Final Project Report: Weakly Supervised Semantic Segmentation on Pascal VOC Dataset

LIU Vincent
ENS Paris Saclay
liuvincent25@gmail.com

Abstract

Our project focuses on image-level weakly supervised semantic segmentation (WSSS). It is an challenging problem which consist of assigning a label to each pixel of an image using only image-level annotations at training time.

We carry out an empirical experimental evaluation on Pascal VOC 2012 dataset [10] of two existing WSSS algorithms, namely Mixup-CAM [5] and Sub-category Exploration [6]. These methods try to improve the initial class activation maps [24] generated by convolutional neural networks (CNN) trained for image classification.

We propose another variants of these algorithms, which we named FMix-CAM, Manifold Mixup-CAM and Sub-category Teaching. These methods follow the same ideas of [5], [6] with slight modifications. As the names suggest, the additional modifications are inspired by Fmix [12], Manifold Mixup [21] and Knowledge Distillation [14].

We discovered that FMix-CAM can achieve encouraging results compared to the ones reported in the literature.

1. Introduction

The goal of semantic segmentation is to assign each pixel of an image to a corresponding class label. In a standard supervised setting, we learn from a dataset of images where each pixel is manually labeled to a class category. Deep learning methods to solve the segmentation problem has shown great advances, and it is quite exciting to see it can be applied to a broad range of problems including scene understanding for self-driving cars [4] [3], land cover mapping from satellite images [19] or inference on medical images diagnostics [18].

Because pixel-level annotation is expensive and time-consuming, weakly supervised approach has become an active area of research to address this issue. In weakly supervised learning, we are given a dataset of images and we only have access to the image-label annotations, which are usually cheaper to obtain. In this setting, most of the methods use class activation maps (CAM) [24] as a starting point to localize the object(s) of interest. The area of interest is refined in order to create pseudo labels assigned to each pixel, then the image can be fed to a convolutional neural network (CNN) [16] to perform supervised segmentation where the pseudo labels act as ground truth.

This problem is quite challenging because the pixel-wise pseudo labels obtained in a weakly supervised manner can be noisy and be prone to error, possibility affecting the performance of the subsequent segmentation task. While the gap between supervision and semi supervision performances for image classification is becoming closer [22] [9], there is still a lot of work to be done in semantic segmentation in weakly supervised learning.

My work relies on prior methods, namely Mixup-CAM [5] as well as Sub-category Exploration [6]. We evaluate three variants, which we name FMix-CAM, Manifold Mixup-CAM and Sub-category Teaching. The purpose of the project is to verify if the methods of [5] and [6] can be further improved using respectively Fmix [12], Manifold Mixup [21] and Knowledge Distillation [14].

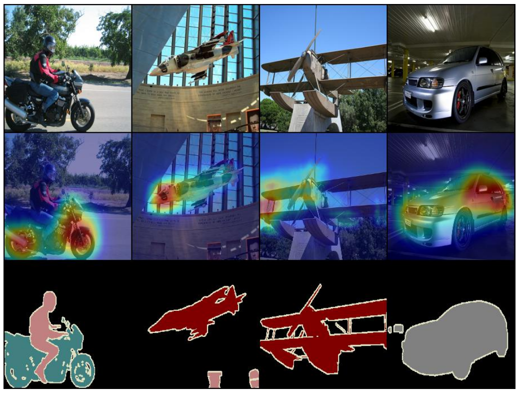

Figure 1. 1st row: Input Images from Pascal VOC dataset [10]. 2nd row: Class activation maps (CAMs). 3rd row: Segmentation masks. CAMs provide cues of the localization of objects of interest. Weakly Supervised Semantic Segmentation (WSSS) methods leverage this information to predict the segmentation mask. [24]

The project report is constructed as follows. In section 2, we discuss about existing methods that are related to our project. In section 3, we present FMix-CAM, Manifold Mixup-CAM and Sub-category Teaching. Section 4 and 5 report our quantitative and qualitative results on Pascal VOC dataset [10].

2. Related works

Using image-level labels, we train a standard convolutional neural network for classification. It allows us to obtain class activation maps which give us some ideas about where the CNN is looking. Most of the WSSS approaches consist of using the class activation maps to generate pixel-wise pseudo labels. The methods we discovered on the literature fall in two categories: the first one consists of improving the **initial** response maps while the second is a **refinement** additional stage, which consists of expanding the response maps.

