Zero-shot Semantic Segmentation Using Relation Network

Yindong Zhang
Visma Consulting Oy
Jyväskylä, Finland
zhangyd96@gmail.com

Oleksiy Khriyenko
University of Jyväskylä
Jyväskylä, Finland
oleksiy.khriyenko@jyu.fi

***Abstract*— Zero-shot learning (ZSL) is widely studied in recent years to solve the problem of lacking annotations. Currently, most studies on ZSL are for image classification and object detection. But, zero-shot semantic segmentation, pixel level classification, is still at its early stage. Therefore, this work proposes to extend a zero-shot image classification model, Relation Network (RN), to semantic segmentation tasks. We modified the structure of RN based on other state-of-the-arts semantic segmentation models (i.e. U-Net and DeepLab) and utilizes word embeddings from Caltech-UCSD Birds 200-2011 attributes and natural language processing models (i.e. word2vec and fastText). Because meta-learning is limited to binary tasks, this work proposes to join multiple binary semantic segmentation pipelines for multi-class semantic segmentation. It is proved by experiments that RN could improve accuracy of U-Net with the help of semantic side information on binary semantic segmentation and it could also be applied on multi-class semantic segmentation with simpler structure than the baseline model, SPNet, but higher accuracy under ZSL setting. However, the capability of RN under generalized zero-shot learning (GZSL) setting still needs improvement. We also studied on how different word embeddings, network structures and data affect RN and what could be done to improve its results.**

I. INTRODUCTION

This paper studies on semantic segmentation tasks, which are pixel-level classification tasks. Similar to image classification, semantic segmentation can also be categorized into binary tasks (i.e. classify pixels into one class or background) and multi-class tasks (i.e. classify pixels into multiple classes and background). To extract visual feature maps, U-Net [1] and DeepLab-v3 [2] framework were adopted. The former one has excellent performance in binary semantic segmentation tasks (e.g. medical images segmentation) and the latter one is more popular in multi-class segmentation tasks.

The obstacle of labelling is also studied in this paper, especially zero-shot learning (ZSL), because it is impossible to label everything. Currently, there are increasing studies on few-shot learning (FSL) and ZSL, which means predicting on classes that have been seen only a few times or never during training. FSL tasks are denoted as k-shot c-way problems, where k means the number of sample images from each class and c means the number of classes [3]. ZSL tasks are denoted as 0-shot c-way problems, because ZSL does not have sample images, which is more common in practice, such as recognizing objects on the street and recognizing handwriting text in an unseen language.

In another word, the goal of ZSL is to transfer the knowledge learnt from seen classes to unseen classes. To achieve this goal, we import side information, which refers to attributes and natural language processing (NLP) models. Attributes are visual characteristics of a class or an object. For example, in 200-2011 (CUB) [4] class attributes have continuous value, which is "the percentage of the time (between 0 and 100) when a human thinks that the attribute is present for a given class [4]". CUB image attributes are Boolean values where 1 means "is present" and 0 means "is not present".

Study of zero-shot semantic segmentation is still at its early stage. So, new tasks of zero-label semantic segmentation (ZLSS) and general zero-label semantic segmentation (GZLSS) are adopted. They are proposed in [5] with a baseline model called Semantic Projection Network (SPNet). In ZLSS it is assumed that there are only unseen classes in testing data. For example, a segmentation model is trained on a huge dataset of forest animals, but it will be applied on sea animals, which does not have any annotations. GZLSS is more practical where both seen and unseen classes could appear during testing, e,g, the forest animal trained model will be applied on jungle animals.

As shown in Table I, this work categorized all tasks into binary ZLSS, multi-class ZLSS and multi-class GZLSS. Same to ordinary binary segmentation tasks, in binary ZLSS, the model only needs to distinguish background and the target class in one image, and its training data and testing data has different target classes. As binary tasks are limited to 2-way, they do not apply to GZLSS. As for multi-class tasks, the model segments multiple classes and background in one image. In multi-class ZLSS, training data and testing data has disjointed label space, whereas multi-class GZLSS have jointed label space.

Studying on ZLSS and GZLSS, authors of [5] construct SPNet by removing the last classifier layer of DeepLab and appending a projection layer to it. Inspired by this, this paper proposes to extend a meta-learning based few-shot image classification model called Relation Network (RN) [3] to a zero-shot semantic segmentation model. Meta-learning aims to

TABLE I. CHARACTERISTICS OF TASKS

	K-shot	C-way	Testing classes
Binary ZLSS	K = 0	C = 2	C_u
Multi-class ZLSS	K = 0	C >= 2	C_u
Multi-class GZLSS	K = 0	C >= 2	$C_u \cup C_s$

help models learn to adapt to new classes by feeding models random samples and random label space in every episode. In meta-learning, the concept of epoch is substituted by the same amount of iterations - "episodes". A successful method of meta-learning is to learn the best initial weights from training data, and then fine tune models based on testing data. However, it is also possible to train a model, which does not require fine tuning [3].

In the following research, we aim to explore how RN could be extended, whether it has advantages over baseline models and how to improve its capability. Following Section II first introduces recent state-of-the-art semantic segmentation models, ZSL models and word embedding techniques, as well as illustrating RN and current ZLSS models. In Section III, highlighting research questions and methods, we show how to extend RN and how to train extended models. Section IV presents implementation details and experiments results, as well as comparison of quantitative results of RN extended models with baseline models and their qualitative results analysis. Finally, we conclude the work and highlight future direction.

II. COMPUTER VISION AND ZERO-SHOT LEARNING

A. Semantic segmentation

Semantic image segmentation refers to pixel classification, where each pixel is labelled with an object category. Recent state-of-the-art semantic segmentation models all benefit from convolutional neural networks (CNNs) [6],[1],[7]. With CNNs, a basic image classification model consists of multiple CNN layers for extracting dense features and ends with a fully connected (FC) layer for classification. As for pixel level classification, CNN can be applied to extract features and classify each pixel. But, there is an accuracy and computation dilemma: complete predictions require CNN layers to keep the original size to contain each pixel, however faster computation needs smaller layer size. A common solution is encoder-decoder structure. Fully Convolutional Networks (FCN) is the first one to utilize this structure for semantic segmentation tasks [6]. FCN replaces last fully connected layers of pre-trained image classification models with convolutional layers in order to generate heat maps and then uses deconvolution (also called upsampling) to classify each pixel. They also propose skip connections in upsampling layers to improve FCN's accuracy.

Based on FCN, U-Net is designed for medical image segmentation with more flexible network structure [1]. U-Net has a U shape, where the left part is the encoder and the right part is the decoder. Medical images usually are in large size, lack of labelling and only have one target class. Therefore, U-Net is popular in few-shot binary semantic segmentation tasks. Furthermore, because U-Net does not contain any pre-trained model, it can be easily adjusted according to image size.

Furthermore, DeepLab is one of the most important semantic segmentation models because it adopts ResNet as its backbone and proposes atrous/dilated convolution and pyramid pooling [7]. ResNet allows models to be deep without reducing their accuracy. As for atrous convolution, because neighbour pixels usually are very similar to each other, the kernel expands its cover space with ignoring some input. By doing this, DeepLab is able to expand the view scope of kernels. From another point of view, it could reduce the number of parameters and save memory usage. However, ignoring some input causes losing contextual information. So, pyramid pooling was introduced to solve this problem.

B. Zero-shot learning

Collecting and annotating data is expensive and time-consuming which motivates researchers to study on few-shot learning (FSL) and zero-shot learning (ZSL). In practice, ZSL, where models have to recognize classes that they have never seen during training based on seen classes, is more common than FSL [3].

