SeeDS: Semantic Separable Diffusion Synthesizer for Zero-shot Food Detection

Pengfei Zhou
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences (CAS)
University of Chinese Academy of Sciences
Beijing, China
pengfei.zhou@vipl.ict.ac.cn

Weiqing Min*†
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS
University of Chinese Academy of Sciences
Beijing, China
minweiqing@ict.ac.cn

Yang Zhang
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS
University of Chinese Academy of Sciences
Beijing, China
yang.zhang@vipl.ict.ac.cn

Jiajun Song
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS
University of Chinese Academy of Sciences
Beijing, China
jiajun.song@vipl.ict.ac.cn

Ying Jin
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS
University of Chinese Academy of Sciences
Beijing, China
yingyuan0226@gmail.com

Shuqiang Jiang†
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS
University of Chinese Academy of Sciences
Beijing, China
sqjiang@ict.ac.cn

ABSTRACT

Food detection is becoming a fundamental task in food computing that supports various multimedia applications, including food recommendation and dietary monitoring. To deal with real-world scenarios, food detection needs to localize and recognize novel food objects that are not seen during training, demanding Zero-Shot Detection (ZSD). However, the complexity of semantic attributes and intra-class feature diversity poses challenges for ZSD methods in distinguishing fine-grained food classes. To tackle this, we propose the **Se**mantic S**e**parable **D**iffusion **S**ynthesizer (SeeDS) framework for Zero-Shot Food Detection (ZSFD). SeeDS consists of two modules: a Semantic Separable Synthesizing Module (S^3M) and a Region Feature Denoising Diffusion Model (RFDDM). The S^3M learns the disentangled semantic representation for complex food attributes from ingredients and cuisines, and synthesizes discriminative food features via enhanced semantic information. The RFDDM utilizes a novel diffusion model to generate diversified region features and enhances ZSFD via fine-grained synthesized features. Extensive experiments show the state-of-the-art ZSFD performance of our proposed method on two food datasets, ZSFooD and UECFOOD-256. Moreover, SeeDS also maintains effectiveness on general ZSD datasets, PASCAL VOC and MS COCO. The code and dataset can be found at https://github.com/LanceZPF/SeeDS.

CCS CONCEPTS

• **Computing methodologies** → **Object detection**.

KEYWORDS

food detection; zero-shot detection; food computing; zero-shot learning; diffusion model

ACM Reference Format:
Pengfei Zhou, Weiqing Min, Yang Zhang, Jiajun Song, Ying Jin, and Shuqiang Jiang†. 2023. SeeDS: Semantic Separable Diffusion Synthesizer for Zero-shot Food Detection. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3581783.3612661

1 INTRODUCTION

Food computing [28], as an interdisciplinary field, utilizes computational methods to understand, model, and enhance human-food interactions, thereby offering a wide range of applications in health and nutrition areas [24, 25, 47]. As one key task in food computing, food detection aims to locate and recognize food objects simultaneously [1, 23, 29]. Food detection can enable various applications such as food recommendation, dietary assessment, and robotics control [27, 45, 48]. However, it is challenging to detect food objects under real-world scenarios due to the constant emergence of novel food classes, such as the continued updates of food categories in restaurants [40]. In this case, continuously collecting and annotating new food objects is unrealistic. To address this, food detection needs the ability of Zero-Shot Detection (ZSD) to detect novel food classes that have no samples during training.

*Corresponding author.
†Weiqing Min and Shuqiang Jiang are also with the Institute of Intelligent Computing Technology, Chinese Academy of Sciences, Suzhou, China.

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10.
https://doi.org/10.1145/3581783.3612661

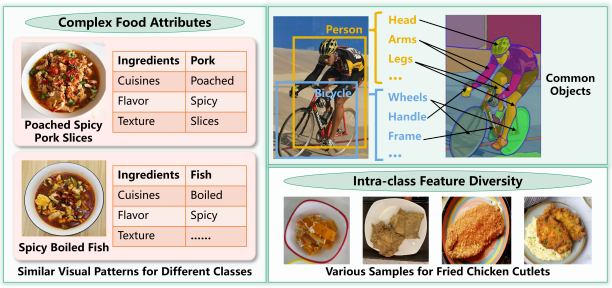

Figure 1: Our motivation: common objects possess certain semantic parts, and food objects lack such structural patterns to match with complex food attributes. Moreover, intra-class features diversify in the same food category.

ZSD emerges with the ability to detect unseen objects belonging to novel categories in real-world scenarios [4, 53]. It enables the transfer of knowledge from seen to unseen classes by incorporating semantic information from external sources like word embeddings. Based on this, ZSD researchers develop mapping-based methods [5, 50] and generation-based methods [12, 15]. The former learns a mapping function to align visual features and semantic features in a common space, tackling unknown objects by neighbor searching in this space. However, in a more complex setting of Generalized Zero-Shot Detection (GZSD) [4], where both seen and unseen objects appear during inference, mapping-based methods would suffer from bias toward seen objects. To address this, generation-based methods are introduced with more robust GZSD performance. These methods use generative models (e.g., VAEs [19] and GANs [9]) to synthesize unseen features and train the zero-shot detector on these synthesized features. This paper reimplements general ZSD methods to food scenarios as an initial attempt at Zero-Shot Food Detection (ZSFD). However, current ZSD methods still meet difficulties when detecting unseen food categories since they are designed without food domain knowledge.

Compared with general ZSD, ZSFD is more challenging due to fine-grained issues of food categories, resulting in the complexity of food attributes and diversity of intra-class features. One of the major problems introduced by complex food attributes for ZSFD is semantic confusion. Similar to the concept of ZSD, ZSFD uses semantic information to bridge the gap between seen and unseen classes due to the lack of visual data for unseen objects. However, unlike common objects that often have distinct semantic parts for each class (e.g., head and limbs for *Person*), food objects have no such structural visual patterns, making it harder to distinguish between different food objects with similar semantic attributes. For instance, *Poached Spicy Pork Slices* and *Spicy Boiled Fish* in Fig. 1 share the same visual pattern, leading to difficulties distinguishing them using mere word embeddings. To address this semantic confusion, we need to extract multi-source semantic information, including ingredients and cuisines, to distinguish between similar food categories effectively.

Another challenge in ZSFD is the intra-class feature diversity, complicating fine-grained food detection. Specifically, dishes within the same food category can have completely different visual patterns. For example, instances of *Fried Chicken Cutlets* have different appearances in Fig. 1, resulting in various visual features. Consequently, we need to synthesize diversified features for food classes to ensure the accuracy of zero-shot food detectors. However, existing generative models used in ZSD like GANs, have limitations in generating realistic and diversified food features for unseen food classes [49]. For example, the training of GANs is unstable and easily suffers from the model collapse problem since it is difficult to converge, which results in similar generated samples for each food class. Therefore, a stable generation model that can generate more diversified and realistic features for ZSFD tasks is to be explored.