These two approaches are in fact independent so in practice we can imagine using both in order to achieve maximum performances. In this paper, we focus on methods improving the first initial response maps. In this section, we present CAM generation procedure, some WSSS approaches (Sub-category Exploration, Mixup-CAM and AffinityNet) as well some deep learning techniques (Fmix, Manifold Mixup, Knowledge distillation and Vision Transformer).

2.1. Class activation map

CAM [24] is a method that helps human visually understand the decisions that CNN are making. In a nutshell, it produces response localization maps for an image classification CNN where global average pooled convolutional feature maps are the input of the last softmax layer. Most methods for the WSSS task in the literature use CAMs to leverage localization of the object, as shown on Figure 1. They are used to generate the segmentation pseudo labels of the images. To come up with segmentation label, each pixel is assigned to the label corresponding to the maximum activation score in the upsampled map.

However, it has been observed that class activation model fires on at the most discriminating portion of the object instead of the entire object area. As a result, the segmentation mask does not fit well object boundaries. Hence, it makes them quite ineffective to use raw response maps as pixel-wise pseudo labels. However, it is still a reliable starting point of most of the existing WSSS approaches.

2.2. Sub-category Exploration

In the paper of sub-category exploration [6], the authors propose to improve the prediction of the initial response maps. Their approach is effective, simple and computationally light. First, they start training a CNN for the image classification task. For each group of images belonging to a category c, they cluster the corresponding feature vector representation. They assign each image a new label corresponding to the cluster assignment, which they call the sub-category of c. Afterwards, they introduce a second classifier for an auxiliary task, that is to teach the network to discriminate the sub-categories. The two classifiers are jointly optimized with two cross entropy loss functions to predict the original parent category as well as the sub category. By trying to predict the sub-category, the activation map does not limit on the most discriminative part of the image and the network learn global aspect of the object of interest. At the end, the object mask can be extracted and be refined in order to have a better object boundary.

2.3. Mixup-CAM

This paper [5] tackles in an elegant way the problem of improving the initial cues given by the network. The authors propose to train the classification network with mixup data augmentation together with two uncertainty regularization losses. Mixup data augmentation [23] is the process of training a neural network on a convex combination space of image pairs and their corresponding labels. In order to further improve the response map, they use semi-supervised learning loss functions which are self entropy loss [11] and concentration loss [15]. Similarly, the CAMs can then be refined and used to train a segmentation network as a second step.

2.4. AffinityNet

AffinityNet [1], in contrast to other methods, focuses on refining the CAMs. The authors trained AffinityNet which learns categorical semantic affinity between a pair of adjacent coordinates. The predicted affinities are used to apply random walk to expand the area of the CAMs. The affinity between two pixels is measured by their distance in the feature space. The expanded CAMs can also be used as segmentation ground truths afterwards.

2.5. FMix

FMix [12] is a Mixed Sample Data Augmentation (MSDA) method related to Mixup, that uses binary masks obtained by applying a threshold to low frequency images sampled from Fourier space.

2.6. Manifold Mixup

Manifold Mixup [21] is also a MSDA method. It prevents neural networks to predict too confident predictions on interpolations of hidden representations.

2.7. Knowledge distillation

Knowledge distillation [14] refers to the process in which a student model is trained to match a teacher model.

Knowledge is transferred from the teacher model to the student by minimizing a loss function between the student and teacher outputs.

2.8. Vision transformers

Vision transformer (ViT) [9] is recent and is quickly becoming a breakthrough in Computer Vision. It has achieved remarkable results on image classification on ImageNet [8]. It is based on attention mechanism [2] and Transformers [20]. We think that being able to leverage information learned by ViT for semantic segmentation could be an area to explore.

3. Methodology

In this project, we study the methods that focus on improving the initial predictions of the class activation maps.

We propose to use Fmix [12] or Manifold Mixup [21] instead of Mixup [23], which are more general approaches. Second, we propose another framework, close to the Subcategory exploration method. It is based on knowledge distillation [14], we named it Sub-category Teaching.

3.1. CAM Generation

In order to generate class activation maps, we follow the process described by [24] [1]. We train a network with a CNN backbone with global average pooling (GAP) followed by a fully connected layer. The network is trained with image-level supervision. At testing time, we generate the feature maps f before GAP, the activation map M associated to class category C_i can be computed using the following equation:

$$M_i(x,y) = W_i^T f(x,y) \tag{1}$$

where W_i is the weight of the fully connected layer associated to class C_i and f is the feature at position x, y.