Based on attributes, researchers try to directly map from visual space to semantic space without intermediate tasks. For example, authors in [8] propose an approach called Attribute Label Embedding (ALE) to use a bilinear function to measure the compatibility between image embeddings and label embeddings. ALE uses attributes as side information for the label embedding and measures the "compatibility" between the embedded inputs and outputs with a function F. During training, image embeddings are the input, label embeddings are the output, the goal is to optimize w to maximize the image-label pairs' compatibility. In this way, ALE is able to directly predict on labels without intermediate tasks.

Then, Structured Joint Embedding (SJE) framework [9] was developed based on ALE by replacing manually annotated attributes with multiple side information (e.g. attributes, hierarchical models, and text-based models). SJE includes multiple bilinear functions F for multiple semantic information sources. In another word, it uses multiple Wi to capture different visual features. For more complex tasks, authors in [10] suggest Latent Embedding Model (LatEm) where a single bilinear function is extended to multiple linear functions to capture diverse visual features. To directly project visual space into semantic space, it is also possible to map both into a third space and optimize their embeddings together. Zhang and Saligrama [11] propose a method called semantic similarity embedding (SSE) where both visual features and semantic features are projected into histograms separately.

Above models all belong to metric-based approaches: learn fixed metrics (e.g. ALE) to embed images (and labels) and then classify images by fixed classifiers (e.g. k-nearest neighbour). Therefore, research on metric-based approaches usually focus on how to embed input data or how to recognize embedded features. Additionally, generative ZSL is also a popular direction, because it is closer to how humans think [12]. When a human reads a description, he will have a fuzzy

image in mind and recognize the target based on this fuzzy image. Similarly, generative models generate synthetic features from texts and classify query image features based on synthetic features. However, complex models, including generative ones, may be too memory and battery consuming, and slow in production.

C. Word embeddings

As mentioned above, attribute is one of the popular semantic side information ZSL [4][13]. Attributes are visual characteristics annotated manually, such as "hooked seabird shape" has 59.85% probability to be recognized in class "black footed albatross". However, because of the nature of attribute datasets, extending them always requires human effort.

Compared with attributes, NLP models are easier to apply to generalized tasks because they use neural networks to learn word vectors (i.e. word representing) from free text in unsupervised learning methods. For example, when Word2vec [14] learns a new word, it actually learns from its neighbor words. The other way around, word2vec could predict a word by its neighbor words. In this way, the distance between two words are not affected by their letters but their neighbor words. For example, NLP models are able to learn the implicit relationships between countries and their capital cities because they are in similar contexts [14].

Furthermore, fastText [15] utilizes characters n-gram to extract word vectors within the target word itself. Different from word2vec, fastText represents a word by a set of characters instead of words. For example, the word where with $n = 3$ can be represented by character n-grams: "wh", "whe", "her", "ere", "re" and itself "where" [15]. With character n-gram, fastText is not limited to the training corpus, because it could represent unseen words.

D. Relation Network

In [3] authors propose to utilize CNN to learn embeddings for query images and semantic side information and a rather flexible classifier. They stress that because meta-learning keeps feeding RN random label space during training, it could focus on learning a classifier (i.e. how to compare images and semantic information), instead of learning features of training data. Fig. 1 displays that RN consists of an image embedding module and a relation module. In FSL, the image embedding module extracts feature maps from both query images and support images. Then the relation module computes relation scores of each class based on the concatenations of query features and support features. In the end, the class with the highest relation score is the prediction class for query images. It is claimed that RN could learn to adapt to new classes through meta-learning, where the model is fed by different support data in each episode. In another word, each episode selects random label space to help RN learn metrics and a classifier that can be shared by all classes. They also stress a well-established meta-learning approach is to learn a set of proper initial parameters, and then fine-tune them with new data. But, RN does not require fine-tuning making it easier to be applied on general tasks.

Furthermore, they state and prove RN could be easily extended to ZSL tasks by adding a word embedding module. Semantic feature maps are concatenated with visual feature maps as input of the relation module (see Fig. 2). Their experiments results demonstrate that the RN performs better than most classical zero-shot image classification models (e.g. SJE and SSE).

Inspired by RN, in [16] authors build a baseline model for their few-shot segmentation dataset (i.e. FSS-1000) based on U-Net and RN. As Fig. 3 displays, their model consists of encoder module, relation module and decoder module. Compared with the original U-Net, their baseline model does not only embed query images and concatenate query feature maps, but also support images and support feature maps. Their proposed dataset (i.e. FSS-1000) only has binary segmentation labels, so they only implement their baseline model on binary semantic segmentation. However, since U-Net itself is designed for binary semantic segmentation tasks, their experiments cannot prove whether the RN improves their baseline model's accuracy.

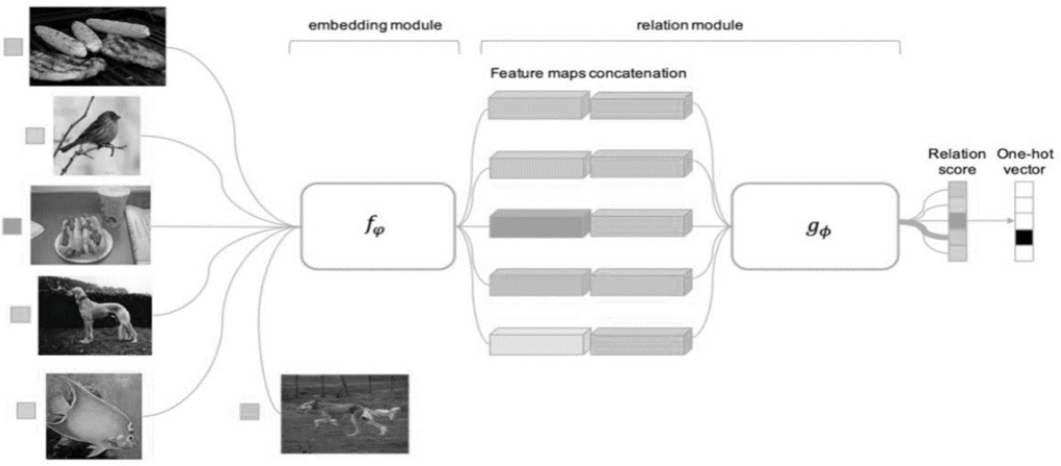

Fig. 1. RN pipeline for a 5-way 1-shot image classification problem with one query example

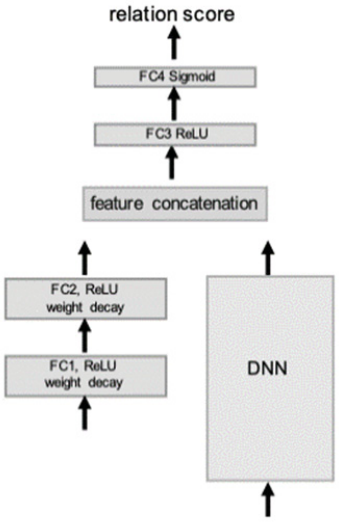

Fig. 2. RN architecture for zero-shot learning [3]

E. Current zero-shot semantic segmentation

Although zero-shot semantic segmentation is still at its early stage, researchers have proven that this task is possible to complete by extending current ZSL and semantic segmentation models. For example, in [5] authors utilize the structure of DeepLab to design a zero-shot semantic segmentation model called SPNet. As shown in Fig. 4 they remove the last classifier layer of a DNN (i.e. DeepLab or FCN) as their visual-semantic embedding, where a, b is width and height, dw is the size of word vectors. For example, the channel size of word2vec word vectors is 300, so dw should be 300 too. Afterwards, in semantic projection these visual-semantic features are projected into the fixed semantic space (i.e. fastText or word2vec) where word vectors are utilized as weights of the projection layer. Projection layer's output has the same depth as label space (i.e. $|S|$ or $|U|$). In the end SPNet outputs the probability of each class for each pixel. SPNet is similar to ALE as they both project visual space directly into semantic space. In addition to this, authors explain the reason why they do not use meta-learning is because it is limited to binary tasks.

