Harvesting Information from Captions for Weakly Supervised Semantic Segmentation

Johann Sawatzky Debayan Banerjee Juergen Gall
University of Bonn
{jsawatzk, debayan, jgall} @ uni-bonn.de

Abstract

Since acquiring pixel-wise annotations for training convolutional neural networks for semantic image segmentation is time-consuming, weakly supervised approaches that only require class tags have been proposed. In this work, we propose another form of supervision, namely image captions as they can be found on the Internet. These captions have two advantages. They do not require additional curation as it is the case for the clean class tags used by current weakly supervised approaches and they provide textual context for the classes present in an image. To leverage such textual context, we deploy a multi-modal network that learns a joint embedding of the visual representation of the image and the textual representation of the caption. The network estimates text activation maps (TAMs) for class names as well as compound concepts, i.e. *combinations of nouns and their attributes. The TAMs of compound concepts describing classes of interest substantially improve the quality of the estimated class activation maps which are then used to train a network for semantic segmentation. We evaluate our method on the COCO dataset where it achieves state of the art results for weakly supervised image segmentation.*

1. Introduction

Fully convolutional networks have shown to perform very well on the task of semantic image segmentation, but training such networks requires data annotated on a pixel level. Obtaining such annotations, however, is very time consuming and expensive. For this reason, approaches for weakly supervised semantic segmentation have been proposed that require less supervision for training. While the type of supervision ranges from class tags, key points, scribbles to bounding boxes, the vast majority of these approaches rely on class tags since it is assumed that these tags can be more easily acquired. A popular approach for weakly supervised learning from image tags [21, 52, 1] consists of estimating for each image in the training set so-called class activation maps (CAMs) [52], which indicate where certain semantic categories occur in an image. In a second step, pixelwise class labels are extracted from the CAMs and a neural network for image segmentation is trained. These approaches, however, still assume curated image tags, *i.e.* all classes of interest in the image must be tagged. This is not guaranteed for tags retrieved together with images from the Internet.

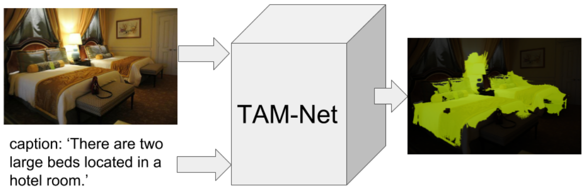

caption: 'There are two large beds located in a hotel room.'
TAM-Net

Figure 1. Given one or multiple captions per image in the training set, our network predicts text activation maps (TAMs) for each image which are then converted into class activation maps (CAMs) as illustrated in Figure 3. The text activation maps are more general than class activation maps since they localize compound concepts like 'two large beds' as well as categories like 'bed'. The example shows the class activation map estimated for 'bed' for this training example. Using the estimated CAMs of all training images, a standard convolutional neural network for semantic segmentation can then be trained.

Another form of supervision are readily available textual descriptions. Such textual descriptions can be either from image captions or text surrounding or referring to an image in an article or blog. Such textual descriptions are more general than class tags and it is possible to reduce these textual descriptions to class tags by parsing the texts for the names of the categories of interest. However, textual descriptions are richer in the description of the image content than just class tags as illustrated in Figure 1. In case of class tags, the image would be only labeled by the relevant category 'bed' and the only information we have is that the image contains at least one bed, but we do not know if the bed is large or small or if many instances are in the image. The caption

"There are two large beds located in a hotel room" provides much more information. In particular, 'two large beds' indicates that many pixels of the image should be labeled by the category 'bed'.

In this work, we therefore propose an approach that uses non curated image captions as weak supervision for training a convolutional neural network for semantic image segmentation. Our contribution focuses on the research question how class activation maps (CAMs) can be estimated from image captions and we show that these class activation maps are substantially more accurate than CAMs estimated from class tags. In order to estimate the class activation maps for the training images from captions, we learn a joint embedding of the visual representation of the image and the textual representation of the caption as illustrated in Figure 3. Using such a joint embedding, our network predicts text activation maps (TAMs), which locate categories like 'bed' as well compound concepts like 'two large beds'. For each class, the TAM of the class as well as the TAMs of the compounds which contain the class name are fused to generate the CAM for the class.

We provide a thorough ablation study that analyses the benefit of the additional textual context that is provided by the compound concepts. On the COCO dataset [26], the proposed approach outperforms the current state of the art in weakly supervised semantic segmentation.

2. Related Work

Fully supervised semantic segmentation has been studied in many works, e.g., [7], [25], [27], [50], [51]. More recently, weakly-supervised semantic segmentation has come to the fore. Early work such as [42] relied on hand-crafted features, such as color, texture, and histogram information to build a graphical model. However, with the advent of convolutional neural networks (CNN), this conventional approach has been gradually replaced because of its lower performance on challenging benchmarks [11].

A natural step to less supervision are more coarse spatial cues like bounding boxes, key points and scribbles. In [30], Papandreou et al. use the expectation-maximization algorithm to perform weakly-supervised semantic segmentation based on annotated bounding boxes and image-level labels. Another more sophisticated approach based on bounding boxes was proposed in [20]. The authors use region proposal generated by Multiscale Combinatorial Grouping (MCG) [33] and Grabcut [35] to localize the objects more precisely within the bounding box. More recently, Li et al. [23] used bounding boxes for object classes and image level supervision for stuff classes. In [24], Lin et al. made use of a region-based graphical model, with scribbles providing ground-truth annotations to train the segmentation network. Scribbles also served as supervision for the works of [40, 41], which investigate loss regularizations. Human annotated keypoints were used by Bearman et al. [2] for weakly supervised class segmentation and by Sawatzky et al. [38] for weakly supervised affordance segmentation.