To address these challenges, we propose Semantic Separable Diffusion Synthesizer (SeeDS), a novel ZSFD approach that overcomes these limitations by generating high-quality fine-grained features based on the advanced generative framework. SeeDS consists of two main modules: a Semantic Separable Synthesizing Module (S^3M) and a Region Feature Denoising Diffusion Model (RFDDM). The S^3M aims to enhance the semantic information by separating the food attributes according to two different domains: ingredient attributes and cuisine attributes. It can further learn disentangled semantic representation separately and synthesize discriminative food features for unseen classes via aggregating more abundant semantic information. The RFDDM leverages the latest diffusion model to generate food region features by reversing a Markov chain from noise to data. It takes the synthesized visual contents from the S^3M as the condition and applies a denoising process to generate more diverse and realistic food region features that can better capture the fine-grained characteristics of food items. Finally, a robust zero-shot food detector in SeeDS can be trained on the discriminative and diversified unseen food features.

Overall, our main contributions can be summarized as follows:

- We propose a novel Semantic Separable Diffusion Synthesizer (SeeDS), which overcomes the limitations of general ZSD frameworks by generating high-quality fine-grained features for detecting unseen food objects.
- We introduce two modules in SeeDS. To address complex attribute issues, we present a Semantic Separable Synthesizing Module (S^3M) that enhances semantic information by learning disentangled ingredient and cuisine representation. To tackle intra-class feature diversity, we propose a Region Feature Denoising Diffusion Model (RFDDM) that leverages a novel diffusion model to generate more diverse and realistic food region features.
- We evaluate our proposed framework on two food datasets ZSFooD and UECFOOD-256 [17], resulting in state-of-the-art ZSFD performance. Additionally, experiments on widely-used PASCAL VOC and MS COCO demonstrate the effectiveness of our approach for general ZSD.

2 RELATED WORK

2.1 Zero-shot Learning

Zero-Shot Learning (ZSL) is a machine learning branch enabling models to recognize unseen images [39]. Existing ZSL methods utilize semantic information about novel images and primarily follow two zero-shot strategies: mapping-based approach [41] and generation-based approach [39]. Mapping-based approach projects extracted visual and semantic features into the same space and

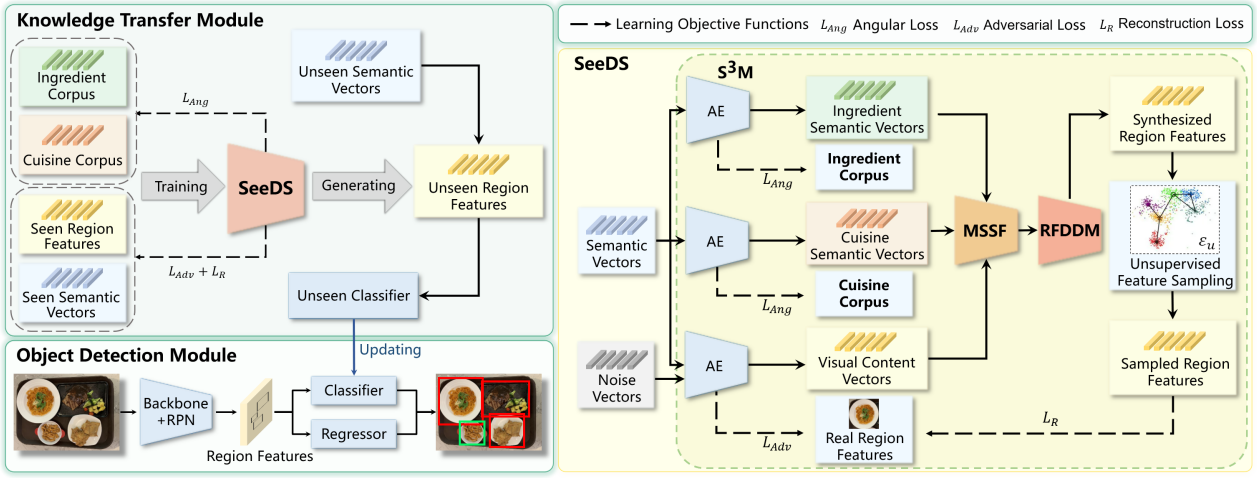

Figure 2: Framework of our approach. SeeDS consisting of the Semantic Separable Synthesizing Module (S^3M) and the Region Feature Denoising Diffusion Model (RFDDM) works as the knowledge transfer module. A zero-shot food detector can be obtained by combining an unseen classifier trained on the generated unseen features.

searches the nearest neighbor in the embedding space for input samples [20]. Generation-based approach transforms zero-shot problems into supervised learning via synthesized features [10, 11]. Specifically, generative models including VAEs [19] and GANs [9] are used to generate samples of novel classes based on their annotated attributes or semantic embeddings, which are obtained from large-scale pretrained language models like BERT [6] and CLIP [32]. Furthermore, the generated samples can then be used to train classifiers for recognizing novel unseen classes.

2.2 Zero-shot Detection

Compared to the ZSL task, Zero-shot Detection (ZSD) presents greater challenges [34, 44]. ZSD approaches, including mapping-based [21, 50, 52] and generation-based [12, 15, 51], have been proposed, grounded in ZSL theory. For example, Bensal *et al.* [4] propose two mapping-based approaches that use background-aware representations to improve the ZSD performance. They also provide a new evaluation metric called Generalized Zero-Shot Detection (GZSD), which aims to detect both seen and unseen classes during evaluation. Generation-based ZSD methods are developed with better GZSD performance [12]. For example, Zhu *et al.* [53] propose an unseen feature generation framework based on VAE. Huang *et al.* [15] synthesize unseen features by a structure-aware GAN. However, due to the limitations of GANs and VAEs in generating diverse and realistic features, their real-world application proves difficult. Recently, diffusion models emerge as powerful generative models [14, 38, 42]. Unlike GANs, they avoid training instability and model collapse and allow diversity control during generation. Therefore, we are the first to apply diffusion models to ZSD tasks.

2.3 Food Detection

Food detection [1] is an essential task in the field of food computing [28], attracting significant interest for its potential applications in the computer vision and multimedia community [3, 7, 31]. However, food detection is a challenging task due to the complex characteristics of their ingredients, cuisines, flavors, etc. Additionally, it is difficult to distinguish between different food objects due to the intra-class variability and inter-class similarity of fine-grained food features [35]. One common approach for food detection is to implement general object detection frameworks. For example, Sun *et al.* [43] propose a mobile application to detect food items based on YOLOv2 [36]. Shimoda *et al.* [40] propose a weakly-supervised food region proposal method using fully convolutional networks trained on web images. However, these food detection models meet difficulties when applied to real-world tasks. A major reason is that classes of meals are constantly updated in real-world scenarios (e.g., in restaurants), and food detectors trained with fixed classes can barely handle novel classes. Therefore, we introduce a novel zero-shot food detection framework SeeDS, which enables the effective zero-shot detection of novel food objects.

