD500 All	*BSD500 Fine*	*BSD500 Coarse*	*BSD500 Mean*	*PASCAL VOC2012*
IIC [16]	0.172	0.151	0.207	0.177	0.273
k-means clustering	0.240	0.221	0.265	0.242	0.317
Graph-based Segmentation [11]	0.313	0.295	0.325	0.311	0.365
CNN-based + superpixels [36]	0.226	0.169	0.324	0.240	0.308
CNN-based + weighted loss, $\mu = 5$ [17]	0.305	0.259	0.374	0.313	0.352
Self-supervised Multi-view Clustering [53]	0.316	0.266	0.391	0.339	0.383
Double Clustering with Superpixel Fitting [51]	0.338	0.291	0.385	0.348	0.376
DynaSeg - Spatial Continuity Focus (SCF)	0.330	0.290	0.407	0.342	**0.396**
DynaSeg - Feature Similarity Focus (FSF)	**0.349**	**0.307**	**0.420**	**0.359**	0.391

4.4. Computational efficiency experiments

To empirically demonstrate the computational efficiency of DynaSeg, we conducted experiments to compare the computational resources required by different segmentation methods. We measured the total parameters, and floating point operations per second (FLOPs) for each method. The results are presented in Table 4.

The results demonstrate that DynaSeg with both CNN-based and ResNet-18 backbones maintains competitive computational efficiency, with lower parameter counts and FLOPs compared to some state-of-the-art methods. Notably, the CNN-based version of DynaSeg demonstrates a good balance between computational cost and performance, making it suitable for applications requiring efficient processing. The results demonstrate that DynaSeg with ResNet-18 is highly efficient, requiring significantly fewer parameters and FLOPs compared to other state-of-the-art methods. For instance, DynaSeg (ResNet-18) uses only 1.84 GFLOPs and 12,046,272. This is significantly lower than methods such as DINO + HP [58], which demands 164.15 GFLOPs and 39,641,952 parameters, and STEGO [49], which requires 67.42 GFLOPs and 86,614,089 parameters. The considerable reduction in computational requirements underscores DynaSeg's capability to deliver effective performance with much lower computational costs, making it particularly suitable for resource-constrained environments. When examining the CNN-based version of DynaSeg, it maintains a competitive edge with only 193,900 parameters and 9.75 GFLOPs. Although methods like PiCIE [33] and DenseSiam [34] show lower GFLOPs at 4.32 and 4.38 respectively, these efficiencies are achieved through the use of pretrained models, which DynaSeg does not utilize. This distinction highlights the robustness and inherent efficiency of DynaSeg, as it achieves comparable or better performance without the need for extensive pretraining. The CNN-based DynaSeg, therefore, strikes an excellent balance between computational cost and segmentation performance, further validating its practical applicability.

In summary, the analysis reveals that DynaSeg, in both its ResNet-18 and CNN-based configurations, offers substantial computational advantages over other leading methods. The ResNet-18 version stands out for its minimal computational demands, positioning it as an optimal choice for various segmentation tasks where efficiency is paramount.

Table 2
Comparison of mIOU for unsupervised segmentation on COCO-All.

Method	Backbone	mIoU All
Modified DC [54]	–	9.8
IIC [16]	ResNet18	6.7
Picie [33]	ResNet18	14.4
DenseSiam [34]	ResNet18	16.4
DynaSeg - FSF	CNN-Based	30.51
DynaSeg - SCF	CNN-Based	27.57
DynaSeg - FSF	ResNet-18 + FPN	30.07
DynaSeg - SCF	**ResNet-18 + FPN**	**30.52**

Table 3
Comparison of different methods based on their mean Intersection over Union (mIoU) for'Stuff' categories and pixel accuracy (pAcc) on COCO-Stuff.

Method	Backbone	mIoU Stuff	pAcc
IIC [16]	ResNet18	27.7	21.8
Picie [33]	ResNet18	31.48	50.0
DenseSiam [34]	ResNet18	24.5	–
HSG [55]	ResNet50	23.8	57.6
ReCo+ [56]	DeiT-B/8	32.6	54.1
STEGO [49]	ViT-B/16	23.7	52.5
DINO + ACSeg [57]	ViT-B/8	16.4	–
DINO + HP [58]	ViT-B/8	24.6	57.2
DINO + CAUSE-TR [50]	ViT-B/8	41.9	74.9
DynaSeg - SCF	CNN-Based	42.41	76.7
DynaSeg - SCF	ResNet-18 FPN	42.41	76.7
DynaSeg - FSF	CNN-Based	42.37	76.6
DynaSeg - FSF	**ResNet-18 FPN**	**54.10**	**81.1**

Table 4
Comparison of computational efficiency of different methods.

Method	Total Parameters	GFLOPs
DINO + HP [58]	39,641,952	164.15
DINO_Vit_base_16 [59]	86,415,592	16.86
IIC [16]	4,521,024	17.94
PiCIE [33]	23,631,424	4.32
DenseSiam [34]	23,741,558	4.38
STEGO [49]	86,614,089	67.42
HSG [55]	27,968,704	7.13
DynaSeg (CNN_based)	193,900	9.75
DynaSeg (ResNet-18)	12,046,272	1.84

Table 5
Silhouette score impact.

