Adversarial Complementary Learning for Weakly Supervised Object Localization

Xiaolin Zhang[1] Yunchao Wei[2] * Jiashi Feng[3] Yi Yang[1] Thomas Huang[2]
[1]CAI, University of Technology Sydney [2]University of Illinois Urbana-Champaign
[3]National University of Singapore
{Xiaolin.Zhang-3@student, Yi.Yang@}uts.edu.au elefjia@nus.edu.sg
{yunchao,t-huang1}@illinois.edu

Abstract

*In this work, we propose **A**dversarial **Co**mplementary **L**earning (ACoL) to automatically localize integral objects of semantic interest with weak supervision. We first mathematically prove that class localization maps can be obtained by directly selecting the class-specific feature maps of the last convolutional layer, which paves a simple way to identify object regions. We then present a simple network architecture including two parallel-classifiers for object localization. Specifically, we leverage one classification branch to dynamically localize some discriminative object regions during the forward pass. Although it is usually responsive to sparse parts of the target objects, this classifier can drive the counterpart classifier to discover new and complementary object regions by erasing its discovered regions from the feature maps. With such an adversarial learning, the two parallel-classifiers are forced to leverage complementary object regions for classification and can finally generate integral object localization together. The merits of ACoL are mainly two-fold: 1) it can be trained in an end-to-end manner; 2) dynamically erasing enables the counterpart classifier to discover complementary object regions more effectively. We demonstrate the superiority of our ACoL approach in a variety of experiments. In particular, the Top-1 localization error rate on the ILSVRC dataset is 45.14%, which is the new state-of-the-art.*

1. Introduction

Weakly Supervised Object Localization (WSOL) refers to learning object locations in a given image using the image-level labels. Currently, WSOL has drawn increasing attention since it does not require expensive bounding box annotations for training and thus can save much labour compared to fully-supervised counterparts [32, 13, 12].

*Corresponding Author

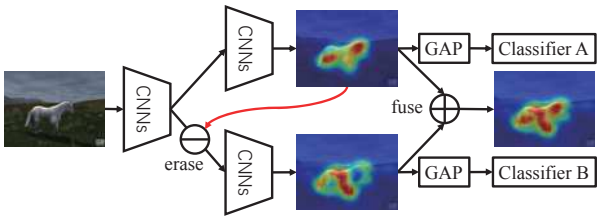


Figure 1: An illustration of the proposed ACoL method. We prove object localization maps can be conveniently obtained during feed-forward pass. Based on this, we design the parallel adversarial classifier architecture, where complementary regions (the head and hind legs *vs.* forelegs) are discovered by two classifiers (A and B) via adversarial erasing feature maps. GAP refers to global average pooling.

It is a very challenging task to learn deep models for locating objects of interest using only image-level supervision. Some pioneer works [48, 45] have been proposed to generate class-specific localization maps according to pre-trained convolutional classification networks. For example, Zhou *et al*. [48] modified classification networks (*e.g*., AlexNet [21] and VGG-16 [34]) via replacing a few high-level layers by a global average pooling layer [23] and a fully connected layer, which can aggregate the features of the last convolutional layer to generate discriminative class activation maps (CAM) for the localization purpose. However, we observe that some critical issues exist in such solutions, mainly including: 1) over-relying on category-wise discriminative features for image classification; 2) failing to localize integral regions of the target objects densely within an image. The two issues are mainly due to the classification networks are inclined to identify patterns from the most discriminative parts for recognition, which inevitably leads to the second issue. For instance, given an image containing a cat, the network can recognize it by identifying the head regardless of the remaining parts such as body and legs.

To tackle such issues, Wei *et al*. [39] proposed an ad-

versarial erasing (AE) approach to discover integral object regions by training additional classification networks on images whose discriminative object regions have partially been erased. Nevertheless, one main disadvantage of AE is that it needs to train several independent classification networks for obtaining integral object regions, which costs more training time and computing resources. Recently, Singh *et al*. [35] enhanced CAM by randomly hiding the patches of input images so as to force the network to look for other discriminative parts. However, randomly hiding patches without any high-level guidance is inefficient and cannot guarantee that networks always discover new object regions.

In this paper, we propose a novel **A**dversarial **Co**mplementary **L**earning (ACoL) approach for discovering entire objects of interest via end-to-end weakly supervised training. The key idea of ACoL is to find the complementary object regions by two adversary classifiers motivated by AE [39]. In particular, one classifier is firstly leveraged to identify the most discriminative regions and guide the erasing operation on the intermediate feature maps. Then, we feed the erased features into its counterpart classifier for discovering new and complementary object-related regions. Such a strategy drives the two classifiers to mine complementary object regions and finally obtain integral object localization as desired. To easily conduct end-to-end training for ACoL, we mathematically prove that object localization maps can be obtained by directly selecting from the class-specific feature maps of the last convolutional layer, rather than using a post-inference manner in [48]. Thus discriminative object regions can be identified in a convenient way during the training forward pass according to the online inferred object localization maps.

Our approach offers multiple appealing advantages over AE [39]. First, AE trains three networks independently for adversarial erasing. ACoL trains two adversarial branches jointly by integrating them into a single network. The proposed joint training framework is more capable of integrating the complementary information among the two branches. Second, AE adopts a recursive method to generate localization maps, and it has to forward the networks for multiple times. Instead, our method generates localization map by forwarding the network only once. This advantage greatly improves the efficiency and have our method much easier for implementation. Third, AE directly adopts CAM [48] to generate localization maps. Thus AE generates localization maps in two steps. Differently, our method generates localization maps in one step, by selecting the feature map which best matches the groundtruth as the localization map. We have also provided detailed proof with theoretical rigor that our method is simpler and more efficient, but yields identical results to CAM [48] (see Section 3.1).