Another related model, zero-shot semantic segmentation network (ZS3Net) [12], is a generative ZSL model which produces synthetic image features of unseen classes. However ZS3Net is designed for semi-supervised learning where some labels of test instances are available during training, which means train classes and test classes are not disjointed in ZS3Net. Therefore, it could only be applied to GZLSS tasks. Besides, what is worth noting is that in their experiments [12] authors adopt a baseline model that is similar to SPNet and ALE. In their baseline model, they replace the last classifier layer of DeepLab-v3 with a CNN layer, which produces word vectors. Then they calculate similarities between predicted vectors and each class's vectors to classify query images.

III. RN FOR ZERO-SHOT SEMANTIC SEGMENTATION

A. Problem definition and methodology

In ZSL, each episode's input data consists of a query set and a support set that share the same label space. In this study, the query set refers to query images $Q = \{(x_{i,j,k}, y_{i,j,k})\}_{i,j,k=1}^{w,h,n}$, where $x_{i,j,k}$ is the pixel at coordinate i, j of image k, $y_{i,j,k}$ is the label of that pixel, and w, h, n is the width, height, batch size of images. And let denote seen classes as C_s and unseen classes as C_u. The support set refers to semantic side information $S_{train} = \{v_c; c \in C_s\}$ where v_c is attributes or word vectors of class c. The goal of model is to predict scores $r_{i,j,k,c}$ of each pixel over each class, and minimize the distance f_d between prediction scores and ground truth $y_{i,j,k,c}$:

$$r_{i,j,k,c} \leftarrow argmin \sum_{i=1}^{w} \sum_{j=1}^{h} \sum_{k=1}^{n} \sum_{c=1}^{C} f_d(r_{i,j,k,c}, y_{i,j,k,c}) \quad (1)$$

Because of the meta-learning setting, in each episode a random combination of Q and S_{train} is selected. And classes, that are not included in S_{train}, are ignored in this episode [5]. In another word, pixels belonging to unselected classes should not affect loss calculation and accuracy calculation.

In multi-class training, each query image features are combined with each class's semantic information as a pair. For example, if there are k images and c classes in one episode, this episode has $k \times c$ pairs. Fig. 5 displays an example pipeline of one episode, where 1 query image and 5 classes are selected.

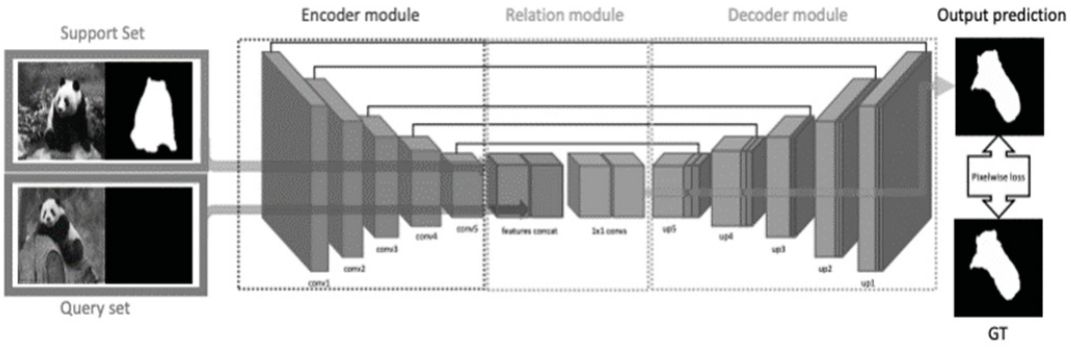

Fig. 3. Baseline network architecture of FSS-1000, VGG-16 as Backbone [16]

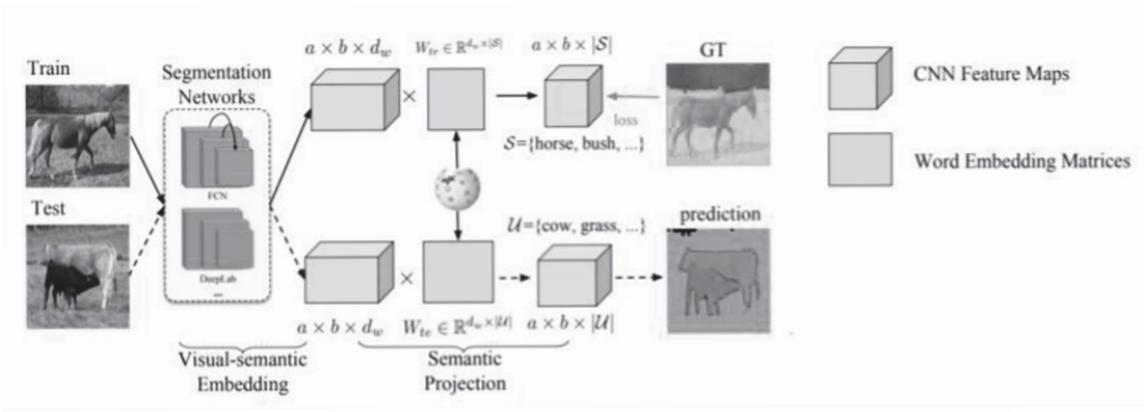

Fig. 4. An example of SPNet pipeline including visual semantic embedding and semantic projection under ZLSS setting [5]

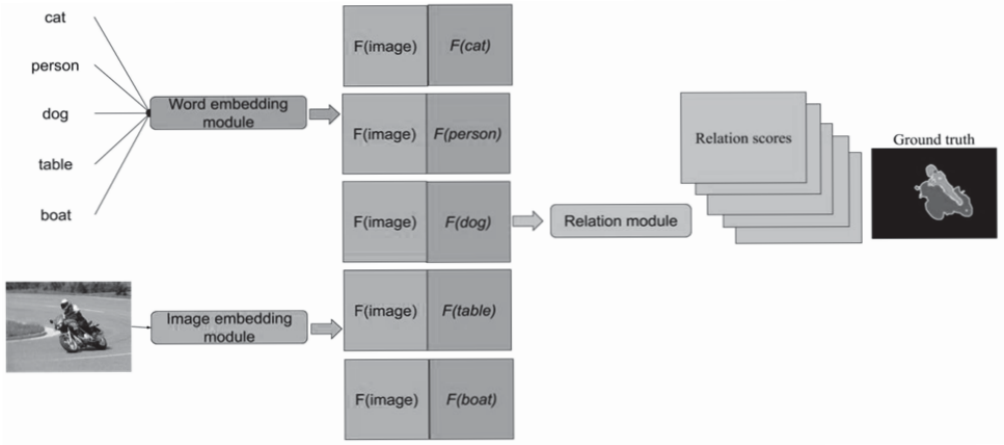

Fig. 5. An example pipeline of one episode in multi-class tasks

Firstly, semantic and visual feature maps are concatenated as the input of the relation module. Then the relation module computes relation scores for each class over each pixel. During multi-class testing, under ZLSS setting the support set $S_{train} = \{v_c; c \in C_u\}$, but under ZLSS setting $S_{train} = \{v_c; c \in C_s \cup C_u\}$. In the binary ZLSS task, training classes and testing classes are disjointed, and each image is only paired with its target class (i.e. k query images result in k pairs).