Framework	Backbone	Thr	mIoU		
			All	Things	Stuff
DynaSeg - FSF	CNN-Based	3	26.03	61.73	34.82
DynaSeg - FSF	CNN-Based	Silh.S	**30.51**	74.10	**42.39**
DynaSeg - SCF	ResNet-18	3	30.52	74.06	42.41
DynaSeg - SCF	ResNet-18	Silh.S	30.52	74.06	42.37

Table 6
Comparison of framework and backbone variations.

Framework	Backbone	mIoU			pAcc
		All	Things	Stuff	
DynaSeg - FSF	CNN-Based	**30.51**	74.10	42.39	76.65
DynaSeg - FSF	ResNet-18	30.07	62.87	**54.10**	**81.08**
DynaSeg - SCF	CNN-Based	27.57	63.55	35.08	76.34
DynaSeg - SCF	ResNet-18	**30.52**	74.06	42.37	76.67

Meanwhile, the CNN-based version provides a balanced approach, achieving efficiency and high performance independently of pre-trained models. These findings collectively affirm DynaSeg's effectiveness as a highly efficient and robust solution for unsupervised image segmentation, capable of meeting diverse application requirements while maintaining low computational overhead.

4.5. Qualitative results

We also provide qualitative results on a few images as done in [17]. The qualitative evaluation showcases the segmentation results for the Spatial Continuity Focus (SCF) method, comparing it with the Differentiable Feature-based Segmentation model (Diff) [17] and the ground truth. In Fig. 4, SCF exhibits its strength in accurately capturing complex details of the cat, including fine contours, legs, and tail. The method achieves high-resolution segmentation with nuanced object boundaries, aligning effectively with the ground truth.

Notably, SCF outperforms Diff by providing a clean and accurate representation of the object. Diff, on the other hand, introduces noise in the background and struggles to accurately segment the tail of the cat, often blending it with the background. This emphasizes SCF's robust performance in preserving true object shapes and minimizing unwanted artifacts, making it particularly effective in complex scenes with detailed structures.

Additionally. As shown in Fig. 3, our model is more effective in bringing out segmentation regions that are semantically related. For example, For the "Show Jumping" image (column 5), the horse and the obstacle are classified as the same class by [17] (both yellow). However, for both FSF and SCF, the horse and the obstacle are appropriately distinguished. For the "ship" image (column 2), [17] fails to differentiate between the sky and the body of the ship (both red), but both proposed SCF and FSF can do it successfully.

Further qualitative results are shown for select Icoseg [47] and Pixabay [48] datasets that were also used in [17]. These results can be seen in Fig. 5. The qualitative results on these datasets are presented to further demonstrate that our proposed approaches do not require as much parameter tuning as [17] does. Fig. 5 highlights that the baseline Differentiable Feature Clustering [17] is quite parameter sensitive. For each dataset, the weighting balance μ must be tuned extensively to obtain a more semantically meaningful result. For instance, PASCAL VOC 2012 and BSD500 datasets require a small balancing value $\mu = 5$. While, Icoseg [47] and Pixabay [48] datasets need a much larger balance value $\mu = 50$ and $\mu = 100$, respectively. On the other hand, As illustrated in Figs. 3 and 5, our proposed method has proven effective in dealing with different datasets using the same weight for both FSF and SCF. The proposed methods also bring out details in an unsupervised manner that is semantically more meaningful. For example, using the Feature Similarity Focus method (FSF), the red car from the iCoseg dataset (column 2 row 4 in Fig. 5) displays more detail on the tires and more precise building outlines than the details extracted by [17] (column 2 row 2), where the car is partially blended into the road. Similarly, for the peppers image (column 4 in Fig. 5), [17] was unable to identify the shapes of the individual peppers accurately. Both of our proposed method do a much better job, even with the same value of μ as the other images.

In the comparative analysis of the proposed SCF and FSF methods, SCF exhibits notable proficiency in class segmentation, excelling in delineating distinct categories. Conversely, FSF demonstrates superior performance in instance segmentation tasks. Fig. 3 column 3 illustrates this distinction. SCF accurately segments the image into well-defined classes such as water, the surface of the sea, and sea coral. On the other hand, FSF not only identifies these classes but also provides more details, capturing the nuances of sea coral structures, including the main trunk and branches. Notably, in Fig. 3 column 1, FSF showcases its capability by revealing fine details of butterfly wing veins and forewing structures with remarkable precision. This precision makes FSF particularly well-suited for applications in the medical domain. The choice between SCF and FSF should be guided by the specific requirements of the given application, considering the emphasis on either class segmentation or detailed object representation.

The experimental results clearly demonstrate the superiority of our proposed dynamic weighting scheme over existing methods. The SCF and FSF methods consistently outperform other techniques in both quantitative and qualitative assessments. Specifically, the adaptability of the proposed approach to varying datasets and scenarios is evident in the significant improvements achieved in mIoU scores across different datasets and evaluation metrics.