3 METHOD

Problem Formulation. The aim of the ZSFD task is to learn a detector on the training set $\mathcal{X}_s$ with semantic vectors and detect unseen objects in the test set. We define $\mathcal{X}_s$ that includes A_s images, B_s bounding box annotations, and N_s seen food categories as the training set, and $\mathcal{O}_s$ as the available annotation set. Each food object $\boldsymbol{o}_m^i \in \mathcal{O}_s$ is annotated with a bounding box and a class label $y_m^i \in \mathcal{Y}_s$. $\mathcal{Y}_s = \{Y_1, ..., Y_{N_s}\}$ and $\mathcal{Y}_u = \{Y_{N_s+1}, ..., Y_N\}$ are class label sets of seen classes and unseen classes respectively, where $\mathcal{Y}_s \cap \mathcal{Y}_u = \emptyset$, $N = N_s + N_u$ is the number of all classes and N_u is the number of unseen classes. Corresponding to class labels, semantic vector set $\mathcal{V} = \mathcal{V}_s \cup \mathcal{V}_u$ is given, where $\mathcal{V}_s$ and $\mathcal{V}_u$ are semantic vector sets of seen and unseen classes, respectively. The semantic vector $\boldsymbol{v} \in \mathcal{V}$ is word embeddings extracted from language models. During the inference, a test set $\mathcal{X}_t$ that contains both N_s seen classes and N_u unseen classes is given. ZSFD also evaluates methods on an unseen set $\mathcal{X}_u \subset \mathcal{X}_t$ that only contains unseen classes.

Framework Overview. As shown on the left of Fig. 2, the proposed framework consists of two parts. An object detection module ϕ_d based on the backbone detector is first trained with images in $\mathcal{X}_s$ containing food object annotations of seen classes, and then used

Algorithm 1 The framework of our ZSFD approach

Require: Training set $\mathcal{X}_s$ with food images and annotations, seen semantic vector set $\mathcal{V}_s$ and unseen semantic vector set $\mathcal{V}_u$
Ensure: Zero-shot food detector with parameters ϕ_d
1: $\phi_d \leftarrow$ Train detector on $\mathcal{X}_s$ with annotations
2: $\mathcal{E}_s \leftarrow$ Extract $\mathcal{E}_s$ region features from $\mathcal{X}_s$ via ϕ_d
3: $G \leftarrow$ Train Semantic Separable Diffusion Synthesizer G on $\mathcal{E}_s$ with corresponding semantic vectors from $\mathcal{V}_s$
4: $\mathcal{E}_u \leftarrow$ Synthesize unseen region features using G and $\mathcal{V}_u$
5: $\phi_{uc} \leftarrow$ Train unseen classifier ϕ_{uc} using $\mathcal{E}_u$ with class labels
6: $\phi_d \leftarrow$ Update parameters in ϕ_d with ϕ_{uc}
7: **return** ϕ_d

to extract region features $\mathcal{E}_s$ of seen food objects. In the knowledge transfer module, we train a Semantic Separable Diffusion Synthesizer (SeeDS) G utilizing the extracted region features $\mathcal{E}_s$, the semantic vectors $\mathcal{V}_s$ according to their food classes, and corpora of ingredient and cuisine. Furthermore, we use G for generating robust and diverse unseen features $\mathcal{E}_u$ via unseen semantic vectors $\mathcal{V}_u$. An unseen classier ϕ_{uc} is further trained on the generated features $\mathcal{E}_u$ and combined into the original detector ϕ_d. Updating the parameters in the detector with the parameters of the unseen classifier, an efficient zero-shot food detector that can locate and recognize unseen food objects is obtained. The framework of our approach is also summarized in Algorithm 1.

3.1 Semantic Separable Synthesizing Module

Disentangled Semantic Knowledge Learning. In our proposed SeeDS, the Semantic Separable Synthesizing Module (S^3M) first learns the semantic representation of ingredients and cuisines based on Disentangled Semantic Knowledge Learning. To learn the semantic representation for fine-grained food classes with domain knowledge, we adopt a disentangled framework consisting of three branches. Two branches of these correspond to the semantic information of ingredients and cuisines. On each branch, an Auto-Encoder (AE) takes the word embeddings as the input semantic vectors, encodes them into latent semantic vectors, and decodes them into reconstructed semantic vectors.

We introduce two domain-knowledge corpora including an ingredient corpus and a cuisine corpus. Each corpus contains a bag of words that are relevant to the specific domain. For example, the ingredient corpus contains ingredient words like *Tomato*, *Eggs*, and *Onion*, while the cuisine corpus contains cuisine words like *Scrambled*, *Stewed*, and *Fried*. The objective of each branch in the disentangle framework is to minimize the angular loss between the input vector and the specific domain knowledge corpus. We also construct two learnable attention masks on both branches: an ingredient attention mask and a cuisine attention mask. Each attention mask is learned as a binary vector representing which words in the corpus are close to the class embedding vector. For example, if the class embedding vector is from *Scrambled Eggs with Onion*, then the ingredient attention mask is learned to be [0, 1, 1] and the cuisine attention mask is learned to be [1, 0, 0]. We apply these learnable attention masks as the objective vectors $\boldsymbol{M}^p \in \mathbb{R}^{n \times a^p}$ in the calculation of the training objective on the p-th branch:

$$\mathcal{L}^p_{Ang} = -\frac{1}{n \cdot a_p} \sum_{i=1}^{n} \sum_{j=1}^{a^p} M^p_{i,j} \cdot \log(\mathrm{Sig}(\mathrm{Ang}(\tilde{\boldsymbol{V}}_i, \boldsymbol{K}^p_j))) + (1 - M^p_{i,j}) \cdot \log(1 - \mathrm{Sig}(\mathrm{Ang}(\tilde{\boldsymbol{V}}_i, \boldsymbol{K}^p_j))), \tag{1}$$

where n is the batch size, a^p is the size of a domain-knowledge corpus, $\tilde{\boldsymbol{V}}_i \in \mathbb{R}^s$ is the reconstructed semantic vector, $\boldsymbol{K}^p_j \in \mathbb{R}^s$ is the j-th word vector in the p-th domain-knowledge corpus, s is the dimension of word embeddings, $\mathrm{Sig}(\cdot)$ is the sigmoid function, and $\mathrm{Ang}(\tilde{\boldsymbol{V}}_i, \boldsymbol{K}^p_j)$ is the cosine similarity between the decoded semantic vector and the corresponding corpus:

$$\mathrm{Ang}(\tilde{\boldsymbol{V}}_i, \boldsymbol{K}^p_j) = \frac{\tilde{\boldsymbol{V}}_i \cdot \boldsymbol{K}^p_j}{\max(\|\tilde{\boldsymbol{V}}_i\|_2 \cdot \|\boldsymbol{K}^p_j\|_2, \epsilon)}, \tag{2}$$

where ϵ is a small value to avoid division by zero. By optimizing the objective for decoded semantic embeddings and domain-knowledge corpora with learnable attention masks, two sets of semantic vectors are obtained: ingredient semantic vectors $\mathbf{V}_I \in \mathbb{R}^{n \times s}$ and cuisine semantic vectors $\mathbf{V}_C \in \mathbb{R}^{n \times s}$.

Multi-Semantic Synthesis Fusion. The main challenge in our proposed S^3M is how to combine the separately generated semantic vector into a unified synthesized feature with rich knowledge. To address this challenge, we introduce Multi-Semantic Synthesis Fusion (MSSF) with a Content Encoder and a Fusion Decoder.