The process of ACoL is illustrated in Figure 1, where an image is processed to estimate the regions of a horse. We can observe that Classifier A leverages some discriminative regions (the horse's head and hind legs) for recognition. By erasing such discriminative regions in feature maps, Classifier B is guided to use features of new and complementary object regions (the horse's forelegs) for classification. Finally, the integral target regions are obtained by fusing the object localization maps from both branches. To validate the effectiveness of the proposed ACoL, we conduct a series of object localization experiments using the bounding boxes inferred from the generated localization maps.

To sum up, our main contributions are three-fold:

- We provide theoretical support of producing class-specific feature maps during the forward pass, so that object regions can be simply identified in a convenient way, which can benefit future relevant researches.
- We propose a novel ACoL approach to efficiently mine different discriminative regions by two adversary classifiers in a weakly supervised manner, which discover integral target regions of objects for localization.
- This work achieves the current state-of-the-art with the error rate of Top-1 45.14% and Top-5 30.03% on the ILSVRC 2016 dataset in weakly supervised setting.

2. Related Work

Fully supervised detection has been intensively studied and achieved extraordinary successes. One of the earliest deep networks to detect objects in a one-stage manner is OverFeat [32], which employs a multiscale and sliding window approach to predict object boundaries. These boundaries are then applied for accumulating bounding boxes. SSD [25] and YOLO [28] use a similar one-stage method, and they are specifically designed for speeding up the detection. Faster-RCNN [29] utilize a novel two-stage approach and has achieved great success in the object detection. It generates region proposals using sliding windows and predicts highly reliable object locations in a unified network in real time. Lin *et al*. [24] presented that the performance of Faster-RCNN can be significantly improved by constructing feature pyramids with marginal extra cost.

Weakly supervised detection and localization aims to apply an alternative cheaper way by only using image-level supervision [2, 35, 1, 38, 30, 19, 10, 9, 18, 22, 26]. Oquab *et al*. [26] and Wei *et al*. [42] adopted a similar strategy to learn multi-label classification networks with max-pooling MIL. The networks are then applied to coarse object localization [26]. Bency *et al*. [2] applied a beam search method to leverage local spatial patterns, which progressively localizes bounding box candidates. Singh *et al*. [35] proposed a method to augment the input images by randomly hiding patches so as to look for more object regions. Similarly, Bazzani *et al*. [1] analysed the scores of a classification network by randomly masking regions of input images

and proposed a clustering technique to generate self-taught localization hypotheses. Deselaers *et al*. [7] used extra images with available location annotations to learn object features and then applied a conditional random field to generally adapt the generic knowledge to specific detection tasks.

Weakly supervised segmentation applies similar techniques to predict pixel-level labels [40, 41, 39, 16, 20, 27, 43]. Wei *et al*. [40] utilized extra images with simple scenes and proposed a simple to complex approach to progressively learn better pixel annotations. Kolesnikov *et al*. [20] proposed SEC that integrates three loss functions *i.e*., seeding, expansion and boundary constrain, into a unified framework to learn a segmentation network. Wei *et al*. [39] proposed a similar idea as ours to find more discriminative regions, they trained extra independent networks for generating class-specific activation maps with the assistance of the pre-trained networks in a post-processing step.

3. Adversarial Complementary Learning

In this section, we describe details of the proposed Adversarial Complementary Learning (ACoL) approach for WSOL. We first revisit CAM [48] and introduce a more convenient way for producing localization maps. Then, the details of the proposed ACoL, founded on the above finding, are presented for mining high-quality object localization maps, and locating integral object regions.

3.1. Revisiting CAM

Object localization maps have been widely used in many tasks [26, 40, 1, 45], offering a promising way to visualize where deep neural networks focus on for recognition. Zhou *et al*. [48] proposed a two-step approach which can produce object localization maps by multiplying the weights from the last fully connected layer to feature maps in a classification network.

Suppose we are given a Fully Convolutional Network (FCN) with last convolutional feature maps denoted as $S \in \mathbb{R}^{H\times H\times K}$, where $H \times H$ is the spatial size and K is the number of channels. In [48], the feature maps are fed into a Global Average Pooling (GAP) [23] layer followed by a fully connected layer. A softmax layer is applied on the top for classification. We denote the average value of the k_{th} feature map as $s_k = \frac{\sum_{i,j}(S_k)_{i,j}}{H\times H}, k = 0, 1, ..., K-1$, where $(S_k)_{i,j}$ is the element of the k_{th} feature map S_k at the i_{th} row and the j_{th} column. The weight matrix of the fully connected layer is denoted as $W^{fc} \in \mathbb{R}^{K\times C}$, where C is the number of target classes. Here, we ignore the bias term for convenience. Therefore, for the target class c, the input of the c_{th} softmax node y_c^{fc} can be defined as

$$y_c^{fc} = \sum_{k=0}^{K-1} s_k W_{k,c}^{fc}, \tag{1}$$

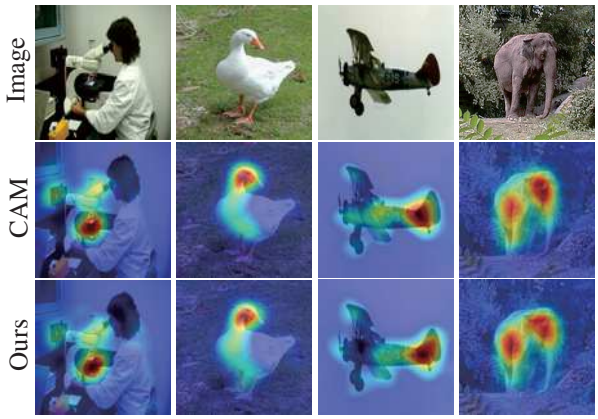


Figure 2: Comparison of methods for generating localization maps. Our method can produce the same-quality maps as CAM [48] but in a more convenient way.

