# Improving Zero-Shot Semantic Segmentation using Dynamic Kernels

by

**Tauseef Tajwar**
**Muftiqur Rahman**
**Taukir Azam Chowdhury**

**Supervisor**
Dr. Md. Hasanul Kabir
Professor, Department of CSE

**Co-Supervisor**
Sabbir Ahmed
Assistant Professor, Department of CSE

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)
A Subsidiary organ of the Organization of Islamic Cooperation (OIC)
Academic Year: 2021-2022

May, 2023

# Improving Zero-Shot Semantic Segmentation using Dynamic Kernels

by

Tauseef Tajwar, 180041109

Muftiqur Rahman, 180041111

Taukir Azam Chowdhury, 180041126

**Supervisor**

Dr. Md. Hasanul Kabir

Professor, Department of CSE

**Co-Supervisor**

Sabbir Ahmed

Assistant Professor, Department of CSE

*A thesis submitted to the Department of CSE*
*in partial fulfillment of the requirements for the degree of B.Sc.*

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)
A Subsidiary organ of the Organization of Islamic Cooperation (OIC)
Academic Year: 2021-2022

May, 2023

# Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Tauseef Tajwar, Muftiqur Rahman, and Taukir Azam Chowdhury under the supervision of Dr. Md. Hasanul Kabir, Professor of the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Gazipur, Dhaka, Bangladesh, and Sabbir Ahmed, Assistant Professor of the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Gazipur, Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Formation derived from the published and unpublished work of others have been acknowledged in the text and a list of references is given.

| | | |
|---|---|---|
| Tauseef Tajwar | Muftiqur Rahman | Taukir Azam Chowdhury |
| Student ID: 180041109 | Student ID: 180041111 | Student ID: 180041126 |

**Supervisor**

Dr. Md. Hasanul Kabir
Professor
Department of Computer Science and Engineering
Islamic University of Technology

**Co-Supervisor**

Sabbir Ahmed
Assistant Professor
Department of Computer Science and Engineering
Islamic University of Technology

## Abstract

Zero-shot Semantic Segmentation (ZS3) is a daunting task since it requires segmenting items into classes that were never seen during training. One popular method is to divide ZS3 into two sub-tasks: creating mask suggestions and assigning class labels to individual pixels inside those regions. However, many existing approaches have difficulty producing masks with sufficient generalization capabilities, resulting in notable performance constraints, particularly on unknown classes. In this regard, we propose using "Dynamic Kernels" to improve object understanding within a ZS3 model during the training phase. We want to produce superior mask suggestions that permit a more accurate representation of the objects by harnessing the intrinsic inductive biases of these kernels. These specialized agents, known as dynamic kernels, adjust based on data taken from visible classes, allowing them to obtain insights on unseen things. In addition, for segment classification, our proposed system utilizes the Contrastive Language-Image Pre-Training (CLIP) architecture. This integration improves the model's generalizability by utilizing its cross-modal training capabilities. The utilization of dynamic kernels in conjunction with CLIP proves to be advantageous as it allows for finer granularity in processing, enabling performance enhancements for both seen and unseen classes. Our proposed ZSK-Net surpasses the existing state-of-the-art methods by achieving a remarkable improvement of **+10.4** and **+0.9** in hIoU on the Pascal VOC and COCO-Stuff datasets, respectively.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

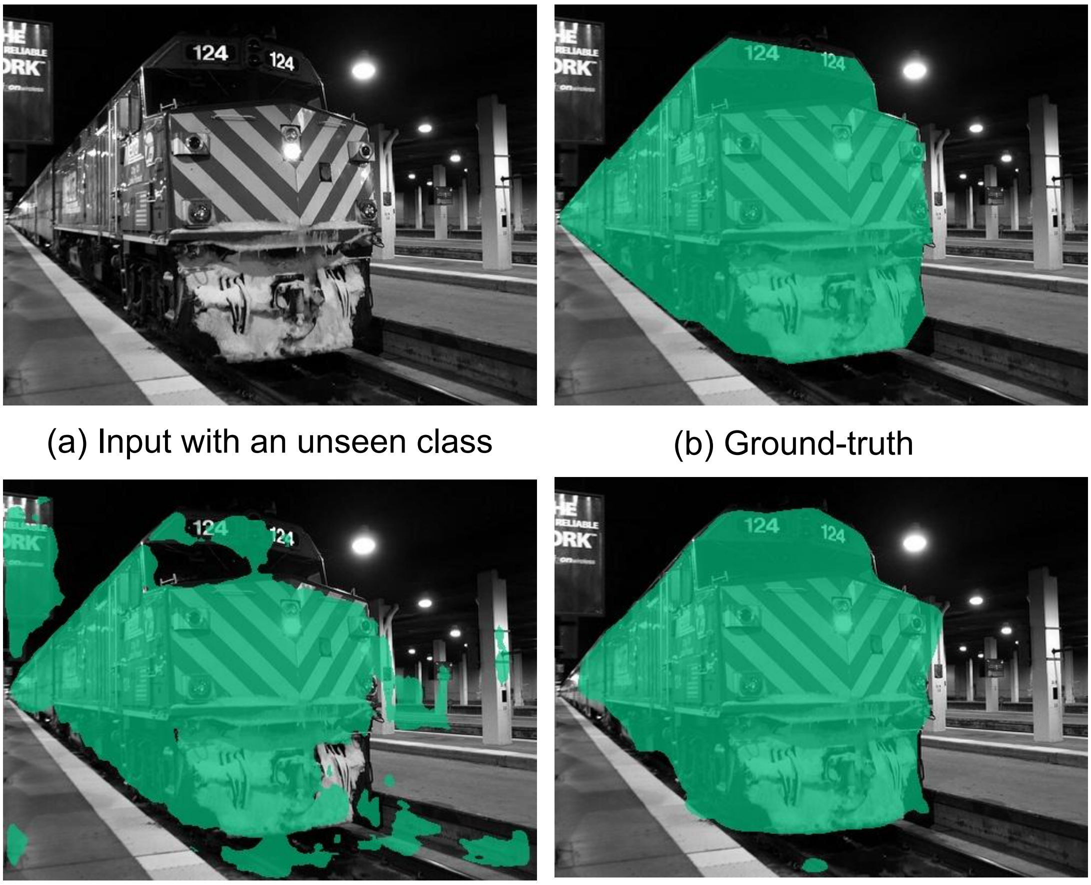
## 1.1 Overview

Semantic segmentation is a task where a model is required to assign a class label to each pixel of an image. Despite several works being proposed to solve this task [37, 25, 9, 21, 2], their applicability is often limited to constrained environments where the model was trained in. In an *open-world* setting, novel objects and concepts appear during inference which the model has never seen in training. From a design perspective, it is also impossible to simulate an *open-world* training paradigm that contains all novel objects and segmentation labels. This is an open research problem to better segment objects that the model has `seen` during training, and also identify novel objects from the background that is `unseen` during training.

Zero-shot semantic segmentation (ZS3) [4] has emerged as a groundbreaking task that aims to segment both observed and unseen objects, unlocking a realm of possibilities across various real-world applications. This innovative approach has the potential to revolutionize sectors such as autonomous driving, medical image segmentation, robot navigation, scene interpretation, and content-based image retrieval.

ZS3 extends the capabilities of computer vision systems in diverse domains by enabling the segmentation of objects that have never been encountered during training. For instance, in the context of autonomous driving, ZS3 can assist in identifying and segmenting unexpected objects or road conditions that were not observed in the training data. In medical image segmentation, ZS3 can contribute to the accurate delineation of novel anatomical structures or the detection of previously unseen anomalies. Similarly, ZS3 can empower robots to learn and navigate in unfamiliar environments.

The applications of ZS3 transcend specialized fields and can benefit any sce-



(a) Input with an unseen class

(b) Ground-truth

(c) Prediction by the current state-of-the-art

(d) Prediction by our proposed pipeline

Figure 1.1: Our proposed model, ‘ZSK-Net’, is able to better predict a contagious segmentation mask for the unseen class `train`.

nario where segmenting both visible and unseen objects is essential for comprehending and evaluating visual data. This pursuit opens up new avenues for research and development, paving the way for groundbreaking advancements in computer vision technology and its practical applications.

## 1.2 Problem Formulation

For a classical fully-supervised semantic segmentation task, there exists a dataset, $\mathcal{D}$, containing images $\{\mathcal{I}_i\}_{i=1}^{T}$ where $T$ corresponds to the total number of images in the dataset. Each $\mathcal{I}_i$ has a corresponding ground-truth mask $\mathcal{M}_i^{gt}$ consisting of $n$ homogenous regions with a set of labels $\mathcal{N}_i^{gt}$=$\{$c $\mid$ c $\in \mathcal{C}\}$ that classify each region into one of the $\mathcal{C}$ predefined classes in that dataset. The goal of a semantic segmentation model is to predict $\{\mathcal{M}_i^{pred}, \mathcal{N}_i^{pred}\}$ given $\mathcal{I}_i$ where $\mathcal{M}_i^{pred}$ and $\mathcal{N}_i^{pred}$ are the predicted mask and set of labels respectively for $\mathcal{I}_i$.

Zero-shot semantic segmentation, on the other hand, is a task defined such that the classes, $\mathcal{C}$, in the dataset can be split into $\mathcal{C}^S$ and $\mathcal{C}^U$ where $S$ denotes seen classes and $U$ denotes unseen classes with the condition that $\mathcal{C}^S \cap \mathcal{C}^U = \emptyset$. For a generalized zero-shot semantic segmentation (GZS3) setting, the dataset can be expressed into train and test set as:

$$\begin{aligned}\mathcal{D}_{train} &= \{\mathcal{I}_i^S, \mathcal{M}_i^{S,gt}, \mathcal{N}_i^{S,gt}\}_{i=1}^{T^{train}} \\ \mathcal{D}_{test} &= \{\mathcal{I}_i^{S\cup U}, \mathcal{M}_i^{S\cup U,gt}, \mathcal{N}_i^{S\cup U,gt}\}_{i=1}^{T^{test}}\end{aligned} \tag{1.1}$$

where the test set contains samples of both seen and unseen classes. This type of setting where the test set contains examples of both seen and unseen classes is called generalized zero-shot semantic segmentation. This is a very challenging setting since the model has to make predictions on classes it has never seen during training. The performance of such models then boils down to how well they are able to generalize from $\mathcal{D}_{train}$ to produce an acceptable result on $\mathcal{D}_{test}$.

## 1.3 Research Challenges

Several approaches have been put forward in the field of ZS3 with the objective of developing a mapping function that establishes a relationship between visual features and semantic features. These methods involve utilizing language models that have been pre-trained on textual data [4, 41, 48, 26]. It is important to note that this problem formulation tends to prioritize enhancing the performance on seen classes. This bias arises because the model learns to establish connections between visual and semantic concepts based on the training dataset. As a result, improving performance on unseen classes is often constrained by the limited variety of objects present in the dataset. Moreover, recent advancements in multimodal vision-language models have demonstrated significant potential for ZS3. These models utilize a combination of structured training, involving image-pixel pairs with annotations, and unstructured training, utilizing pre-trained language models and unpaired images, to gain knowledge about both seen and unseen objects. Building on this progress, a more intuitive strategy was proposed by researchers [13] by separating ZS3 into two sub-tasks: firstly, grouping pixels into segments without considering their class affiliation, and secondly, determining the class for each segment in a zero-shot manner [45, 13]. This approach aligns more closely with human perception, as humans typically identify objects as cohesive wholes rather than focusing on individual parts. The decoupled formulation offers two major challenges: 1) identifying an effective approach for comprehending and recognizing the attributes that distinguish individual objects, resulting in improved mask suggestions, and 2) improving the categorization of pixel groups separated by the mask generator. Our study tries to solve these issues by first offering a method for generating improved mask ideas for objects independent of their class labels. As a result, we aim to improve the overall performance of a ZS3 model by transferring information from visible to unseen classes.

## 1.4 Contribution

The research on decoupled ZS3 formulation [13, 45] employs a pixel grouping mechanism based on transformers [10]. This mechanism applies $N$ queries to generate $N$ proposals from an input image, which are subsequently utilized by the segment classifier. However, this approach has limitations in learning discriminative features essential for generalizing from seen to unseen classes. Moreover, the performance on small-scale datasets becomes even more difficult with this approach [13], leading to suboptimal results in terms of harmonic intersection-over-union (hIoU), a robust metric for assessing the model's generalization ability. We propose that by incorporating the inductive biases inherent in Convolutional Neural Networks (CNN), which have proven highly effective in numerous computer vision tasks, we can enhance the overall accuracy and generalization performance of a ZS3 model. CNN models incorporate two fundamental inductive biases: proximity (e.g., the assumption of pixel interdependence within neighboring regions) and weight sharing (e.g., identification of recurring patterns). To address this, we propose the integration of Dynamic Kernels, enabling enhanced representational capabilities during the mask proposal generation stage of a zero-shot semantic segmentation pipeline. By incorporating these similar inductive biases, our method aims to augment the model's capacity to capture essential information and improve overall performance.