5. Ablation study

This section presents supplementary results examining the influence

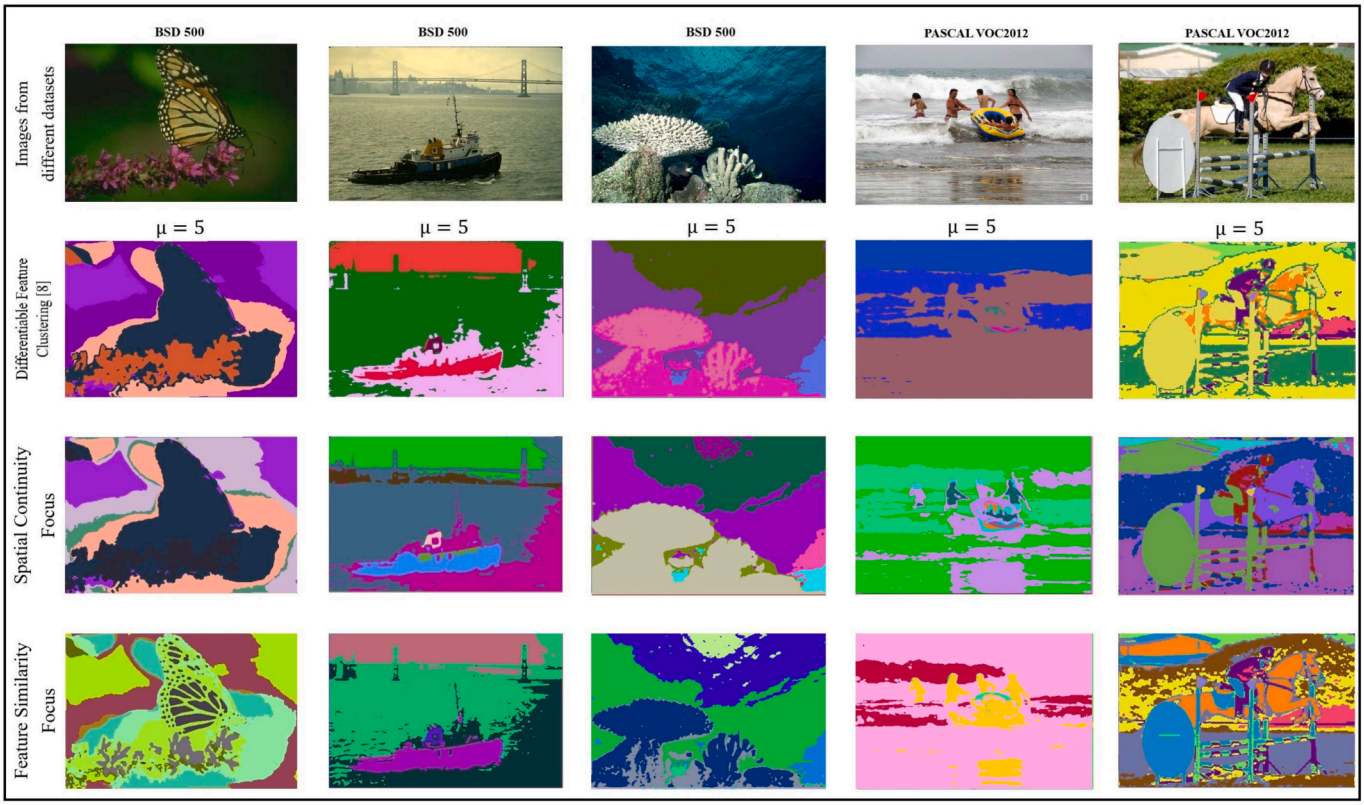

Fig. 3. Qualitative Results on select BSD500 and PASCAL VOC2012 images. Same color corresponds to the pixels being assigned the same clustering label by the algorithm. Please read Section 4.5 for discussion on these results.

of the Silhouette Score Phase and the Feature Extractor through ablation on the COCO dataset. Additionally, we provide a comparison between FSF and SCF to further analyze their respective contributions. Furthermore, the integration of ResNet with the Feature Pyramid Network (FPN) significantly enhances feature extraction, leading to superior segmentation performance. This contribution highlights the effectiveness of our approach, particularly in handling objects at different scales and complexities.

5.1. Silhouette score impact

To evaluate the impact of integrating the Silhouette Score into our proposed dynamic weighting scheme, we conducted an ablation study, comparing segmentation results with and without the inclusion of Silhouette Score. Table 5 presents the comparative results across different configurations.

In the case of the FSF method with a CNN-based backbone, integrating Silhouette Score at threshold 3 significantly improves the mIoU for "All" from 26.03 to **30.51** and for "Stuff" from 34.82 to **42.39**, demonstrating substantial enhancements.

For the SCF method with a ResNet-18 backbone, the segmentation results remain consistent for "All," "Things," and "Stuff" across both threshold 3 and Silh.S configurations, indicating that the model itself maintains the number of clusters, which doesn't reach the specified threshold. This implies that the SCF method inherently adjusts its clustering strategy, rendering the additional Silhouette Score less influential in this scenario. This nuanced understanding of the SCF behavior underscores the adaptability of the model and its ability to maintain segmentation quality without explicit reliance on external thresholds. It showcases the model's self-adjusting clustering mechanism, which may result in consistent mIoU values in certain scenarios.

The accompanying graph in Fig. 6 provides additional insights into the distribution of ground truth cluster numbers, complementing the tabular results. The graph illustrates a significant disparity when comparing the distribution with a random threshold of 3. Conversely, when comparing the distribution with the optimal number of clusters derived from the Silhouette Score, a strong correlation is observed. This finding underscores the effectiveness of the proposed dynamic weighting scheme.