where $W_{k,c}^{fc} \in \mathbb{R}$ denotes the element of the matrix W^{fc} at the k_{th} row and the c_{th} column. The row $W_{k,c}^{fc}, k = 0, 1, ..., K-1$ contributes to calculating the value y_c^{fc}. Therefore, the object localization map A_c^{fc} of class c proposed in [48] can be obtained by aggregating the feature map S as follows,

$$A_c^{fc} = \sum_{k=0}^{K-1} S_k \cdot W_{k,c}^{fc}. \tag{2}$$

CAM provides a useful way to inspect and locate the target object regions, but it needs an extra step to generate object localization maps after the forward pass. In this work, we reveal that object localization maps can be conveniently obtained by directly selecting from the feature maps of the last convolutional layer. Recently, some methods [17, 4] have already obtained localization maps like this, but we are the first to prove this convenient approach can generate same-quality localization maps with CAM, which is meaningful and contributes to embedding localization maps into complex networks. In the following, we provide both theoretical proof and visualized comparison to support our discovery. Given the output feature maps S of an FCN, we add a convolutional layer of C channels with the kernel size of 1×1, stride 1 on top of the feature maps S. Then, the output is fed into a GAP layer followed by a softmax layer for classification. Suppose the weight matrix of the 1×1 convolutional layer is $W^{conv} \in \mathbb{R}^{K\times C}$. We define the localization maps $A_c^{conv}, c = 0, 1, ..., C-1$ as the output feature maps of the 1×1 convolutional layer and A_c^{conv} can be calulated by

$$A_c^{conv} = \sum_{k=0}^{K-1} S_k \cdot W_{k,c}^{conv}, \tag{3}$$

where $W_{k,c}^{conv} \in \mathbb{R}$ denotes the element of the matrix W^{conv} at the k_{th} row and the c_{th} column. Therefore, the c_{th} in-

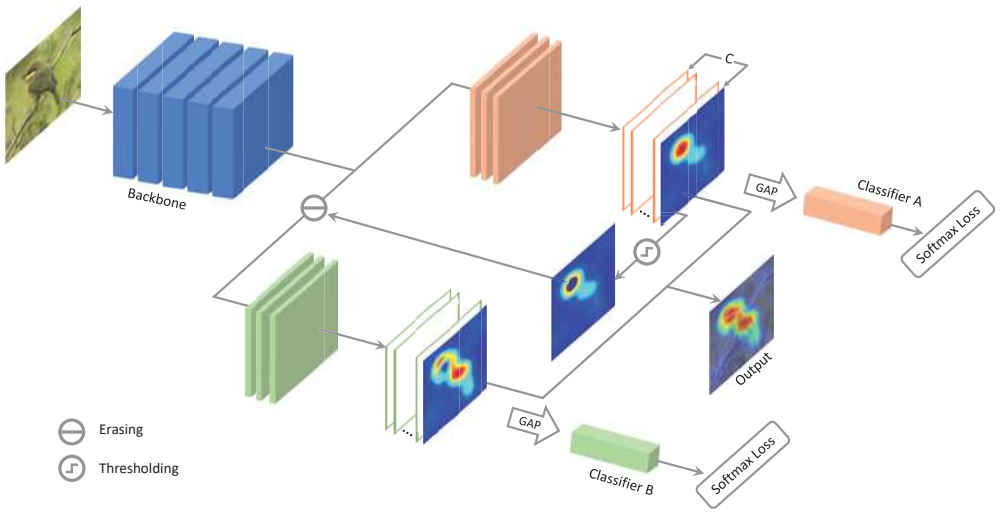


Figure 3: Overview of the proposed ACoL approach. The input images are processed by Backbone to extract mid-level feature maps, which are then fed into two parallel-classifiers for discovering complementary object regions. Each classifier consists of several convolutional layers followed by a global average pooling (GAP) layer and a softmax layer. Different from Classifier A, the input feature maps of Classifier B are erased with the guidance of the object localization maps from Classifier A. Finally, the object maps from the two classifiers are fused for localization.

put value y_c^{conv} of the softmax layer is the average value of A_c^{conv}. So, y_c^{conv} can be calculated by

$$y_c^{conv} = \frac{\sum_{i,j}(A_c^{conv})_{i,j}}{H \times H}. \quad (4)$$

It is observed that the y_c^{fc} and y_c^{conv} are equal if we initialize the parameters of the both networks in the same way. Also, A_c^{fc} and A_c^{conv} have the same mathematical form. Therefore, we get the same-quality object localization maps A_c^{fc} and A_c^{conv} after the networks are convergent. In practice, the object localization maps from both methods are very similar and highlight the same target regions expect for some marginal differences caused by the stochastic optimization process. Figure 2 compares the object localization maps generated by CAM and our revised approach. We observe that the both approaches can generate the same quality maps and highlight the same regions in a given image. However, with our revised method, the object localization maps can be directly obtained in the forward pass rather than a post-processing step proposed in CAM.

3.2. The proposed ACoL

The mathematical proof in Section 3.1 provides theoretical support of the proposed ACoL. We identify that deep classification networks usually leverage the unique pattern of a specific category for recognition and the generated object localization maps can only highlight a small region of the target object instead of the entire object. Our proposed ACoL aims at discovering the integral object regions through an adversarial learning manner. In particular, it includes two classifiers, which can mine different but complementary regions of the target object in a given image.