While the works mentioned above require some type of explicit spatial hints, others only rely on the list of present classes in the image. Qi et al. [34] used proposals generated by MCG [33] to localize semantically meaningful objects. Recently, Fan et al. [12] leveraged saliency to obtain object proposals and link objects of the same class across images with a graph partitioning approach. Pathak et al. [31] addressed the weakly-supervised semantic segmentation problem by introducing a series of constraints.

In absence of explicit location cues provided by humans, class activation maps (CAMs) [52] proved to be a seminal supervision source. Pinheiro et al. [32, 39] pioneered in this area. In [21], three loss functions are designed to gradually expand the high confidence areas of CAMs. This approach was first improved by Wei et al. [44] who use an adversarial erasing scheme to acquire more meaningful regions that provide more accurate heuristic cues for training. Recently, Huang et al. [17] proposed deep seeded region growing of CAMs with image level supervision. In [45], Wei et al. presented a simple-to-complex framework which used saliency maps produced by the methods [8] and [18] as initial guides. Hou et al. [16] advanced this approach by combining the saliency maps [15] with attention maps [48]. Oh et al. [29] and Chaudhry et al. [5] considered linking saliency and attention cues together, but they adopted different strategies to acquire semantic objects. Roy and Todorovic [36] leveraged both bottom-up and top-down attention cues and fused them via a conditional random field as a recurrent network. Ahn et al. [1] use image level class labels to generate an initial set of CAMs and then propagate those CAMs by using random walk predictions from AffinityNet. Wei et al. [46] use image level supervision with dilated convolutions with varying levels of dilations to generate weakly supervised segmentations. Wang et al. [43] use image level supervision along with a bottom-up and top-down framework, which alternatively expands object regions and optimizes the segmentation network. In Briq et al. [4] a convolutional simplex projection network is used for weakly supervised image segmentation. Tang et al. [41] integrate standard regularizers directly into the loss functions over partial input for semantic segmentation. Ge et al. [13] use a four stage process that combines object localization with filtering and fusion of object categories. Hong et al. [14] and Jin et al. [19] tackle the weakly-supervised semantic segmentation problem using images or videos from the Internet. While the mentioned works mainly focus on steps following the generation of class activation maps, we focus on the CAMs themselves by leveraging object attributes from the captions. Note that while the class tags are typically expected to be clean, *i.e.* require human curation, our cap-

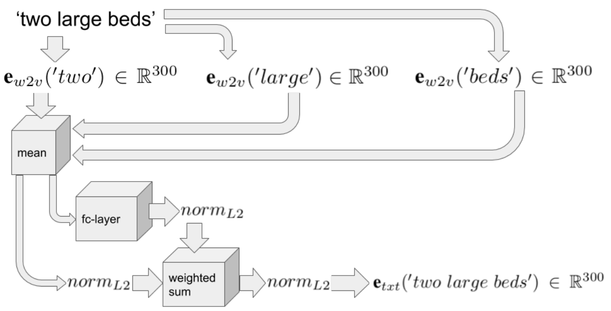

'two large beds'
$\mathbf{e}_{w2v}('two') \in \mathbb{R}^{300}$
$\mathbf{e}_{w2v}('large') \in \mathbb{R}^{300}$
$\mathbf{e}_{w2v}('beds') \in \mathbb{R}^{300}$
mean
fc-layer
$norm_{L2}$
$norm_{L2}$
weighted sum
$norm_{L2}$
$\mathbf{e}_{txt}('two\ large\ beds') \in \mathbb{R}^{300}$

Figure 2. To obtain the textual embedding of an arbitrary snippet of text, we first encode each word with a word2vec model and average over words, which gives us the input embedding. Feeding it into a single fully connected layer with a residual connection yields the output embedding

tions are not expected to mention all relevant classes. This is closer to the realistic scenario of retrieving information from the internet.

The task of weakly supervised visual grounding [6, 3, 49, 10, 9, 47] is related but a different task. Instead of training a network for semantic image segmentation, the goal is to localize a given phrase in an image and the challenge is to handle phrases that are not part of the training data. This means that they require a phrase for inference, while in our case the captions are only given for training but not for inference on the test dataset.

3. Method

Generating class activation maps [52] on training images is a crucial intermediate step for the majority of current state of the art weakly supervised image segmentation methods. In our work, we therefore focus on improving them. To this end, we harvest information from image captions instead of relying on class tags only. Our TAM-network is a multimodal architecture, which maps image pixels and text snippets into a common semantic space which allows us to calculate activation maps for class names as well as compound concepts that include the class name as illustrated in Figure 3. From TAMs of class names and class relevant compound concepts, we obtain class activation maps. From these CAMs, we directly estimate the pixelwise class labels as described in Section 3.3. We finally train the widely used VGG16-deeplab model [7] for semantic segmentation on the estimated labels which yields us the final segmentation model.