This combined analysis emphasizes the significance of incorporating the Silhouette Score in our framework, showcasing its positive influence on segmentation accuracy across different scenarios and backbone architectures.

5.2. Effect of feature extractor

To analyze the impact of the choice of feature extractor, we compare the performance of CNN-Based and ResNet-18 with FPN in the FSF and SCF methods. Table 6 comprehensively compares segmentation results across different combinations of frameworks and backbones.

For the FSF method, the choice of the feature extractor has a substantial impact on segmentation performance. When using a CNN-Based backbone, the mIoU for "All" is 30.52, and for "Stuff," it is 42.39. In contrast, with a ResNet-18 backbone, the mIoU for "All" decreases slightly to 30.07, but there is a notable improvement in "Stuff" with a mIoU of 54.10. This suggests that the ResNet-18 backbone with FPN captures semantic information related to "Stuff" categories more effectively.

Similarly, for the SCF method, the impact of the feature extractor is evident. With a CNN-Based backbone, the mIoU for "All" is 27.57, and for "Stuff," it is 35.08. Switching to a ResNet-18 backbone results in an improvement, with the mIoU for "All" reaching 30.52 and for "Stuff" reaching 42.37. This highlights that the ResNet-18 backbone, even with its deeper architecture, contributes to better segmentation performance for both the "All" and "Stuff" categories.

The pixel accuracy (pAcc) values further complement these findings,

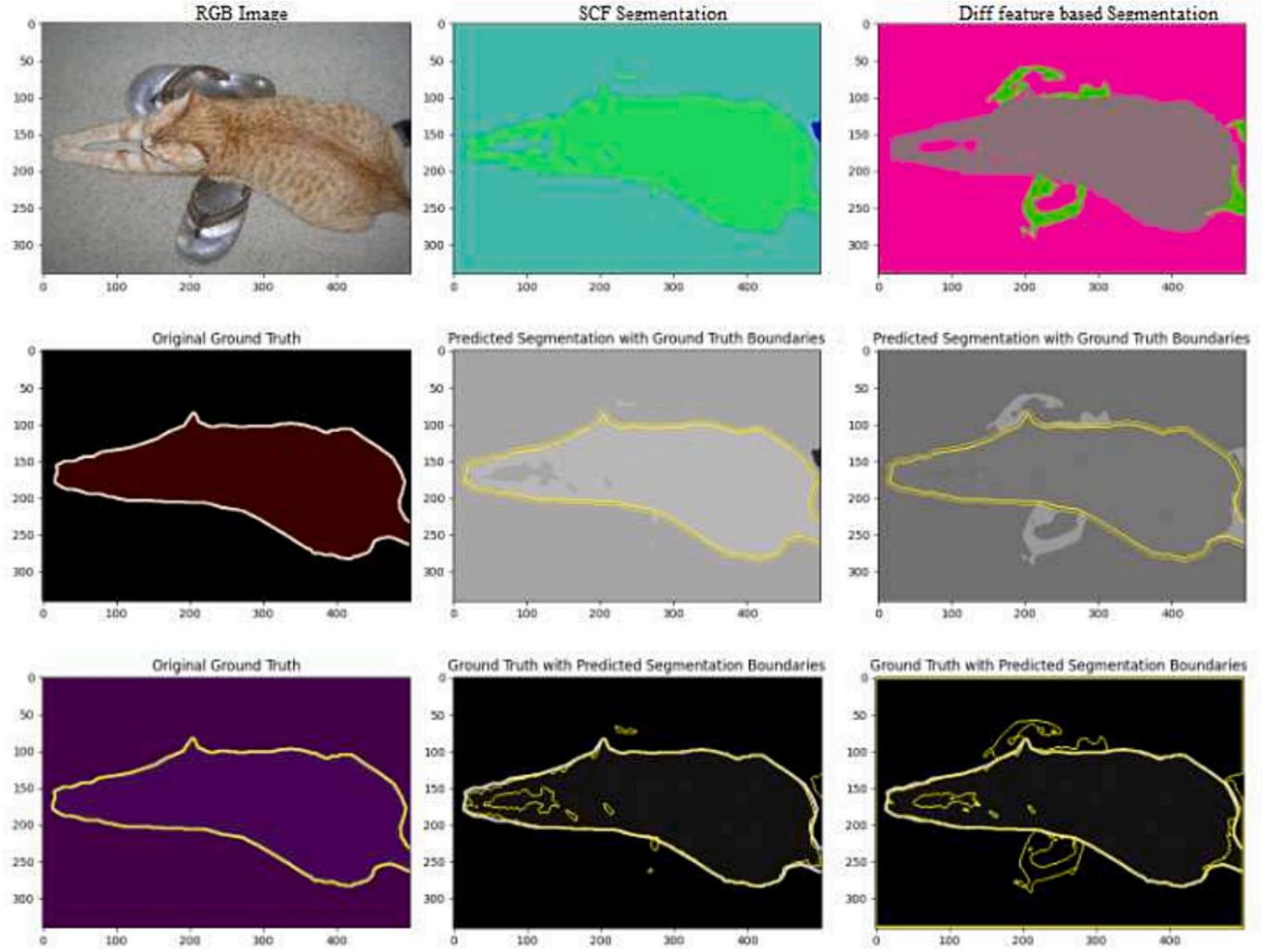

Fig. 4. Qualitative results on Pascal VOC 2012: Original image, DynaSeg - SCF predicted segmentation, and Diff predicted segmentation.

showcasing the overall accuracy of pixel-level predictions. In summary, the choice of feature extractor, particularly the ResNet-18 backbone, plays a crucial role in enhancing segmentation performance, especially for specific categories like "Stuff."