Figure 3 shows the architecture of the proposed ACoL, including three components, Backbone, Classifier A and Classifier B. Backbone is a fully convolutional network acting as a feature extractor, which takes the original RGB images as input and produces high-level position-aware feature maps of multiply channels. The feature maps from Backbone are then fed into the following parallel classification branches. The object localization maps for each classifier can be conveniently obtained as described in Section 3.1. Both branches consist of the same number of convolutional layers followed by a GAP layer and a softmax layer for classification. The input feature maps of the two classifiers are different. In particular, the input features of Classifier B are erased with the guidance of the mined discriminative regions produced by Classifier A. We identify the discriminative regions by conducting a threshold on the localization maps of Classifier A. The corresponding regions within the input feature maps for Classifier B are then erased in an adversarial manner via replacing the values by zeros. Such an operation encourage Classifier B to leverage features from other regions of the target object for supporting image-level labels. Finally, the integral localization map of the target object will be obtained by combining the localization maps produced by the two branches.

Formally, we denote the training image set as $I = \{(I_i, y_i)\}_{i=0}^{N-1}$, where y_i is the label of the image I_i and N is the number of images. The input image I_i is firstly transformed by Backbone $f(\theta_0)$ to the spatial feature maps $S \in \mathbb{R}^{H_1 \times H_1 \times K}$ with K channels and $H_1 \times H_1$ resolution. We use θ to denote the learnable parameters of the CNN. Classifier A is denoted as $f(\theta_A)$ which can generate object

map $M^A \in \mathbb{R}^{H_2 \times H_2}$ of the size $H_2 \times H_2$ given the input feature maps S in a weakly supervised manner, as explained in Section 3.1. M^A usually highlights the unique discriminative regions for the target class.

We identify the most discriminative region as the set of pixels whose value is larger than the given threshold δ in object localization maps. M^A is resized by linear interpolation to $H_1 \times H_1$ if $H_1 \neq H_2$. We erase the discriminative regions in S according to the mined discriminative regions. Let $\tilde{S}$ denote the erased feature maps, which can be generated via replacing the pixel values of the identified discriminative regions by zeros. Classifier B $f(\theta_B)$ can generate the object localization maps $M^B \in \mathbb{R}^{H_2 \times H_2}$ with the input $\tilde{S}$. Then, the parameters θ of the network can be updated by back-propagation. Finally, we can obtain the integral object map for the class c by merging the two maps M^A and M^B. Concretely, we normalize both maps to the range $[0, 1]$ and denote them as $\bar{M}^A$ and $\bar{M}^B$. The fused object localization map $\bar{M}^{fuse}$ is calculated by $\bar{M}^{fuse}_{i,j} = \max(\bar{M}^A_{i,j}, \bar{M}^B_{i,j})$, where $\bar{M}_{i,j}$ is the element of the normalized map $\bar{M}$ at the i_{th} row and j_{th} column. The whole process is trained in an end-to-end way. Both classifiers adopt the cross entropy loss function for training. Algorithm 1 illustrates the training procedure of the proposed ACoL approach.

Algorithm 1 Training algorithm for ACoL

Input: Training data $I = \{(I_i, y_i)\}_{i=1}^N$, threshold δ
1: **while** training is not convergent **do**
2: Update feature maps $S \leftarrow f(\theta_0, I_i)$
3: Extract localization map $M^A \leftarrow f(\theta_A, S, y_i)$
4: Discover the discriminative region $R = \bar{M}^A > \delta$
5: Obtain erased feature maps $\tilde{S} \leftarrow erase(S, R)$
6: Extract localization map $M^B \leftarrow f(\theta_A, S, y_i)$
7: Obtain fused map $\bar{M}^{fuse}_{i,j} = \max(\bar{M}^A_{i,j}, \bar{M}^A_{i,j})$
8: Update θ_0, θ_A and θ_B
9: **end while**
Output: $\bar{M}^{fuse}$

During testing, we extract the fused object maps according to the predicted class and resize them to the same size with the original images by linear interpolation. For fair comparison, we apply the same strategy detailed in [48] to produce object bounding boxes based on the generated object localization maps. In particular, we firstly segment the foreground and background by a fixed threshold. Then, we seek the tight bounding boxes covering the largest connected area in the foreground pixels. For more details please refer to [48].

4. Experiments

4.1. Experiment setup

Datasets and evaluation metrics We evaluate the classification and localization accuracy of ACoL on two datasets, *i.e.*, ILSVRC 2016 [6, 31] and CUB-200-2011 [37]. ILSVRC 2016 contains 1.2 million images of 1,000 categories for training. We compare our approach with other approaches on the *validation* set which has 50,000 images. CUB-200-2011 [37] has 11,788 images of 200 categories with 5,994 images for training and 5,794 for testing. We leverage the localization metric suggested by [31] for comparison. The metric calculates the percentage of the images whose bounding boxes have over 50% IoU with the ground-truth. In addition, we also implement our approach on Caltech-256 [14] to visualize the outstanding performance in locating the integral target object.

Methods	top-1 err.	top-5 err.
GoogLeNet-GAP [48]	35.0	13.2
GoogLeNet	30.6	10.5
GoogLeNet-ACoL(Ours)	29.0	11.8
VGGnet-GAP [48]	33.4	12.2
VGGnet	31.2	11.4
VGGnet-ACoL(Ours)	32.5	12.0

Table 1: Classification error on ILSVRC validation set.