3.1. Parsing Captions

In our work, we distinguish three different types of text snippets: *Class names* contain the names of the classes of interest in the particular dataset as well as their plural form. *Compound concepts* are snippets of a sentence between beginning of sentence, prepositions, verbs and end of sentence. These snippets have to contain multiple words excluding articles. Essentially these are combinations of numbers, adjectives, adverbs and nouns like *two completely black dogs*. They are split into two categories: *Class related compound concepts* contain a class name inside them and *class unrelated compound concepts* do not. All of them are used during training of the TAM network, but the type of the snippet determines its weight in the loss. For CAM inference, we use class names and class related compound concepts. We use 300-dimensional word to vec (w2v) [28] embeddings to convert text snippets to numerical vectors of equal length. The embedding of a single word is given by the word to vec dictionary. For text snippets containing multiple words, we take the arithmetical mean of the normalized embeddings of individual words. These embeddings are used as input to the textual path of the TAM-network.

We use the classes mentioned in the captions to determine what classes are present in a training image. If the class name is present in at least one of the image captions, the class is considered as present, otherwise not. Note that in contrast to curated image tags, the captions do not necessarily contain all classes that are present in an image.

3.2. Multi-Modal TAM Network

Our TAM network comprises a visual path and a textual path which map visual and textual information into a common 300-dimensional semantic embedding space.

Visual Path. Our visual embedding path maps an image I with P pixels to a pixelwise visual embedding $\mathbf{E}_{vis}(I) \in \mathbb{R}^{P \times 300}$. It is a modification of the VGG16 architectures, but we will also report results for a ResNet38 architecture. In both cases, we change the output dimension of the last fully connected layer to the dimension of the common semantic embedding space.

Textual path. As shown in Figure 2, our textual path first obtains the word to vec embedding $\mathbf{e}_{w2v}(t) \in \mathbb{R}^{300}$ of a text snippet t by taking the average of the word to vec embeddings of the single words. Then $\mathbf{e}_{w2v}(t) \in \mathbb{R}^{300}$ is mapped to a vector $\mathbf{e}_{txt}(t) \in \mathbb{R}^{300}$ in the common semantic embedding space via:

$$\begin{aligned} \mathbf{e}_{txt}(t) =& norm_{L2}(norm_{L2}(\mathbf{e}_{w2v}(t)) \\ &+ w_{res} norm_{L2}(\mathbf{M}_{txt}\mathbf{e}_{w2v}(t))) \end{aligned} \tag{1}$$

where $norm_{L2}()$ denotes normalization by the L_2 norm, $\mathbf{M}_{txt} \in \mathbb{R}^{300 \times 300}$ is the weight matrix of a fully connected layer and $w_{res} \in \mathbb{R}$ is a hyperparameter. We also investigated RNNs that take the word to vec embeddings of the individual words as input and output $\mathbf{e}_{txt}$, but they performed slightly worse than our approach. This is probably due to short length of our text snippets which contain mostly 2 or 3 words.

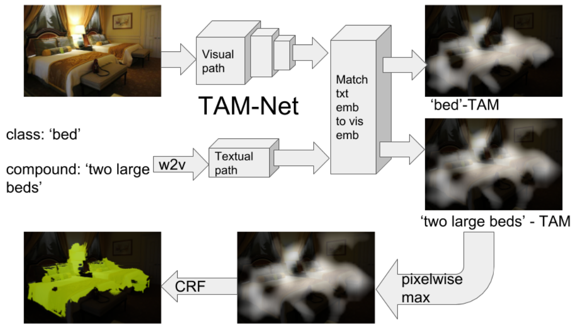


Figure 3. For each present class, we locate the class name as well as all compound concepts related to this class in the image (estimate their TAMs). We then normalize these TAMs and take for each class the pixelwise maximum over them to arive at the CAM. Finally, we estimate pixelwise class labels from these.

Textual Activation Map. Given an image I with P pixels and a text snippet t, we generate the textual activation map (TAM), which we denote by $\mathbf{x}(I,t) \in \mathbb{R}^P$, from the visual embedding $\mathbf{E}_{vis}(I) \in \mathbb{R}^{P\times 300}$ and the textual embedding $\mathbf{e}_{txt}(t) \in \mathbb{R}^{300}$ by:

$$\mathbf{x}(I,t) = \mathbf{E}_{vis}(I)\mathbf{e}_{txt}(t). \tag{2}$$

To obtain the normalized TAM, we apply $relu$ to discard negative values and normalize it by

$$\mathbf{x}_{norm}(I,t) = \frac{\sqrt{relu(\mathbf{x}(I,t))}}{\underset{p\in pixels}{max}\sqrt{relu(x(I,t,p))}}. \tag{3}$$

3.3. Direct Estimation of Pixelwise Class Labels from TAMs

Since we aim to learn a model for semantic segmentation, we need to estimate the pixelwise class labels for each training image I. To obtain these, we first calculate normalized class activation maps for all present classes as shown in Figure 3. To this end, for each class c mentioned in the captions of image I, we collect a set of text snippets $\Phi(c)$ which comprise the class name and all compound concepts related to it. Then we combine the normalized TAMs for all $t \in \Phi(c)$ into a normalized CAM $\mathbf{y}_{norm}(I,c) \in \mathbb{R}^P$ for class c by taking the pixelwise maximum over the TAMs:

$$y_{norm}(I,c,p) = \underset{t\in\Phi(c)}{max}\{x_{norm}(I,t,p)\} \tag{4}$$

We obtain the background activation map $\mathbf{b}(I)$ as in [1] by

$$b(I,p) = (1 - \underset{c\in C(I)}{max}\{y_{norm}(I,c,p)\})^{\alpha} \tag{5}$$

where $C(I)$ are the classes present in image I. We keep $\alpha = 4$ which is the value suggested in [1]. We then finally refine the normalized CAMs with a CRF [22] to estimate pixelwise class labels.