These kernels are intended to prioritize various parts of the picture, similar to how CNNs favor locality. By directing attention to other picture regions, the model is encouraged to broaden its focus beyond the annotated viewed items, making it easier to identify more things in the background. To do this, we use a formulation of dynamic kernels that change their parameters adaptively through repeated rounds, responding to different feature representations rather than being independent of the input. During the training process, all dynamic kernels are encouraged to interact and share their knowledge with one another. This interaction enables them to comprehend how different features collaborate to form a cohesive object representation. Consequently, the dynamic kernels become adept at identifying and grouping relevant features together, even for unseen classes. This capability makes them an effective means of transferring feature-specific knowledge from seen to unseen classes. The underlying concept is akin to the functionality of deeper layers in CNNs, which act as filters specialized in recognizing patterns such as edges. Similarly, the formulation of dynamic kernels in our proposed Zero Shot dynamic Kernel (ZSK) semantic model can be seen as a collection of specialized agents finely tuned to detect discriminative spatial features of object instances.

By using these intrinsic inductive biases, our ZSK semantic model outperforms techniques relying exclusively on transformers, especially when dealing with restricted datasets. This benefit arises from the model's effective use of dynamic kernels, which resemble the localized filtering properties found in CNNs.

To enhance the segment classification aspect, we utilize a CLIP-inspired architecture, which has been trained on extensive image and text datasets. This architecture generates image and text embeddings from the mask proposals generated by the model. By incorporating external knowledge, our approach refines the mask proposal predictions and effectively identifies discriminative features of previously overlooked `unseen` objects.

To showcase the efficacy of our proposed ZSK-Net, we conducted evaluations on two widely recognized ZS3 benchmark datasets: Pascal VOC and COCO-Stuff. Our model exhibited superior quantitative accuracy metrics, as well as qualitative performance by generating coherent and localized masks for both `unseen` and `seen` object instances. The key contributions of our research can be summarized as follows:

1. To enhance the mask proposal generation process of zero-shot semantic segmentation, we present the notion of dynamic kernels, which resemble

inductive biases of convolutional operators.

2. We use a CLIP-like image and text encoder to provide more discriminative image and class embedding, allowing our model to learn features from seen classes more effectively and transfer that information to segment unseen classes in a zero-shot environment.

3. We demonstrate higher performance on benchmark ZS3 datasets by improving on the state-of-the-art ($10.4uparrow$ hIoU on Pascal VOC and $0.9uparrow$ hIoU on COCO-Stuff).

# Chapter 2

# Literature Review

## 2.1 Semantic Segmentation

During the emergence of semantic segmentation, initial approaches relied on simple Fully Convolutional Network (FCN)-based formulations [33, 40, 47]. Subsequently, advancements were made by incorporating convolutional-based models such as DeepLab [8], Deconvnet [34], and U-Net [37]. The U-shaped architecture proposed by Ronnenberger et al. [37] gained popularity due to its efficient computation, which replaced traditional sliding window-based methods. This architecture adopted an encoder-decoder-based design with a bottleneck connecting the upsample and downsample streams. It also introduced skip-connections to establish a link between the encoder and decoder, enabling the learning of low and high-level features in an end-to-end manner. As a result, it achieved impressive performance and sparked the development of several variants in subsequent years [50, 29, 52, 44, 24].

The introduction of transformers into computer vision, spearheaded by Dosovitskiy et al. [14], presented a revolutionary technique to image processing. It envisioned pictures as sequences of patches represented by positional embeddings, opening up new study opportunities. This paradigm change extended to the field of semantic segmentation [51], where transformers' intrinsic capacity to represent dependencies over a vast receptive field proved extremely useful. This capacity is useful for semantic segmentation tasks since distant pixels can influence one other, affecting the total prediction.

However, the utilization of transformer-based architectures presented a significant challenge in terms of data requirements. Additionally, these models were computationally expensive, limiting their applicability as versatile backbones. Nevertheless, the SWIN transformer architecture [32] addressed these issues by introducing a shifted window-based approach that efficiently captures patch re-

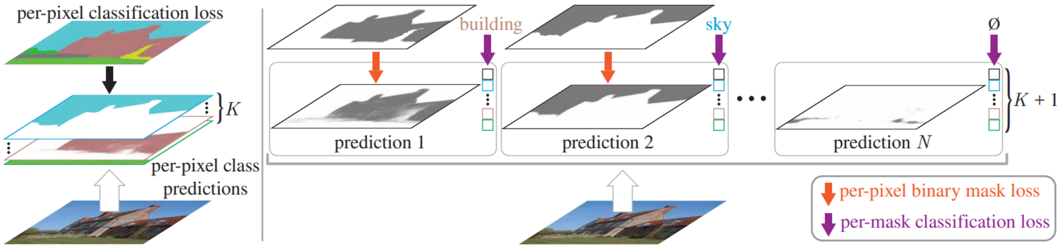

per-pixel classification loss
K
per-pixel class predictions
building
sky
Ø
prediction 1
prediction 2
prediction N
K + 1
per-pixel binary mask loss
per-mask classification loss

Figure 2.1: **Per-pixel classification vs. mask classification [10].**

lationships within an image. Consequently, the Swin-Unet model [6] emerged, incorporating the concepts of U-Net and SWIN transformers specifically tailored for semantic segmentation tasks.

### 2.1.1 MaskFormer [10]

MaskFormer is a straightforward model for classifying masks, enabling the prediction of binary masks accompanied by a single global class label. It offers a unified approach to address both semantic and instance-level segmentation tasks, utilizing the same model, loss function, and training process. In particular, MaskFormer demonstrates superior performance compared to per-pixel classification baselines, especially when dealing with a large number of classes. The mask classification-based technique surpasses the current state-of-the-art semantic method, achieving an impressive mIoU score of 55.6 on the ADE20K dataset.

Semantic Segmentation is a task that divides the image into separate regions with different semantic categories. There are two approaches to semantic segmentation. The traditional approach, based on Fully Convolutional Networks (FCNs), treats semantic segmentation as per-pixel classification, where a classification loss is applied to each pixel to determine its semantic category. This approach naturally partitions the image into different classes based on per-pixel predictions. However, there exists an alternative concept known as binary mask classification. Techniques based on binary mask categorization anticipate multiple binary masks, where each mask corresponds to a single class prediction. This strategy separates the segmentation process into distinct components: image partitioning and classification. Mask classification is particularly useful for instance-level segmentation, where the goal is to identify and categorize individual objects or instances within an image. The concept of Per Pixel Classification operates under the assumption of a predetermined number of outputs and lacks the ability to generate a variable number of predicted regions. The proposed approach, called

MaskFormer, aims to address the challenges of per-pixel classification and convert it into mask classification. It seamlessly integrates with existing per-pixel classification models by utilizing the set prediction mechanism from DETR and a Transformer decoder. So there are two predictions, a binary masks and their corresponding class.MaskFormer utilizes the Transformer decoder to calculate pairs of mask predictions and specify each mask with a unique class. The mask embedding vectors are employed to create binary mask predictions by applying a dot product operation with per-pixel embeddings derived from a fully-convolutional network. This approach effectively addresses both semantic-level and instance-level segmentation tasks seamlessly, without the need for any modifications to the model architecture, loss functions, or training procedures. MaskFormer first takes the input image and creates N binary masks. After that, it classifies each mask with a global class prediction. So MaskFormer has two predictions associated with it. One is a binary mask and the second is its corresponding class. The traditional per-pixel classification models take input images and assign each pixel to a specific class.

The MaskFormer jointly embeds two predictions into z. In order to train a mask classification model effectively, it is crucial to establish a correspondence between the set of predictions, denoted as "z," and the set of ground truth segments, referred to as "$z_{groundtruth}$".To train a mask classification model, it is necessary to establish a matching between the set of predictions z and the set of ground truth segments $z_{groundtruth}$. The ground truth set z ground truth consists of pairs ($c_{groundtruth}$, $m_{groundtruth}$), where $c_{groundtruth}$ represents the ground truth class of the ith segment and $m_{groundtruth}$ represents its binary mask in the shape of (H×W). Padding is applied to the ground truth labels using "null object" denotes as Ø ) to enable one-to-one matching. The assumption is that N (the size of the prediction set) is greater than or equal to $N_{groundtruth}$ (the size of the ground truth set). For semantic segmentation, when the number of predictions (N) matches the number of category labels (K), a simple and fixed matching can be used. Each prediction is matched to a ground truth region with the same class label. If there is no ground truth region with the same class label, it is matched to a special label $\phi$ representing "no object." However, the authors discovered that using a bipartite matching approach, which considers more factors, provides better results compared to the fixed matching method based on their experiments.

There are three modules in MaskFormer architecture.

1. Pixel Level Module

2. Transformer Module

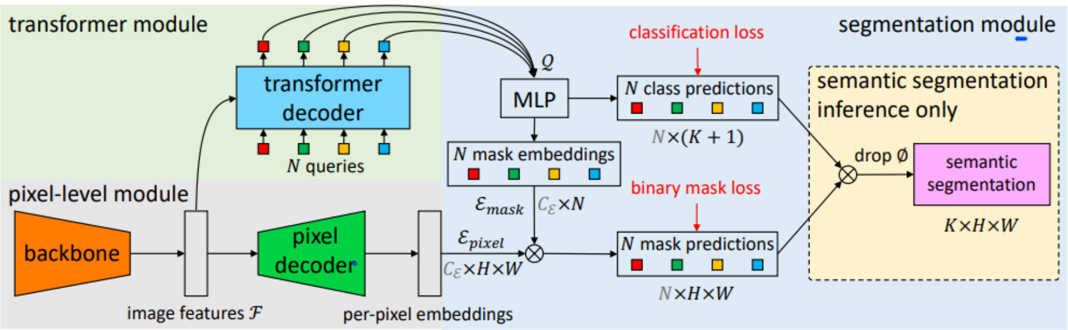


Figure 2.2: **MaskFormer Architecture** [10].

3. Segmentation Module

This module takes an input image of size HxW. First, the image is passed to the backbone. The backbone can be any feature extraction model. The backbone extracts information from the image and produces low-resolution feature maps. The image feature is then passed into two regions. One will go to the transformer module and another part will go to the pixel decoder. The pixel decoder will upsample the image features and create per-pixel embeddings. Then these per-pixel embeddings are passed to the segmentation module. The transformer module in this paper utilizes a transformer decoder, specifically the DETR model. The transformer decoder consists of N queries, each searching for different features in different parts of the images. For instance, one query may seek to identify if a specific class (x) is present at the upper right corner of the input images. In this way, each query asks specific questions and covers the entire image individually. The author of the paper employs N=100 queries for their approach. To encode the global information of the entire image, the author utilizes multi-headed self-Attention. This module allows each query to share information with the other queries.

Consequently, the Transformer Module is capable of capturing the global context of the image. The queries are initialized as zero vectors, and the author employs 6 layers of DETR in their implementation. This module takes the global information from the Transformer Module and passes that context to the multi-layer perceptron. The MLP layer produces class predictions for each mask prediction. The output is Nx(K+1), with the extra class representing the ”no object” category. For the mask predictions, the MLP utilizes two additional hidden layers to generate mask embeddings. These embeddings are then used in the process of creating a binary mask. The binary mask is produced by taking the dot product between the embedding that is found by MLP and another embedding that is found from the first module. Their product similarity result is passed through an activation function named sigmoid, resulting in the generation of the Binary Mask. The author employs a two-hidden-layer architecture with 256 units to predict mask embeddings.

The MaskFormer approach is designed to be compatible with any backbone architecture. In this particular research paper, the author employs widely adopted backbones, such as various ResNet architectures, which are commonly used for extracting information in convolutional tasks. These architectures include ResNet models with 50 and 101 layers respectively, as well as the Swin-Transformer backbones, which have been recently proposed. Additionally, they utilize the R101c model, which is a variant of R101 commonly used in the semantic segmentation community. R101c replaces the initial 7x7 convolution layer with three consecutive 3x3 convolutions. The pixel decoder can be implemented using various semantic segmentation decoders.