Implementation details We evaluate the proposed ACoL using VGGnet [34] and GoogLeNet [36]. Particularly, we remove the layers after *conv5-3* (from *pool5* to *prob*) of VGG-16 network and the last *inception* block of GoogLeNet. Then, we add two convolutional layers of kernel size 3×3, stride 1, pad 1 with 1024 units and a convolutional layer of size 1×1, stride 1 with 1000 units (200 and 256 units for CUB-200-2011 and Caltech-256 datasets, respectively). As the proof in Section 3.1, localization maps can be conveniently obtained from the feature maps of the 1×1 convolutional layer. Finally, a GAP layer and a softmax layer are added on the top of the convolutional layers. Both networks are fine-tuned on the pre-trained weights of ILSVRC [31]. The input images are randomly cropped to 224×224 pixels after being resized to 256×256 pixels. We test different erasing thresholds δ from 0.5 to 0.9. In testing, the threshold δ maintains constant w.r.t. the value in training. For classification results, we average the scores from the softmax layer with 10 crops (4 corners plus center, same with horizontal flip). We train the networks on NVIDIA GeForce TITAN X GPU with 12GB memory.

4.2. Comparisons with the state-of-the-arts

Classification: Table 1 shows the Top-1 and Top-5 error on the ILSVRC validation set. Our proposed methods GoogLeNet-ACoL and VGGnet-ACoL achieve sightly better classification results than GoogLeNet-GAP and VGGnet-GAP, respectively, and are comparable to the original GoogLeNet and VGGnet. For the fine-grained recognition dataset CUB-200-2011, it also achieves remarkable performance. Table 2 summarizes the benchmark approaches for classification with or without (w/o) bounding

Methods	Train/Test anno.	err.
Alignments [11]	w/o	46.4
Alignments [11]	BBox	33.0
DPD [47]	BBox+Parts	49.0
DeCAF+DPD [8]	BBox+Parts	35.0
PANDA R-CNN [46]	BBox+Parts	23.6
GoogLeNet-GAP on full image [48]	w/o	37.0
GoogLeNet-GAP on crop [48]	w/o	32.2
GoogLeNet-GAP on BBox [48]	BBox	29.5
VGGnet-ACoL(Ours)	w/o	28.1

Table 2: Classification error on fine-grained CUB-200-2011 test set.

Methods	top-1 err.	top-5 err.
Backprop on GoogLeNet [33]	61.31	50.55
GoogLeNet-GAP [48]	56.40	43.00
GoogLeNet-HaS-32 [35]	54.53	-
GoogLeNet-ACoL(Ours)	53.28	42.58
GoogLeNet-ACoL*(Ours)	53.28	35.22
Backprop on VGGnet [33]	61.12	51.46
VGGnet-GAP [48]	57.20	45.14
VGGnet-ACoL(Ours)	54.17	40.57
VGGnet-ACoL*(Ours)	54.17	36.66

Table 3: Localization error on ILSVRC validation set (* indicates methods which improve the Top-5 performance only using predictions with high scores).

Methods	top-1 err.	top-5 err.
GoogLeNet-GAP [48]	59.00	-
VGGnet-ACoL(Ours)	54.08	43.49
VGGnet-ACoL*(Ours)	54.08	39.05

Table 4: Localization error on CUB-200-2011 test set.

box annotations. We find our VGGnet-ACoL achieves the lowest error 28.1% among all the methods without using bounding box.

To summarize, the proposed method can enable the networks to achieve equivalent classification performance with the original networks though our modified networks actually do not use fully connected layers. We attribute it to the erasing operation which guides the network to discover more discriminative patterns so as to obtain better classification performance.

Localization: Table 3 illustrates the localization error on the ILSVRC *val* set. We observe that our ACoL approach outperforms all baselines. VGGnet-ACoL is significantly better than VGGnet-GAP and GoogLeNet-ACoL also achieves better performance than GoogLeNet-HaS-32 which adopts the strategy of randomly erasing the input images. We illustrate the localization performance on the CUB-200-2011 dataset in Table 4. Our method outperforms GoogLeNet-GAP by 4.92% in Top-1 error.

We further improve the localization performance by combining our localization results with the state-of-the-art classification results, *i.e.*, ResNet [15] and DPN [5], to break the limitation of classification when calculating localization accuracy. As shown in Table 5, the localization accuracy constantly improves with the classification results getting better. We have a boost to 45.14% in Top-1 error and 38.45% in Top-5 error when applying the classification results generated from the ensemble DPN. In addition, we boost the Top-5 localization performance (indicated by *) by only selecting the bounding boxes from the top three predicted classes following [48] and VGGnet-ACoL-DPN-ensemble* achieves 30.03% on ILSVRC.