5.3. Comparison between FSF and SCF

To gain insights into the differences between the Feature Similarity Focus (FSF) and Spatial Continuity Focus (SCF) methods, we conduct a comparative analysis using Table 6.

For the FSF method, with a CNN-Based backbone, the mIoU for "All" is 30.52, and for "Stuff," it is 42.39. On the other hand, the ResNet-18 backbone yields an mIoU of 30.07 for "All" and an impressive 54.10 for "Stuff." This indicates that FSF, particularly with a ResNet-18 backbone, excels in capturing semantic information related to complex and varied categories, such as "Stuff."

Moving to the SCF method, the mIoU for "All" with a CNN-Based backbone is 27.57, and for "Stuff," it is 35.08. Transitioning to a ResNet-18 backbone results in an improved mIoU of 30.52 for "All" and 42.37 for "Stuff." This suggests that SCF, like FSF, benefits from a more sophisticated backbone, and the ResNet-18 architecture contributes to better segmentation performance, especially for challenging categories like "Stuff."

To further refine the comparative analysis between the Feature Similarity Focus (FSF) and Spatial Continuity Focus (SCF) methods, we consider additional mIOU scores on BSD500 and PASCAL VOC2012 datasets, as presented in Table 1. In the BSD500 dataset, across different segmentation scenarios (All, Fine, and Coarse), FSF consistently outperforms SCF in terms of mIOU scores. Notably, for BSD500 All, FSF achieves an mIOU of 0.349 compared to SCF's 0.330. This pattern is observed in other scenarios, reinforcing the effectiveness of FSF in capturing fine-grained details and semantic nuances. On the PASCAL VOC2012 dataset, both FSF and SCF demonstrate competitive performance, with FSF achieving an mIOU of 0.391 and SCF's 0.396. The marginal difference in mIOU scores on PASCAL VOC2012 suggests that both methods perform comparably on this specific dataset, indicating their capability to handle diverse segmentation challenges effectively.

Comparing the two methods across different datasets and segmentation scenarios, FSF consistently delivers higher mIoU scores, indicating its robustness and effectiveness in capturing both global and fine-grained semantic information. The choice of FSF over SCF is particularly beneficial when targeting scenarios with varied and intricate segmentation requirements. Both FSF and SCF perform better with a ResNet-18 backbone compared to CNN-based backbones. FSF, in particular, achieves higher mIoU values for "Stuff" across both backbones, emphasizing its effectiveness in capturing detailed and intricate features. Therefore, FSF shows promise for scenarios involving complex and diverse categories, especially when coupled with a ResNet-18 backbone.