Modules like ASPP or PSP are commonly used in per-pixel classification methods to encode contextual information across different regions. Nonetheless, as the second module focuses on all image features and captures comprehensive global knowledge to make a specific category, the per-pixel module necessitates less intensive context aggregation. To meet the needs of MaskFormer, the researchers have devised a lightweight pixel decoder, leveraging the widely adopted FPN architecture. Following the FPN methodology, the researchers amplify the feature map, which initially has a low resolution due to the backbone, by a fixed factor within the decoder. This upsampled map is then merged with the projected feature map obtained from the backbone. The projection process involves applying a 1x1 convolution layer followed by GroupNorm (GN) to ensure the alignment of channel dimensions. Subsequently, the summed features are fused using a 3x3 convolution layer, GN, and ReLU activation. By iterating the process starting from a stride of 32, the resulting map is obtained where the stride is reduced to 4. This process is repeated starting from the stride 32 feature map until a final feature map with a stride of 4 is obtained. Finally, a single 1x1 convolution layer is applied to generate per-pixel embeddings. Notably, After passing through the pixel decoder, the features demonstrate a consistent dimension throughout. The process of segmenting an image involves assigning each pixel in the image to a specific category.

The process of segmenting an image involves assigning each pixel in the image's height and width to the highest-valued prediction pairs. This can be found by applying a dot product between every mask prediction and every class prediction. It must be noted that the class prediction can belong to the "null class".Segmenting an image involves deciding which category a pixel belongs to based on the highest values obtained from comparing probabilities and masks.

This method assigns a pixel to a specific probability-mask pair only if both the most likely class probability and the mask prediction probability are sufficiently high. If we get the highest dot product, we will pair that class with that corresponding mask. This is how the masks and their corresponding class probability pair can be calculated. Consequently, to assign a pixel to a specific class, it will check the prediction pair. As a result, pixels are assigned to a specific class when it is determined that they correspond to the pair with the highest probability. In the context of semantic segmentation tasks, segments with identical category labels are combined or merged together. In their approach, the authors apply a filtering process before inference to eliminate low-confidence predictions. In addition to the above, the authors have an additional criterion to consider when evaluating predicted segments. If a significant portion of the binary masks associated with these segments, indicated by the value of mi being greater than the threshold, is obscured or covered by other predictions, they exclude them from the final selection. This helps ensure that only accurately segmented regions, without substantial occlusion, are retained for further analysis or processing.

The process of inference is tailored specifically for semantic segmentation and is executed using straightforward matrix multiplication techniques. Through our empirical observations, we have found that achieving better results can be accomplished by performing marginalization over probability-mask pairs. In other words, the authors take each of the binary masks and class predictions which are predicted by the second module, and the dot productions are calculated between each of the binary masks with every other class prediction. After that, the one with the highest value is taken as the class prediction for that particular mask. It is worth noting that the operation above does not consider the "null object" category as traditional semantic segmentation necessitates a label for each output pixel. It is important to highlight that this strategy provides to calculate every pixel class probability by taking every produced binary mask and its corresponding class probability. Then a dot product is calculated between every binary mask with every class prediction. Nevertheless, their observations have indicated that directly maximizing the per-pixel class likelihood results in suboptimal performance. The authors give a hypothesis that this occurs due to the even distribution of gradients to every query, which complicates the training process.

### 2.1.2 K-Net [49]

This paper introduces a framework called K-Net (Kernels Network) for image segmentation tasks, including semantic segmentation, instance segmentation, and panoptic segmentation. The main idea behind K-Net is to unify these different segmentation tasks by utilizing the concept of dynamic kernels. Kernels are convolutional filters that are typically used in image processing and deep learning for feature extraction. In K-Net, a set of convolutional kernels is randomly initialized and then learned according to the specific segmentation targets. The segmentation targets in K-Net consist of semantic categories (e.g., identifying objects or regions with specific labels) and instance identities (e.g., differentiating individual instances of objects). Semantic kernels are learned to capture the characteristics of different semantic categories, while instance kernels are trained to recognize and distinguish different instances within each category. By combining the outputs of semantic and instance kernels, panoptic segmentation, which aims to provide a unified representation of both semantic and instance information, can be achieved. The forward pass of K-Net involves applying the learned kernels to convolve with image features, resulting in segmentation predictions. This process allows the network to assign labels or segment individual instances within an image.

K-Net's effectiveness and versatility are attributed to two key design aspects. Firstly, the framework incorporates a content-aware mechanism that dynamically updates the kernels based on their activations on the image. By dynamically changing the kernels based on their responses to the image features, this kernel update strategy makes sure that each kernel, especially the ones that separate different object instances, can better identify and differentiate various objects and segments in the image, which improves the kernels' ability to discriminate and the segmentation performance. Secondly, K-Net adopts a bipartite matching strategy for assigning learning targets to each kernel. This strategy, influenced by object detection methods, creates a one-to-one correspondence between the instances in an image and the kernels. This method solves the problem of dealing with different numbers of instances across images. Moreover, it is purely mask-driven, meaning it does not rely on bounding boxes. As a result, K-Net is naturally free from non-maximum suppression (NMS) and does not require bounding box annotations, making it suitable for real-time applications.

To achieve this, the maximum number of groups is assumed to be N, where N represents the number of semantic classes for semantic segmentation or the maximum number of instances in an image. In panoptic segmentation, N represents total stuff classes (e.g., background, sky, road) and objects in the image. The method uses N convolutional filters (K) to divide the image into N regions. Each filter is in charge of finding the pixels that belong to its respective region. The segmentation output (M) is computed by convolving the filters (K) to the input feature map (F) generated by a deep neural network, as illustrated in equation

2.1.

$$M = \sigma(K * F) \tag{2.1}$$

The activation function ($\sigma$) applied to the convolution result determines how each pixel is assigned to the kernels. In semantic segmentation, the softmax function is typically used to assign each pixel to only one kernel, representing a single class. In instance segmentation, the sigmoid function is used, allowing one pixel to belong to multiple masks (instances) by setting a threshold on the activation map. This formulation has been widely used in semantic segmentation, where each kernel finds pixels of a similar class across images. However, this paper aims to explore if the concept of kernels can be equally applied to instance segmentation and panoptic segmentation. In K-Net, the instance segmentation is performed in one pass without extra steps by using separate instance kernels that segment at most one object per image. By combining the outputs of instance kernels and semantic kernels, panoptic segmentation is achieved, where pixels are assigned to either an instance ID or a class of stuff. So, the formulation uses kernels to assign pixels to predefined meaningful groups in segmentation tasks. By applying appropriate activation functions and combining semantic and instance kernels, K-Net achieves semantic, instance, and panoptic segmentation efficiently in a unified framework.

Unlike semantic categories, which provide a clear and explicit characteristic for semantic kernels, instance kernels need to distinguish objects that have different sizes and looks both inside and outside images. Therefore, instance kernels require a stronger ability to discriminate between objects than static kernels. To address this challenge, the paper proposes an approach that makes the instance kernels dependent on the group of pixels that correspond to them. This is achieved through a kernel update head, as depicted in Figure 2.3. The kernel update head, denoted as $f_i$, consists of three main steps: group feature assembling, adaptive kernel update, and kernel interaction. In the first step, the group features $F_k$ are gathered for each pixel group using the mask prediction $M_{i-1}$. The group feature captures the content of each individual group, which is crucial for distinguishing them from one another. This assembled group feature, $F_k$, is then used to adaptively update the corresponding kernel $K_{i-1}$.

Next, the updated kernel communicates with other kernels to fully capture the image context. This allows the kernels to learn from the relationships and dependencies between different instances and semantic categories. Finally, the group-aware kernels $K_i$, which have been updated based on the assembled group features, perform convolution over the feature map F to get more precise mask

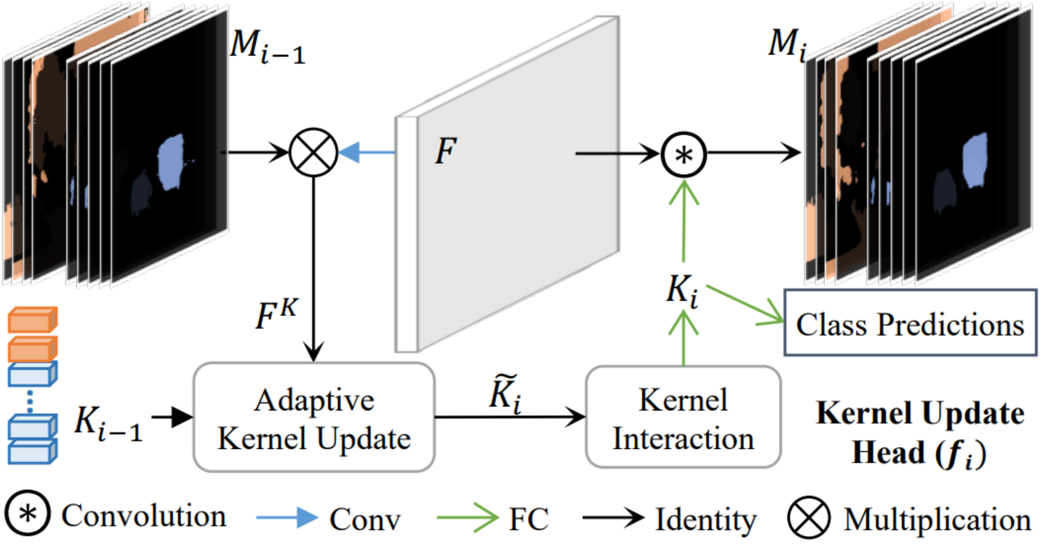


Figure 2.3: **Kernel Update Head [49].**

predictions Mi. This iterative process can be repeated to refine the partitioning and improve the discriminative ability of the kernels. Finer partitions tend to reduce noise in the group features, leading to more precise and discriminative kernels.

The kernel update head module converts the static kernels to dynamic ones following three crucial steps:

1. **Group Feature Assembling:** The first step involves assembling the features for each group, which will later be used to make group-aware kernels. The feature $F_k$ for each kernel $K_{i-1}$ is assembled by element-wise multiplication of the feature map F with the corresponding mask $M_{i-1}$. The resulting assembled feature $F_k$ is a tensor of size B×N×C.

2. **Adaptive Feature Update:** In the second step, the kernels are updated by the kernel update head using the assembled feature $F_k$ to enhance their representation ability. Since the mask $M_{i-1}$ may not be completely accurate and the group features may contain noise introduced by pixels from other groups, an adaptive kernel update strategy is devised. This strategy involves element-wise multiplication between $F_k$ and $K_{i-1}$, followed by linear transformations. The head then learns two gates, $G_F$ and $G_K$, which control the contribution of $F_k$ and $K_{i-1}$ to the updated kernel K'. The gates determine the weights assigned to $F_k$ and $K_{i-1}$, allowing for a weighted summation of their features.

3. **Kernel Interaction:** The final step involves kernel interaction, which fa-

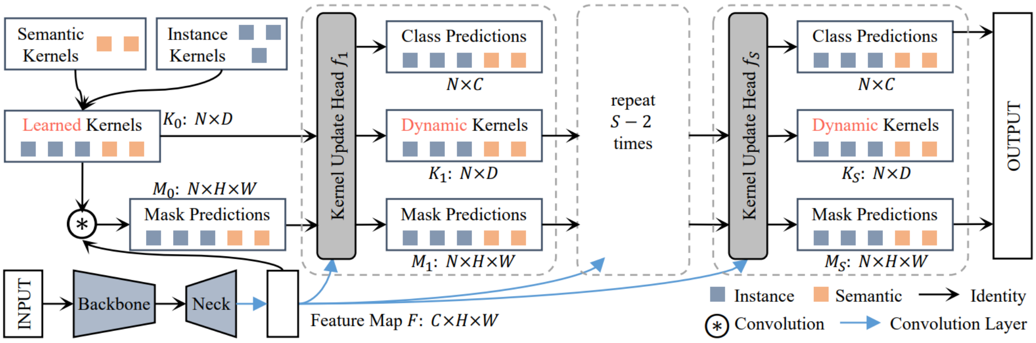


Figure 2.4: **K-Net for panoptic segmentation** [49].

cilitates the exchange of contextual information among the kernels. This interaction process enables each kernel to implicitly model and exploits the relationships between different groups in an image. To perform kernel interaction, the approach employs Multi-Head Attention followed by a Feed-Forward Neural Network. The output of the kernel interaction denoted as $K_i$, is then used to generate a new mask prediction $M_i$ through the function $g_i$. Additionally, $K_i$ is used to estimate classification scores in instance and panoptic segmentation tasks.

In summary, K-Net provides a unified framework for image segmentation tasks by leveraging convolutional kernels. Its adaptive kernel update strategy and the use of a bipartite matching strategy for learning targets contribute to its effectiveness and flexibility in handling different segmentation tasks, making it appealing for real-world applications. The model is depicted in figure 2.4.