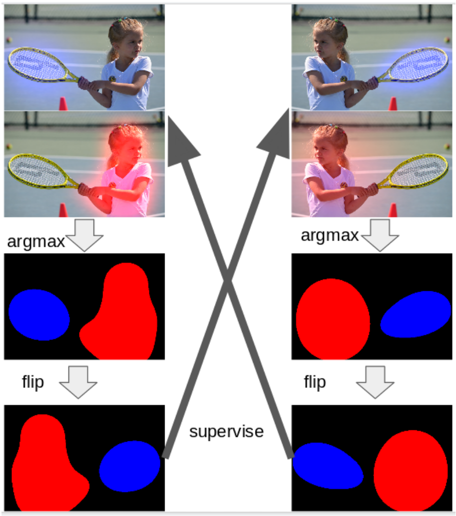


Figure 4. The auto consistency loss enforces invariance under geometric transformations like flipping. To this end, labels are estimated in an online manner from the TAMs of class names for the image and the flipped image. Then these are flipped and the estimated labels of the image are used to supervise the embedding of the flipped image and vice versa.

3.4. Training of Embedding Architecture

We finally describe how we train our network. For training, we propose the following loss

$$L = \lambda_{cls}L_{cls} + \lambda_{cpt}L_{cpt} + \lambda_{ac}L_{ac}. \tag{6}$$

The class loss L_{cls} and concepts loss L_{cpt} ensure that the textual activation maps have high values for classes and compounds that are present in a training image and low values if they are not present. The auto consistency loss L_{ac} ensures that the TAMs are geometric consistent if an image is vertically flipped as illustrated in Figure 4.

Class Loss. For each class c, we apply average pooling over the pixels of the TAM for its class name t_c, *i.e.*, $x_{pool}(I,t_c) = \frac{1}{P}\sum_p \mathbf{x}(I,t_c)$. These values can be seen as logits for the probability of this class to be present in the image. We use average pooling as in [17, 5, 1] since it typically leads to activation maps that cover the complete class and not only its most distinctive areas as it is the case for max pooling. The class loss L_{cls} is then given by the multi-label binary cross entropy loss:

$$\begin{aligned} L_{cls} = &-\sum_{c\in C(I)} log\left(\frac{1}{1+e^{-x_{pool}(I,t_c)}}\right) \\ &-\sum_{c\notin C(I)} log\left(\frac{e^{-x_{pool}(I,t_c)}}{1+e^{-x_{pool}(I,t_c)}}\right). \end{aligned} \tag{7}$$

The loss is minimized if the TAMs for the present classes show a strong signal while the TAMs of the classes that are not present in the image are close to zero.

Concepts Loss. The concepts loss is calculated in the same way as the class loss, except that we use the TAMs $\mathbf{x}(I,t)$ of compound concepts instead of using TAMs of class names. As for the class loss, we apply average pooling $x_{pool}(I,t) = \frac{1}{P}\sum_p \mathbf{x}(I,t)$. While we use the multilabel binary cross entropy loss as for the class loss, we have to subsample the missing concepts since there are otherwise too many. The concepts loss is then given by

$$\begin{aligned} L_{cpt} = &- \sum_{t \in Comp(I)} log\left(\frac{1}{1+e^{-x_{pool}(I,t)}}\right) \\ &- \sum_{t \in ContrComp(I)} log\left(\frac{e^{-x_{pool}(I,t)}}{1+e^{-x_{pool}(I,t)}}\right) \end{aligned} \tag{8}$$

where $Comp(I)$ are the compound concepts present in image I and $ContrComp(I)$ are contrastive compound concepts that are randomly sampled from other images.

Auto-Consistency Loss. The purpose of the auto-consistency loss is to ensure that geometric transformations of the image do not alter the accordingly transformed TAMs as illustrated in Figure 4. We use the image I and the flipped image I_f for training and obtain the corresponding TAMs $\mathbf{x}(I,t_c)$ and $\mathbf{x}(I_f,t_c)$ for all classes. Note that there is no TAM for the background class, we therefore set $\mathbf{x}(I,t_{bg}) = 0$ for the background class. We convert them into pixel-wise class probabilities $\mathbf{Z}$ and $\mathbf{Z}_f$ by applying the softmax for each pixel p, *i.e.*, $\mathbf{Z}(p,c) = \text{softmax}_{c'}(\mathbf{x}(I,t_{c'},p))$.

We also compute a pixel-wise labeling for $\mathbf{x}(I,t_c)$ and $\mathbf{x}(I_f,t_c)$ as in Section 3.3 but without CRF. Instead, we simply use the class with the highest activation per pixel

$$\hat{c}(I,p) = \underset{c \in \{C(I) \cup bg\}}{argmax}\ x_{norm}(I,t_c,p) \tag{9}$$

where the background activation is estimated as in (5). We finally mirror the pixel-wise segmentations as shown in Figure 4. The auto-consistency loss L_{ac} is then given by

$$\begin{aligned} L_{ac}(I) = &-\frac{1}{2}\sum_{p \in pixel} log\left(\frac{1}{1+e^{-Z_f(p,\hat{c}(I,p))}}\right) \\ &-\frac{1}{2}\sum_{p \in pixel} log\left(\frac{1}{1+e^{-Z(p,\hat{c}_f(I,p))}}\right). \end{aligned} \tag{10}$$

4. Experiments

For our experiments, we use the COCO dataset [26], since it provides several captions for each image and instance level segmentations for 80 object classes which we convert to class level segmentations. As train set we use the COCO train2014 split containing 83k images. To evaluate the final semantic segmentation models, we use the COCO val2014 split containing 40k images as our test set. Our evaluation metric is intersection over union averaged over 81 classes (80 object classes and background).