In this work, we have studied on the following questions: how to extend RN from zero-shot image classification to binary ZLSS, multi-class ZLSS, and multi-class GZLSS; does RN have advantages over baseline models; and what factors would affect the capability of RN and how could it be improved?

For binary zero-shot semantic segmentation tasks U-Net is set as the baseline model. Because in these tasks, similar classes could make them be considered as few-shot tasks as well. Moreover, U-Net is designed for few-shot binary semantic segmentation which concatenate downsampling feature maps with upsampling ones. Thus, we aim to investigate if semantic side information could improve the accuracy of U-Net.

SPNet (see Fig. 3) is set as the baseline model for multiclass semantic segmentation tasks. The crucial part of SPNet is projecting image features into semantic space, which is flexible in terms of label space and does not require fine-tuning. But, using DeepLab framework makes their model heavy to train and apply to applications, so in this work we studied if RN could outperform it with a simpler architecture.

All models are evaluated by the average intersection over union ($mIoU$) (2) over classes. I_i refers to the intersection area of prediction and ground truth of class i, and U_i refers to their union area.

$$mIoU = \frac{\sum_{i=0}^{n} \frac{I_i}{U_i}}{n} \quad (2)$$

Under GZLSS setting, the models are evaluated by $mIoU$ of unseen classes, $mIoU$ of seen classes, and harmonic mean (H) of them:

$$H = \frac{2 * mIoU_{seen} * mIoU_{unseen}}{mIoU_{seen} + mIoU_{unseen}} \quad (3)$$

B. Network architecture

RN consists of three modules: a word embedding module $f_{\phi 1}(v_c)$ for extracting semantic features from classes, an image embedding module $f_{\phi 2}(x_{i,j,k})$ for extracting visual features from query images, and a relation module $g_{\varphi}(f_1 \oplus f_2)$ for computing relation scores for each visual features and semantic features pair. Therefore, RN predicts relation scores

$r_{i,j,k,c}$ for each pixel over each class (see Equation 4). The goal of RN is to learn metrics (i.e. embeddings $f_{\phi 1}$ and $f_{\phi 2}$) and a classifier (i.e. relation module g_{φ}) for categorizing the concatenation of feature maps into a class. In semantic segmentation, besides metrics, it also learns a set of classifiers for categorizing concatenations on pixels into classes.

$$r_{i,j,k,c} = g_{\varphi}(f_{\phi 1}(v_c) \oplus f_{\phi 2}(x_{i,j,k})) \quad (4)$$

Relation module is the key point of how to extend RN from image classification to semantic segmentation. Inspired by U-Net and baseline model of FSS-1000 dataset [16], relation module uses the same structure as the decoder of U-Net. Based on this, Fig. 6 demonstrates RN architecture for the binary ZLSS task (also called 0-shot 2-way task): word embedding module is on the top which transforms semantic information into proper channels for concatenation, image embedding is on the left which is ImageNet pretrained VGG16 [11], and relation module is on the right which predicts scores for each pixel. As it shows, the relation module consists of upsampling and CNN layers, where feature maps from image embedding module and word embedding module are utilized to improve its accuracy.

Inspired by SPNet, another method is using a DNN as image embeddings and directly classify on the concatenation of semantic feature maps and visual feature maps by CNN layers. Fig. 7 displays RN architecture of using DeepLab-v3 (without its last 4 classifier layers) as image embeddings, ImageNet pretrained ResNet101 [17] as its backbone, FC layers as word embeddings and upsampling and CNN layers as relation module. Moreover, the word embedding module could be composed by FC layers or upsampling and CNN layers. Neural networks learn feature maps from input data by kernels. In fully connected (FC) layers, kernels have the same size as the input, but in convolutional neural networks (CNN) layers kernel size is smaller than input (e.g. 3 × 3). As Fig. 8 demonstrates, with

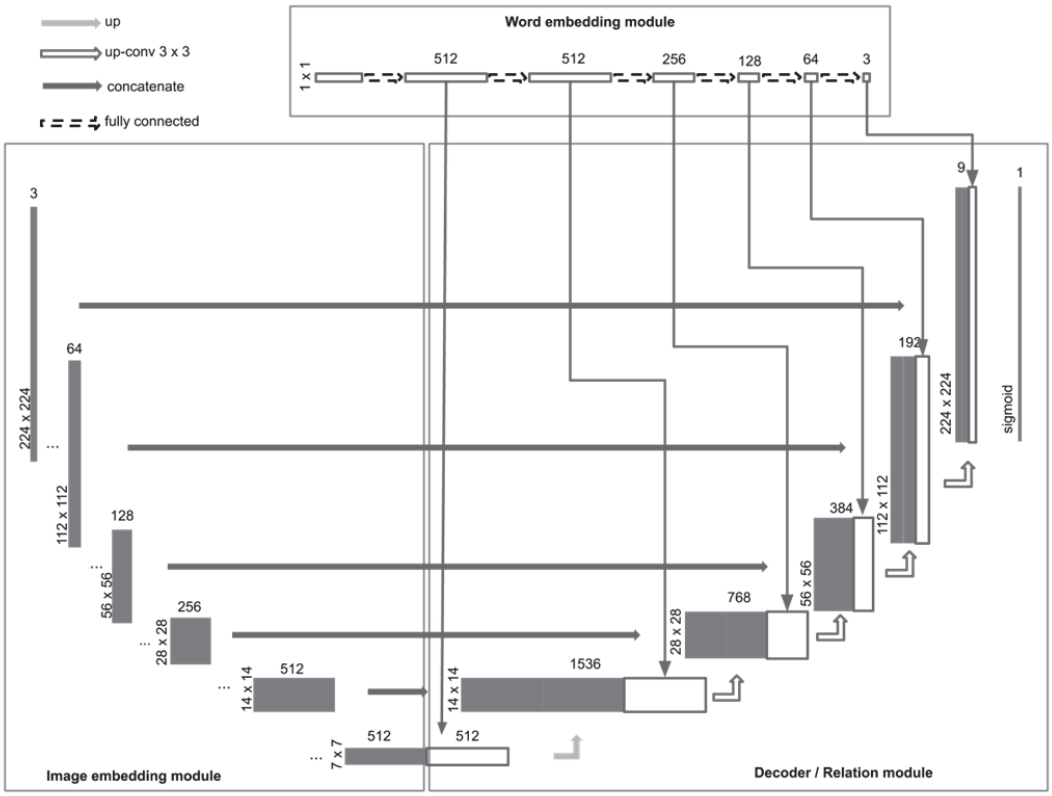

Fig. 6. Relation Network architecture for binary semantic segmentation tasks with pre-trained VGG16 and U-Net decoder

the same input (i.e. 5 × 5) FC kernel on the left has the same size of input but CNN kernel on the right is 3 × 3, and each kernel multiplies with all channels' feature maps and sums up its results. Therefore, when the input data size is 10 × 15 × 5 × 5, both FC and CNN layers have 5 kernels, step size is 1 and no padding, the output of FC layer would be 10 × 5 × 5 × 5, but the output of CNN layer would be 10 × 5 × 3 × 3. Compared with FC, CNN focuses on a part of the input data at a time, which is helpful in Computer Vision. For example, when we recognize a cat from an image, what helps us are those key features (e.g., ears and eyes), instead of the whole image. Therefore, using a smaller kernel helps models to learn more useful feature maps.

With FC layers its input is 2-dimensional data (i.e. batch size × channel size). FC layers transform the channel size of semantic feature maps (e.g. from 300 to 512), so semantic feature maps need to be repeated in width and height before concatenation.