6. Conclusion

Our study introduces a state-of-the-art unsupervised image

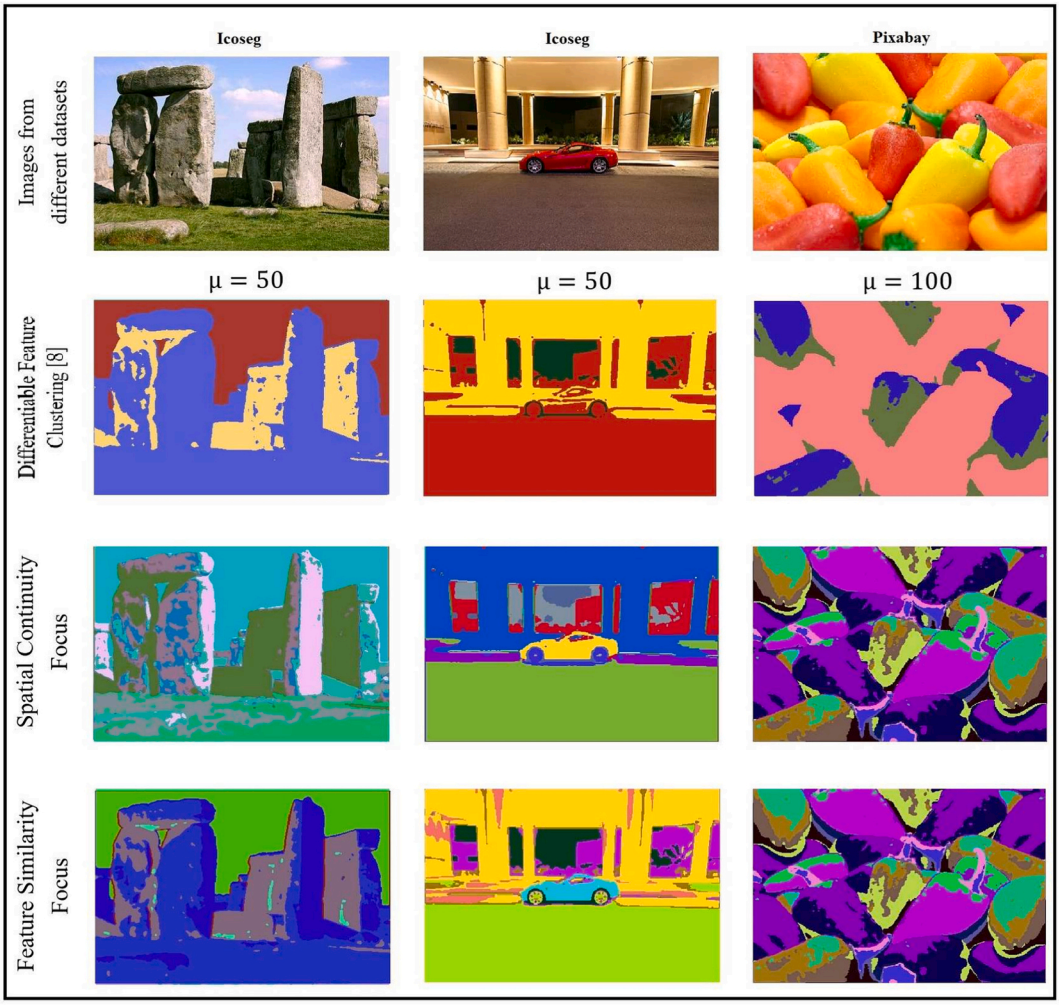

Fig. 5. Qualitative Results on select Icoseg and Pixabay images. Same color corresponds to the pixels being assigned the same clustering label by the algorithm. Please read Section 4.5 for discussion on these results.

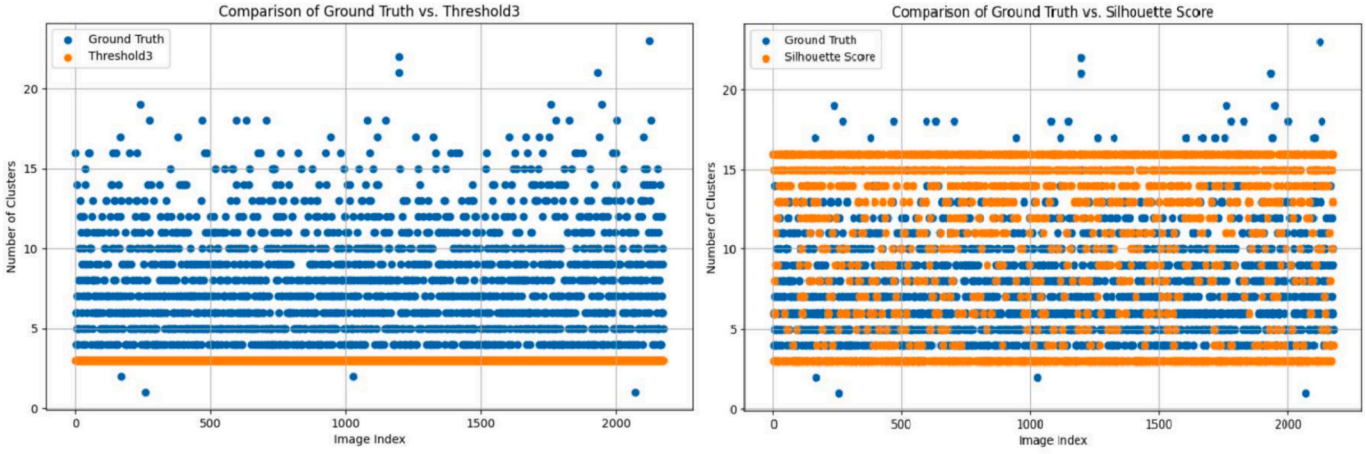

Fig. 6. Comparison of cluster distribution: Left graph shows the comparison of the distribution of the number of clusters in the ground truth (blue) to a fixed threshold of three clusters (orange). Right graph displays the distribution of the number of clusters in the ground truth (blue) and the number of clusters predicted by the silhouette score (orange). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

segmentation approach. The dynamic weighting scheme, Silhouette Score Phase, and integration of a pre-trained ResNet feature extraction and Feature Pyramid Network (FPN) decoding collectively contribute to superior performance. The dynamic weighting scheme, in particular, enhances segmentation accuracy and flexibility by dynamically adjusting loss weights and cluster numbers during training, simplifying implementation and improving adaptability across diverse datasets. Additionally, the joint optimization framework and spatial continuity

loss promote coherent and balanced segmentations by preserving high-frequency details and ensuring homogeneity within clusters. Our method achieves high segmentation accuracy without extensive parameter tuning, showcasing its adaptability across diverse datasets.

Supplementary data to this article can be found online at https://doi.org/10.1016/j.imavis.2024.105206.

CRediT authorship contribution statement

Boujemaa Guermazi: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Riadh Ksantini:** Writing – review & editing, Supervision. **Naimul Khan:** Writing – review & editing, Supervision, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors acknowledge financial support from the Natural Sciences and Engineering Research Council of Canada (#RGPIN-2020-05471).

