Greedy Layerwise Training for Weakly-Supervised Object Localization and Segmentation

Zhecheng Wang
Department of Mechanical Engineering
Stanford University
zhecheng@stanford.edu

Chi Zhang
Department of Mechanical Engineering
Stanford University
czhang94@stanford.edu

Abstract

In this work, we propose an end-to-end model for weakly-supervised object localization and segmentation. We build our model on CNN architectures which are originally used for classification. Firstly, we take advantage of the global average pooling (GAP) layer which enabled convolutional layers to preserve their ability of localization. Meanwhile, using layerwise training, we find that our model can extract features greedily, endowing our model with a remarkable ability to segment objects with clear boundaries. Our model is evaluated on a newly collected dataset (WhereIsSolar) which contains 475k remote sensing images for solar panel detection. Experiments show that our model achieves competitive results compared with the current state-of-the-art weakly-supervised approach.

1. Introduction

Convolutional Neural Networks (CNNs) has seen a rapid resurgence after Krizhevsky *et al.*[8] demonstrated their significant capability in visual recognition tasks on ImageNet[2]. CNN can be used for not only image-level classification, but also object localization, object detection, semantic segmentation and instance segmentation. The target of object localization or detection is to draw the bounding box of an object in an image, while segmentation task is to make pixel-level classification in an image. All these algorithms or models have broad applications in real world, such as pedestrian detection, face recognition, vehicle detection and object identification/segmentation in satellite imagery, *etc*.

Current state-of-art semantic segmentation methods are mainly based on Fully Convolutional Network (FCN)[11], which is trained with pixel-to-pixel supervision. With VGG-16 framework, the test time speed is lower than 5 images/second, which is not high enough for real-time segmentation. Current state-of-art object detection methods can be divided into two categories: one category is the R-CNN based methods[5, 4, 14], which first generate regions of interest (RoI) and then do classification and bounding box regression on these regions. One disadvantage of this type of models is their slow speed. Even Faster R-CNN[4], armed with Region Proposal Network, can only achieve a speed of 5 images/second with VGG-16 framework at test time. Another category includes end-to-end methods without region proposal such as YOLO[13] and SSD[10], which output all class scores and bounding box information with a single network. While these methods are much more time-efficient, their performances are not comparable to R-CNN based methods.

Though different, the two categories of methods mentioned above share a common problem that they are all fully-supervised. Concretely, bounding box annotations are needed for object detection or localization and pixel-level class labels are required for segmentation. Although these annotations are available for benchmark computer vision datasets, such as ImageNet, PASCAL-VOC[3], it is very expensive to collect them in real-world applications because annotating bounding boxes or object boundaries can be very time-consuming compared with image-level labeling. Therefore, it is necessary to develop an unsupervised, or semi-supervised object detection and segmentation method based on image-level labels, which are much easier to be available in most cases.

2. Related Work

Recent work has revealed that CNN itself has strong ability in localizing objects. Zhou *et al*. [21] pointed out that the convolutional units of various layers of CNNs behaved as object detectors despite no supervision on the location of the object was provided. However, this remarkable ability to localize objects in convolutional layers will be lost when fully-connected layers are used for classification tasks. Instead, with fully-connected layers replaced with global average pooling (GAP) layers, even the last con-

volutional layer in the network can preserve the ability of localization, thus the model trained for classification with image-level labels can also be used for object localization. Other work [12] also used global max pooling (GMP) to replace fully-connected layers for localization. Compared with GMP-based method, which can only localize one point in the boundary of an object, the GAP-based can find nearly the full extent of the object.

Although Zhou's best model proposed in [21] achieved 37.1% top-5 test error, close to the 34.2% top-5 test error achieved by fully supervised AlexNet[8] for weakly supervised object localization on ILSVRC 2014 benchmark[15], there still exist two problems. Firstly, replacing fully-connected layer with GAP reduces classification performance. Top-1 classification error increased 2.2% for VGG-16-based model after this modification and 3.8% for the GoogLeNet-based model. Secondly, with the network going deeper, spatial dimensions of feature maps decrease and activations will be attenuated, resulting in a low resolution activation map with blurred object boundary. And only the salient part of an object, which is the most activated, can be localized. In fact, this is a major trade-off in CNN: the features in upstream layers are complete, generic but too noisy for accurate classification and object localization, while features in downstream layers are pure and specific for classification. During the feed-forward process, a large part of the features are filtered out as activations decay layer by layer, leaving only the most salient features remained, which is good for classification but not good for extracting the full region of an object.

In our work, we aim to greedily extract features at mid-level stage layers using layerwise training. This method will not only preserve activations but also make features gradually specific at the same time. We also aim to keep high classification accuracy while utilizing it for object localization and segmentation.

3. Methodology

Our model is built on well-designed classification CNN. We keep the original framework for classification but design branches for localization. Although the branches are still trained to minimize the classification loss, with global average pooling (GAP) layer, class activation map (CAM) can be generated for object localization and segmentation with only the supervision of image-level labels. On the other hand, we use greedy layerwise training to extract features for improving localization and segmentation results.

3.1. Global Average Pooling

In [9], global average pooling is used as a structural regularizer to prevent overfitting during training. For our purpose, global average pooling acts not only a way of regularization but also a tool for building class activation map. Also, it will make the CNN invulnerable to different input image size, while fully-connected layers cannot adapt to different input size.