With CNN layers, its input is 4-dimensional data (i.e. batch size × channel size × width × height) which can be upsampled and transformed to the proper size. Using FC layers is closer

to the original concept of RN where each pixel is concatenated with the same semantic feature maps and makes model size smaller in this case. But, the other one may provide more context information for semantic segmentation models. In the end, the relation module uses sigmoid at its last layer to restrict relation scores between 0 and 1 representing the probability of this pixel belonging to the target class. In the binary ZLSS task, if a score is higher than or equal to 0.5, this pixel is predicted as target class, otherwise it is predicted as background. In C-way 0-shot semantic segmentation tasks, as shown in Fig. 9, RN predicts relation scores for each word-image pair and compares relation scores among classes. Then for each pixel the class with the highest relation score is its prediction class. With this network architecture, a multi-class segmentation task is essentially a series of binary segmentation tasks.

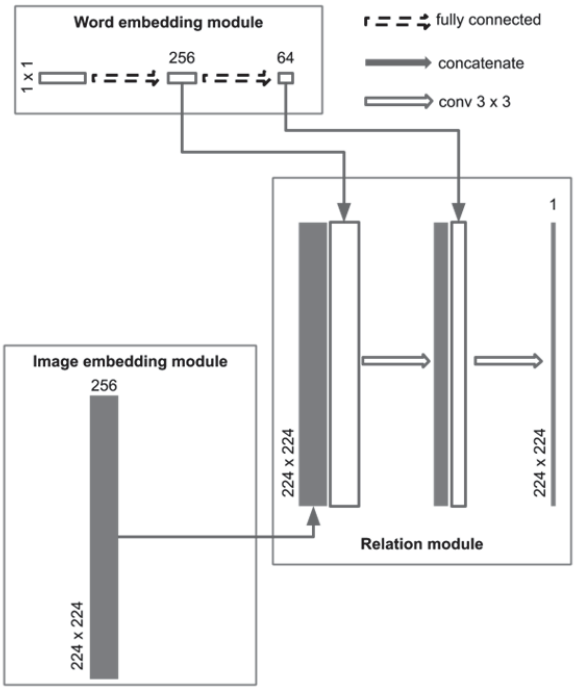

Fig. 7. Relation Network architecture for binary semantic segmentation tasks with DeepLab-v3 removing its last 4 classifier layers

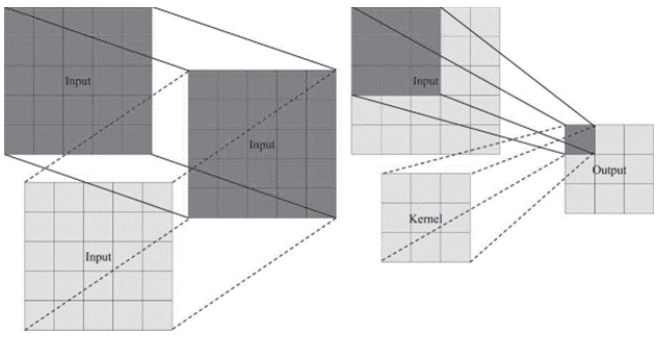

Fig. 8. Comparison of FC kernel (left) and CNN kernel (right)

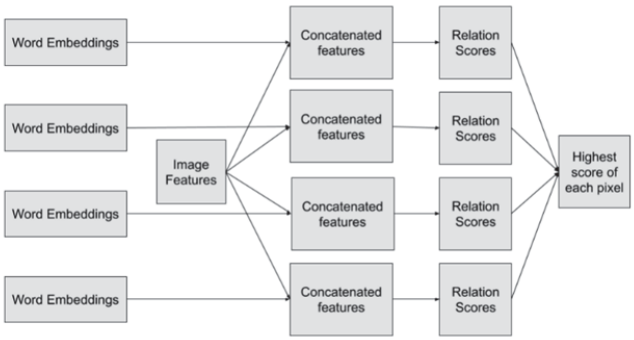

Fig. 9. C-way 0-shot semantic segmentation pipeline

C. Loss function

In both binary segmentation training and multiclass segmentation training, binary cross entropy (BCE) loss (5) is adopted for backpropagation, where $r_{i,j,k,c}$ is the relation score of a pixel over class c and $y_{i,j,k,c}$ is the ground truth of it. If this pixel belongs to class c, its ground truth is 1, otherwise it is 0. Mean square error (MSE) could also be used here but does not make much difference. As for categorical cross entropy, experiments show that convergence speed is not as fast as BCE.

IV. Experiments

A. Datasets and splits

In [3] authors used Animals with Attributes (AwA2) [13] and CUB for zero-shot image classification. They both provide attributes information which could be input of the word embedding module. However, AwA2 does not have segmentation labels. Whereas, besides 200 bird species and corresponding 312 numeric attributes values, CUB also provides rough segmentation labels for each image. And PASCAL VOC 2012 [18] and COCO-Stuff are also popular semantic segmentation datasets [5][12]. PASCAL contains 2913 images with segmentation annotations, covering 20 classes and 1 background class. COCO has 123K annotated images, including 182 classes and 1 background class.

In this work, experiments are made on CUB, PASCAL and COCO. The train/test split of CUB is provided by [3] from their GitHub repository: 150 seen classes and 50 unseen classes. Because there is only one target class in each image, its class split is the same as its image split and this dataset can be utilized in both binary segmentation and multiclass segmentation. As for PASCAL and COCO, we directly adopt the class train/test splits proposed by [5]: split classes based on whether they are included in ImageNet 1K because the pre-trained models are trained on it. In PASCAL the last 5 classes are unseen classes (i.e., potted plant, sheep, sofa, train and tv/monitor) and the rest 15 ones are seen classes. In COCO 15 classes are unseen classes (i.e. cow, giraffe, suitcase, frisbee, skateboard, carrot, scissors, cardboard, clouds, grass, playing field, river, road, tree and wall-concrete) and the rest 167 are seen classes. The train/test splits from PASCALand COCO are

directly used as the image split. Unseen classes are ignored in loss function during training, seen classes are ignored in accuracy calculation under ZLSS setting, and no class is ignored in accuracy calculation under GZLSS setting.

As for word embeddings, all datasets could be trained with NLP models (i.e., word2vec and fastText). Following choices of [5], our experiments use Google News [19] pre-trained word2vec model, Common Crawl [20] pre-trained fastText model and their concatenation. When a class has multiple words, their word vectors are summed.

B. Implementation details

Experiments are implemented with PyTorch [21]. Pre-trained models and DeepLab-v3 framework are imported from PyTorch sub-package "models". The U-Net based model uses VGG16 as its image embeddings module and DeepLab-v3 uses ResNet101 as its backbone. Both VGG16 and ResNet101 are pre-trained on ImageNet 1K.

For pre-processing the data, all the input images, which are in JPG format, are resized to size (224,224) and normalized with mean value [0.485, 0.456, 0.406] and std value [0.229, 0.224, 0.225] to be aligned with ImageNet pre-trained models implemented by PyTorch team. All the labels, which are in PNG format, are converted to classes index. Specifically, each pixel in binary segmentation labels (i.e. CUB labels) are converted to 0 as background or 1 as target classes. As for multi-class segmentation labels (i.e. PASCAL and COCO-Stuff) 0 indicates background, -1 indicates difficult or ambiguous pixels, positive numbers represent classes. PASCAL annotations are represented by 22 different colors, and during preprocessing, they are converted to numbers from -1 to 20. COCO annotations are indexed images where 0-181 refers to classes and 255 refers to background (COCO do not annotate difficult pixels). Besides, it is worth noting that annotation images are resized by OpenCV [22] with "INTER_NEAREST" parameter to avoid generating new pixels.