3.2. FMix-CAM

The first method consists of modifying the approach proposed by [5]. We use FMix [12] instead of the traditional Mixup data augmentation [23]. The procedure is described as follows:

1. We generate a binary mask $\in \{0,1\}^{w\times h}$ sampled from Fourier Space, as described in [12].

2. For two data samples (I,t) and (I',t'), we generate a new mixed sample:

$$I_{mix} = mask \odot I + (1 - mask) \odot I'$$
$$t_{mix} = \lambda t + (1-\lambda)t'$$

where I is an image, t is the target vector, $\odot$ is the element wise multiplication and λ is the proportion of 1 in the binary mask.

Following [5], we use multi-class classification loss L_{cls}, and we add two regularization loss functions. The first one comes from [11], which is the self entropy loss.

$$L_{ent} = -\sum_i P_i log(P_i)$$

where P is the predicted vector. The second one is the concentration loss proposed by [15] [5], applied directly to the class activation maps.

$$L_{con}(M) = \sum_{C_i}\sum_{h,w} || < h,w > - < \mu_h^c, \mu_w^c > ||^2 \hat{M_i}(h,w)$$

where $\mu_h^c = \sum_{h,w} h\hat{M_i}(h,w)$ is the center in height for category C_i, $\hat{M_i}$ is the normalized response of M_i to represent a spatially distributed probability map, according to [5].

The desired effect is to give some consistency to the class activation maps. At the end, we look to minimize jointly the three loss functions.

$$L = L_{cls} + \lambda_{ent}L_{ent} + \lambda_{con}L_{con} \tag{2}$$

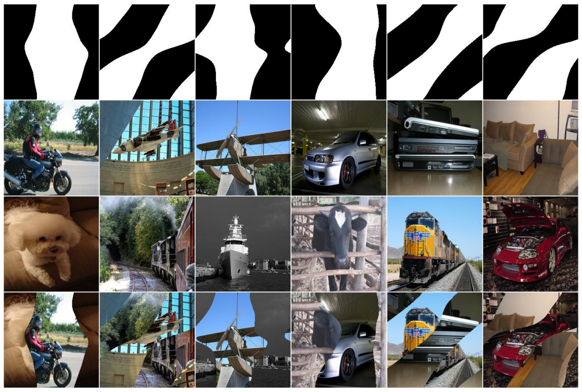

Figure 2. 1st row: Mask generated using FMix method, as described in [12]. 2nd row: Sample I. 3rd row: Sample I'. 4th row: New mixed Sample I_{mix}.

3.3. Manifold Mixup-CAM

The second method consists of using a generalized version of Mixup which is Manifold Mixup. According to [21], the procedure is summarized below.

1. Select a random layer k from a set of eligible layers in the neural network.

2. Input two mini batches (I,t) and (I',t') of data. Forward until layer k to obtain feature representation $f_k(I)$ and $f_k(I')$.

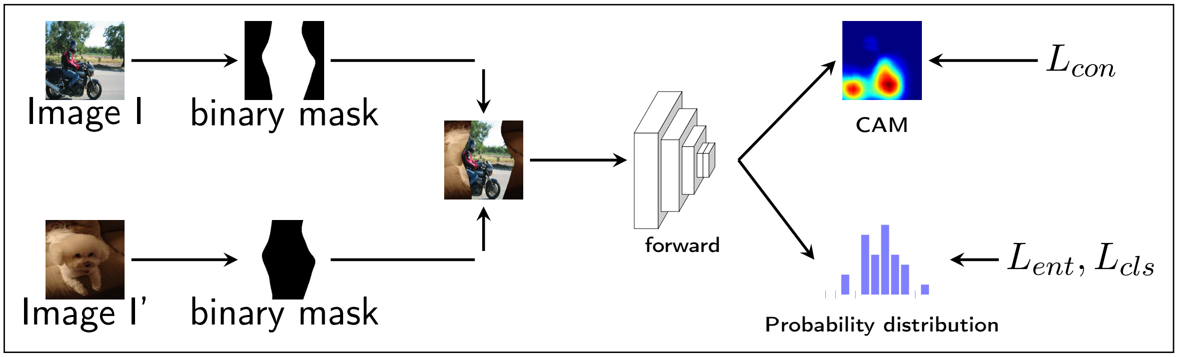

Figure 3. FMix CAM. Figure is inspired from [5]. At step 1, we sample two images as well as a binary mask. At step 2, we generate a mixed sample based on the inputs. At step3, we forward the mixed image into the neural network, and compute classification loss and self entropy loss to our probability outputs, together with a concentration loss to our class activation maps.