Implementation Details. The visual paths of our VGG16 and ResNet38 architecture is identical to the respective architectures from [1] up to the last layer. While we use mostly the VGG architecture for our experiments, we show some results using the ResNet at the end. We train the VGG architecture as well as the ResNet architecture for 15 epochs. For VGG, the learning rate is 0.1 for weights and 0.2 for biases, for ResNet it is 0.01 and 0.02, respectively. For VGG the batch size is 16, for ResNet it is 8. Weight decay equals 0.0005 for both architectures. The first two convolutional blocks are not finetuned at all and for the fc8 layer and the textual path, we scale up the learning rate by 10. During training, the learning rate decays to 0 according to the polynomial policy. The data augmentation techniques include random scaling, cropping and mirroring. We set $\lambda_{cls} = 1.0$, $\lambda_{cpt} = 0.3$ and $\lambda_{ac} = 0.001$ so that each loss term is roughly in the same order of magnitude.

For the concepts loss (8), we sample a maximum of 10 compound concepts mentioned in the captions of an image. To collect contrastive compound concepts which are absent in the image, we first sample 10 random images and extract the compound concepts from their captions. Then we randomly sample 50 of these concepts.

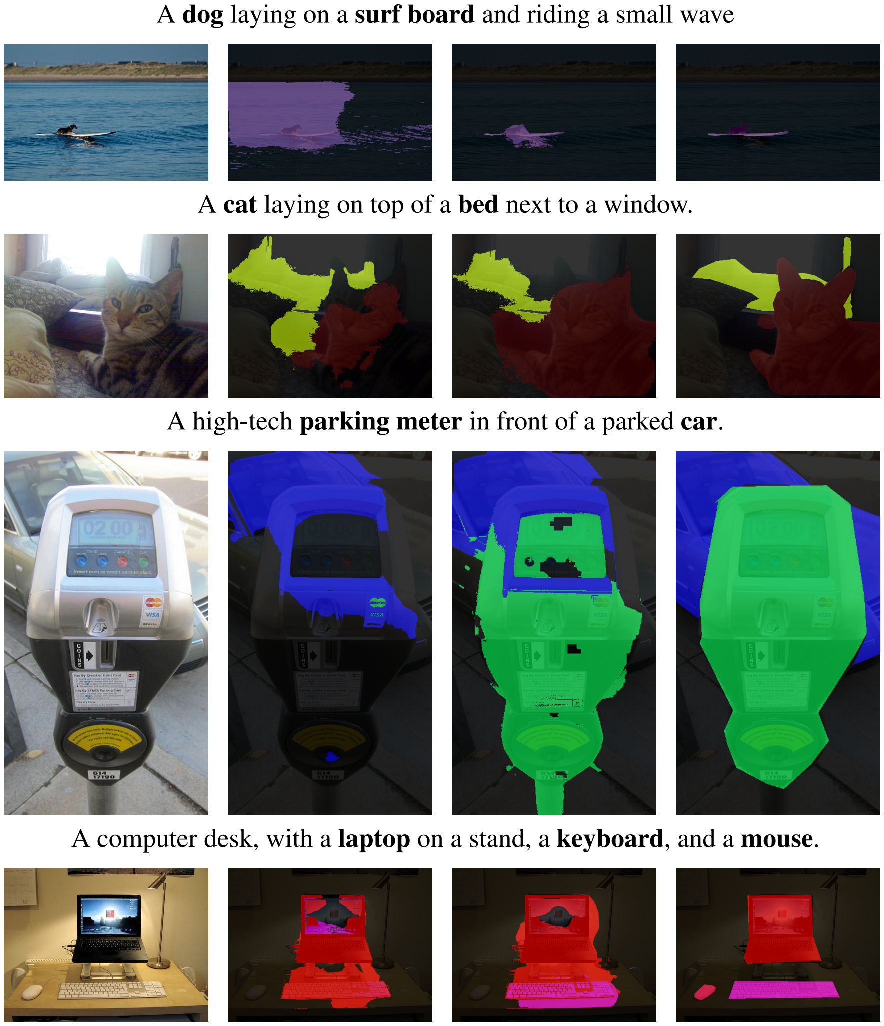
For semantic segmentation we use the baseline VGG16 deeplab model [7]. We keep the hyperparameters but increase the number of iterations by a factor of 3 to account for the bigger size of COCO as compared to Pascal VOC2012.

4.1. Evaluation of System Components

For our ablation studies, we first evaluate the accuracy of the pixel labels that are estimated on the training images from the captions as described in Section 3.3. Simply using a one-hot encoding of the classes retrieved from captions instead of the textual path gives an IoU of 14.6% as can be seen in Table 1. If we use only the class loss L_{cls}, the accuracy is similar to the baseline. Using the concepts loss L_{cpt}, however, improves mean IoU substantially to 18.5%. While the recall increases, the precision decreases. This is expected since the compound concepts provide mainly the textual context and are less class focused. Using both loss functions, alleviates this effect and improves the mean IoU to 19.9%. Including auto consistency during training leads to further improvement from 19.9% to 20.3%. We show some qualitative results in Figure 5. If we use a ResNet38 architecture instead of a VGG16 architecture, our results improve to 30.5% IoU.