## 2.2 Zero-Shot Learning

Zero-Shot Learning (ZSL) works in a setting where the training and test sets are disjoint with the goal of teaching a model to work with objects that it has never seen before [27]. Akata et al. [1] tried to achieve this goal by learning a function that measures the compatibility between the image and its corresponding label hence contributing to the mapping between visual and semantic feature space. DeViSE[17] attempts to do the same by utilizing textual data to learn the relationship between the two spaces and map the images to a rich embedding space. Romera et al. [36] takes a simple approach to the whole problem by creating a simple random forest algorithm based on relative attributes. All these methods face a common problem known as the prototype sparsity problem that Fu et al. [18] try to improve by introducing an auxiliary dataset alongside a

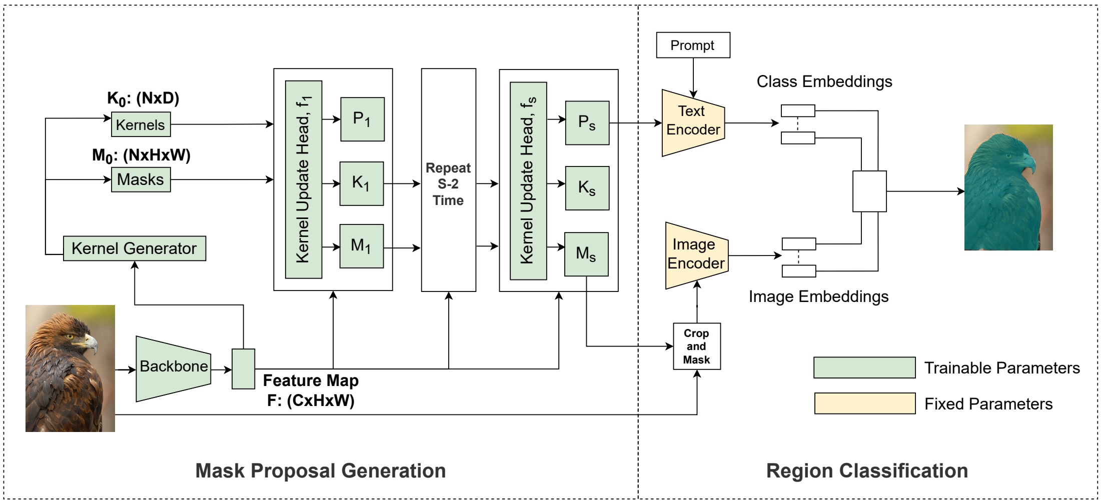


Figure 2.5: **Overview of our proposed ZSK-Net.**

target dataset. Authors in [46, 20] introduced ways to use binary codes in ZSL by trying to preserve the local structural property and discrete nature in binary codes and using boolean operations.

A different approach from this was taken by [43] where the authors proposed a novel conditional GAN that can synthesize CNN features of unseen classes. An improvement over this approach was proposed in [16] where the model is trained with multi-modal cycle consistent semantic compatibility. LisGAN[28] is another method based on GANs that can produce unseen features from random noises guided by semantic descriptions. Sariyildiz [38] introduced a new loss function called the Gradient Matching (GM) loss that evaluates the quality of the gradient signal obtained from the generated examples.

### 2.2.1 Contrastive Language-Image Pre-Training (CLIP) [35]

Cutting-edge computer vision systems usually depend on forecasting a predefined collection of object categories. However, this constraint limits their versatility and necessitates the acquisition of supplementary labeled data when dealing with novel concepts. A different strategy involves leveraging the textual information associated with images, enabling a wider range of supervision to enhance learning. In this research paper, the authors illustrate training models to predict image-caption pairs using a large dataset. This technique leads to effective and scalable learning of cutting-edge image representations. The acquired visual concepts can subsequently serve as references or be employed to depict novel concepts. This approach enables smooth adaptation to various downstream tasks without requiring explicit training, even for previously unseen objects. To evaluate its effectiveness, the performance of this technique is assessed across a broad spectrum of vision datasets, covering diverse aspects of visual understanding. The outcomes indicate that the model successfully adapts to a wide range of tasks and frequently achieves competitive performance compared to fully supervised baselines. It even demonstrates comparable accuracy to the original backbones model in the traditional dataset, despite not utilizing its extensive training dataset. Advances in pre-training methods for learning from the raw text have had a significant impact on NLP.

The introduction of the "text-to-text" input-output interface has further enhanced the flexibility of task-agnostic architectures, allowing them to transfer knowledge to downstream datasets without specialized adaptations or customization. The findings indicate that the level of supervision obtained through pre-training methods on large-scale text collections surpasses that of meticulously labeled NLP datasets created by crowdsourcing. On the contrary, within the realm of computer vision, Crowdsourcing methods are used to pre-train models. The question arises as to whether pre-trained models that learn from the internet can achieve comparable performance to traditional vision models. The authors also contemplate harnessing the potential of natural language supervision for image representation, which is still relatively uncommon. This is because the supervision in this specific field does not yield superior performance on standard datasets when compared to other approaches.

There are several approaches in this paper.

1. **Natural Language Supervision**: Learning perception through natural language supervision is not new, but there is a variety of terminology used to describe similar work. Examples include unsupervised, self-supervised, weakly supervised, and supervised methods. The unifying factor among these diverse studies is the recognition of the importance of linguistic information as a valuable indicator for training purposes. While each method may differ in specific details, they all utilize natural language supervision for learning. Topic models and n-gram representations can help address natural language complexities. However, advancements in context learning have provided us with effective tools to harness the rich source of supervision provided by natural language.

2. **Preparing a Large Dataset in Size:** Previous studies have relied on three datasets, namely MS-COCO, Visual Genome, and YFCC100M. While the first two datasets mentioned are of high quality, they are relatively small in comparison to the extensive amount of data that is currently available.

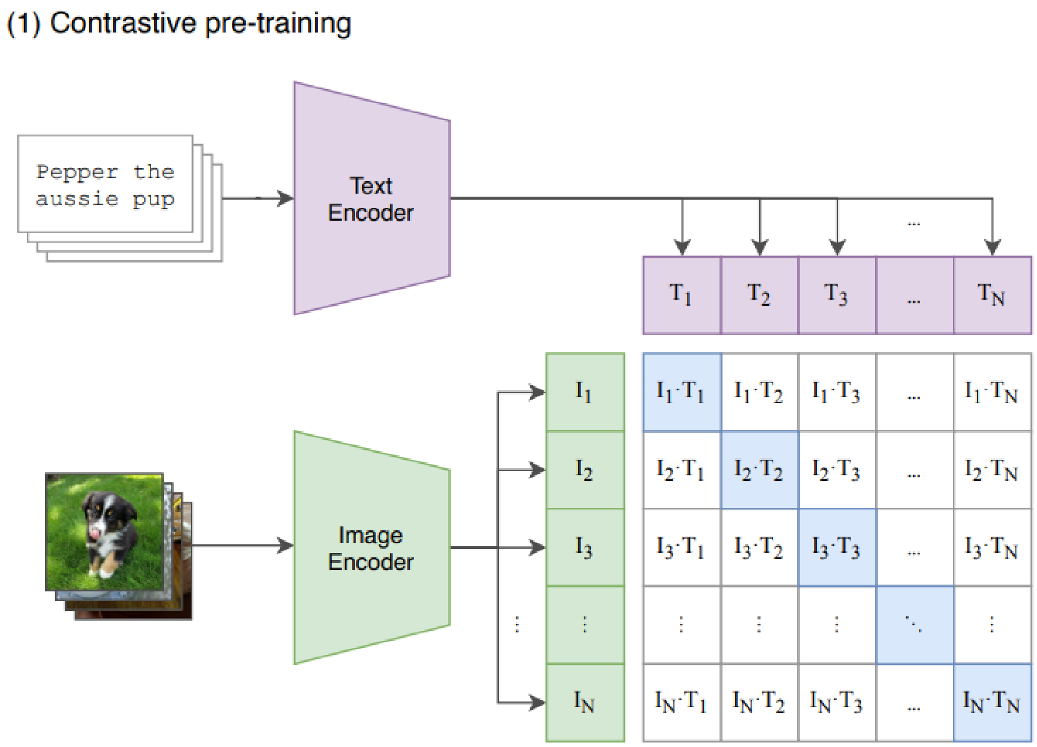

(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
$T_1$ $T_2$ $T_3$ ... $T_N$
$I_1$ $I_2$ $I_3$ ⋮ $I_N$
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
⋮ ⋮ ⋮ ⋱ ⋮
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$

Figure 2.6: **CLIP training. The image and text encoders are trained jointly using image-text pairs [35].**

On the other hand, the third dataset serves as an alternative option, importantly, it should be noted that the available information for each image varies and is limited in quality. Following the application of specific filters, the dataset underwent a reduction in size. The authors of this paper created a new dataset by collecting images with the captions from internet. By collecting this, the dataset becomes huge. Their focus was to incorporate a wide variety of image-text pairs by searching for combinations that aligned with 500,000 predefined queries. They aimed to achieve balance among the classes in the dataset, which led them to set a significantly high number for each query. As a result, they included up to that number of image-text pairs for each query.

3. **Choosing an efficient training method** The author utilizes both a CNN and a text encoder in their approach. The CNN is responsible for extracting information from the images, while the text encoder is utilized to extract textual information. These components are then integrated to collectively predict image captions. But this approach is difficult because of the large parameter model. CLIP is a technique used to determine which (image, text) pairs actually occurred within a batch of image text pairs. During the training process, both the image encoder and the text encoder are trained using a maximization strategy for the diagonal elements. Additionally, a contrastive approach is employed to minimize the off-diagonal elements. This training method ensures optimal learning for both encoders. They simplify the training process of CLIP by using a large pre-training dataset without pre-trained weights, and a linear projection to map the encoders' representations to the multi-modal embedding space. The authors employ the technique of randomly selecting crops from the images. By utilizing these two techniques, data augmentation is done for training. and a log-parameterized multiplicative scalar optimization strategy.

4. **Choosing Scaled Model** ResNet-50 is modified with ResNet-D improvements, antialiased rect-2 blur pooling, and attention pooling, using a transformer-style multi-head QKV attention mechanism. It holds the global context of the image and gives attention to the particular part. For ViT, the authors adopt its implementation with slight adjustments, including the addition of layer normalization to combined embeddings. The author of this paper uses a modified Transformer as a Text Encoder. It consists of 12 layers, a width of 512, and 8 attention heads, totaling 63 million parameters. The feature representation is obtained from the highest layer's activations at the [EOS] token. Masked self-attention is employed for potential pre-training

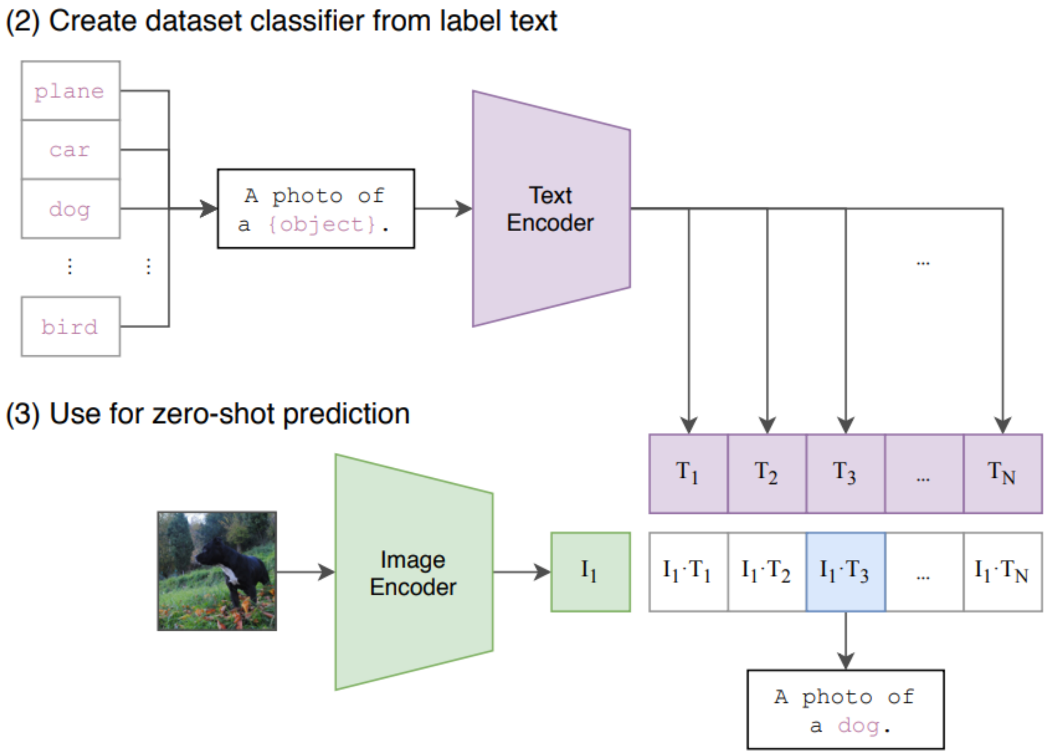


Figure 2.7: **CLIP at test time** [35].

or auxiliary language modeling objectives.