C. Binary semantic segmentation

While training each model, the total episode number is 50,000, batch size is 32, learning rate is 1e-6 and is reduced by 50% every 5,000 episodes, and Adam is chosen as optimizer.

$$\omega, r_{i,j,k,c} \leftarrow argmin \sum_{i=1}^{w} \sum_{j=1}^{h} \sum_{k=1}^{n} \sum_{c=1}^{C} w\big(- y_{i,j,k,c} \log(r_{i,j,k,c}) - (1 - y_{i,j,k,c}) \log(1 - r_{i,j,k,c})\big) \quad (5)$$

1) Effect of word embeddings

Compared with U-Net, combination of U-Net and RN has higher mIoU over classes, as demonstrated in Table II. Among four different word embeddings, the model with CUB attributes has the best accuracy, which achieves 81.18% on unseen classes and 81.84% on seen classes. But, the accuracy difference between the attributes-based model and other three is smaller than 3%, so this may be not enough to imply that in semantic segmentation tasks, human annotated visual side information is more helpful than NLP models. Besides, different from experiments of [5], concatenation of word2vec and fastText does not have obvious advantages compared with other word embeddings.

2) Effect of network structure

When use FC layers in the word-embedding module, semantic feature map size is always $channels \times 1 \times 1$, and then they are repeated to the same width and height as the visual features. When used CNN layers, word embeddings are upsampled and then transformed by CNN layers with 3×3 size kernels. As Table III displays, after training two U-Net based RN models with the same semantic information (i.e. CUB attributes), the one using FC layers performs slightly better than the one uses CNN layers. Besides, using FC layers decreases model size, because its kernel size is only 1×1. Therefore, following experiments all adopt FC layers in their word embeddings. It is obvious that the U-Net framework performs better than DeepLab in this binary task. Table IV shows that the U-Net based model outperforms DeepLab based model in all criteria (both models are trained and evaluated with CUB images and CUB attributes).

D. Multi-class semantic segmentation

In multi-class training, the total episode number is 50,000 for PASCAL, 100,000 for COCO. The initial learning rate is 1e-4 and is reduced by 50% every 10,000 episodes, and optimizer is Adam. The batch size for U-Net based models is 32, but for DeepLab based models is 8 because of the limit of random access memory (RAM) (i.e. 16 GB). After trying using checkpoints and adjusting activation functions' parameters to reduce memory usage, we still had to reduce its batch size. In each episode, 10 classes are randomly selected from all classes, and then unseen classes are excluded from the label space. Because COCO has many more classes (i.e. 182 classes), to speed its converging speed, labels are randomly selected within the label space of each episode. When calculating the loss, unselected classes are ignored (i.e. unselected seen classes and all unseen classes). During testing, under ZLSS setting label space is unseen classes (i.e. 5 classes in PASCAL and 15 classes in COCO) and under GZLSS setting label space is both seen and unseen classes.

1) Effect of word embeddings

As Table V and Table VI show, three kinds of word vectors are adopted to study the effect of word embeddings on PASCAL. With U-Net based models, similar experiment results as authors in [5] are collected: the concatenation of word2vec and fastText has the best performance among them. However, the concatenated vectors have poor performance with the DeepLab based model.

TABLE II. EFFECT OF WORD EMBEDDINGS: MIOU OF BINARY SEMANTIC SEGMENTATION ON CUB WITH U-NET BASED MODEL

	unseen class mIoU (%)	seen class mIoU (%)	H (%)
No word embeddings	74.69	74.41	74.55
CUB attributes	81.18	81.84	81.51
word2vec	78.69	78.82	78.75
fastText	79.79	79.92	79.86
word2vec + fastText	78.51	78.69	78.60

TABLE III. EFFECT OF WORD EMBEDDINGS MODULE STRUCTURE: MIOU OF BINARY SEMANTIC SEGMENTATION ON CUB AND ITS ATTRIBUTES WITH U-NET BASED MODELS

	unseen class mIoU (%)	seen class mIoU (%)	H (%)
FC layers	81.18	81.84	81.51
CNN layers	78.12	78.79	78.45

TABLE IV. EFFECT OF NETWORK STRUCTURE: IOU OF BINARY SEMANTIC SEGMENTATION ON CUB AND CUB ATTRIBUTES WITH U-NET BASED MODEL

	unseen class mIoU (%)	seen class mIoU (%)	H (%)
U-Net_VGG16	81.18	81.84	81.51
DeepLab_Resnet101	72.85	74.17	73.51

Moreover, although label space is randomly chosen during training, their loss is still able to decrease to 1e-7 and the best ZLSS mIoU is reached within the first 15,000 episodes. Furthermore, in GLZSS the mIoU over seen classes is much higher than the mIoU over unseen classes. These indicate models are able to converge with seen classes, but is overfitting on PASCAL, causing poor performance with unseen classes. Besides, the model with concatenated vectors ends with higher loss than the other two, which indicates its overfitting is less serious (see Table V). One reason for overfitting is that training data is not diverse enough because there are only 20 classes in PASCAL.

2) Effect of network structure

Although DeepLab has a more complex structure, it does not always have better results than U-Net in this case. By comparing Table V and Table VI, it is observed that DeepLab based models are more difficult to converge, which indicates its overfitting problem as less serious than U-Net based models. The first model (i.e. the one uses word2vec) in Table V does not have the highest ZLSS mIoU, but it reaches the best harmonic mean among above two tables. However, the other two DeepLab based models have rather poor performance, so it is hard to say whether DeepLab has better capability on multi-class tasks.

1) Effect of data

As there are only 20 classes in PASCAL, the label spaces are highly overlapped among episodes. However, meta learning aims to feed random and various classes to the model to make it adapt to new classes. As mentioned above, RN is overfitting on PASCAL, so to investigate whether more diverse training data could boost its performance, RN is also trained and evaluated on COCO. With word2vec as semantic side information, VGG16 as image embedding module, U-Net as relation module, RN is able to reach 37.37% of ZLSS mIoU, 6.42% of GZLSS harmonic mean (3.80% over unseen classes and 20.55% over seen classes). Furthermore, models trained on these two datasets are cross evaluated on each other. As shown in Table VII, COCO trained model surpasses PASCAL trained model in ZLSS of both datasets.

Meanwhile, size of objects could also affect prediction scores. Below figures (see Fig. 10 and Fig. 11) demonstrate the relationship between IoU and average object size of unseen classes from PASCAL and COCO under ZLSS setting. For PASCAL classes and most COCO classes, it is obvious that IoU has a significantly positive correlation with object size.

TABLE V. EFFECT OF WORD EMBEDDINGS: MIOU OF MULTI-CLASS SEMANTIC SEGMENTATION ON PASCAL WITH U-NET AND VGG16 BASED MODELS

	ZLSS mIoU (%)	GZLSS unseen mIoU (%)	GZLSS seen mIoU (%)	GZLSS H (%)	Lowest loss
word2vec	40.09	10.87	51.03	17.92	1e-7
fastText	41.84	15.75	52.56	24.23	1e-7
w2v + ft	48.93	16.09	50.49	24.40	1e-4

TABLE VI. EFFECT OF WORD EMBEDDINGS: MIOU OF MULTI-CLASS SEMANTIC SEGMENTATION ON PASCAL WITH DEEPLAB AND RESNET101 BASED MODELS

	ZLSS mIoU (%)	GZLSS unseen mIoU (%)	GZLSS seen mIoU (%)	GZLSS H (%)	Lowest loss
word2vec	41.38	17.13	67.80	27.35	1e-3
fastText	40.02	7.6	58.66	13.45	1e-3
w2v + ft	36.18	5.27	39.14	9.29	1e-3

E. Compare with baseline models

In binary ZLSS, compared with the baseline model U-Net, RN is able to improve the accuracy with the help of semantic side information. But in multi-class tasks, accuracy of RN depends on how diverse training data is, as shown in Table VIII and Table IX where their best results are picked up for comparison. In both tables, SPNet adopts concatenation of word2vec and fastText as its word embeddings and DeepLab as its framework. But RN uses VGG16 and U-Net as its image embedding module and relation module separately. In Table VIII RN uses concatenation of word2vec and fastText as well, but in Table IX it uses word2vec only.