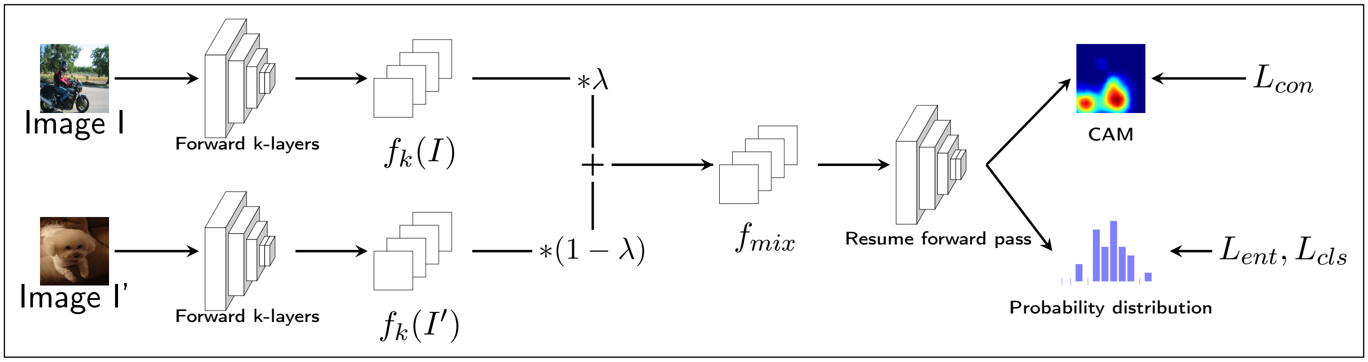

Figure 4. Manifold Mixup CAM. Figure is inspired from [5]. At step 1, we sample two images and forward the activation maps $f_k(.)$. At step 2, we compute a convex combination of the activation maps to obtain f_{mix}. We resume the forward pass using f_{mix} as input. At step 3, we compute the loss functions, same as above.

3. Use Mixup on $(f_k(I), t)$ and $(f_k(I'), t')$:

$$f_{mix} = \lambda f_k(I) + (1-\lambda) f_k(I')$$
$$t_{mix} = \lambda t + (1-\lambda) t'$$

4. Resume the forward pass from layer k with input f_{mix} associated to target vector t_{mix}.

The difference with traditional Mixup is that the convex combination of input appears randomly during the forward pass. We add also the self entropy loss as well as the concentration loss, similarly to our first proposed method.

3.4. Sub-Category Teaching

Our motivation for the last approach is that ViT [9] is known to achieve excellent performances on image classification over existing CNN models. As CAMs are specific to CNN architecture with GAP layer, we cannot compute CAMs for ViT. Using attention maps produced by ViT can be an alternative, however, we don't know to get a "CAM-equivalent" of attention maps, that is a spatial class probability map associated to each class category. In order to leverage the powerful representation learned by ViT, we can teach a second CNN model to learn the feature representations. We follow the Sub-category exploration framework so that the student can learn about subcategories discovered by the Teacher. We hope to obtain a richer subcategory discovery compared to [6] since we use a specific auxiliary network for this purpose.

Teacher Training. We train a teacher model M_1 which is a vision transformer (ViT). After training, we use ViT as a feature extractor to obtain the embedding $E(I)$ of each image I.

Sub-category discovery. Afterwards, we apply clustering on the features $E(I)$, for each subset of points belonging to class category C_i. We have $K_i \in \mathbb{N}^*$ number of clusters for each parent class category C_i. The cluster assignment can be interpreted as the subcategory of class category C_i.

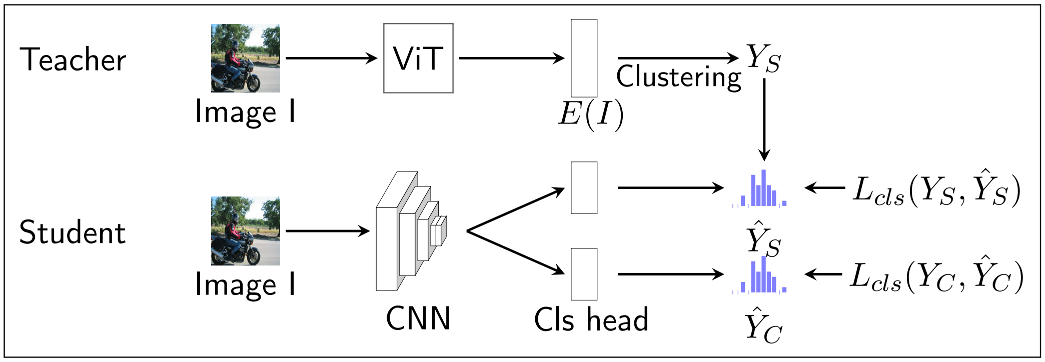

Figure 5. Sub-Category Teaching. Step 1 refers to the process of fine-tuning a vision transformer pre-trained on ImageNet (teacher). Step 2 consists of computing the feature vectors, and applying clustering on images corresponding to each parent category. At step 3, we train a convolutional neural network (student) to classify both parent category labels and corresponding subcategory labels.