4.2. Comparison to Visual Grounding

We also compare our approach for generating CAMs with the state of the art approach for weakly supervised visual grounding [10]. The authors use complete image captions from the COCO dataset as supervision for train-


A **dog** laying on a **surf board** and riding a small wave

A **cat** laying on top of a **bed** next to a window.

A high-tech **parking meter** in front of a parked **car**.

A computer desk, with a **laptop** on a stand, a **keyboard**, and a **mouse**.

Figure 5. Examples of estimated pixelwise class labels on the training set. From left to right: Image, baseline, proposed method, ground truth. Above each image we provide the caption and highlight the class names of the COCO classes.

Results on training set							
method	TAM-Net	L_{cls}	L_{cpt}	L_{ac}	precision	recall	IoU
baseline class tags from captions	VGG16				0.370	0.207	0.146
baseline ground-truth class tags	VGG16				0.337	0.230	0.158
prop. L_{cls}	VGG16	√			0.378	0.205	0.144
prop. L_{cpt}	VGG16		√		0.288	0.383	0.185
prop. $L_{cls} + L_{cpt}$	VGG16	√	√		0.382	0.316	0.199
prop. $L_{cls} + L_{cpt} + L_{ac}$	VGG16	√	√	√	0.383	0.329	0.203
prop. $L_{cls} + L_{cpt} + L_{ac}$	ResNet38	√	√	√	0.468	0.509	0.305
VisGround[10]	ResNet151				0.375	0.432	0.231

Table 1. Results for estimated pixel labels on the training set.

ing. During inference, this model receives an image and a single text snippet referring to an object or to stuff in this image and returns an activation map for this snippet. To segment the image region the snippet refers to, the authors suggest to threshold the activation map with the average of the minimum and maximum value of this map. We can not use this method out of the shelf since locating multiple text snippets simultaneously is not intended for visual grounding and there is no policy to select the label of a pixel if it is assigned to multiple segments. To adapt the model to our setting and estimate pixelwise class labels for the training images, we first generate the activation maps for the classes retrieved from the captions. Then, we subtract from each activation map the average of its minimum and maximum

Impact of different types of concepts							
type of concepts	cls. rel. comp. cpt.	other comp. cpt.	no adj.	sgl. adj+nouns	precision	recall	IoU
cls. rel. only	√				0.404	0.306	0.199
all comp. conc.	√	√			0.382	0.316	0.199
all concepts	√	√		√	0.383	0.306	0.193
no adjectives	√	√	√		0.380	0.283	0.183

Table 2. Using compound concepts only for training leads to best results. Results are reported without auto-consistency loss.

Performance dependence on w_{res}			
size of w_{res}	precision	recall	IoU
0.0	0.395	0.287	0.190
0.2 (prop.)	0.382	0.316	0.199
0.4	0.277	0.494	0.201
0.6	0.255	0.523	0.194

Table 3. Small values of w_{res} allow the textual path to adjust the w2v embeddings to the needs of visual grounding, while large values lead to degeneration during training and inferior performance. Results are reported without auto-consistency loss.

Performance dependence on λ_{cpt}			
size of λ_{cpt}	precision	recall	IoU
0.1	0.362	0.301	0.186
0.3 (prop.)	0.382	0.316	0.199
0.5	0.396	0.319	0.204

Table 4. Performance grows when the impact of compound concepts is increased. Results are reported without auto-consistency loss.

value. Finally, we set the activations for background to 0 and apply argmax over the classes. This yields 23.1% IoU as shown in Table 1. This is better than our results for the VGG16 architecture but far inferior to our ResNet38 architecture, although the visual grounding model uses a deeper ResNet151 architecture.

4.3. Evaluating Concept Types

We compare the performance of our system when using different types of sentence snippets during training and report the results in Table 2. In our proposed method, all compound concepts are fed into the concept loss. Conceptually, adjectives provide additional cues during class activation map calculation. If we discard the adjectives, the accuracy decreases as shown in Table 2. Using only compounds containing COCO class names increases the precision but reduces the recall, the IoU remains unchanged. We therefore use all compound concepts from the captions in all other experiments. Feeding all nouns and adjectives additionally to the compounds into the loss decreases the performance. This shows that the compound concepts better encapsulate the context of the captions than single words.

4.4. Evaluating Textual Embedding

The hyperparameter w_{res} (1) controls the extent to which w2v embeddings [28] are adjusted in the textual path. Intuitively high values add flexibility to the network and allow the textual path to adjust the w2v embeddings to the task at hand. Table 3 reports the results without the auto-consistency loss. As can be seen, increasing w_{res} from 0 improves the performance with a peak at 0.4. However, for higher values, the performance decreases again. This is because higher flexibility allows the network to find degenerate solutions: If w_{res} becomes too high, the textual path maps all class names and compounds appearing frequently to one vector v and all other class names and compounds to the negative vector $-v$. The visual path then maps everything to v.

4.5. Evaluating Concept Loss Weight

In Table 4, we also evaluate the impact of the parameter λ_{cpt} which weights the concept loss in (6). As in the previous experiment, we do not use the auto consistency loss. Even a small concept loss already improves the baseline approach giving 18.6%. By further increasing the weight of the concepts to 0.5, the performances raises to 20.4%.