5. **Training:** The authors trained a series of models consisting of ResNets and 3 ViT. All models in the study were trained using identical optimization techniques, including the same optimizer, weight decay method, and the number of training epochs. This approach ensured that the process of optimizing the models, which involves adjusting their parameters during training, remained consistent across all experiments. Furthermore, the weight decay regularization technique, employed to mitigate overfitting, was consistently applied to ensure fairness and comparability in the training process. Lastly, every model underwent an equal number of training epochs, reflecting the complete passes made through the training data, ensuring uniformity in training duration and convergence. A large minibatch size and mixed-precision training and gradient checkpointing techniques were used to accelerate training and save memory.

CLIP offers the capability to function as a classifier that can classify objects that are not seen in the training dataset. In the inference time, the authors take all the class names in the dataset and insert them in the text prompt. To perform this prediction, the image is passed through the Image Encoder and we get the image representation of the image. The text prompt with the class names is inserted in the Text Encoder. The Text Encoder gives us the class representation in a single vector. The cosine similarity between image representation and class representations is computed. The class of an image is determined by its highest cosine similarity. By leveraging this approach, CLIP can serve as a zero-shot classifier, allowing the prediction of class labels for unseen classes based on their textual descriptions. The investigation conducted in this study aimed to explore the potential of transferring knowledge from internet text to vision tasks. The findings suggest that this approach can lead to similar behaviors emerging in computer vision. The CLIP models underwent optimization during pre-training to acquire skills across various tasks. Leveraging natural language prompting, these models enable zero-shot transfer to existing datasets.

## 2.3 Zero-Shot Semantic Segmentation

The main task of semantic segmentation is to classify an image into pixel level, which has limited ability to scale a large number of object classes. Whereas Zero-Shot Semantic Segmentation can overcome this constraint by segmenting things that the model has never seen in the training stages. Zero-shot semantic segmentation [4] has two approaches to its training process, which are, transductive and inductive method [48]. In the transductive approach, the unseen classes are allowed to appear without labels during training which seems counter-intuitive to the concept of ZSL. The inductive approach, on the other hand, does not allow unseen classes to appear during the training phase at all.

Butcher , in their model ZS3-Net [4], utilized Generative Moment Matching Network (GMNN)[30] to refine the classifier part by generating pseudo-features in order to obtain the features of unseen classes at inference. A semantic projection layer was introduced in the SPNet architecture [42] to utilize the information gained from seen classes to segment unseen classes using a similarity function. However, both ZS3-Net and SPNet are biased towards the seen classes. Another issue that comes with the ZS3-Net is that the generator uses random embedding vectors to generate fake features which makes it prone to mode collapse- a problem that was solved by the CaGNet architecture [19] by incorporating the neighborhood information while generating pixel-wise features.

In Joint Embedding Space Network [3], semantic and visual encoders create a joint embedding space where the semantic encoder takes word embeddings and extracts semantic prototypes and the visual features can be found from the visual encoders. The semantic prototypes and the visual features are both responsible for making the joint embedding space. Kato [26] used variational mapping to learn semantic features using a dual-branch network. The segmentation branch has the encoder that generates visual feature maps from the input and the conditioning branch provides the semantic feature maps from word embedding vectors. The decoder concatenates the semantic and the visual feature maps to produce the output. Hu [23] introduced Gaussian and Laplacian distributions to reduce the effect of noisy samples. The variance is responsible for finding the noise in the input image. The novel uncertainty-aware Bayesian model estimates the variance from the noisy samples in order to enhance the generalization ability between training samples and the noisy ones.

### 2.3.1 ZS3-Net [4]

In this paper, a new architecture called ZS3Net is introduced to tackle this problem. ZS3Net integrates a deep visual segmentation model with a method for creating visual representations from semantic word embeddings. This fusion enables the model to handle pixel classification tasks that involve both seen and unseen categories during testing, which is known as "generalized" zero-shot classification. Additionally, the authors introduce a self-training step that utilizes automatic pseudo-labeling of pixels from unseen classes, further improving the model's performance. The authors emphasized that their approach tackles the zero-shot setting for semantic segmentation, which was not addressed by any existing methods at the time of their proposal. They introduce a new task called zero-shot semantic segmentation (ZS3) and present ZS3Net as an effective architecture to address this challenge. Taking inspiration from state-of-the-art zero-shot classification approaches, the authors combine a deep neural network backbone for image embedding with a generative model that captures class-dependent features. This unique combination enables the generation of visual samples belonging to unseen classes. These generated samples are then incorporated into the training process, along with real visual samples from seen classes. By leveraging both real and synthetic samples, the authors train a final classifier that can effectively handle both seen and unseen classes in the zero-shot semantic segmentation task.

This method utilizes fully convolutional networks (FCNs) for the semantic segmentation of images. The approach involves mapping each category in the set of classes to a vector representation using a word embedding model like word2vec. The word embedding model learns to represent words as vectors in a high-dimensional space by training on a large text corpus. These vector representations are then used to train an FCN.

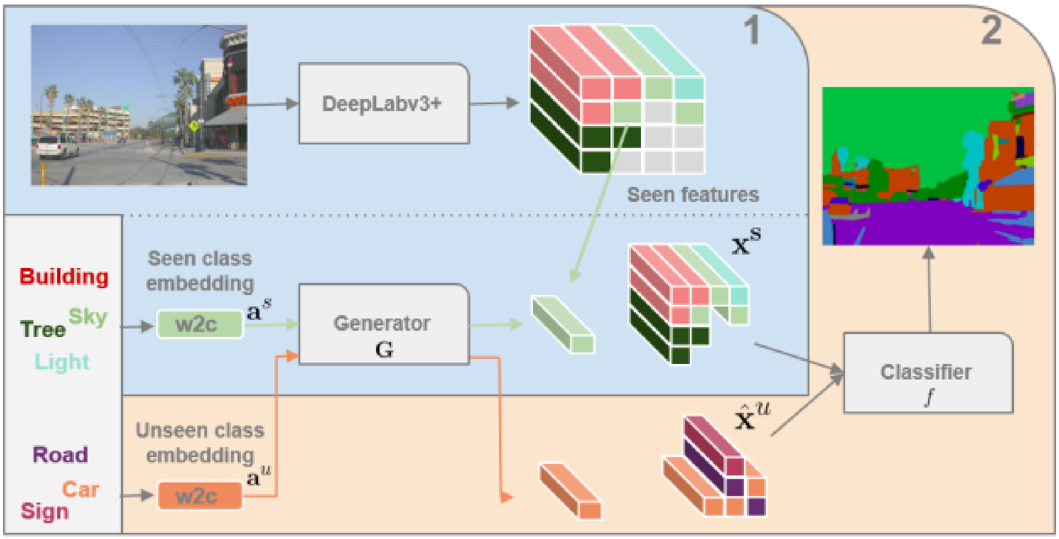


Figure 2.8: **Architecture of ZS3-Net [4].**

The first step is to define and collect pixel-wise data. We start with a DeepLabv3+ model that has been pre-trained on seen class data with a supervised loss function. We then have to select appropriate features from the model, from several feature maps that can be used separately for classification. The classifier that will be fine-tuned later must be able to work on individual features at the pixel level. In the second step, a classification model is trained to predict the class of each pixel in an image. The classification model is similar to DeepLabv3+, and it consists of a 1x1 convolutional layer. Once the generator is trained in the first step, arbitrarily many pixel-level features can be sampled for any classes, including unseen ones. These features can be used to create a fake unseen training set, which is then mixed with the real features from seen classes to fine-tune the classification layer. The new pixel-level classifier can then be used to perform semantic segmentation of images that show objects from both seen and unseen classes.

To further enhance the performance of ZS3-Net, the authors propose a novel approach called ZS5-Net (ZS3-Net with Self-Supervision) in the context of relaxed zero-shot learning. In this setup, unlabelled pixels belonging to unseen classes are already available during training. The ZS5-Net pipeline incorporates a self-training step, which enhances the performance of the model. The self-training process leverages these unlabelled pixels to improve the model's understanding of unseen classes, leading to improved segmentation results. Furthermore, the authors extend their model by incorporating contextual cues derived from spatial region relationships. They observe that objects with similar properties often

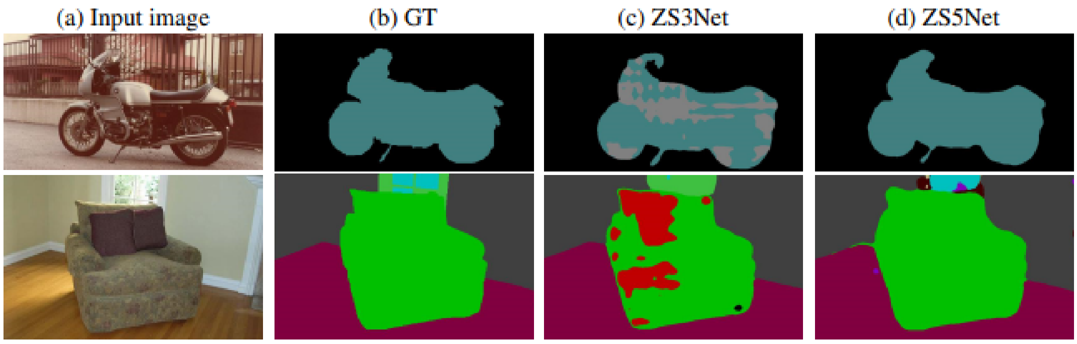


Figure 2.9: **Improvement: ZS5-Net** [4].

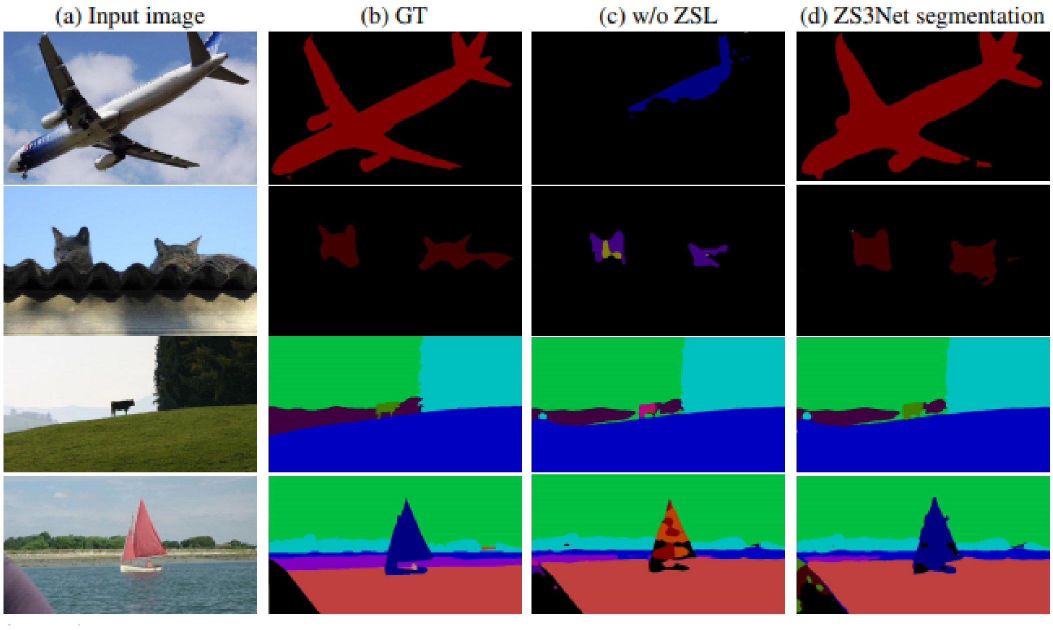


Figure 2.10: **Qualitative analysis of ZS3-Net** [4].

exhibit similar contexts. For example, animals like cows and horses are commonly found in fields, while vehicles like motorbikes and bicycles are predominantly seen in urban scenes. Exploiting this observation, the model integrates semantic contextual cues to improve its understanding of object relationships within a scene. By considering the context in which objects appear, the model can make more informed and accurate predictions for semantic segmentation.

The performance of ZS3-Net is evaluated on two popular datasets, namely Pascal-VOC and Pascal-Context, using different numbers of unseen classes in the zero-shot setup. Comparative analysis against a zero-shot learning baseline demonstrates the superior performance of the proposed method. Moreover, the incorporation of self-training and semantic contextual cues further enhances the model's performance, showcasing the effectiveness of these additional techniques.

The architecture combines a deep visual segmentation model with a generative model for synthetic data generation and leverages both real and synthetic samples to handle seen and unseen classes effectively. The incorporation of self-training and semantic contextual cues enhances the model's performance. The paper makes significant contributions to the field of zero-shot learning and semantic segmentation, providing promising avenues for future research and advancements in the field. The proposed ZS3-Net architecture addresses the fundamental challenge of zero-shot semantic segmentation by enabling the model to recognize and classify objects from both seen and unseen classes. By leveraging generative models and semantic word embeddings, the architecture generates synthetic visual samples for unseen classes, effectively expanding the training data and enabling the model to learn robust representations for these classes. This approach goes beyond traditional embedding-based methods by incorporating pixel-level classifiers and fine-tuning strategies specific to semantic segmentation. One of the key advantages of ZS3Net is its ability to incorporate self-training, which utilizes unlabelled pixels from unseen classes during the training phase. By pseudo-labeling these pixels based on the model's predictions, the architecture further refines its understanding of unseen classes and improves segmentation performance. This iterative process allows the model to learn from its own predictions and adapt to challenging scenarios.