If the image does belong to the parent the class category C_i, then we assign a subcategory $k^\star$ to the image (the cluster assignment), so that $S_i^{k^\star} = 1$ and $S_i^k = 0$ if $k \neq k^\star$. If the image does not belong to the class category C_i, then $S_i^k = 0 \quad \forall k$.

Student Teaching. We aim to train a student which is a CNN model M_2. It consists of a CNN backbone with two head classifiers. The first head aims to classify the parent category C_i as usual. The second tries to classify the sub-category assignment S_i^k. We seek at optimizing jointly:

$$L = L_{cls}(Y_C, \hat{Y_C}) + L_{cls}(Y_S, \hat{Y_S}) \tag{3}$$

where L_{cls} can be any multi-classification loss.

3.5. Implementation Details

We run the code on personal laptop with NVIDIA RTX2090 GPU. We did not use directly the Sub-category exploration Github repository because we wanted to practice our coding skills. However, we still used some existing functions that are available on Github.

Networks. For the network backbone, we decided to fine tune the same one as [5] [6] in their experiments. It is a ResNet-38 [13] architecture trained on ImageNet. The weights are available at: https://github.com/Juliachang/SC-CAM.

FMix. We use the official FMix implementation [12], available at: https://github.com/ecs-vlc/FMix.

Manifold Mixup. We adapted the Mixup implementation from https://github.com/facebookresearch/mixup-cifar10 and the Manifold Mixup implementation from https://github.com/vikasverma1077/manifold_mixup. The concentration loss is adapted from https://github.com/NVlabs/SCOPS.

Sub Category Teaching. We use timm pytorch implementation of vision transformers: https://github.com/rwightman/pytorch-image-models/tree/master/timm. For clustering the feature representations, we use FINCH [17] https://github.com/ssarfraz/FINCH-Clustering, as suggested by [6]. The advantage is that we do not have to deal with the number of clusters.

Our project only focuses on improving the initial cues of the object localization. But it is interesting to see the impact on subsequent refinement steps. To this end, similarly to [5], [6], we use directly the code of AffinityNet in order to refine the class activation maps (for the second part). We use also the Deeplab-v2 framework which uses the ResNet-101 architecture for the final segmentation.

Affinity Net. https://github.com/jiwoon-ahn/psa **Deeplab-v2.** [7] https://github.com/kazuto1011/deeplab-pytorch

Dataset. We evaluate our algorithms on Pascal VOC 2012 Dataset. It is composed of 1,464 train images and 1,449 val images [10]. It contains 21 categories, an image can have multiple labels, e.g. a *person* sitting on a *chair* with a *dog* can be on the same image.

Hyperparameters. We use a batch size of 8 for classification and 2 for AffinityNet and Deeplab-v2, learning rate of 0.1 with SGD optimizer. Data augmentation such as Random Flipping, Color Jiterring has been used.

Additional Dataset. The authors of [1], [5], [6] used additional data set to further improve the mIoU. The additional data set is the Semantic Boundary Dataset, which can be found at: http://home.bharathh.info/pubs/codes/SBD/download.html. I used it only for my last test since training time was significantly longer.

Our code. Our final implementation for generating the CAMs can be found at : https://github.com/liuvince/mva-wsss.

4. Quantitative evaluation

To quantify our result, we will use the semantic segmentation metric mIoU which is the mean intersection over union of our predictions compared to the ground truth segmentation maps provided by Pascal VOC 2012 dataset. The process is depicted as follows. First we evaluate the mIoU of our initial response maps against ground truths provided by Pascal VOC 2012 dataset on train set. We also evaluate the same response maps which have been refined using Affinity Net. Secondly, we train a segmentation neural network using the predictions as pseudo ground truths. In part 4.1 and 4.2, we use only Pascal VOC 1,464 training images to test our methods while in part 4.3, we evaluate our best method with augmented 10,528 training images, in the same settings as [1] [5] and [6]. The training time with 10,528 training images was significantly longer so I could not do it for all the methods, I decided to report the result with additional data only for my best method.