As shown in Fig. 3, an AE is applied as the Content Encoder for each branch. Each AE adopts two linear layers as the encoder $\mathrm{ENC}(\cdot)$ and two linear layers as the decoder $\mathrm{DEC}(\cdot)$, activated by LeakyReLU. For example, the Content Encoder takes $\mathbf{V}_I$ as input on the ingredient branch, encodes it with the visual content vectors $\boldsymbol{X}$ generated from the AE in the adversarial branch, and maps the latent representation into the ingredient content vectors:

$$\boldsymbol{N}_I = \mathrm{DEC}(\mathrm{ENC}(\boldsymbol{Z} \otimes \boldsymbol{X} \otimes \boldsymbol{V}_I) \otimes \boldsymbol{X} \otimes \boldsymbol{V}_I), \tag{3}$$

where $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, $\boldsymbol{Z} \in \mathbb{R}^{n \times d}$ are sampled noise vectors used for expanding the spanning space of synthesizing, d is the dimension of the region feature, $\boldsymbol{V} \in \mathbb{R}^{n \times s}$ denotes input word embeddings, $\otimes$ denotes the concatenate operation, $\boldsymbol{N}_I \in \mathbb{R}^{n \times e}$ and e is the new embedding dimension. The cuisine content vectors $\boldsymbol{N}_C \in \mathbb{R}^{n \times e}$ on the cuisine branch are obtained following the same pipeline.

The Fusion Decoder further takes separate content vectors as inputs and decodes them into synthesized features that combine content information from ingredients and cuisines. Two Adaptive Instance Normalization (AdaIN) [16] blocks with two linear transformations are adopted to normalize the content with the semantic representation from the different branches:

$$\mathrm{AdaIN}(\boldsymbol{N}_I, \boldsymbol{N}_C) = \sigma(\boldsymbol{N}_C) \left(\frac{\boldsymbol{N}_I - \mu(\boldsymbol{N}_I)}{\sigma(\boldsymbol{N}_I)} \right) + \mu(\boldsymbol{N}_C), \tag{4}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are the mean and standard deviation of vectors, respectively. Finally, we obtain synthesized features $\boldsymbol{E} \in \mathbb{R}^{b \times d}$, where b is the synthesis number controlled by sample times.

Unsupervised Feature Sampling. ZSFD needs high-quality features to learn a detector that can distinguish various unseen food

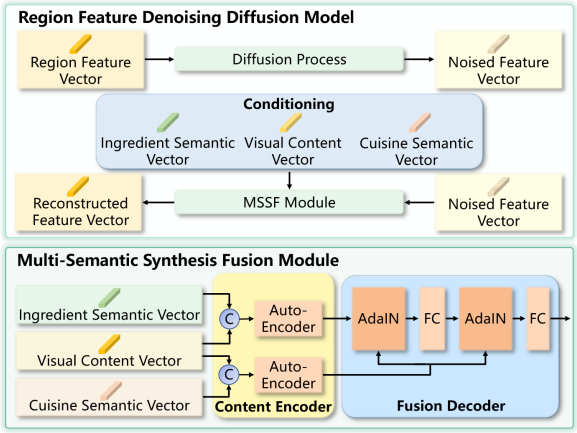

Figure 3: Detailed architecture of RFDDM and MSSF module. MSSF is used to predict the noise based on the conditions.

objects. We should maximize the inter-class differentiation of synthesized features to deal with the fine-grained food problem. We use K-Means clustering to sample representative features $\boldsymbol{E}' \in \mathbb{R}^{b' \times d}$ from all generated data to balance the trade-off between the diversity and quality of generated samples and reduce the computational cost for further training. Specifically, we cluster synthesized features into S clusters and select $b' = S \cdot P$ features from all clusters. We rank features in each cluster by their distance to the center. Assuming that the feature closer to the cluster center has higher quality and better generalization ability, we choose the top P features according to their distance score in each cluster. We ensure that selected samples are evenly chosen from each cluster.

3.2 Region Feature Denoising Diffusion Model

The core generator in our SeeDS is a newly proposed Region Feature Denoising Diffusion Model (RFDDM). RFDDM can be used to generate 1D feature vectors, which can improve the diversity of synthesized region features in SeeDS. As shown in Fig. 3, the RFDDM is based on the idea of modeling the data distribution learns to generate samples by applying a series of denoising steps to reverse the diffusion process that removes the sampled noise and recovers the region feature.

Let $\boldsymbol{x} \in \mathbb{R}^d$ be a 1D region feature vector. As illustrated in 4, we assume that $\boldsymbol{x}_T$ is generated by a diffusion process that starts from the sample $\boldsymbol{x}_0 \sim p_0(\boldsymbol{x})$, where p_0 is the data distribution, and Gaussian noise is added at each timestep $t = 1, \ldots, T$ according to the Markovian process. The noise level at each timestep is controlled by a scalar $\beta_t \in (0, 1)$. The forward diffusion process $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ for each timestep can be described as:

$$\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{z}_t, \tag{5}$$

where $\boldsymbol{z}_t \in \mathbb{R}^d$ is sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{x}_0 = \boldsymbol{x}$. The RFDDM aims to reverse this process, which is given by:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = N(x_{t-1}; \mu_\theta(\boldsymbol{x}_t, t), \Sigma_\theta(\boldsymbol{x}_t, t)). \tag{6}$$

RFDDM uses the parameters of the MSSF module, which is shown in Fig. 3, to predict the recover region feature utilizing the covariance $\Sigma_\theta(\boldsymbol{x}_t, t)$ and the mean $\mu_\theta(\boldsymbol{x}_t, t)$:

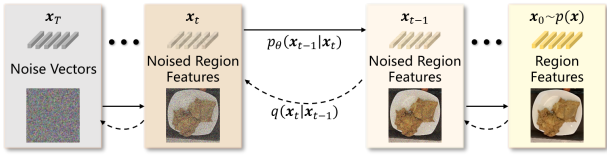

Figure 4: The visual illustration of the diffusion process in RFDDM. The forward process $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ continually add Gaussian noise to $\boldsymbol{x}_{t-1}$ (from right to left), the reverse process $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ aims to denoise the noised feature vector $\boldsymbol{x}_t$.

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_t - \frac{1-\alpha}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{z}_\theta(\boldsymbol{x}_t, t)), \tag{7}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ is the accumulated noise scalars, and $\boldsymbol{z}_\theta(\boldsymbol{x}_t, t)$ is the predicted noise parameterized by RFDDM. Thus we can map $\boldsymbol{x}_t$ to $\boldsymbol{x}_0$ by applying a series of denoising functions $\boldsymbol{F}_t$:

$$\boldsymbol{x}_{t-1} = \boldsymbol{F}_t(\boldsymbol{x}_t, t, \boldsymbol{z}_\theta(\boldsymbol{x}_t, t); \theta), \tag{8}$$

where θ are the parameters of the MSSF module in RFDDM. The denoising functions $\boldsymbol{F}_t$ are implemented by MSSF modules that share the same architecture but have different parameters for each timestep. The RFDDM is trained by minimizing the mean squared error between the real noise $\boldsymbol{z}_t$ and $\boldsymbol{z}_\theta(\boldsymbol{x}_t, t)$ for all timesteps:

$$\begin{aligned}\mathcal{L}_R &= \mathbb{E}_{\boldsymbol{x}, \boldsymbol{z}_t}[\sum_{t=1}^{T} ||\boldsymbol{z}_t - \boldsymbol{z}_\theta(\boldsymbol{x}_t, t)||^2] \\ &= \mathbb{E}_{\boldsymbol{x}, \boldsymbol{z}}[\sum_{t=1}^{T} ||\boldsymbol{z}_t - \boldsymbol{F}_t(\boldsymbol{x}_t, t, \boldsymbol{z}_\theta(\boldsymbol{x}_t, t); \theta)||^2].\end{aligned} \tag{9}$$