The integration of semantic contextual cues into the model is another significant contribution of this work. By considering the spatial relationships and contextual information of objects, the model gains a deeper understanding of scene semantics. This knowledge enhances its ability to accurately segment objects within complex scenes and improves the overall quality of the segmentation results. The evaluation of ZS3Net on benchmark datasets demonstrates its effectiveness in handling zero-shot semantic segmentation tasks. By comparing against a zero-shot learning baseline, the authors provide quantitative evidence of the superiority of their proposed approach. Moreover, the incorporation of self-training and semantic contextual cues leads to additional performance improvements, indicating the potential for further enhancements in this field. The proposed architecture opens up several avenues for future research in zero-shot semantic segmentation.

One potential direction is the exploration of alternative generative models to improve the quality of synthetic samples for unseen classes. Advancements in generative adversarial networks (GANs) or other generative techniques can be leveraged to generate more realistic and diverse visual samples, thereby enhancing the model's ability to generalize to unseen classes. Another promising direction is the investigation of more effective self-training strategies. The current approach pseudo-labels unlabelled pixels based on the model's predictions, but exploring alternative techniques such as co-training or active learning could potentially yield further improvements in performance. Additionally, investigating methods to mitigate the effects of noisy pseudo-labels during self-training could enhance the robustness of the model. Furthermore, the integration of additional sources of information, such as textual descriptions or auxiliary datasets, could be explored to further enhance the model's understanding of unseen classes. By leveraging multimodal data and incorporating cross-modal embeddings, the model may be able to capture richer representations and achieve even better performance in zero-shot semantic segmentation tasks.

In conclusion, the research paper presents the ZS3-Net architecture as a novel approach to address the challenging task of zero-shot semantic segmentation. By combining deep visual segmentation models with generative techniques, self-training, and semantic contextual cues, the proposed architecture achieves impressive results on benchmark datasets. The work opens up new possibilities for advancing zero-shot learning and semantic segmentation, with potential future directions including the exploration of alternative generative models, more effective self-training strategies, and integration of additional modalities. These advancements have the potential to contribute significantly to the field of computer vision and push the boundaries of semantic segmentation in unseen class scenarios.

### 2.3.2 Zero-Shot MaskFormer [45]

This study introduces an innovative approach to tackle the zero-shot semantic segmentation challenge. The method leverages the CLIP (Contrastive Language-Image Pretraining)[35] model, which has been trained extensively on image-caption data to establish a robust connection between visual and textual information. However, one of the main difficulties in leveraging CLIP for pixel-level semantic segmentation is the difference in granularity between the image-level representation learned by CLIP and the pixel-level segmentation task. In order to address the discrepancy in granularity, the authors propose a modification to the CLIP model. Their proposed approach involves modifying the original image-level CLIP model to operate at a more detailed level. Specifically, they adapt the model to work with individual pixels using a pixel-level module. This transformation would facilitate the alignment of visual features at the pixel level with corresponding textual features. It enables the utilization of a vision-category alignment model specifically designed for FCN-based zero-shot semantic segmentation. However, taking the information from pixels alone may not perform satisfactory results, given that the original model was trained using image-level data. To leverage the alignment capabilities of the CLIP model in connecting vision and category, the authors propose the adoption of a semantic segmentation technique that incorporates mask proposals.

In this study, the Maskformer[10] method is employed to fulfill this task. This method produces a set of masks that do not depend on any class and then assigns a specific label to each mask. Through the separation of the semantic segmentation task into these distinct sub-tasks, it becomes more adaptable for handling novel categories. The process of producing masks that are not specific to any class, using seen categories for training, shows a strong ability to adapt to new categories. This model exhibits a classification stage that is comparable to the CLIP model. After generating binary masks, they are classified to determine their corresponding classes. This classification stage, where the model assigns classes to each proposed mask, operates at a recognition level similar to that of the CLIP model, enabling alignment between the two approaches. The authors also use the CLIP model to assign zero-shot labels to each proposal by feeding it the image crop that masks out the rest of the image. Additionally, a learned prompt-based approach is employed to improve ZS classification accuracy based on the pre-trained CLIP model. These strategies help bridge the gap between the pixel-level semantic segmentation task and the image-level CLIP model, enabling zero-shot semantic segmentation with the support of the learned vision-category alignment capabilities of CLIP.

The authors propose two strategies for performing region classification using the CLIP pre-trained model:

1. **Direct Application of CLIP Vision Encoder:** In this strategy, the CLIP vision encoder is applied directly to each mask proposal for classification. The mask proposal is binarized using a threshold of 0.5, and the foreground area is extracted by applying the binarized mask to the image. The resulting masked image crop undergoes resizing before being utilized in the classification process and these are passed to CLIP for classification purposes. Nevertheless, due to the absence of supplementary training procedures, the available data for training from known classes remains unused. Consequently, during the inference stage, this methodology may exhibit subpar performance when dealing with known classes.

2. **Retraining an Image Encoder on Seen Classes:** In order to make use of the training data from familiar classes, the authors suggest retraining an image encoder specifically on these classes. However, when training new classifiers using the data from known classes, the retrained image encoder

may not possess the capacity to generalize effectively to classes that have not been previously encountered. To address this, the authors suggest that the image encoder be retrained using the features generated from the pre-trained CLIP text encoder as fixed classifier weights. The image encoder captures vision features, while the text encoder generates representations for the seen classes. The author's objective is to enhance the alignment between these vision features and the corresponding text representations within the shared embedding space. This method gives the image encoder gains some generalization ability to unseen classes. By using the text feature from CLIP as the classifier weights for MaskFormer, this method can be seamlessly combined with the MaskFormer training procedure. So the MaskFormer does not need to train additionally on the image encoder separately.

These two approaches work in synergy, addressing the fact that they tackle different aspects of the problem Therefore, the authors use the results of these two strategies by an ensembling mechanism, combining their outputs to obtain improved performance. This ensemble approach helps leverage the strengths of both strategies for region classification using the CLIP model.

Within the framework of zero-shot semantic segmentation with the CLIP model, the authors investigate two distinct approaches for creating viable text prompts.:

1. **Hand-crafted Prompting:** This crafting approach involves leveraging the preexisting prompts specifically designed for image classification tasks within the CLIP framework. The ImageNet-1K dataset is utilized for this purpose These prompts consist of natural sentences with a vacant space. That vacant space can be filled with the class names corresponding to a specific dataset. However, considering that these prompts were not originally tailored for semantic segmentation, their suitability for this task may vary. To identify the most effective prompt for ZS3, the authors thoroughly handcrafted to evaluate and experiment with all available prompts using the training data.

2. **Learning-based Prompt:** The learning-based prompt approach utilizes techniques that have demonstrated potential in customizing large-scale pre-trained language and vision-language models for targeted tasks further along the processing pipeline. In this approach, the input sentence is passed to the text encoder in the CLIP model. The input sentence can be utilized as a sequence of tokens, which consists of two distinct token types: the first is the prompt token and the second one is the category token. The formulation of the generalized prompted text follows a pattern of

[p0]...[pi][cat0]...[catn]...[pm]. Here, 'n' represents the count of category tokens, and 'm' represents the count of prompt tokens. In the learned prompt approach, tokens [p0][p1]...[pm] are treated as learnable parameters. These parameters can be made to generalize on novel classes by training them on the seen data.

These two approaches offer different ways to construct text prompts for zero-shot semantic segmentation. The hand-crafted prompting approach relies on predefined prompts, while the learning-based prompt approach allows the prompt tokens to be learned from the data. Both approaches strive to enhance the effectiveness of the CLIP model when utilized for zero-shot semantic segmentation, focusing on achieving better performance.

In addition to the two-stage framework presented in the paper, a more conventional approach for semantic segmentation is to use a fully convolutional network(FCN). FCN is a widely used method for supervising an image segmentation task. The FCN generates a map of features with spatial information based on the input image, and a group of trained classifiers is applied to all the pixels of the feature map to produce segmentation maps. When incorporating the CLIP model into the FCN framework, there are two strategies outlined When incorporating the CLIP model into the FCN framework, there are two strategies outlined:

1. **Direct Pixel-wise Classification:** In this specific method, the CLIP vision encoder generates a feature map, which serves as the image representation of the input image. This image representation can be utilized for performing classification at the pixel level. However, the initial CLIP model represents the visual features of an image using the information encoded in the token [cls], which may not directly align with the requirements of the FCN framework. Additionally, there might be a resolution mismatch between the image size used in CLIP pre-training (224x224) and the higher image resolution typically required for semantic segmentation (e.g., 640x640). To address this, the sliding window technique is employed, where multi-scale inference is performed by sliding a window over the high-resolution image. This technique has been used in previous works and has been shown to improve performance.

2. **Retrained FCN-based Vision Encoder:** In this approach, a vision encoder constructed using a fully-connected network undergoes retraining on the classes that have been observed using a comparable technique. The CLIP text encoder is employed to derive consistent weights for the classifier, enabling the retrained model to possess a level of capability to generalize towards classes that have not been encountered before. By utilizing

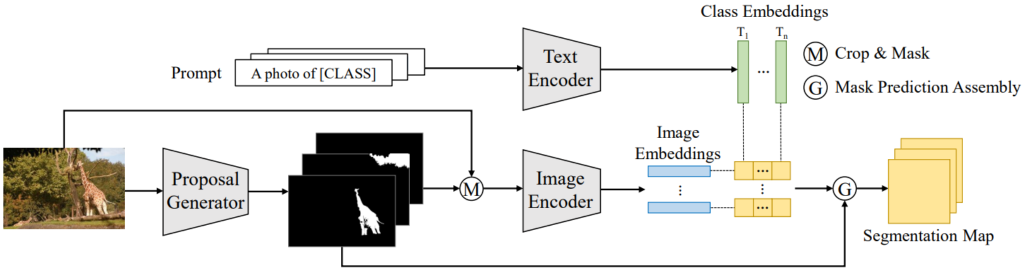


Figure 2.11: **ZS-MaskFormer for Zero-Shot Semantic Segmentation [45].**

the training data from previously encountered classes, this approach enhances the model's capacity to effectively handle categories that have been observed as well as those that are new or unfamiliar.

Similar to the two-stage framework, the predictions from these two strategies can be ensembled by default unless specified otherwise. This ensemble strategy aims to combine the strengths of both approaches and improve overall performance. The model architecture of Zero-Shot MaskFormer is depicted in figure 2.11.

# Chapter 3

# Methodology

## 3.1 Architecture Overview

Fig. 2.10 depicts our suggested ZSK-Net model. It is divided into two parts: (a) mask proposal creation and (b) region categorization. To create mask prediction and accompanying class probability, we use the notion of dynamic kernels[32, 7, 22, 12, 51], which are a collection of learnable parameters updated according to their corresponding characteristics in the picture. These are sent into the area classifier, which uses a CLIP-like[35] model to categorize the regions as visible or unseen.

### 3.1.1 Mask Proposal Generation

The mask proposal generator's purpose is to generate binary masks for all objects in the given picture. We reformulate K-Net [49] as a mask proposal generator to do this assignment.

The model begins by receiving an input picture $mathcalI$ and sending it to the backbone feature extractor.

When we consider the backbone as a function of $mathcalI$, we obtain the extracted features $\mathcal{F} in \mathbb{R}^{C\times H\times W}$ as an output of $\mathcal{B}(\mathcal{I})$, where $C$ is the number of output channels and $H$, $W$ are the height and width of the feature map. For our pipeline, we utilize an Atrous Spatial Pyramid Pooling (ASPP) head [8] as the kernel generator, which employs which uses $\mathcal{F}$ to generate $N$ initial D-dimensional kernels, $\mathcal{K}_0 \in \mathbb{R}^{N\times D}$, from which we obtain $N$ binary masks applying $\mathcal{M}_0^{pred} = \mathcal{K}_0 * \mathcal{F}$. The initial kernels $\mathcal{K}_0$ are iteratively updated through S stages to get the final set of $N$ refined, discriminative, and group-aware kernels $\mathcal{K}_s$.