4.1. Improving initial response maps

Table 1 shows the mIoU of class activation maps on train set with different methods that we tested. We observe that for all methods, the mIoU is increased compared to the baseline.

In our experiments, the MSDA methods out perform subcategory methods, while it is not the case in the reported results from original papers [5] [6]. We believe that our implemented version of sub-category methods in practice over-fit a little. MSDA methods, on the other hand, are more straightforward to use.

In the initial step, we achieve the best results using Fmix CAM, by a large margin. Note in practice it took longer to converge while the other methods had converged faster.

We see that using Affinity Net in order to refine the object mask improves significantly the final mIoU. However, we still have overall lower results compared to the published papers' [5][6][1]. The reason might be that we did not use the additional data and the exact hyper-parameters like the authors did.

Methods	mIoU (%)	mIoU (%) (refined)
CAM (baseline)	45.0	55.6
Sub-Category Exploration [6]	46.5	56.7
Sub-Category Teaching (ours)	46.4	55.2
Mixup CAM [5]	46.8	55.8
Manifold Mixup CAM (ours)	46.8	56.6
Fmix CAM (ours)	**48.2**	**58.4**

Table 1. mIoU of class activation maps on *train* set during the initial step using only 1,464 training images

As a side note, in order to reproduce good results, we need to perform *test time augmentation* for computing the CAMs. Basically, in [1], they compute CAMs for different scales of input images. They average the results which helped a lot improving performances (3 % mIoU gain).

4.2. Impact on subsequent steps

Table 2 shows the final mIoU of segmentation masks on val set with different methods. We use the refined class activation maps of the previous step, and use it as pseudo ground truths to train Deeplab-v2 [7], a segmentation network.

[1] states that using supervised learning on noisy labels can improve even further the mIoU, compared to the previous step. However, we did not observe an improvement. The reason could be that we use a small dataset together with a small batch size (due to small memory gpu), so the segmentation network is prone to noise and over-fitting. Note that for CAMs, the comparison can also be unfair since we use image-level labels to filter out CAMs of irrelevant classes. In practice, even if the mIoU presented here is lower, the segmentation network is needed since in a realistic setup, we would not filter out irrelevant classes.

Methods	mIoU (%)
CAM (baseline)	52.7
Sub-Category Exploration [6]	54.3
Sub-Category Teaching (ours)	53.0
Mixup CAM [5]	**55.1**
Manifold Mixup CAM (ours)	53.5
Fmix CAM (ours)	54.0

Table 2. Final mIoU of class activation maps on *val* set after training DeepLabv2 segmentation network using only 1,464 training images

4.3. Supplementary results on FMix-CAM

In this section, we use the additional dataset Semantic Boundary Dataset, in order to evaluate our final results on the same settings as [1] [5] [6]. The results of Affinity Net, Sub-Category Exploration and Mixup CAM in Table 3 are the real ones published in the original papers (on previous section, I re-implemented my own version and did the experiments). We report the results of our best method, which is FMix-CAM. We remark that using more images improves significantly the results (from 54.0 to 61.2). An entire process (training a classification network, then AffinityNet, and finally Deeplab-v2 together with producing the evaluation) took more than 15 hours, so we did not tune the hyper parameters. Even with more data, we still have trouble in the second step of training a segmentation network. We believe with further tuning of the segmentation network, we can increase the results to be at least as competitive as the reported Mixup CAM from the original paper, since we got encouraging results for the first step of CAMs generation.