References

[1] K. Ramesh, G.K. Kumar, K. Swapna, D. Datta, S.S. Rajest, A review of medical image segmentation algorithms, EAI Endors. Trans. Pervas. Health Technol. 7 (27) (2021) e6.
[2] L. Zhao, H. Zhou, X. Zhu, X. Song, H. Li, W. Tao, Lif-seg: Lidar and camera image fusion for 3d lidar semantic segmentation, IEEE Trans. Multimed. 26 (2023) 1158–1168.
[3] I. Ruban, V. Khudov, O. Makoveichuk, H. Khudov, I. Khizhnyak, A swarm method for segmentation of images obtained from on-board optoelectronic surveillance systems, in: 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), IEEE, 2018, pp. 613–618.
[4] Ç. Kaymak, A. Uçar, A brief survey and an application of semantic image segmentation for autonomous driving, Handb. Deep Learn. Appl. (2019) 161–200.
[5] J. Sánchez, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the fisher vector: theory and practice, Int. J. Comput. Vis. 105 (2013) 222–245.
[6] G. Gao, G. Xu, J. Li, Y. Yu, H. Lu, J. Yang, Fbsnet: a fast bilateral symmetrical network for real-time semantic segmentation, IEEE Trans. Multimed. 25 (2022) 3273–3283.
[7] A.M. Hafiz, G.M. Bhat, A survey on instance segmentation: state of the art, Int. J. Multimed. Inf. Retr. 9 (3) (2020) 171–189.
[8] A. Kirillov, K. He, R. Girshick, C. Rother, P. Dollár, Panoptic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9404–9413.
[9] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, Int. J. Comput. Vis. 1 (4) (1988) 321–331, https://doi.org/10.1007/BF00133570.
[10] J, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability 1(14), 1967, pp. 281–297.
[11] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient graph-based image segmentation, Int. J. Comput. Vis. 59 (2) (2004) 167–181, https://doi.org/10.1023/B:VISI.0000022288.19776.77.
[12] Z. Liu, R. Shi, L. Shen, Y. Xue, K.N. Ngan, Z. Zhang, Unsupervised salient object segmentation based on kernel density estimation and two-phase graph cut, IEEE Trans. Multimed. 14 (4) (2012) 1275–1289.
[13] K. He, G. Gkioxari, P. Doll´ar, R. Girshick, Mask r-cnn, Mask r-cnn, in: Proceedings ofthe IEEE international conference on computer vision, 2017, pp. 2961–2969.
[14] S.E. Mirsadeghi, A. Royat, H. Rezatofighi, Unsupervised image segmentation by mutual information maximization and adversarial regularization, IEEE Robot. Automat. Lett. 6 (4) (2021) 6931–6938, https://doi.org/10.1109/LRA.2021.3095311.
[15] L. Zhou, W. Wei, DIC: deep image clustering for unsupervised image segmentation, IEEE Access 8 (2020) 34481–34491, https://doi.org/10.1109/ACCESS.2020.2974496.
[16] X. Ji, J.F. Henriques, A. Vedaldi, Invariant information clustering for unsupervised image classification and segmentation, in: Proceedings of the IEEE/CVF International Conference On Computer Vision, 2019, pp. 9865–9874.
[17] W. Kim, A. Kanezaki, M. Tanaka, Unsupervised learning of image segmentation based on differentiable feature clustering, IEEE Trans. Image Process. 29 (2020) 8055–8068. arXiv:2007.09990. https://doi.org/10.1109/TIP.2020.3011269.
[18] N. Navab, J. Hornegger, W.M. Wells, A. Frangi, Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III vol. 9351, Springer, 2015.
[19] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495.
[20] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 834–848.
[21] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ade20k dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 633–641.
[22] Y. LeCun, C. Cortes, C. Burges, et al., Mnist handwritten digit database, 2010.
[23] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Tech. rep, University of Toronto, 2009.
[24] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 603–619.
[25] A.L. Rezaabad, S. Vishwanath, Learning representations by maximizing mutual information in variational autoencoders, in: 2020 IEEE International Symposium on Information Theory (ISIT), IEEE, 2020, pp. 2729–2734.
[26] Y. Ouali, C. Hudelot, M. Tami, Autoregressive unsupervised image segmentation, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16, Springer, 2020, pp. 142–158.
[27] C. Han, L. Rundo, R. Araki, Y. Nagano, Y. Furukawa, G. Mauri, H. Nakayama, H. Hayashi, Combining noise-to-image and image-to-image GANs: brain MR image augmentation for tumor detection, IEEE Access 7 (2019) 156966–156977. arXiv:1905.13456. https://doi.org/10.1109/ACCESS.2019.2947606.
[28] J.J. Hwang, S. Yu, J. Shi, M. Collins, T.J. Yang, X. Zhang, L.C. Chen, SegSort: Segmentation by discriminative sorting of segments, in: Proceedings of the IEEE International Conference on Computer Vision 2019-Octob, 2019, pp. 7333–7343, arXiv:1910.06962, https://doi.org/10.1109/ICCV.2019.00743.
[29] L. Melas-Kyriazi, C. Rupprecht, I. Laina, A. Vedaldi, Finding an unsupervised image segmenter in each of your deep generative models, arXiv preprint arXiv:2105.08127, 2021.
[30] J. Donahue, K. Simonyan, Large scale adversarial representation learning, in: Advances in Neural Information Processing Systems 32 (NeurIPS), 2019, pp. 1–11, arXiv:1907.02544.
[31] R. Abdal, P. Zhu, N.J. Mitra, P. Wonka, Labels4Free: unsupervised segmentation using StyleGAN, in: Proceedings of the IEEE International Conference on Computer Vision, 2021, pp. 13950–13959, arXiv:2103.14968, https://doi.org/10.1109/ICCV48922.2021.01371.
[32] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, StyleGANv2, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2020, pp. 8107–8116, arXiv:1912.04958.
[33] J.H. Cho, U. Mall, K. Bala, B. Hariharan, Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16794–16804.
[34] W. Zhang, J. Pang, K. Chen, C.C. Loy, Dense siamese network for dense unsupervised learning, in: European Conference on Computer Vision, Springer, 2022, pp. 464–480.
[35] X. Xia, B. Kulis, W-Net: A Deep Model for Fully Unsupervised Image Segmentation, arXiv. arXiv:1711.08506. URL, http://arxiv.org/abs/1711.08506, Nov 2017.
[36] A. Kanezaki, Unsupervised Image Segmentation by Backpropagation, Icassp, 2018, pp. 2–4.
[37] S. Du, N. Bayasi, G. Hamarneh, R. Garbi, Mdvit: Multi-domain vision transformer for small medical image segmentation datasets, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2023, pp. 448–458.
[38] N. Mrabah, N.M. Khan, R. Ksantini, Z. Lachiri, Deep clustering with adynamic autoencoder: From reconstruction towards centroids construc-tion, Neural Networks 130 (2020) 206–228.
[39] W. Jiang, Y. Wu, L. Guan, J. Zhao, Dfnet: Semantic segmentation onpanoramic images with dynamic loss weights and residual fusion block, in: in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 5887–5892.
[40] B. Guermazi, R. Ksantini, N. Khan, A dynamically weighted loss function for unsupervised image segmentation, in: 2022 IEEE Eighth International Conference on Multimedia Big Data (BigMM), IEEE, 2022, pp. 73–78.
[41] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[42] J. He, Z. Deng, L. Zhou, Y. Wang, Y. Qiao, Adaptive pyramid contextnetwork for semantic segmentation, in: in: Proceedings of the IEEE/CVFconference on computer vision and pattern recognition, 2019, pp. 7519–7528.
[43] S. Seferbekov, V. Iglovikov, A. Buslaev, A. Shvets, Feature pyramid net-work for multi-class land segmentation, in: in: Proceedings of the IEEE con-ference on computer vision and pattern recognition workshops, 2018, pp. 272–275.