3.3 Loss Functions

Given the seen feature collection $\mathcal{E}_s$ with semantic vector set $\mathcal{V}$ from $\mathcal{X}_s$ and Gaussian noise set $\mathcal{Z}$, our goal is to learn a synthesizer G: $(\mathcal{V} \times \mathcal{Z}) \mapsto \mathcal{E}$, which takes a semantic vector $\boldsymbol{v} \in \mathcal{V}_u$ and a sampled random noise $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$ as inputs and outputs the synthesized region feature $\tilde{\boldsymbol{e}} \in \mathcal{E}_u$. Specifically, the total training objective in SeeDS comprises three parts: the angular loss $\mathcal{L}_{Ang}^p$ that is used to learn domain-specific semantic representations, the adversarial loss $\mathcal{L}_{Adv}$ that is used to learn the visual content and the reconstruction loss $\mathcal{L}_R$ that is used to train the RFDDM.

$$\mathcal{L}_{Total} = \lambda_1(\mathcal{L}_{Ang}^1 + \mathcal{L}_{Ang}^2) + \lambda_2\mathcal{L}_{Adv} + \lambda_3\mathcal{L}_R, \tag{10}$$

where $\mathcal{L}_{Ang}^1$ and $\mathcal{L}_{Ang}^2$ are angular losses on ingredient and cuisine branches respectively. λ_1, λ_2 and λ_3 control the weights of the three losses. Specifically, the proposed $\mathcal{L}_{Adv}$ is used to adversarially train an AE that generates the visual content, which is one of the condition vectors for RFDDM to synthesize diverse samples:

$$\mathcal{L}_{Adv} = \min_G \max_D \mathcal{L}_W + \mathcal{L}_C + \mathcal{L}_S, \tag{11}$$

where $\mathcal{L}_W$ is the conditional Wasserstein GAN loss [2], $\mathcal{L}_C$ is the classifier alignment loss [12], $\mathcal{L}_S$ is the semantic diverging loss [15].

Table 1: Statistics of the proposed ZSFooD and existing datasets that are widely used for ZSD.

Dataset	Classes			Images		
	S	U	Total	Training	Test	Total
PASCAL VOC [8]	16	4	20	10,728	10,834	21,562
MS COCO [22]	65	15	80	82,783	40,504	123,287
UECFOOD-256 [17]	205	51	256	20,452	5,732	26,184
ZSFooD	184	44	228	10,463	10,140	20,603

Table 2: Comparison with baseline methods on ZSFooD (%).

Metric	Method	ZSD	GZSD		
			Seen	Unseen	HM
Recall@100	ConSE [30]	39.7	58.0	38.1	46.4
	BLC [52]	41.2	55.3	40.5	46.8
	CZSD [50]	48.0	86.1	44.8	58.9
	SU [12]	45.3	82.3	44.1	57.4
	RRFS [15]	48.8	86.6	47.6	61.4
	SeeDS	**52.9**	**87.0**	**49.8**	**63.3**
mAP	ConSE [30]	0.8	54.3	0.7	1.4
	BLC [52]	1.1	51.1	0.9	1.8
	CZSD [50]	4.0	81.2	2.1	4.1
	SU [12]	3.9	79.1	2.3	4.5
	RRFS [15]	4.3	82.7	2.7	5.2
	SeeDS	**5.9**	**82.8**	**3.5**	**6.7**

Table 3: Comparison with baselines on UECFOOD-256 (%).

Metric	Method	ZSD	GZSD		
			Seen	Unseen	HM
Recall@100	CZSD [50]	60.7	**57.6**	45.5	50.8
	SU [12]	61.9	52.5	52.8	52.6
	RRFS [15]	64.8	54.9	55.1	55.0
	SeeDS	**74.0**	55.2	**61.4**	**58.1**
mAP	CZSD [50]	22.0	**20.8**	16.2	18.2
	SU [12]	22.4	19.3	20.1	19.7
	RRFS [15]	23.6	20.1	22.9	21.4
	SeeDS	**27.1**	20.2	**26.0**	**22.7**

Table 4: Comparison of mAP on PASCAL VOC (%).

Model	ZSD	GZSD		
		Seen	Unseen	HM
SAN [34]	59.1	48.0	37.0	41.8
HRE [5]	54.2	**62.4**	25.5	36.2
PL [33]	62.1	-	-	-
BLC [52]	55.2	58.2	22.9	32.9
SU [12]	64.9	-	-	-
RRFS [15]	65.5	47.1	49.1	48.1
SeeDS	**68.9**	48.5	**50.6**	**49.5**

Table 5: Comparison of Class-wise AP and mAP for different methods on unseen classes of PASCAL VOC (%).

Method	car	dog	sofa	train	mAP
SAN [34]	56.2	85.3	62.6	26.4	57.6
HRE [5]	55.0	82.0	55.0	26.0	54.5
PL [33]	**63.7**	87.2	53.2	44.1	62.1
BLC [52]	43.7	86.0	60.8	30.1	55.2
SU [12]	59.6	92.7	62.3	45.2	64.9
RRFS [15]	60.1	93.0	59.7	49.1	65.5
SeeDS	60.4	**95.3**	**65.9**	**53.8**	**68.9**

4 EXPERIMENTS

4.1 Experimental Settings

Dataset Splittings. We adopt two datasets to evaluate the ZSFD performance: our constructed ZSFooD and widely-used UECFOOD-256 [17]. UECFOOD-256 is a food detection dataset reformable into a ZSFD benchmark but mainly contains images with a single food object, and thus only provides 28,429 bounding box annotations. Compared with it, ZSFooD has 20,603 food images collected in 10 restaurant scenarios, each with multiple food objects annotated with bounding boxes. ZSFooD is more challenging with 95,322 bounding boxes and 291 classes. Following the setting in [4, 33], categories in ZSFooD and UECFOOD-256 are split into 184 seen classes and 44 unseen classes, and 205 seen classes and 51 unseen classes, respectively. We also compare our method with ZSD baselines on PASCAL VOC 2007+2012 [8] and MS COCO 2014 [22] using the given splitting [12, 15]. Note that two different splits are adopted for MS COCO: 48/17 seen/unseen split and 65/15 seen/unseen split. We replace the ingredient corpus with the corpus of texture and color words and replace the cuisine corpus with the corpus of shape and edge words when implementing SeeDS for general ZSD.

Evaluation Metrics. Similar to previous works [4, 15], we use mean Average Precision (mAP) and Recall@100 with IoU threshold 0.5 for the evaluation on ZSFooD, UECFOOD-256 and PASCAL VOC. For MS COCO, we report mAP and Recall@100 with IoU thresholds of 0.4, 0.5, and 0.6. We also report the performance of methods under the setting of GZSD. The Harmonic Mean (HM) of seen and unseen is the key metric used for GZSD performance.