4.6. Comparison to Ground-Truth Image Tags

The class tags we get from the captions are not perfect as can be seen from Table 5. Although for some classes the recall is very low, using parsed tags instead of clean tags does not harm the weakly supervised segmentation performance significantly. Training the baseline model with clean tags instead of retrieved tags improves the mean IoU from 14.6% to 15.8% as shown in Table 1. Surprisingly, the precision of CAMs obtained with clean tags is even smaller than with retrieved tags. It seems that COCO object classes not mentioned in the captions are more difficult to locate since captions only mention the most important objects which tend to be large. If an approach for weakly supervised image segmentation fails to locate an object that is expected to be present in an image, the precision decreases. It is remarkable that the gain from captions is substantially higher than the gain from ground-truth class tags, which shows that the captions provide more information than just tags.

4.7. Results on Test Set

To demonstrate the effect of improved CAMs on the final segmentation results, we train deeplab [7] for semantic segmentation on the estimated pixel-wise labels and evaluate its performance on the test set. We first evaluate the impact

Class tag retrieval precision and recall		
class type	precision	recall
person and accessory	95.0%	33.0%
vehicles	89.5%	63.6%
outdoor	95.4%	56.0%
animal	87.1%	92.3%
sport	96.2%	64.2%
kitchenware	89.4%	20.3%
food	85.9%	68.7%
furniture	90.3%	44.0%
electronics	92.3%	48.3%
appliance	79.8%	46.0%
indoor	97.0%	43.7%

Table 5. Recall and precision of image class tags retrieved from image captions.

Results on the test set.		
method	TAM-Net	IoU test set
Baseline gt class tags no CRF	VGG16	0.161
Proposed no CRF	VGG16	0.210
Proposed	VGG16	0.216
Proposed	ResNet38	0.285

Table 6. Results of the final semantic segmentation model on the test set. The gain in accuracy of estimated pixel labels on the train set transfers well to the test set.

Comparison to DSRG[17]	
method	IoU test set
DSRG[17]	0.260
Proposed (VGG16)	0.216
Proposed (VGG16) + DSRG[17]	0.269
Proposed (ResNet38)	**0.285**
Proposed (ResNet38)+DSRG[17]	0.277

Table 7. DSRG[17] is a saliency based approach for weakly supervised semantic segmentation. It can be combined with our approach.

Comparison to the state of the art	
method	IoU test set
BFBP[37]	0.204
SEC[21]	0.224
DSRG[17]	0.260
VisGround[10] adapted	0.275
Proposed	**0.285**

Table 8. Comparison of the final semantic segmentation model with the state of the art on the test set.

of the CRF and compare the results obtained by the baseline approach with ground truth class tags to our proposed approach using VGG16 and ResNet38. The results reported in Table 6 indicate that the quality of the estimated labels is a strong predictor for the quality of the final segmentation. The performance gap of 4.5% on the train set between the baseline and the proposed method results in 4.9% on the test set. Applying a CRF on top of the test set prediction gives a slight improvement of 0.6%. We use a CRF in all remaining experiments. If we use a ResNet38 architecture for our TAM network, we obtain 28.5%.

We also compare our method to deep seeded region growing (DSRG) [17], which is a state of the art weakly supervised semantic segmentation method and it achieves 26.0% IoU on COCO. To train their model, the authors take CAMs and background cues from a strongly supervised saliency model as input. We can therefore combine this approach with our method by feeding our generated CAMs to DSRG. We report the results in Table 7. For VGG16, DSRG improves IoU from 21.6% to 26.9%. This is also a better result than the number reported by the authors demonstrating the benefit of our CAMs. For ResNet38, however, DSRG decreases the accuracy of our approach. A possible explanation is that DSRG reduces the information from CAMs by estimating high confidence regions first and expands them using the saliency maps. If the CAMs are inaccurate, DSRG substantially improves the results. However, if the CAMs become more accurate, DSRG discards too much information from the CAMs.

We finally compare our approach with other approaches for weakly supervised semantic segmentation. As can be seen from Table 8, our approach outperforms the other weakly supervised semantic segmentation models as reported by [17]. We also include our adapted version of the visual grounding approach of Engilberge et al. [10]. Interestingly, this approach also achieves a higher IoU than [17], which shows the rich information that is present in image captions. Nevertheless, our approach achieves the highest IoU despite of using a ResNet38 instead of a deeper ResNet151.

5. Conclusion

We presented an approach that uses image captions as supervision for weakly supervised semantic image segmentation. Inspired by weakly supervised approaches that estimate class activation maps from class tags and deduce localization cues from them, our approach estimates text activation maps for the class names as well as compound concepts and fuses them to obtain better class activation maps. We evaluated our approach on the COCO dataset and demonstrated that our approach outperforms the state of the art for weakly supervised image segmentation.

Acknowledgement The work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) GA 1927/5-1 (FOR 2535 Anticipating Human Behavior) and the ERC Starting Grant ARCA (677650).