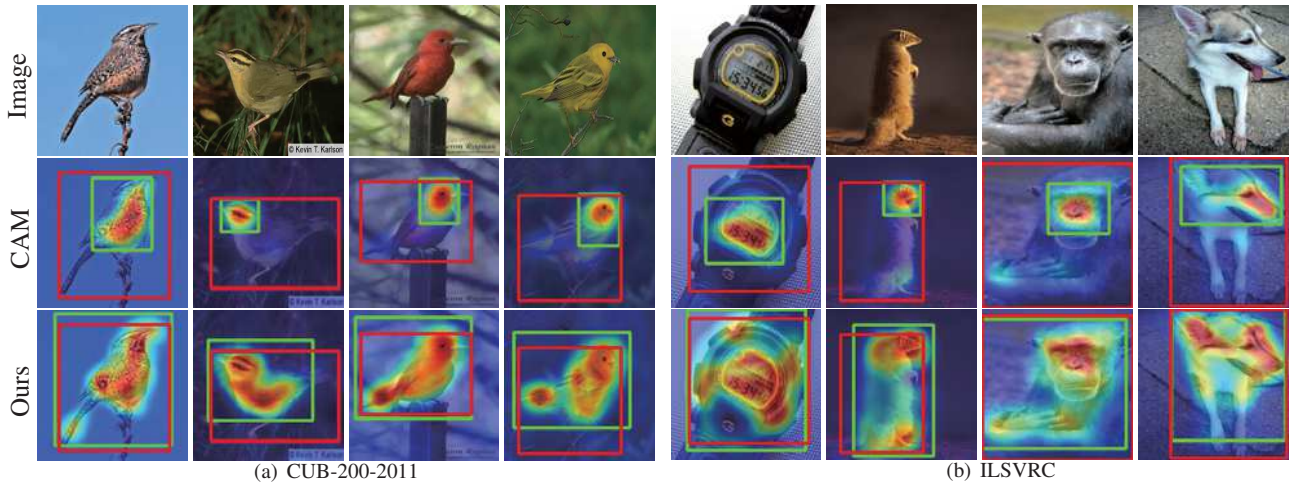
Methods	top-1 err.	top-5 err.
VGGnet-ACoL-ResNet-50	49.82/26.22	40.38/8.47
VGGnet-ACoL-ResNet-101	49.26/24.90	40.08/7.80
VGGnet-ACoL-ResNet-152	48.96/24.39	39.97/7.59
VGGnet-ACoL-DPN-92	46.30/17.70	38.96/3.83
VGGnet-ACoL-DPN-98	46.16/17.42	38.99/3.67
VGGnet-ACoL-DPN-131	46.06/17.08	38.85/3.42
VGGnet-ACoL-DPN-ensemble	45.14/15.47	38.45/2.70
VGGnet-ACoL-DPN-ensemble*	45.14/15.47	30.03/2.70

Table 5: Localization/Classification error on ILSVRC validation set with the state-of-the-art classification results.

Figure 4 visualizes the localization bounding boxes of the proposed method and CAM method [48]. The object localization maps generated by ACoL can cover larger object regions to obtain more accurate bounding boxes. For example, our method can discover nearly entire parts of a bird, *e.g.*, the wing and head, while the CAM method [48] can only find a small part of a bird, *e.g.*, the head. Figure 5 compares the object localization maps of the two classifiers in mining object regions. We observe that Classifier A and Classifier B are successful in discovering different but complementary target regions. The localization maps from the two classifiers can finally fuse into a robust one, in which the integral object is effectively highlighted. Consequently, we get boosted localization performance.

4.3. Ablation study

In the proposed method, the two classifiers locate different regions of interest via erasing the input feature maps of Classifier B. We identify the discriminative regions by a hard threshold δ. In order to inspect its influence on localization accuracy, we test different threshold values $\delta \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ shown in Table 6. We obtain the best performance in Top-1 error when the threshold $\delta = 0.6$ on ILSVRC, and it becomes worse when the erasing threshold is larger or smaller. We can conclude: 1) The proposed complementary branch (Classifier B) successfully works collaboratively with Classifier A, because the former can mine complementary object regions so as to generate integral object regions; 2) a well-designed thresh-



(a) CUB-200-2011 (b) ILSVRC

Figure 4: Comparison with CAM method. Our method can locate larger object regions to improve localization performance (ground-truth bounding boxes are in red and the predicted are in green).

Dataset	threshold	top-1 err.	top-5 err.
CUB-200-2011	0.5	58.34	48.11
	0.6	54.15	**42.79**
	0.7	**54.08**	43.49
	0.8	55.78	45.17
	0.9	55.22	45.76
ILSVRC	0.5	62.62	52.03
	0.6	**54.17**	**40.57**
	0.7	54.55	42.53
	0.8	56.61	45.45
	0.9	55.72	44.42

Table 6: Localization error with different erasing thresholds.

Methods	GT-known loc. err.
AlexNet-GAP [48]	45.01
AlexNet-HaS [35]	41.26
AlexNet-GAP-ensemble [48]	42.98
AlexNet-HaS-emsemble [35]	39.67
GoogLeNet-GAP [48]	41.34
GoogLeNet-HaS [35]	39.43
Deconv [44]	41.6
Feedback [3]	38.8
MWP [45]	38.7
ACoL (Ours)	**37.04**

Table 7: Localization error on ILSVRC validation data with ground-truth labels.

old can improve the performance as a too large threshold cannot effectively encourage Classifier B to discover more useful regions and a too small threshold may bring background noises.

We also test a cascade network of three classifiers. In particular, we add the third classifier and erase its input feature maps guided by the fused object localization maps from both Classifier A and B. We observe there is no significant improvement in both classification and localization performance. Therefore, adding the third branch does not necessarily improve the performance and two branches are usually enough for locating the integral object regions.