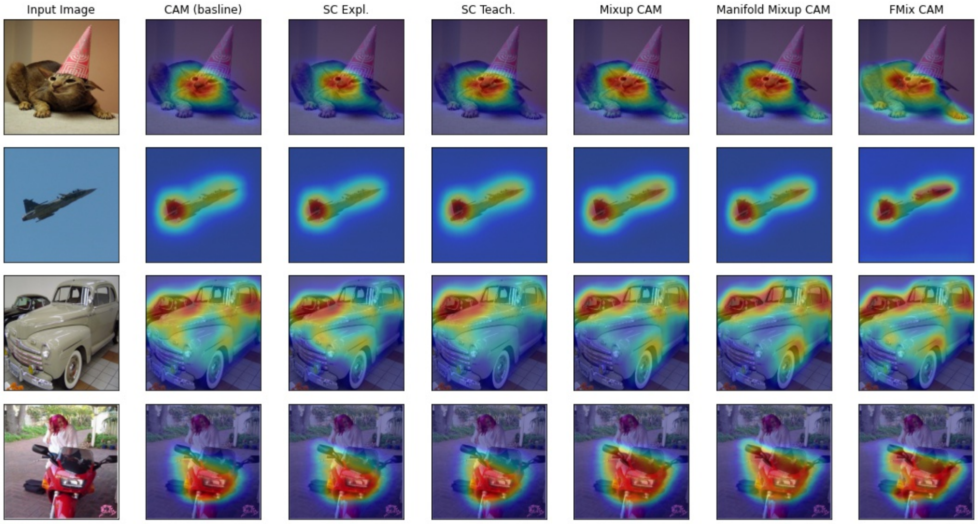

Figure 6. Qualitative Comparison of Initial Class Activation Maps for different methods.

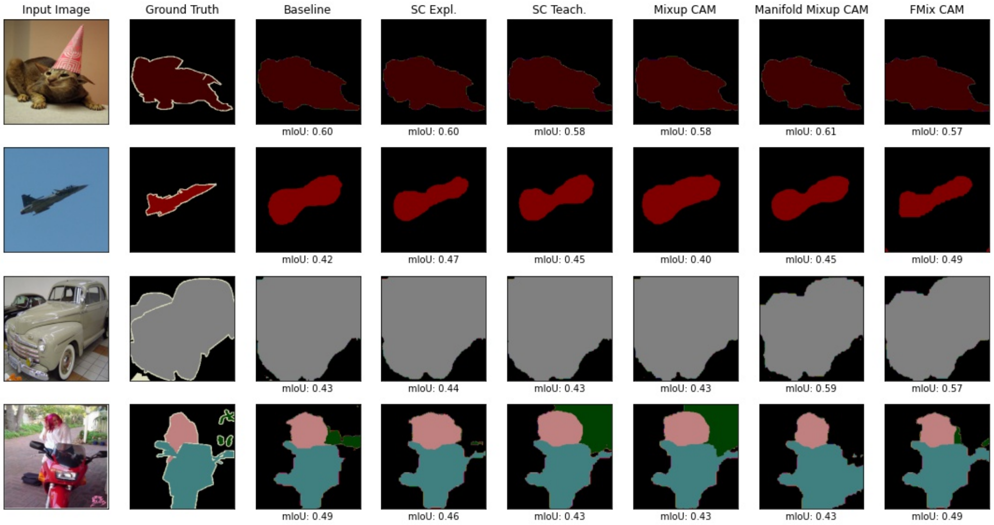

Figure 7. Qualitative Comparison of segmentation maps obtained after refinement for different methods.

Methods	mIoU
(reported by real paper against ours)	(%)
Affinity Net [1]	61.7
Sub-Category Exploration [6]	66.1
Mixup CAM [5]	65.6
Fmix CAM (ours)	61.2

Table 3. Final mIoU of class activation maps on *val* set after training DeepLabv2 segmentation network using additional 10,528 training images

5. Qualitative evaluation

5.1. Initial class activation maps

Figure 6 shows the differences between response maps for different methods. We see that the standard CAMs tend to focus on the most discriminative part of the object.

For the first cat example, we remark that the focus is on the cat's head. However, in the MSDA methods, the response maps are more spread over the cat body. For Mixup-CAM and Manifold Mixup-CAM, the maps cover the head as well as the paws while for Fmix CAM, it covers impressively the whole body.