Implementation Details. We adopt the Faster-RCNN [37] with the ResNet-101 [13] as the backbone for fair comparisons. For training the synthesizer, Adam [18] is used with a learning rate of 1e-4 with a weight decay of 1e-5 for all experiments. To align the experimental settings with baselines [12, 15], we synthesize 500/500/500/250 features for each unseen class of ZSFooD/UECFOOD-256/PASCAL VOC/MS COCO to train the classifier. We set $T = 100$ for the noise sampling process of RFDDM. The linear start and the linear end for noise scalars are set to β_1 = 8.5e-4 and β_T = 1.2e-2, respectively. We empirically set $\lambda_1 = 1$, $\lambda_2 = 1$ and $\lambda_3 = 0.1$ without meticulous tuning for ensuring stable training. Word embedding vectors of class names and corpora are extracted by CLIP [32] for ZSFooD and UECFOOD-256, and extracted by FastText [26] for PASCAL VOC and MS COCO.

4.2 Experiments on ZSFD datasets

Evaluation on ZSFooD. We reimplement baseline methods and show ZSFD results on ZSFooD in Table 2. Compared with the second-best method RRFS, "ZSD", "Unseen" and "HM" are improved by 1.6%, 0.8%, and 1.5% mAP, respectively. For Recall@100, "ZSD", "Unseen" and "HM" are improved by 4.1%, 2.2% and 1.9% mAP, respectively. The improvements demonstrate the effectiveness of the

Table 6: ZSD performance comparison on MS COCO (%).

Model	Split	Recall@100			mAP
		IoU=0.4	IoU=0.5	IoU=0.6	IoU=0.5
SB [4]	48/17	34.5	22.1	11.3	0.3
DSES [4]	48/17	40.2	27.2	13.6	0.5
PL [33]	48/17	-	43.5	-	10.1
BLC [52]	48/17	51.3	48.8	45.0	10.6
RRFS [15]	48/17	58.1	53.5	47.9	13.4
SeeDS	48/17	**59.2**	**55.3**	**48.5**	**14.0**
PL [33]	65/15	-	37.7	-	12.4
BLC [52]	65/15	57.2	54.7	51.2	14.7
SU [12]	65/15	54.4	54.0	47.0	19.0
RRFS [15]	65/15	65.3	62.3	55.9	19.8
SeeDS	65/15	**66.5**	**64.0**	**56.8**	**20.6**

Table 7: GZSD performance comparison on MS COCO (%).

Model	Split	Recall@100			mAP		
		S	U	HM	S	U	HM
PL [33]	48/17	38.2	26.3	31.2	35.9	4.1	7.4
BLC [52]	48/17	57.6	46.4	51.4	42.1	4.5	8.1
RRFS [15]	48/17	59.7	58.8	59.2	42.3	13.4	20.4
SeeDS	48/17	**60.1**	**60.8**	**60.5**	**42.5**	**14.5**	**21.6**
PL [33]	65/15	36.4	37.2	36.8	34.1	12.4	18.2
BLC [52]	65/15	56.4	51.7	53.9	36.0	13.1	19.2
SU [12]	65/15	57.7	53.9	55.7	36.9	19.0	25.1
RRFS [15]	65/15	58.6	61.8	60.2	37.4	19.8	26.0
SeeDS	65/15	**59.3**	**62.8**	**61.0**	**37.5**	**20.9**	**26.8**

proposed SeeDS in detecting unseen fine-grained food objects. Our proposed SeeDS enhances feature synthesizer by utilizing the multi-source semantic knowledge from S^3M, and help detectors achieve better ZSFD performance when trained with diverse features generated by RFDDM. We also observe that the mAP performance of "Unseen" for all ZSD methods is much lower than mAP of "Seen", which denotes that the larger number of classes in ZSFooD makes ZSFD on unseen objects extremely challenging. Note that the "Seen" performance in the setting of GZSD has not been improved since classifier parameters for seen classes are mainly influenced by the backbone detector trained on seen objects, while ZSFD specifically focuses on improving detection on unseen classes.

Evaluation on UECFOOD-256. Experimental results on UECFOOD-256 are shown in Table 3. Compared with RRFS, our SeeDS improves mAP by 3.5%, 3.1%, and 1.3%, and Recall@100 by 9.2%, 6.3%, and 3.1% for "ZSD", "Unseen" and "HM", respectively. For Recall@100, "ZSD", "Unseen" and "HM" are improved by 9.2%, 6.3%, and 3.1%, respectively. These results underline the effectiveness of our SeeDS framework, especially in complex GZSD scenarios with simultaneous seen and unseen food objects. We also observe that the mAP for all baseline methods can not reach a similarly high number as in ZSD, indicating ZSFD as a challenging task with potential for further method development.

4.3 Experiments on general ZSD datasets

Evaluations on PASCAL VOC and MS COCO. To further evaluate the performance of SeeDS in general ZSD, we conduct extension

Table 8: Ablation studies measured by mAP (%).

Dataset	Methods		ZSD	GZSD		
	S^3M	RFDDM		S	U	HM
ZSFooD			4.3	82.7	2.7	5.2
	✓		5.0	82.8	3.3	6.3
		✓	5.7	82.8	3.1	6.0
	✓	✓	**5.9**	**82.8**	**3.5**	**6.7**
PASCAL VOC			65.5	47.1	49.1	48.1
	✓		68.0	48.5	49.8	49.1
		✓	68.2	48.4	49.6	49.0
	✓	✓	**68.9**	**48.5**	**50.6**	**49.5**

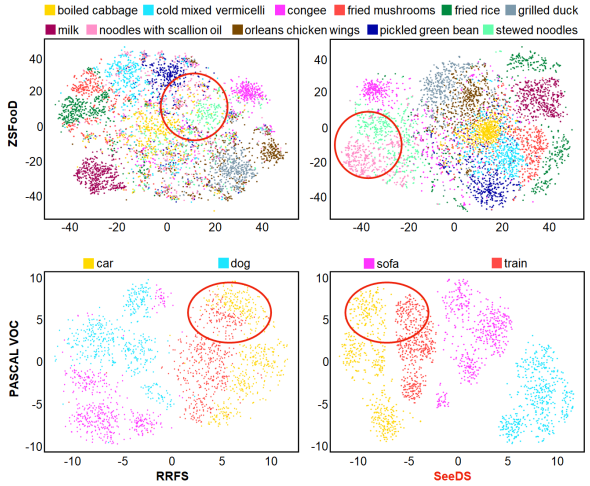

Figure 5: The t-SNE visualization of synthesized features.

experiments on PASCAL VOC and MS COCO. Our SeeDS outperforms all baselines under "ZSD" setting, increasing the mAP by 3.4% compared with the latest ZSD baseline RRFS [15]. Furthermore, our method obtains better performance under a more challenging setting of GZSD. The "Seen", "Unseen" and "HM" are improved by 1.4%, 1.5% and 1.4% compared with the RRFS. Results show that our method achieves a more balanced performance on the seen and unseen classes for GZSD. The class-wise AP performance on PASCAL VOC is reported in Table 5. We can observe that our approach achieves the best performance in most classes.