References

[1] J. Ahn and S. Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic

segmentation. *CVPR*, 2018. 1, 2, 4, 5

[2] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the Point: Semantic Segmentation with Point Supervision. *ECCV*, 2016. 2

[3] G. Bouritsas, P. Koutras, A. Zlatintsi, and P. Maragos. Multimodal visual concept learning with weakly supervised techniques. *CVPR*, 2018. 3

[4] R. Briq, M. Moeller, and J. Gall. Convolutional simplex projection network for weakly supervised semantic segmentation. *BMVC*, 2018. 2

[5] A. Chaudhry, P. K. Dokania, and P. H. S. Torr. Discovering class-specific pixels for weakly-supervised semantic segmentation. *BMVC*, 2017. 2, 4

[6] K. Chen, J. Gao, and R. Nevatia. Knowledge aided consistency for weakly supervised phrase grounding. *CVPR*, 2018. 3

[7] L. Chen, 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*, 2018. 2, 3, 5, 7

[8] M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. 2

[9] T. Durand, T. Mordan, N. Thome, and M. Cord. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. *CVPR*, 2017. 3

[10] M. Engilberge, L. Chevallier, P. Prez, and M. Cord. Finding beans in burgers: Deep semantic-visual embedding with localization. *CVPR*, 2018. 3, 5, 6, 8

[11] 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*, 2015. 2

[12] R. Fan, Q. Hou, M.-M. Cheng, G. Yu, R. R. Martin, and S.-M. Hu. Associating inter-image salient instances for weakly supervised semantic segmentation. *ECCV*, 2018. 2

[13] W. Ge, S. Yang, and Y. Yu. Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. *CVPR*, 2018. 2

[14] S. Hong, D. Yeo, S. Kwak, H. Lee, and B. Han. Weakly supervised semantic segmentation using web-crawled videos. *CVPR*, 2017. 2

[15] Q. Hou, M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr. Deeply supervised salient object detection with short connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 2

[16] Q. Hou, D. Massiceti, P. K. Dokania, Y. Wei, M.-M. Cheng, and P. H. S. Torr. Bottom-up top-down cues for weakly-supervised semantic segmentation. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2018. 2

[17] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. *CVPR*, 2018. 2, 4, 8

[18] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. *CVPR*, 2013. 2

[19] B. Jin, M. V. O. Segovia, and S. Süsstrunk. Webly supervised semantic segmentation. *CVPR*, 2017. 2

[20] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation. *CVPR*, 2017. 2

[21] A. Kolesnikov and C. H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. *ECCV*, 2016. 1, 2, 8

[22] P. Krahenbuhl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*. 4

[23] Q. Li, A. Arnab, and P. H. Torr. Weakly- and semi-supervised panoptic segmentation. *ECCV*, 2018. 2

[24] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. *CVPR*, 2016. 2

[25] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multipath refinement networks for high-resolution semantic segmentation. *CVPR*, 2017. 2

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. *ECCV*, 2014. 2, 5

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *CVPR*, 2015. 2

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *NIPS*, 2013. 3, 7

[29] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele. Exploiting saliency for object segmentation from image level labels. *CVPR*, 2017. 2

[30] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. *ICCV*, 2015. 2

[31] D. Pathak, P. Krähenbühl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. *ICCV*, 2015. 2

[32] P. H. O. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. *CVPR*, 2015. 2

[33] J. Pont-Tuset, P. Arbelez, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2

[34] X. Qi, Z. Liu, J. Shi, H. Zhao, and J. Jia. Augmented feedback in semantic segmentation under image level supervision. *ECCV*, 2016. 2

[35] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 2004. 2

[36] A. Roy and S. Todorovic. Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation. *CVPR*, 2017. 2

[37] F. Saleh, M. S. A. Akbarian, M. Salzmann, L. Petersson, S. Gould, and J. M. Alvarez. Built-in foreground/background

prior for weakly-supervised semantic segmentation. *ECCV*, 2016. 8

[38] J. Sawatzky, A. Srikantha, and J. Gall. Weakly supervised affordance detection. *CVPR*, 2017. 2

[39] W. Shimoda and K. Yanai. Distinct class-specific saliency maps for weakly supervised semantic segmentation. *ECCV*, 2016. 2

[40] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers. Normalized cut loss for weakly-supervised cnn segmentation. *CVPR*, 2018. 2

[41] M. Tang, F. Perazzi, A. Djelouah, I. Ben Ayed, C. Schroers, and Y. Boykov. On regularized losses for weakly-supervised cnn segmentation. *ECCV*, 2018. 2

[42] A. Vezhnevets, V. Ferrari, and J. Buhmann. Weakly supervised structured output learning for semantic segmentation. *CVPR*, 2012. 2

[43] X. Wang, S. You, X. Li, and H. Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. *CVPR*, 2018. 2

[44] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. *CVPR*, 2017. 2

[45] Y. Wei, X. Liang, Y. Chen, X. Shen, M. Cheng, J. Feng, Y. Zhao, and S. Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2

[46] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang. Revisiting dilated convolution: A simple approach for weakly- and semi-supervised semantic segmentation. *CVPR*, 2018. 2

[47] F. Xiao, L. Sigal, and Y. J. Lee. Weakly-supervised visual grounding of phrases with linguistic structures. *CVPR*, 2017. 3

[48] J. Zhang, Z. Lin, S. X. Brandt, Jonathan, and S. Sclaroff. Top-down neural attention by excitation backprop. *ECCV*, 2016. 2

[49] F. Zhao, J. Li, J. Zhao, and J. Feng. Weakly supervised phrase localization with multi-scale anchored transformer network. *CVPR*, 2018. 3

[50] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *CVPR*, 2017. 2

[51] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. *ICCV*, 2015. 2

[52] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. Learning deep features for discriminative localization. *CVPR*, 2016. 1, 2, 3