When RN is trained on PASCAL, its ZLSS mIoU is close to SPNet but its GZLSS accuracy is much better than SPNet. And as expected, a bigger dataset (i.e. COCO-Stuff) could boost performance of RN with meta-learning. After only 50,000 episodes, RN achieves higher mIoU than SPNet in both ZLSS and GZLSS tasks on COCO. But its loss (i.e. 5e-2) is still big and its mIoU on seen classes (i.e. 20.07%) is rather low, compared with other experiments, implying it is underfitting.

However, work done in [5] is able to increase the GZLSS accuracy of SPNet dramatically by reducing its prediction scores on seen classes (i.e. calibration). After calibrating, SPNet-C could maintain its mIoU on seen classes and increase its mIoU on unseen classes. On the other hand, although calibrating RN could improve its unseen mIoU a little, it would decrease seen mIoU dramatically, resulting in lower harmonic mean.

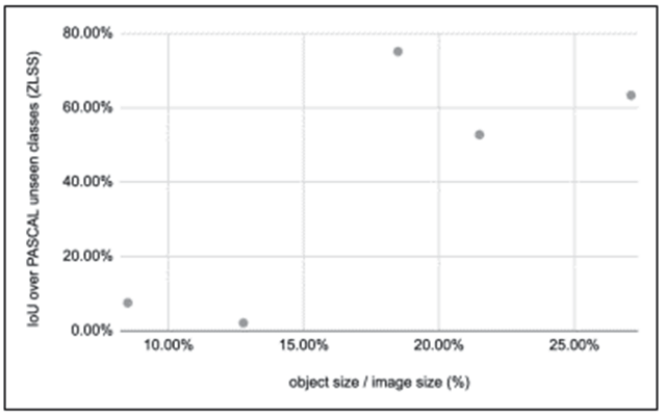

Fig. 10. Relationship between IoU and average object size of unseen classes in PASCAL under ZLSS setting

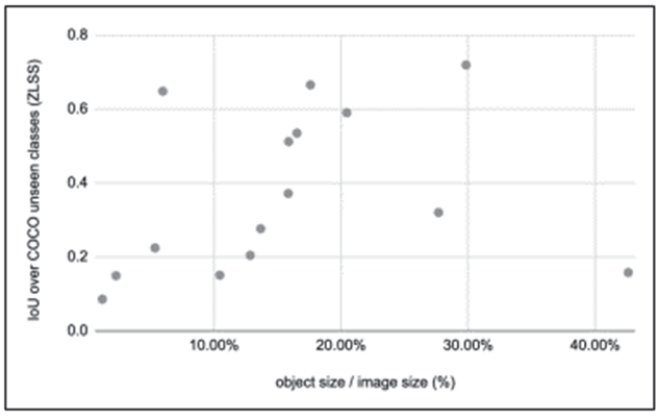

Fig. 11. Relationship between IoU and average object size of unseen classes in COCO under ZLSS setting

TABLE VII. PASCAL AND COCO TRAINED U-NET BASED MODELS WITH WORD2VEC AND VGG16 CROSS EVALUATED ON EACH OTHER'S TESTING DATA

ZLSS mIoU (%)	PASCAL-evaluated	COCO-evaluated
PASCAL-trained	40.09	5.09
COCO-trained	78.69	37.37

TABLE VIII. COMPARISON OF MIOU BETWEEN SPNET ON PASCAL. (SPNET DATA IS FROM [5]

	ZLSS mIoU (%)	GZLSS unseen mIoU (%)	GZLSS seen mIoU (%)	GZLSS H (%)
RN	48.93	16.09	50.49	24.40
SPNet	49.50	0.01	75.51	0.02
SPNet-C		29.33	76.84	42.45

TABLE IX. COMPARISON OF MIOU BETWEEN SPNET ON COCO. (SPNET DATA IS FROM [5]

	ZLSS mIoU (%)	GZLSS unseen mIoU (%)	GZLSS seen mIoU (%)	GZLSS H (%)
RN	37.37	3.80	20.55	6.42
SPNet	35.20	0.20	34.05	0.33
SPNet-C		8.33	34.52	13.42

According to mIoU (see Table II), the effect of different word embeddings is small. But in Fig. 12, it is obvious that attributes boost the model's accuracy and fastText performs better than word2vec. This is because in Fig. 12 prediction scores (i.e. from 0 to 1) are directly shown in images, but in Table II prediction scores are converted to prediction labels (i.e. 0 or 1) to calculate the mIoU.

Fig. 13 displays the segmentation results on PASCAL predicted by different RNs. The first three models from the left are trained on PASCAL with the same U-Net structure but different word embeddings. The fourth one adopts DeepLab structure and word2vec embeddings. The last one has the same setting as the first model, but is trained on COCO. Different from the above binary task, each predicted pixel in Fig. 13 is converted to RGB format by PASCAL palette. Because in ZLSS the search scope is unseen classes, all seen classes are represented by black color as the background. And ZLSS results on COCO are shown in Fig. 14 made by a U-Net based model with wor2vec embeddings. Seen classes are drawn in black, but unseen classes are represented directly by their class indexes. At each row, expected classes are listed on the left.

As analyzed in the previous Section, big objects (e.g. trains and sofas) are easier to be recognized than small objects. Additionally, the poor recognition of monitors may be caused by their colorful screens which is misleading when the data is not diverse enough. Although the COCO trained model could segment monitors well, it still has difficulty on sheep and glass (see the glass part of trains in Fig. 13).

V. CONCLUSION

In this work authors prove that it is feasible to extend RN from zero-shot image classification tasks to ZLSS and GZLSS tasks. Because semantic feature maps of each class are concatenated with visual feature maps separately, RN is limited to binary semantic segmentation. However, this work manages

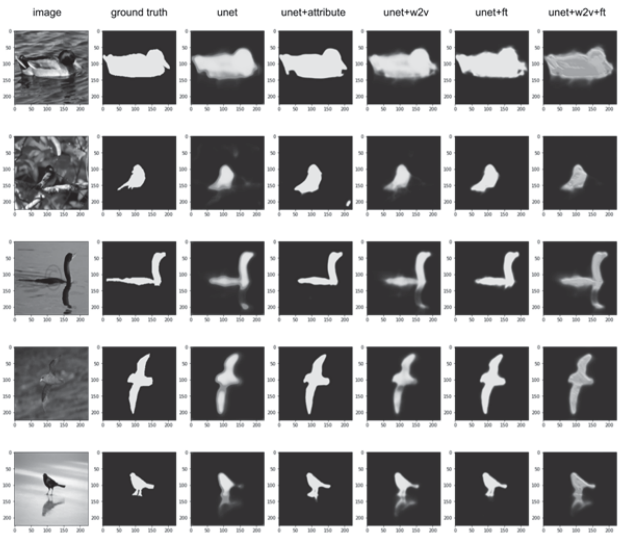

Fig. 12. Qualitative results on CUB unseen classes by U-Net based models with no-vector, attributes, word2vec, fastText and the concatenation of word2vec and fastText