We evaluate the ZSD performance on MS COCO with different IoU thresholds of 0.4, 0.5 and 0.6. As seen in Table 6, our method outperforms all baseline methods, achieving significant gain on both mAP and Recall@100. For the 47/17 split, our method improves the mAP and Recall@100 by 0.6% and by 1.8% at IoU=0.5 compared with RRFS, respectively. For the 65/15 split, our SeeDS improves the mAP and Recall@100 by 0.8% and by 1.7% at IoU=0.5, respectively. As shown in Table 7, our SeeDS also outperforms the RRFS under the GZSD setting, where "S" denotes performance on seen classes and "U" denotes performance on unseen classes. The absolute "HM" performance gain of our method is 1.2% mAP and 1.3% Recall@100 for the 48/17 split, and 0.8% mAP and 0.8% Recall@100 for the 65/15 split. Results demonstrate that our model exceeds existing ZSD methods in terms of both mAP and Recall@100.

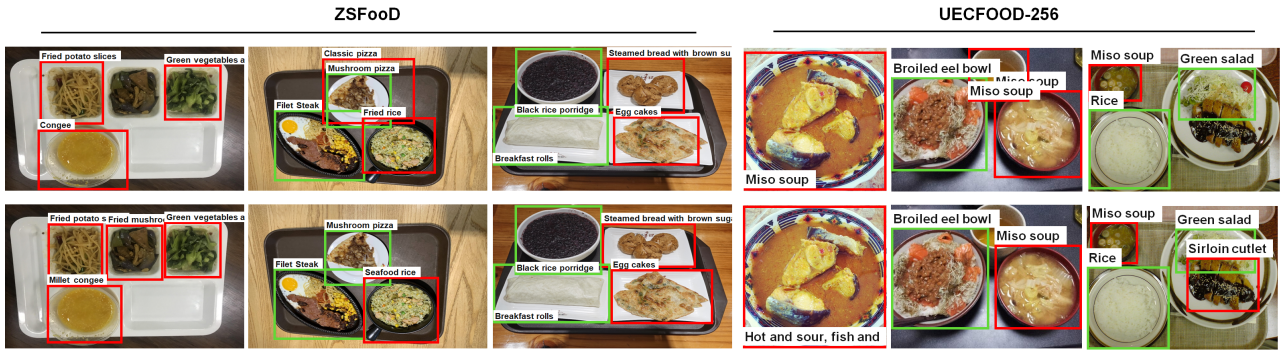

Figure 6: Detection results by baseline RRFS and our approach on ZSFooD, UECFOOD-256, PASCAL VOC and MS COCO. Seen classes are shown in green boxes and unseen in red boxes. Zoom in for a better experience.

4.4 Ablation Study

We conduct quantitative ablation analysis for two key modules, including the S^3M and RFDDM modules. Table 8 reports the "ZSD" and "GZSD" performance of mAP at IoU=0.5 for ZSFooD and PASCAL VOC. We replace the ingredient corpus with the texture corpus and replace the cuisine corpus with the shape corpus when implementing S^3M on PASCAL VOC. As shown in Table 8, the SeeDS incorporating the strategy of S^3M improves the "HM" by 1.1% on ZSFooD and 1.0% on PASCAL VOC. This indicates that the synthesizer with richer semantic information can synthesize more robust visual features for ZSFD and ZSD. We can observe that the "ZSD" performance has been improved by 1.4% on ZSFooD and 1.3% on PASCAL VOC compared with the baseline when the core GAN-based synthesizer is replaced with the RFDDM. These performance gains demonstrate the effectiveness of implementing the diffusion model in the ZSD framework. With respect to the "ZSD", "U" and "HM", the 1.6%, 0.8% and 1.5% mAP improvement on ZSFooD and the 3.5%, 1.5% and 1.4% mAP improvement on PASCAL VOC are obtained by the SeeDS that adopts both S^3M and RFDDM modules compared with the baseline. It shows that all proposed modules are vital for providing more robust synthesized features for training an effective Zero-shot Detector, which is able to improve the ZSFD and ZSD performance by a large margin.

4.5 Qualitative Results

Feature distribution visualization. To further demonstrate the effectiveness of our model in optimizing the distribution structure in generating, we utilize t-SNE [46] to visualize generated unseen features on ZSFooD and PASCAL VOC. Region features generated by the baseline RRFS and our approach are illustrated in Figure 5, where we select a quarter of the categories in ZSFooD to make visualization clear. Generated features for similar classes (e.g., *Stewed Noodles* and *Noodles with Scallion Oil* for ZSFooD, and *Car* and *Train* for PASCAL VOC) are confused with each other by the baseline method because of high similarity in their semantic representation. In this case, we observe that our synthesized features are more discriminative, which form well-separated clusters. Furthermore, discriminative synthesized features can help learn a more robust unseen classifier for ZSFD and ZSD.

Detection Results. We visualize the results of ZSFD on ZSFooD and UECFOOD-256 in Figure 6, where the first row is the output by RRFS and the second row is by our SeeDS. The baseline RRFS fails to predict true class labels for several unseen food objects, while our model provides more accurate ZSFD results. The proposed separable semantic learning in SeeDS effectively leverages domain knowledge of ingredients and cuisines to train an unseen classifier based on inter-class separable synthesized features. Furthermore, the incorporation of RFDDM further improves the performance of the synthesizer in generating fine-grained features that are robust for ZSFD. It is worth noting that ZSD baselines are often affected by similar visual features among fine-grained categories, even when their ingredients differ. For instance, the *Hot and Sour, Fish and Vegetable Ragout* is mistakenly recognized as the *Miso Soup* by RRFS. In contrast, SeeDS is able to discriminate between the *Classic Pizza* and the *Mushroom Pizza* via the difference in ingredients.

5 CONCLUSION

In this paper, we first define Zero-Shot Food Detection (ZSFD) task for tackling real-world problems. Furthermore, We propose a novel ZSFD framework SeeDS to address the challenges posed by complex attributes and diverse features of food. We evaluate our method and baselines on two food benchmark datasets ZSFooD and UECFOOD-256, which demonstrates the effectiveness and robustness of SeeDS on ZSFD. To further explore the performance of our method on general ZSD, we evaluate SeeDS on two widely-used datasets, PASCAL VOC and MS COCO. The results show that SeeDS can generalize well on ZSD. The ablation studies on ZSFooD and UECFOOD-256 demonstrate the effectiveness of the proposed modules, including S^3M and RFDDM. Therefore, our approach shows great potential for ZSFD and can be extended to various multimedia applications. In future research, we need to better understand the feature sampling mechanism and its potential for solving fine-grained problems. Also, with the development and new opportunities, we believe open-vocabulary food detection with a food-specialized visual-language pretrained model could better serve real-world needs and deserves further exploration.

ACKNOWLEDGMENTS

This work was supported by the National Nature Science Foundation of China (61972378, U19B2040, 62125207, U1936203), and was also sponsored by CAAI-Huawei MindSpore Open Fund.

REFERENCES

[1] Eduardo Aguilar, Beatriz Remeseiro, Marc Bolaños, and Petia Radeva. 2018. Grab, pay, and eat: Semantic food detection for smart restaurants. *IEEE Transactions on Multimedia (TMM)* 20, 12 (2018), 3266–3275.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the International Conference on Machine Learning (ICML)*. 214–223.