The process of making kernels group-aware includes three critical steps: 1) grouping the properties unique to each of the $N$ kernels. This is accomplished by the use of binary masks generated by each kernel, which map important pixels of an image to the associated kernel. 2) A weighted summation of the kernels and their associated characteristics determined in the preceding step is used to update the kernels. 3) Obtaining kernel updates only based on their own characteristics appears counter-intuitive, as each kernel must also be aware of what the other $N-1$ kernels have learned. This is made easier by running the upgraded kernels via a multi-head attention (MHA) [39] module. The kernel update head governs the three processes that change static kernels to dynamic kernels in each of the model's S phases. The following equations can be used to define this process mathematically:

$$\mathcal{F}_K = \Sigma_u^H \Sigma_v^W \mathcal{M}_{prev}(u,v) \cdot \mathcal{F}(u,v) \tag{3.1}$$

$$\mathcal{F}_G = FC(\mathcal{F}_K) \otimes FC(\mathcal{K}_{prev}) \tag{3.2}$$

$$\mathcal{K}_{updated} = \mathcal{G}_F \otimes \mathcal{F}_K + \mathcal{G}_K \otimes \mathcal{K}_{prev} \tag{3.3}$$

$$\mathcal{K}_{next} = MHA(\mathcal{K}_{updated}) \tag{3.4}$$

$$\mathcal{M}_{next} = g_i(\mathcal{K}_{next}) * \mathcal{F} \tag{3.5}$$

The equation labeled as 3.1 groups the features for each kernel by multiplying the feature map $\mathcal{F}$ with the masks $\mathcal{M}_{prev}$. To calculate a gate feature $\mathcal{F}_G$, we multiply $\mathcal{F}_K$ with the kernels and pass them through a fully-connected $(FC)$ layer. The update process is performed by taking a weighted sum of the grouped features and kernels, as shown in Equation 3.3. The weights for these terms, denoted as $\mathcal{G}_F$ and $\mathcal{G}K$, are obtained by linearly transforming $\mathcal{F}G$ and applying a sigmoid operation. By convolving the feature map $\mathcal{F}$ with $g_i(\mathcal{K}next)$, we obtain the updated set of masks $\mathcal{M}next$. Here, $g_i$ represents a linear transformation followed by a ReLU activation and Layer Norm.

### 3.1.2 Region Classification

The Mask Proposal generation module generates Mask proposals $\mathcal{M}$ and corresponding class scores $\mathcal{P}$. These outputs are then inputted into the Region Classification module. The design of this module follows a CLIP [35] approach, where the class scores $\mathcal{P}$ are passed through the text encoder, and the Mask proposals $\mathcal{M}_S$ are passed through the image encoder of a CLIP model.

**Text encoder:** To obtain the class embedding $\mathcal{X}cls$, we utilize the text encoder $\mathcal{E}text$ of the CLIP model. Firstly, the text encoder is initialized with the class names provided in the text prompt $\mathcal{T}$. Then, we proceed to forward both the text prompt and class scores $\mathcal{P}$ to the text encoder $\mathcal{E}text$ of CLIP. This process results in the generation of the desired class embedding $\mathcal{X}cls$.

$$\mathcal{X}_{cls} = \mathcal{E}_{text}(\mathcal{P}, \mathcal{T}) \tag{3.6}$$

The $mathcalX_cls$ operate as fixed classifier weights for the mask proposal generator, allowing us to use the training data of seen classes without explicitly training the CLIP image encoder on them.

**Image encoder:** Simultaneously with the text encoder, the image encoder operates to produce the image embedding $\mathcal{X}img$ based on the mask proposal $\mathcal{M}S$. Initially, the mask is converted to binary form and used to mask the image, thereby extracting the foreground region. This process takes place within the "crop and mask" sub-module, depicted in Figure 2.10. The resulting foreground-extracted image is subsequently fed into the image encoder, resulting in the generation of the desired image embedding $\mathcal{X}_{img}$.

The class and image embeddings are integrated as follows to provide a unified embedding that is used to forecast the final segmentation mask:

$$\mathcal{X}_{combined} = \mathcal{X}_{cls} \cdot \mathcal{X}_{img} \tag{3.7}$$

## 3.2 Hungarian Mask Loss

During the training phase of our module, a bipartite matching loss [7, 10, 13] is utilized. In this process, the model generates $N$ mask proposals, where $N$ is often much larger than the actual number of objects present in the image. Consequently, many of these proposals do not correspond to meaningful objects in the image. To address this challenge, the bipartite matching strategy establishes a one-to-one mapping between each mask proposal and the ground truth masks. The set of ground truth masks includes an additional category for "no object" to which the irrelevant proposals are matched. This enables the model to automatically disregard masks that are not relevant to the given input image. This concept is similar to how it is used in object detection to avoid using post-processing techniques such as non-max suppression to reject undesired bounding boxes.

Table 3.1: Performance Comparison of the proposed ZSK-Net architecture on the benchmark Pascal-VOC dataset.

| | Pascal VOC | | |
|---|---|---|---|
| Method | hIoU | mIoU-seen | mIoU-unseen |
| SPNet[42] | 21.8 | 73.3 | 15.0 |
| ZS3-Net [4] | 28.7 | 77.3 | 17.7 |
| CaGNet [19] | 39.7 | 78.4 | 25.6 |
| SIGN [11] | 41.7 | 75.4 | 28.9 |
| ZS-MaskFormer [45] | 75.3 | 86.4 | 66.7 |
| ZSK-Net (Ours) | **85.7** | **91.3** | **80.7** |
| Gain | 10.4 ↑ | 4.9 ↑ | 14 ↑ |

Table 3.2: Performance Comparison of the proposed ZSK-Net architecture on the benchmark COCO-Stuff dataset.

| | COCO-Stuff | | |
|---|---|---|---|
| Method | hIoU | mIoU-seen | mIoU-unseen |
| SPNet[42] | 16.8 | 20.5 | 14.3 |
| ZS3-Net [4] | 15.0 | 34.7 | 9.5 |
| CaGNet [19] | 18.2 | 35.5 | 12.2 |
| SIGN [11] | 32.3 | 15.5 | 20.9 |
| ZS-MaskFormer [45] | 36.4 | **39.8** | 33.5 |
| ZSK-Net (Ours) | **37.3** | 39.7 | **35.4** |
| Gain | 0.9 ↑ | .1 ↓ | 1.9↑ |

# Chapter 4

# Experiments

## 4.1 Dataset

We performed experiments using the two widely utilized benchmark datasets for the Zero-Shot Semantic Segmentation (ZS3) task: the 'Pascal VOC' dataset [15] and the 'COCO-Stuff' dataset [5].

**Pascal VOC**: The Pascal VOC dataset consists of 11,185 training images and 1,449 validation images, encompassing a total of 20 distinct classes. For our experiments, we followed the approach outlined in [45] and divided the classes into 15 "seen" classes and 5 "unseen" classes. The Pascal VOC dataset includes objects such as chairs, bicycles, and dining tables, which are susceptible to occlusion. This presents a challenge for ZS3 models in accurately capturing the intricate details of these objects while generating mask proposals.

**COCO-Stuff**: With a total of 171 classes, which are divided into 156 "seen" classes and 15 "unseen" classes, this dataset is considered large-scale. It comprises 117,000 training images and 5,000 validation images. On average, each example in this dataset contains 7.7 objects, and approximately 90% of the images have multiple categories assigned to them. Consequently, this dataset poses an exceptionally challenging scenario for zero-shot semantic segmentation [31].

## 4.2 Evaluation Metrics

The model was tested using a variety of conventional evaluation techniques, including mean IoU (mIoU), pixel accuracy (pAcc), and harmonic mean IoU (hIoU). We choose hIoU as the primary comparison metric since it is a single number that indicates how the model fared on both the seen and unseen classes. This is a powerful metric that assesses the model's performance across all classes and a better indicator of a ZS3 model's resilience and generalizability. hIoU is defined by the following formula:

$$hIoU = \frac{2 \times mIoU_{seen} \times mIoU_{unseen}}{mIoU_{seen} + mIoU_{unseen}} \tag{4.1}$$

## 4.3 Implementation Details

For comparison purposes with state-of-the-art models, the default backbone used is ResNet-101. During both training and testing, the mask proposal generator generates a total of 100 mask suggestions. The models were trained using an RTX 3090 GPU. Unless specified otherwise, the chosen CLIP version employed the ViT-B/16 backbone. All experiments were conducted with a single cue as the input.

The learning rate is initially set to $1e^{-4}$, accompanied by a weight decay of $1e^{-4}$ and a polynomial learning rate decay with a power of 0.9. The batch size used for both datasets is 8. The training process consists of 40,000 iterations for the Pascal VOC dataset and 240,000 iterations for the COCO-Stuff dataset. The remaining hyperparameters are kept mostly consistent with the ones mentioned in [45].

Table 4.1: Class-wise performance analysis of both seen and unseen classes in Pascal VOC dataset. Unseen classes are highlighted in gray.

| **Metric** | aeroplane | bicycle | bird | boat | bottle | bus | car | cat | chair | cow | dining table | dog | horse | motro bike | person |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mIoU | 98.6 | 87.7 | 98.7 | 93.8 | 94.8 | 98.6 | 96.6 | 97.5 | 57.7 | 88.8 | 82.4 | 94.4 | 91.7 | 93.8 | 94.7 |
| Acc | 99.5 | 98.4 | 99.3 | 95.2 | 95.6 | 99.0 | 97.7 | 98.5 | 76.7 | 97.1 | 90.7 | 98.0 | 93.4 | 97.2 | 97.4 |

| **Metric** | potted plant | sheep | sofa | train | tv monitor |
|---|---|---|---|---|---|
| mIoU | 66.4 | 87.8 | 66.7 | 96.6 | 85.7 |
| Acc | 67.2 | 91.2 | 81.7 | 99.9 | 88.8 |

## 4.4 Comparison with Baselines

We evaluated our model against the top-performing ZS3 models on the Pascal VOC and COCO-Stuff datasets, without utilizing any additional techniques to enhance performance. The comparison results can be found in Table 3.1 for the Pascal VOC dataset and Table 3.2 for the COCO-Stuff dataset.

**Pascal VOC**: Our model demonstrated exceptional performance on the dataset, achieving a harmonic mean Intersection over Union (IoU) score of 85.7. This represents a significant improvement of +10.4 compared to the previous leading method, ZSMaskFormer [45]. Moreover, our pipeline outperformed the previous state-of-the-art approach on both the unseen and seen classes. Specifically, it achieved a score of 91.3 mean IoU (mIoU) on the seen classes and 80.7 on the unseen classes, surpassing the previous best model's scores of 86.4 and 66.7, respectively.

**COCO-Stuff**: Our model performed admirably on the COCO-Stuff validation set, getting a harmonic mean Intersection over Union (hIoU) score of 37.3. This is a significant improvement of +0.9 over the prior state-of-the-art approach. Our model performed significantly better on the unseen classes, with a mIoU of 35.4 compared to the previous best method's score of 33.5, representing a +1.9 improvement. Furthermore, our model's performance on the observed COCO-Stuff classes is comparable to the prior best technique.

**Observations**: The observations from Table 3.1 and 3.2 validate that the pixel-based approaches [42, 4, 19, 11] employed by models addressing ZS3 exhibit a notable bias towards the seen classes. This bias is evident from their inadequate performance on the unseen classes across both datasets. In comparison to the pixel-based techniques stated above, our decoupled formulation outperforms them not just on unseen classes, but also on seen classes. Our technique, ZSK-Net, produces the best results, with the greatest scores for both visible and unseen classes and the smallest variance between the two class types. This means that our model isn't biased toward either class type. These results confirm our hypothesis that dynamic kernels are useful for transferring training knowledge to the inference phase. This investigation also indicates that dynamic kernels may learn discriminative features, making them good segments for both visible and unseen classes.

## 4.5 Seen-to-Unseen Knowledge Transfer

The analysis of class-wise performance on the Pascal VOC validation set is presented in Table 4.1. The dataset was divided into 15 seen classes and 5 unseen classes, namely `potted plant`, `sheep`, `sofa`, `train`, and `TV monitor`. For each class, the table lists the mean Intersection over Union (mIoU) and pixel accuracy (pAcc) scores. The model demonstrates relatively higher scores for the `sheep` and `train` classes in terms of both mIoU and pAcc. Specifically, the mIoU and pAcc scores for `sheep` are 87.8 and 91.2, respectively, while those for `train` are 96.6 and 99.9, respectively. These scores are comparable to some of the best-performing seen classes, highlighting the model's strong performance on these specific classes. The relatively high performance of the model on the `sheep` and `train` classes can be attributed to their visual similarities with some of the seen classes. For instance, `sheep` shares certain features with classes like `horse`, `cat`, and `dog`, which allows the model to leverage its knowledge from these visually similar seen classes. Similarly, the `train` class exhibits similarities to other vehicle classes such as `bus` and `car`, enabling the model to generalize its understanding of vehicle-related features. The remarkable generalization ability of dynamic kernels has facilitated the model in learning visual features from seen classes and effectively transferring this knowledge to unseen classes, leading to high performance. This observation becomes evident when considering other unseen classes, such as `potted plant` and `sofa`, which lack visual similarities with any seen classes. In comparison, the model's performance in these classes is relatively lower than those with visual similarities. This further strengthens the notion that dynamic kernels serve as effective agents for generalization in zero-shot semantic segmentation tasks. Their ability to capture and transfer knowledge across classes that exhibit visual similarities contributes to their success in handling unseen classes.