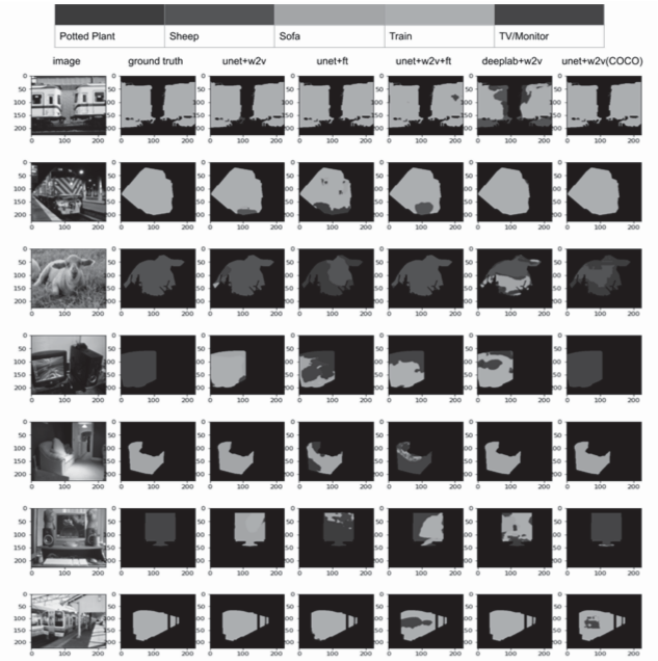

Fig. 13. Qualitative results on PASCAL unseen classes under ZLSS setting by RN

Fig. 14. Qualitative results on COCO unseen classes under ZLSS setting by U-Net based model with word2vec.

to do multi-class semantic segmentation tasks by joining multiple binary semantic segmentation pipelines. U-Net and SPNet frameworks are referred to in network structure designing. ImageNet pre-trained DNNs (i.e. VGG16, ResNet101 and DeepLab-v3) are adopted in image embedding module, attributes and NLP models are utilized as word embeddings.

In the binary ZLSS task on CUB, although there is only one object in each image and classes are similar to each other, RN could improve semantic segmentation accuracy with the help of semantic information. In multi-class tasks, the effect of data is obvious. Because RN is trained by meta-learning, more diverse training data could boost its accuracy. As above experiments shown, in ZLSS tasks, RN has a close mIoU as SPNet on PASCAL, but outperforms it on COCO. However, under GZLSS setting, although RN has obviously higher harmonic mean than SPNet, SPNet can effectively improve its accuracy by calibrating, whereas calibrating does not help RN. The poor performance in GZLSS tasks may be caused by U-Net structure, as U-Net was originally designed for binary segmentation tasks and may not be suitable for multi-class ones. Moreover, DeepLab could achieve better harmonic mean than U-Net but has a bigger model size and needs more time to train.

In terms of practical application, U-Net based RN has a smaller model size and a simpler architecture than SPNet and could maintain a similar ZLSS mIoU. However, as the number of classes grows in GZLSS tasks, U-Net based models have rather low mIoU. DeepLab based models may have better results in multi-class tasks but costs more RAM usage.

In experiments, the effect of different settings is also studied for optimizing the capability of RN. In binary segmentation, although attributes are more helpful than NLP models, they are more expensive to extend and apply on real problems. Overall, NLP models have unstable performance in experiments, so it is hard to conclude which one is the best in this case. While choosing between U-Net and DeepLab, the former one has an advantage when the search space is small.

Nevertheless, while designing network structure, authors simply remove the last 4 classifier layers of DeepLab and appended upsampling and CNN layers to it. There may be other options on how to utilize the DeepLab framework. Moreover, different semantic side information (e.g. Ontologies) could also be utilized to improve their accuracy. Authors consider further investigation of DeepLab based models capability and more diverse semantic information as potential future work. And it is worth noting because of limited GPU RAM, DeepLab based models were trained with batch size = 8, which would influence convergence.

REFERENCES

[1] O. Ronneberger, P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", *in Proc. International Conference on Medical image computing and computer-assisted intervention,* Springer, Cham, 2015, pp. 234-241.

[2] C. Liang-Chieh, Y. Zhu, G. Papandreou, F. Schroff and H. Adam, "Encoder-decoder with atrous separable convolution for semantic

image segmentation", *in Proc. European conference on computer vision (ECCV),* 2018, pp. 801-818.
[3] S. Flood, Y. Yang, L. Zhang, T. Xiang, P. HS Torr and T.M. Hospedales, "Learning to compare: Relation network for few-shot learning", *in Proc. IEEE Conference on Computer Vision and Pattern Recognition,* 2018, pp. 1199-1208.
[4] W. Catherine, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset." 2011.
[5] X. Yongqin, S. Choudhury, Y. He, B. Schiele and Z. Akata, "Semantic projection network for zero-and few-label semantic segmentation", *in Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8256-8265.
[6] L. Jonathan, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation", *in Proc. IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431-3440.
[7] C. Liang-Chieh, G. Papandreou, I. Kokkinos, K. Murphy and A.L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs.", *IEEE transactions on pattern analysis and machine intelligence 40*, no. 4 (2017): 834-848
[8] A. Zeynep, F. Perronnin, Z. Harchaoui and C. Schmid, "Label-embedding for attribute-based classification", *in Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 819-826.
[9] A. Zeynep, S. Reed, D. Walter, H. Lee and B. Schiele, "Evaluation of output embeddings for fine-grained image classification", *in Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2927-2936.
[10] X. Yongqin, A. Zeynep, G. Sharma, Q. Nguyen, M. Hein and B. Schiele. "Latent embeddings for zero-shot classification", *in Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 69-77.
[11] Z. Ziming and V. Saligrama, "Zero-shot learning via semantic similarity embedding", *in Proc. IEEE international conference on computer vision*, 2015, pp. 4166-4174.
[12] M. Bucher, V. U. Tuan-Hung, M. Cord and P. Pérez, "Zero-Shot Semantic Segmentation", *Advances in Neural Information Processing Systems*, 2019, pp. 466-477.
[13] X. Yongqin, C.H. Lampert, B. Schiele and A. Zeynep, "Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly", *IEEE transactions on pattern analysis and machine intelligence 41,* no. 9 (2018): 2251-2265.
[14] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality", *Advances in neural information processing systems*, 2013, pp. 3111-3119.
[15] J. Armand, E. Grave, P. Bojanowski, M. Douze, H. Jégou and T. Mikolov, "Fasttext. zip: Compressing text classification models", arXiv preprint arXiv:1612.03651 (2016).
[16] W. Tianhan, L. Xiang, C. Yau Pun, Y. Tai and C.K. Tang. "Fss-1000: A 1000-class dataset for few-shot segmentation." arXiv preprint arXiv:1907.12347 (2019).
[17] H. Kaiming, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", *in Proc. IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The pascal visual object classes challenge 2012 (voc2012) results (2012)." In URL http://www. pascal-network. org/challenges/VOC/voc2011/workshop/index.html.
[19] T. Mikolov, C. Kai, G. Corrado and J. Dean, "GoogleNews-vectors-negative300.bin.gz - Efficient estimation of word representations in vector space".arXiv preprint arXiv:1301.3781. 2013.
[20] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch and A. Joulin, "Advances in pre-training distributed word representations." arXiv preprint arXiv:1712.09405. 2017.
[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, "PyTorch: An imperative style, high-performance deep learning library", *Advances in Neural Information Processing Systems*, 2019, pp. 8024-8035.
[22] G. Bradski, "The OpenCV Library".Dr. Dobb's Journal of Software Tools. 2000