[44] H. Caesar, J. Uijlings, V. Ferrari, Coco-stuff: Thing and stuff classes in context, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1209–1218.
[45] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proc. 8th Int'l Conf. Computer Vision vol. 2, 2001, pp. 416–423.
[46] M. Everingham, M. Everingham, A. Zisserman, A. Zisserman, C. Williams, C. Williams, The PASCAL visual object classes challenge 2006 (VOC2006) results, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[47] D. Batra, A. Kowdle, D. Parikh, J. Luo, T. Chen, icoseg: Interactive co-segmentation with intelligent scribble guidance, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3169–3176.
[48] A. Kanezaki, Unsupervised image segmentation by backpropagation, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 1543–1547.
[49] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, W.T. Freeman, Unsupervised semantic segmentation by distilling feature correspondences, arXiv preprint arXiv: 2203.08414, 2022.
[50] J. Kim, B.-K. Lee, Y.M. Ro, Causal unsupervised semantic segmentation, arXiv preprint arXiv:2310.07379, 2023.
[51] X. Li, X. Chen, Y. Qiu, C. Tao, P. Zheng, Differentiable double clusteringwith edge-aware superpixel fitting for unsupervised image, segmentation,Displays 83 (2024) 102721.
[52] B. Guermazi, Dynaseg-scf, Online Video, accessed: March 15, 2024. URL, https://drive.google.com/file/d/1qItjtMB9Gf-3opqYb-kaM1_7MogQfctN/view?usp=drive_link, 2024.
[53] T. Fang, Z. Liang, X. Shao, Z. Dong, J. Li, Self-supervised multi-viewclustering for unsupervised image segmentation, in: Artificial Neural Net-works and Machine Learning–ICANN 2021, 30, Springer, 2021, pp. 113–125.
[54] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 132–149.
[55] T.-W. Ke, J.-J. Hwang, Y. Guo, X. Wang, S.X. Yu, Unsupervised hierarchical semantic segmentation with multiview cosegmentation and clustering transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2571–2581.
[56] G. Shin, W. Xie, S. Albanie, Reco: retrieve and co-segment for zero-shot transfer, Adv. Neural Inf. Proces. Syst. 35 (2022) 33754–33767.
[57] K. Li, Z. Wang, Z. Cheng, R. Yu, Y. Zhao, G. Song, C. Liu, L. Yuan, J. Chen, Acseg: Adaptive conceptualization for unsupervised semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7162–7172.
[58] H.S. Seong, W. Moon, S. Lee, J.-P. Heo, Leveraging hidden positives for unsupervised semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19540–19549.
[59] M. Caron, H. Touvron, I. Misra, H. J´egou, J. Mairal, P. Bojanowski, A. Joulin, Emerging properties in self-supervised vision transformers, in: in:Proceedings of the IEEE/CVF international conference on computer vi-sion 15, 2021, pp. 9650–9660.