For the plane example, it seems that Fmix are more con-

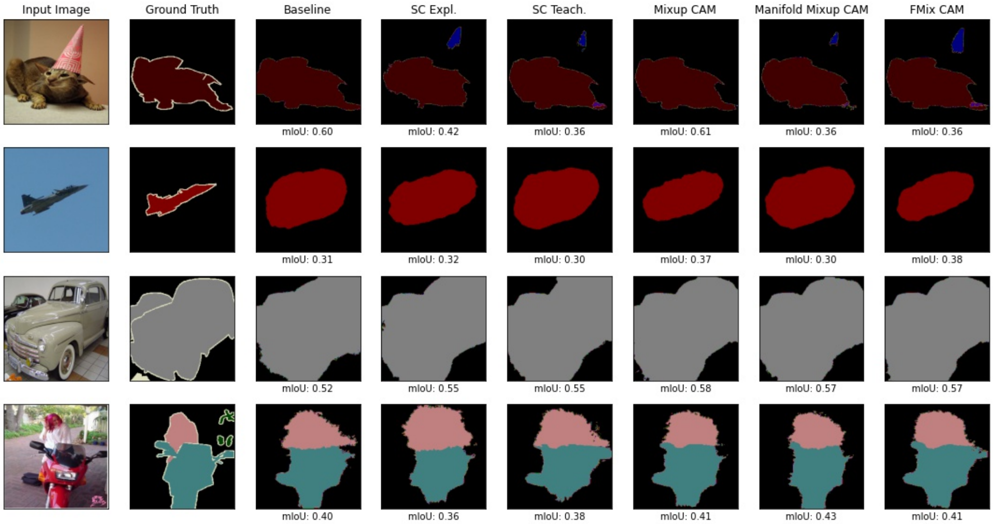

Figure 8. Qualitative comparison of segmentation maps after training a segmentation network using pseudo ground truths.

fident and the object area is more fine grained compared to the other methods.

For the scooter example, it seems to lead to similar results regardless the methods.

5.2. CAMs refined by Affinity Net

We can compare the segmentation maps to the ”real” ground truth since the images in the Pascal VOC 2012 dataset are manually annotated but we do not use them since we fake a weakly-supervised setup. Figure 7 shows the segmentation maps obtained after Affinity Net refinement. We see that the areas are more fine grained compared to the raw response maps obtained during the previous step. Surprisingly, for the cat example, the mIoU obtained by using FMix CAM is less than the baseline. For the plane and the car example however, it fits better the objects' boundary.

5.3. Segmentation maps generated by a segmentation network

Figure 8 shows the results after training a segmentation neural network. We see that the final segmentation maps are a bit noisier than the previous steps which was not observed in the real papers [1], [5] [6]. The issue is that we use a small subset of the dataset so the segmentation network may have over-fitted. Again, please note that the comparison is unfair since we use image-level labels to filter out CAMs of irrelevant classes at the previous step.

6. Discussion and Conclusion

In this project, we have worked on the challenging problem of weakly supervised semantic segmentation. It is an important work for segmentation problems since it can bring down the need of expensive annotated images. While most of the methods in the literature focus on refining the class activation maps like [1], our focus is on improving the initial class activation maps. Our work is based on AffinityNet [1], Mixup-CAM [5] and Sub-category Exploration [6]. We tried some modifications to see if using methods such as Fmix [12], Manifold Mixup [21] and Knowledge Distillation [14] can further improve the original results on WSSS. For a small dataset, we got encouraging results for Fmix CAM since we got better results on the initial generation of the response maps. However, we did not manage to train a good segmentation network for the second step. The reason could be that we used a small dataset, a small batch size, leading to possible overfitting.

As further work, in order to improve the initial class activation maps, we can focus on working on another way to leverage the powerful feature representation of ViT. We can also investigate the impact of different hyperparameters in order improve the segmentation network of the second step. Typically, with more computational power, one could investigate if using a smaller learning rate or bigger batch size could improve the segmentation mIoU.

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation, 2018. 2, 3, 5, 6, 7, 8

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016. 3

[3] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground

truth database. *Pattern Recognition Letters*, xx(x):xx–xx, 2008. 1

[4] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV (1)*, pages 44–57, 2008. 1

[5] Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Mixup-cam: Weakly-supervised semantic segmentation via uncertainty regularization, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[6] Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation via sub-category exploration, 2020. 1, 2, 4, 5, 6, 7, 8

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2017. 5, 6

[8] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. 3

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. 1, 3, 4

[10] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015. 1, 2, 5

[11] Yves Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. volume 17, 01 2004. 2, 3

[12] Ethan Harris, Antonia Marcu, Matthew Painter, Mahesan Niranjan, Adam Prügel-Bennett, and Jonathon Hare. Fmix: Enhancing mixed sample data augmentation, 2020. 1, 2, 3, 5, 8

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 5

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 1, 2, 3, 8

[15] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation, 2019. 2, 3

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012. 1

[17] Vivek Sharma M. Saquib Sarfraz and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943. 5

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 1

[19] Michael Schmitt, Jonathan Prexl, Patrick Ebel, Lukas Liebel, and Xiao Xiang Zhu. Weakly supervised semantic segmentation of satellite images for land cover mapping – challenges and opportunities, 2020. 1

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 3

[21] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states, 2019. 1, 2, 3, 8

[22] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification, 2020. 1

[23] Hongyi Zhang, Moustapha Cisse, Yann Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. 10 2017. 2, 3

[24] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization, 2015. 1, 2, 3