Table 4.2: Considering different values for number of stages.

| Number of Stages | hoU | mIoU-`seen` | mIoU-`unseen` |
|---|---|---|---|
| 2 | 76.6 | 85.0 | 69.7 |
| 3 | 77.1 | 86.0 | 69.8 |
| 4 | 81.1 | 87.1 | 76.0 |
| 5 | 81.6 | 87.5 | 76.3 |
| 6 | 71.0 | 79.0 | 64.5 |

## 4.6 Ablation Study

We conducted an ablation study to analyze the impact of varying the number of stages in the pipeline. This parameter determines the model's size and affects the frequency of kernel updates until reaching the most refined version. By examining the results presented in Table 4.2, we observed a progressive improvement in harmonic mean IoU (hIoU) performance up to 5 stages. However, beyond 5 stages, the performance started to decline, indicating a saturation point. Notably, the performance gain from stage 4 to 5 in terms of hIoU was not substantial, even though a pipeline with 4 stages is computationally less demanding compared to one with 5 stages. Therefore, we determined that utilizing four stages strikes the optimal balance between performance and computational efficiency, and thus, we selected this value for the number of stages in our model.

## 4.7 Qualitative Analysis

In Figure 4.1, we present visualizations of multiple segmentation predictions and compare them with the results obtained using the current state-of-the-art model, ZSMaskFormer [45]. The visualization includes four segmentation results from the Pascal VOC dataset, featuring two unseen classes (`sofa` and `sheep`) and four visible classes (`dining-table`, `chair`, `person`, and `bicycle`). The figure is divided into four rows, each showcasing an example, and four columns displaying the original image, the ground truth, the prediction made by ZSMaskFormer (current state-of-the-art), and the prediction made by our model, ZSK-Net.

In the second case, which involves a `sheep`, our proposed method generates a prediction that closely resembles the ground truth, exhibiting a high level of accuracy. However, ZSMaskFormer struggles to accurately identify certain sheep pixels, leading to misclassifications.

Regarding the first image, which represents the unseen class `sofa`, ZSMaskFormer demonstrates a notable number of misclassified pixels. It mistakenly assigns sofa labels to various areas of the image that are not part of the sofa. In contrast, our model, ZSK-Net, surpasses this performance by reducing the occurrence of false positives. Although there is a minor flaw in labeling the cushion on the sofa as part of the sofa, overall, our model produces more accurate results. Now let's consider the seen classes. In the first example, the image contains a `dining-table` and a `chair`. Table 4.1 indicates that the mIoU score for the `chair` class is comparatively lower than the other seen classes. This could be attributed to chairs being frequently occluded by other objects in the image, making their segmentation challenging. Nevertheless, ZSK-Net performs better

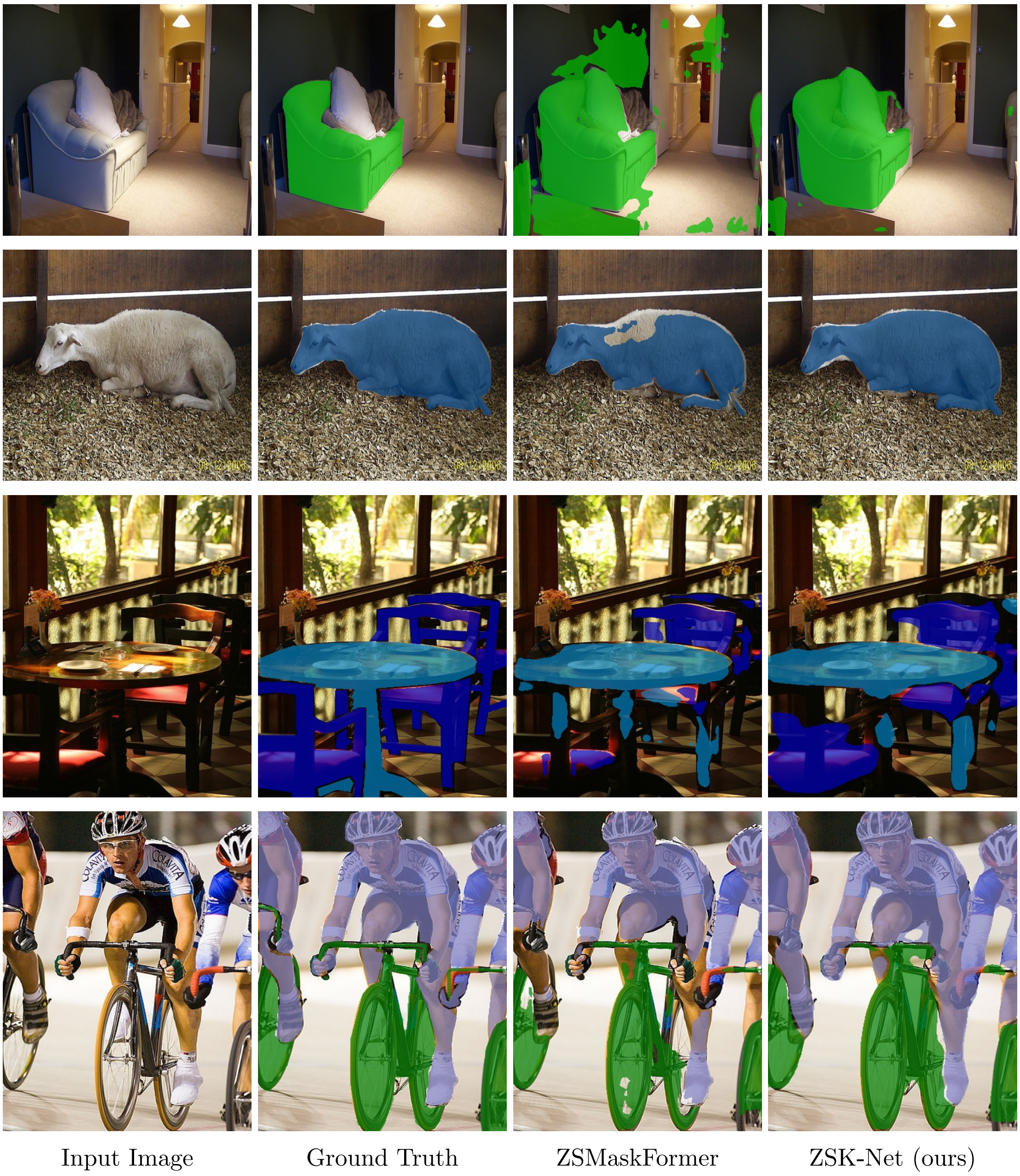


Figure 4.1: Qualitative results comparing the proposed model with the state-of-the-art. Results of both unseen (rows 1-2) and seen classes (rows 3-4) have been presented here.

in maximizing the true positives in this example, resulting in a more accurate segmentation map compared to ZSMaskFormer.

The final example showcases multiple `persons` as seen objects. ZSMaskFormer fails to accurately capture the leg regions, whereas our proposed method successfully incorporates this detail in its prediction. This qualitative experiment demonstrates the capability of dynamic kernels as effective segmentors for both seen and unseen classes. It showcases that our model, ZSK-Net, is able to leverage the advantages of dynamic kernels to improve segmentation performance across different object categories.

# Chapter 5

# Conclusion

This study introduces the concept of dynamic kernels within the realm of zero-shot semantic segmentation. The proposed pipeline leverages the feature understanding capabilities of dynamic kernels to generate region proposals in an input image. These proposals are then classified using a pre-trained vision-language model such as CLIP. Our experimental results on the Pascal VOC and COCO-Stuff datasets demonstrate that this approach achieves state-of-the-art performance in terms of segmentation accuracy. However, it should be noted that achieving very high harmonic mean IoU (hIoU) performance on datasets with a large number of classes still poses a challenge. Addressing this challenge and further improving the performance of the proposed technique on datasets with numerous object categories could be the focus of future research in this field.

# Bibliography

[1] Zeynep Akata, Florent Perronnin, Za&quot;id Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:1425–1438, 2016. 20

[2] Saeid Asgari Taghanaki, Kumar Abhishek, Joseph Paul Cohen, Julien Cohen-Adad, and Ghassan Hamarneh. Deep semantic segmentation of natural and medical images: a review. *Artificial Intelligence Review*, 54(1):137–178, Jan 2021. 5

[3] Donghyeon Baek, Youngmin Oh, and Bumsub Ham. Exploiting a joint embedding space for generalized zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9536–9545, 2021. 26

[4] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 3, 5, 7, 26, 27, 28, 29, 39, 42

[5] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 40

[6] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. *arXiv preprint arXiv:2105.05537*, 2021. 12

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 36, 38

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 11, 36

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 5

[10] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 1, 3, 8, 12, 14, 32, 38

[11] Jiaxin Cheng, Soumyaroop Nandi, Prem Natarajan, and Wael Abd-Almageed. Sign: Spatial-information incorporated generative network for generalized zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9556–9566, 2021. 39, 42

[12] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 36

[13] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11583–11592, 2022. 8, 38

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 11

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 40

[16] Rafael Felix, Ian Reid, Gustavo Carneiro, et al. Multi-modal cycle-consistent generalized zero-shot learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 21–37, 2018. 21

[17] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013. 20

[18] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. Transductive multi-view zero-shot learning. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2332–2345, 2015. 20

[19] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. Context-aware feature generation for zero-shot semantic segmentation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1921–1929, 2020. 26, 39, 42

[20] Amirhossein Habibian, Thomas Mensink, and Cees GM Snoek. Composite concept discovery for zero-shot video event detection. In *Proceedings of International Conference on Multimedia Retrieval*, pages 17–24, 2014. 21

[21] Shijie Hao, Yuan Zhou, and Yanrong Guo. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406:302–321, 2020. 5

[22] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 574–584, 2022. 36

[23] Ping Hu, Stan Sclaroff, and Kate Saenko. Uncertainty-aware learning for zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 33:21713–21724, 2020. 27

[24] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. Unet 3+: A full-scale connected unet for medical image segmentation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1055–1059. IEEE, 2020. 11

[25] Fabian Isensee, Paul F Jäger, Peter M Full, Philipp Vollmuth, and Klaus H Maier-Hein. nnu-net for brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 118–132. Springer, 2020. 5

[26] Naoki Kato, Toshihiko Yamasaki, and Kiyoharu Aizawa. Zero-shot semantic segmentation via variational mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 7, 27

[27] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958, 2009. 20

[28] Jingjing Li, Mengmeng Jing, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. Leveraging the invariant side of generative zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7402–7411, 2019. 21

[29] Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes. *IEEE transactions on medical imaging*, 37(12):2663–2674, 2018. 11

[30] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *International conference on machine learning*, pages 1718–1727. PMLR, 2015. 26

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 40

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 11, 36

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 11

[34] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015. 11

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 3, 21, 23, 25, 31, 36, 37

[36] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *International conference on machine learning*, pages 2152–2161. PMLR, 2015. 20

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5, 11

[38] Mert Bulent Sariyildiz and Ramazan Gokberk Cinbis. Gradient matching generative networks for zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2168–2178, 2019. 21

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 37

[40] Segnet Vijay, A Kendall, and R Cipolla. A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell*, (39):2481, 2015. 11

[41] Ce Wang, Moshiur Farazi, and Nick Barnes. Recursive training for zero-shot semantic segmentation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. 7

[42] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8256–8265, 2019. 26, 39, 42

[43] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551, 2018. 21

[44] Xiao Xiao, Shen Lian, Zhiming Luo, and Shaozi Li. Weighted res-unet for high-quality retina vessel segmentation. In *2018 9th international conference on infor-*

*mation technology in medicine and education (ITME)*, pages 327–331. IEEE, 2018. 11

[45] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for zero-shot semantic segmentation with pre-trained vision-language model. *arXiv preprint arXiv:2112.14757*, 2021. 1, 3, 8, 31, 35, 39, 40, 41, 42, 44

[46] Yang Yang, Yadan Luo, Weilun Chen, Fumin Shen, Jie Shao, and Heng Tao Shen. Zero-shot hashing via transferring supervised knowledge. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1286–1295, 2016. 21

[47] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *European Conference on Computer Vision*, pages 191–207. Springer, 2020. 11

[48] Hui Zhang and Henghui Ding. Prototypical matching and open set rejection for zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6974–6983, 2021. 7, 26

[49] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. *Advances in Neural Information Processing Systems*, 34:10326–10338, 2021. 1, 3, 16, 19, 20, 36

[50] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 11

[51] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 11, 36

[52] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 3–11. Springer, 2018. 11