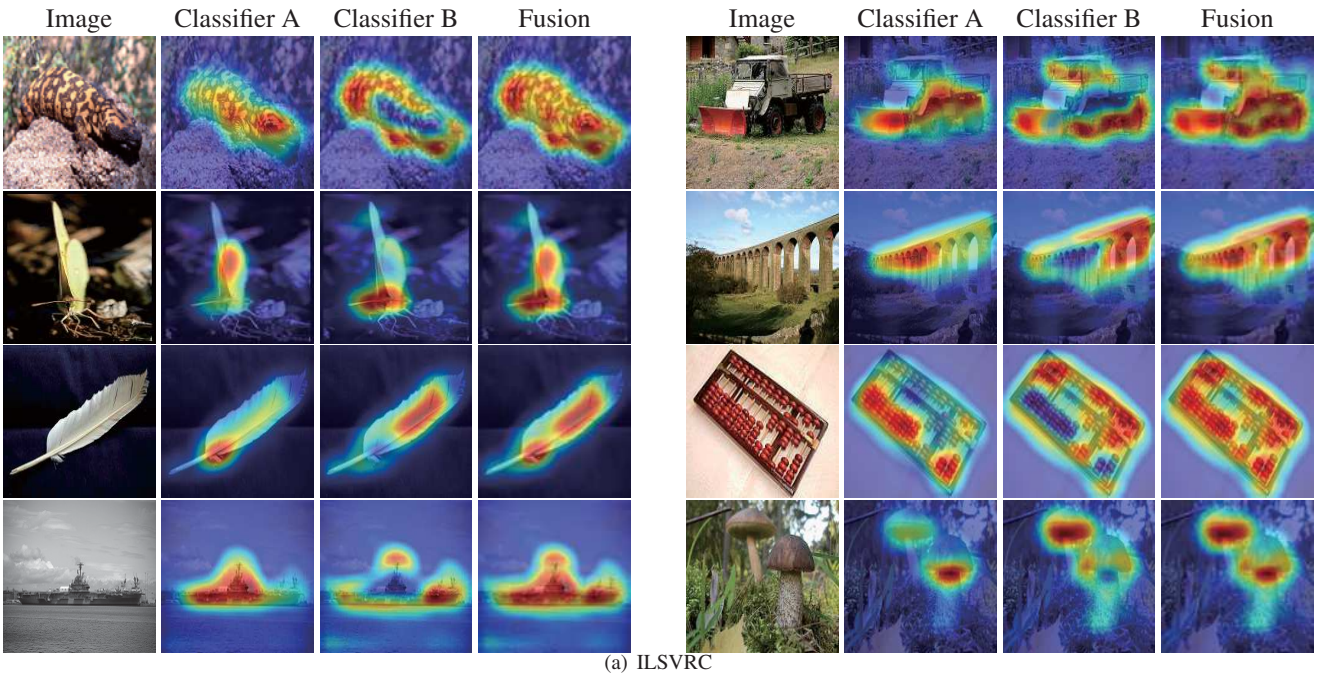
Furthermore, we eliminate the influence caused by classification results and compare the localization accuracy using ground-truth labels. As shown in Table 7, the proposed ACoL approach achieves 37.04% in Top-1 error and surpasses the other approaches. This reveals the superiority of the object localization maps generated by our method, and shows that the proposed two classifiers can successfully locate complementary object regions.

5. Conclusion

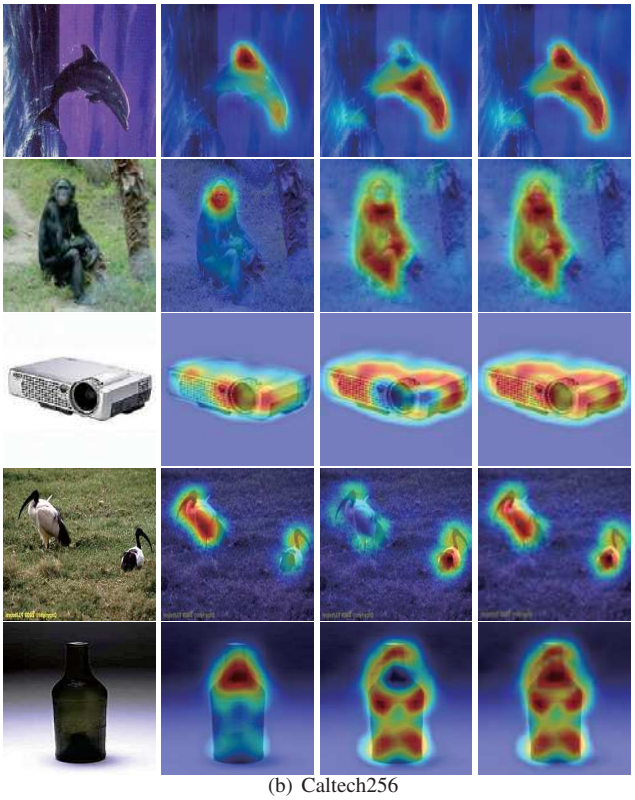
We firstly mathematically prove that object localization maps can be conveniently obtained by selecting from feature maps. Based on it, we proposed Adversarial Complementary Learning for locating target object regions in a weakly supervised manner. The proposed two adversarial classification classifiers can locate different object parts and discover the complementary regions belonging to the same objects or categories. Extensive experiments show the proposed method can successfully mine integral object regions and outperform the state-of-the-art localization methods.

Acknowledgement

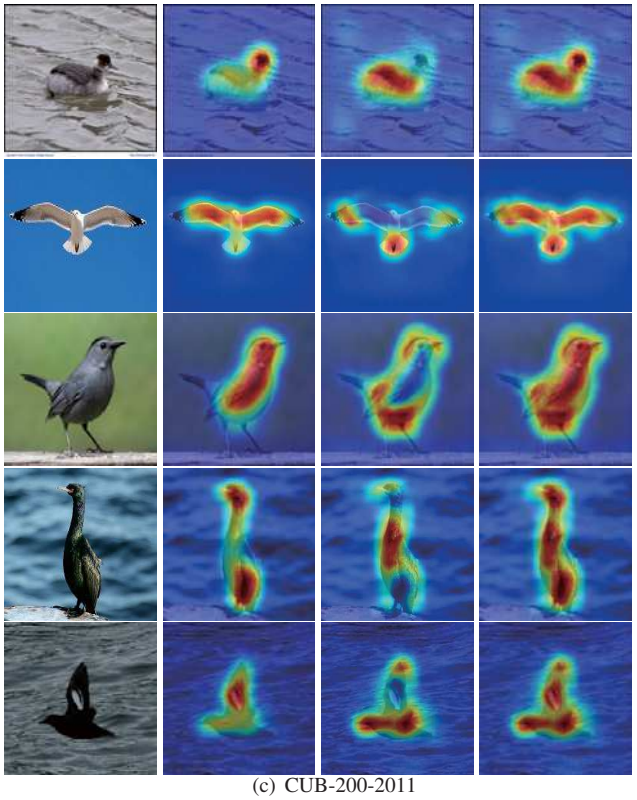
Yi Yang is the recipient of a Google Faculty Research Award. This work is partially supported by IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) - a research collaboration as part of the IBM AI Horizons Network. We acknowledge the Data to Decisions CRC (D2D CRC) and the Cooperative Research Centres Programme for funding this research.