[3] Sinem Aslan, Gianluigi Ciocca, Davide Mazzini, and Raimondo Schettini. 2020. Benchmarking algorithms for food localization and semantic segmentation. *International Journal of Machine Learning and Cybernetics (IJMLC)* 11, 12 (2020), 2827–2847.

[4] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. 2018. Zero-shot object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 384–400.

[5] Demirel Berkan, Ramazan Gökberk Cinbiş, and Nazlı İkizler Cinbiş. 2018. Zero-Shot Object Detection by Hybrid Region Embedding. In *Proceedings of the British Machine Vision Conference (BMVC)*. 56–68.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

[7] Takumi Ege and Keiji Yanai. 2017. Simultaneous estimation of food categories and calories with multi-task CNN. In *Proceedings of the IAPR International Conference on Machine Vision Applications (MVA)*. 198–201.

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)* 88, 2 (2010), 303–338.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM (CACM)* 63, 11 (2020), 139–144.

[10] Omkar Gune, Biplab Banerjee, Subhasis Chaudhuri, and Fabio Cuzzolin. 2020. Generalized zero-shot learning using generated proxy unseen samples and entropy separation. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*. 4262–4270.

[11] Zongyan Han, Zhenyong Fu, Shuo Chen, and Jian Yang. 2021. Contrastive embedding for generalized zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2371–2381.

[12] Nasir Hayat, Munawar Hayat, Shafin Rahman, Salman Khan, Syed Waqas Zamir, and Fahad Shahbaz Khan. 2020. Synthesizing the unseen for zero-shot object detection. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)* 33, 6840–6851.

[15] Peiliang Huang, Junwei Han, De Cheng, and Dingwen Zhang. 2022. Robust Region Feature Synthesizer for Zero-Shot Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7622–7631.

[16] Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the International Conference on Computer Vision (ICCV)*. 1501–1510.

[17] Yoshiyuki Kawano and Keiji Yanai. 2015. Automatic expansion of a food image dataset leveraging existing categories with domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 3–17.

[18] D. Kingma and J. Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*. 1–15.

[19] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[20] Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, et al. 2015. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 4247–4255.

[21] Zhihui Li, Lina Yao, Xiaoqin Zhang, Xianzhi Wang, Salil Kanhere, and Huaxiang Zhang. 2019. Zero-shot object detection with textual descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 8690–8697.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 740–755.

[23] Ya Lu, Thomai Stathopoulou, Maria F Vasiloglou, Stergios Christodoulidis, Zeno Stanga, and Stavroula Mougiakakou. 2020. An artificial intelligence-based system to assess nutrient intake for hospitalised patients. *IEEE Transactions on Multimedia (TMM)* 23 (2020), 1136–1147.

[24] Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2019. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 43, 1 (2019), 187–203.

[25] Austin Meyers, Nick Johnston, Vivek Rathod, Anoop Korattikara, Alex Gorban, Nathan Silberman, Sergio Guadarrama, George Papandreou, Jonathan Huang, and Kevin P Murphy. 2015. Im2Calories: Towards an automated mobile vision food diary. In *Proceedings of the International Conference on Computer Vision (ICCV)*. 1233–1241.

[26] Tomáš Mikolov, Édouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. 52–55.

[27] Weiqing Min, Shuqiang Jiang, and Ramesh Jain. 2019. Food recommendation: Framework, existing solutions, and challenges. *IEEE Transactions on Multimedia (TMM)* 22, 10 (2019), 2659–2671.

[28] Weiqing Min, Shuqiang Jiang, Linhu Liu, Yong Rui, and Ramesh Jain. 2019. A survey on food computing. *ACM Computing Surveys (CSUR)* 52, 5 (2019), 1–36.

[29] Weiqing Min, Zhiling Wang, Yuxin Liu, Mengjiang Luo, Liping Kang, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. 2023. Large scale visual food recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 01 (2023), 1–18.

[30] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. 2014. Zero-shot learning by convex combination of semantic embeddings. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[31] Laavanya Rachakonda, Saraju P Mohanty, and Elias Kougianos. 2020. iLog: An intelligent device for automatic food intake monitoring and stress detection in the IoMT. *IEEE Transactions on Consumer Electronics (TCE)* 66, 2 (2020), 115–124.

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*. 8748–8763.

[33] Shafin Rahman, Salman Khan, and Nick Barnes. 2022. Polarity Loss: Improving Visual-Semantic Alignment for Zero-Shot Detection. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)* (2022), 1–13.

[34] Shafin Rahman, Salman Khan, and Fatih Porikli. 2019. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*. 547–563.

[35] Assyifa Ramdani, Agus Virgono, and Casi Setianingsih. 2020. Food Detection with Image Processing Using Convolutional Neural Network (CNN) Method. In *Proceedings of the IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*. 91–96.

[36] Joseph Redmon and Ali Farhadi. 2017. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7263–7271.

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 28.

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 10684–10695.

[39] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. 2019. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 8247–8255.

[40] Wataru Shimoda and Keiji Yanai. 2019. Webly-Supervised Food Detection with Foodness Proposal. *IEICE Transactions on Information and Systems* 102, 7 (2019), 1230–1239.

[41] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 26.

[42] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[43] Jianing Sun, Katarzyna Radecka, and Zeljko Zilic. 2019. Foodtracker: A real-time food detection mobile application by deep convolutional neural networks. *arXiv preprint arXiv:1909.05994* (2019).

[44] Chufeng Tan, Xing Xu, and Fumin Shen. 2021. A survey of zero shot detection: methods and applications. *Cognitive Robotics* 1 (2021), 159–167.

[45] Avinash Ummadisingu, Kuniyuki Takahashi, and Naoki Fukaya. 2022. Cluttered Food Grasping with Adaptive Fingers and Synthetic-Data Trained Object Detection. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*. 8290–8297.

[46] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research (JMLR)* 9, 86 (2008), 2579–2605.

[47] Hao Wang, Guosheng Lin, Steven CH Hoi, and Chunyan Miao. 2022. Learning structural representations for recipe generation and food retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2022), 1–15.

[48] Wei Wang, Weiqing Min, Tianhao Li, Xiaoxiao Dong, Haisheng Li, and Shuqiang Jiang. 2022. A review on vision-based analysis for automatic dietary assessment. *Trends in Food Science & Technology* 122 (2022), 223–237.

[49] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. 2022. Tackling the Generative Learning Trilemma with Denoising Diffusion GANs. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[50] Caixia Yan, Xiaojun Chang, Minnan Luo, Huan Liu, Xiaoqin Zhang, and Qinghua Zheng. 2022. Semantics-guided contrastive network for zero-shot object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2022), 1–1.

[51] Shizhen Zhao, Changxin Gao, Yuanjie Shao, Lerenhan Li, Changqian Yu, Zhong Ji, and Nong Sang. 2020. GTNet: Generative transfer network for zero-shot object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 12967–12974.

[52] Ye Zheng, Ruoran Huang, Chuanqi Han, Xi Huang, and Li Cui. 2020. Background learnable cascade for zero-shot object detection. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*.

[53] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. 2020. Don't Even Look Once: Synthesizing Features for Zero-Shot Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11693–11702.