(a) ILSVRC



(b) Caltech256



(c) CUB-200-2011

Figure 5: Object localization maps of the proposed method. We compare complementary effects of the two branches on ILSVRC, Caltech256 and CUB-200-2011 datasets. For each image, we show object localization maps from Classifier A (middle left), Classifier B (middle right) and the fused maps (right). The proposed two classifier (A and B) can discover different parts of target objects so as to locate the entire regions of the same category in a given image.

References

[1] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani. Self-taught object localization with deep networks. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016. 2, 3

[2] A. J. Bency, H. Kwon, H. Lee, S. Karthikeyan, and B. Manjunath. Weakly supervised localization using deep feature maps. In *eccv*, pages 714–731. Springer, 2016. 2

[3] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2956–2964, 2015. 7

[4] A. Chaudhry, P. K. Dokania, and P. H. Torr. Discovering class-specific pixels for weakly-supervised semantic segmentation. *arXiv preprint arXiv:1707.05821*, 2017. 3

[5] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng. Dual path networks. *arXiv preprint arXiv:1707.01629*, 2017. 6

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE CVPR*, pages 248–255, 2009. 5

[7] T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *ijcv*, 100(3):275–293, 2012. 3

[8] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014. 6

[9] X. Dong, D. Meng, F. Ma, and Y. Yang. A dual-network progressive approach to weakly supervised object detection. In *ACM Multimedia*, 2017. 2

[10] X. Dong, L. Zheng, F. Ma, Y. Yang, and D. Meng. Few-example object detection with model communication. *arXiv preprint arXiv:1706.08249*, 2017. 2

[11] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars. Local alignments for fine-grained categorization. *International Journal of Computer Vision*, 111(2):191–212, 2015. 6

[12] R. Girshick. Fast r-cnn. In *arXiv preprint arXiv:1504.08083*, 2015. 1

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE CVPR*, pages 580–587, 2014. 1

[14] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007. 5

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[16] Q. Hou, P. K. Dokania, D. Massiceti, Y. Wei, M.-M. Cheng, and P. Torr. Bottom-up top-down cues for weakly-supervised semantic segmentation. *arXiv preprint arXiv:1612.02101*, 2016. 3

[17] S. Hwang and H.-E. Kim. Self-transfer learning for fully weakly supervised object localization. *arXiv preprint arXiv:1602.01625*, 2016. 3

[18] Z. Jie, Y. Wei, X. Jin, and J. Feng. Deep self-taught learning for weakly supervised object localization. In *IEEE CVPR*, 2017. 2

[19] D. Kim, D. Yoo, I. S. Kweon, et al. Two-phase learning for weakly supervised object localization. *arXiv preprint arXiv:1708.02108*, 2017. 2

[20] A. Kolesnikov and C. H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, pages 695–711, 2016. 3

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1

[22] X. Liang, S. Liu, Y. Wei, L. Liu, L. Lin, and S. Yan. Towards computational baby learning: A weakly-supervised approach for object detection. In *IEEE ICCV*, pages 999–1007, 2015. 2

[23] M. Lin, Q. Chen, and S. Yan. Network in network. *ICLR*, 2013. 1, 3

[24] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017. 2

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[26] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 685–694, 2015. 2, 3

[27] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*, 2015. 3

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[29] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[30] O. Russakovsky, A. Bearman, V. Ferrari, and L. Fei-Fei. Whats the point: Semantic segmentation with point supervision. In *ECCV*, pages 549–565, 2016. 2

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[32] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *International Conference on Learning Representations*, 2014. 1, 2

[33] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 6

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015. 1, 5

[35] K. K. Singh and Y. J. Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. *arXiv preprint arXiv:1704.04232*, 2017. 2, 6, 7

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 5

[37] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011. 5

[38] L. Wang, G. Hua, R. Sukthankar, J. Xue, and N. Zheng. Video object discovery and co-segmentation with extremely weak supervision. In *ECCV*, pages 640–655. Springer, 2014. 2

[39] Y. Wei, J. Feng, X. Liang, C. Ming-Ming, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *IEEE CVPR*, 2017. 1, 2, 3

[40] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE TPAMI*, 39(11):2314–2320, 2017. 3

[41] Y. Wei, X. Liang, Y. n. Chen, Z. Jie, Y. Xiao, Y. Zhao, and S. Yan. Learning to segment with image-level annotations. *Pattern Recognition*, 2016. 3

[42] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. Hcp: A flexible cnn framework for multi-label image classification. *IEEE TPAMI*, 38(9):1901–1907, 2016. 2

[43] 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. In *IEEE CVPR*, 2018. 3

[44] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 7

[45] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision*, pages 543–559. Springer, 2016. 1, 3, 7

[46] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *eccv*, pages 834–849. Springer, 2014. 6

[47] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *iccv*, pages 729–736, 2013. 6

[48] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *IEEE CVPR*, 2016. 1, 2, 3, 5, 6, 7