As mentioned in previous section, one similar pooling method is *global max pooling* (GMP), which can also be used for weakly-supervised object localization. Based on prior work[12], we believe that GAP encourages the network to detect the extent of the whole object as GMP encourages it to focus on identifying one discriminative part of the object. Intuitively, this phenomenon can be explained by the difference of average and max function: average of a map takes all activations and during training, the values can be maximized by finding all discriminative parts of an object, while doing a max simply wipes out all low activations except the most discriminative part.

According to Zhou *et al.*'s experiments on ILSVRC dataset, GMP achieves similar performance as GAP in classification but outperforms in localization. In order to obtain stronger ability of localization and better compare our model with theirs, we use GAP as the pooling method.

3.2. Class Activation Mapping

A class activation map (CAM) shows discriminative image regions of a specific category for CNN to identity it.

For a given image, denote the activation of pixel (x, y) on unit i in the last convolutional layer to be $h_i(x, y)$. The height of the feature activation maps is h and the width is w, Then, the result of performing global average pooling is

$$H_i = \frac{1}{hw} \sum_{x,y} h_i(x, y) \tag{1}$$

since hw is a constant in a known network, this term can be simply neglected and H_i then becomes $\sum_{x,y} h_i(x, y)$.

For a given class c, the neuron output S_c before softmax is $\sum_i w_{ic} H_i$ where w_{ic} denotes the weight (importance) H_i relative to class c. By plugging in the expression of H_i into S_c, it can be obtained that

$$\begin{aligned} S_c &= \sum_i w_{ic} H_i \\ &= \sum_i w_{ic} \sum_{x,y} h_i(x, y) \\ &= \sum_{x,y} \sum_i w_{ic} h_i(x, y) \end{aligned} \tag{2}$$

Finally after softmax, the output of class c can be expressed as

$$\frac{exp(S_c)}{\sum_c exp(S_c)} \tag{3}$$

bias terms are ignored in our case. Based on Eq. 2 and Eq. 3, it can be observed that the term

$$M_c = \sum_i w_{ic} h_i(x, y) \tag{4}$$

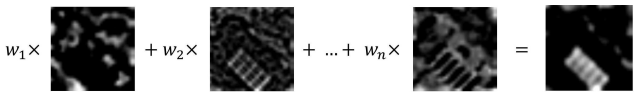

Figure 1. Classification activation map (CAM)

explicitly indicates the importance of the activation at pixel (x, y) leading to classifying an image as class c.

Intuitively, $h_i(x, y)$ is the map of the presence of some visual pattern[20, 19]. As Fig.1 shows, the class activation map M_c is a weighted linear sum of all visual patterns observed in images of class c. To identify the regions on an image mostly related to a particular class, we can simply upsample the class activation map to the size of the image. This is an excellent property of CNN, which make weakly-supervised object localization and segmentation possible.

3.3. Greedy Layerwise Training

Previously, greedy layerwise training were commonly used for unsupervised pre-training of deep neural networks[1] with a target of overcoming the initialization problem in training a deep neural network. It has been proved that greedy layerwise unsupervised training can serve as a good weight initialization for optimization. However, this approach has been no longer necessary after numbers of advanced training techniques emerged, such as ReLU [6], Dropout [17] and Batch Normalization [7], but it sheds light on extracting features greedily layer by layer to generate specific but undiminished representations.

Inspired by this idea, we utilize layerwise training to greedily extract the specific, discriminative but complete region of an object in localization and segmentation tasks. Why greedy layerwise training works can be illustrated with the feature evolution map (as is shown in Fig.2). For any deep feed-forward network, upstream layers learn low-level features such as edges and basic shapes, while downstream layers learn high-level features that are more specific and abstract. Therefore, feature maps at low level are noisy and less relative to specific classes. However, they have two advantages: First, few features are filtered out and most are reserved. Second, the resolution of the feature maps is high. By contrast, the feature maps at high level are more indicative of specific classes. Using such feature maps for classification can often yield good accuracy. However, many features which are not very indicative of specific classes are filtered out and the resolution becomes lower after multiple downsamplings during the feed-forward process. As the CAM is a linear combination of the feature maps, CAM generated at low-level hierarchy is more complete, noisier and has higher resolution, while CAM generated at high-level hierarchy is more specific, discriminative and has lower resolution. This can be regarded as a trade-off in representation learning.

However, with greedily layerwise training for classification, we can break such trade-off and generate complete, clear but also specific and discriminative CAM. This is because minimizing the classification loss can be regarded as a process of extracting features indicative of the target classes. Therefore, training the layers at low-level or mid-level hierarchy for classification can extract specific features from a complete and noisy upstream feature map and thus generate feature map that are both complete and specific. If we repeat this process for several times, we can get better CAM for object localization and segmentation. From Fig.2 we can see that CAMs generated with layerwise training keep the complete object boundary but also reduce the noise. To sum up, the essential intuition behind greedy layerwise training is to greedily extract features to balance the specificity of representations and completeness of activations, and keep a comparably high resolution of the class activation map as well.

Specifically, we train a single convolutional layer plus a GAP layer and a linear classifier for image-level classification at a time, based on a pre-trained network for classification. Then we discard the last GAP and linear classifier and add a new convolutional layer with a GAP layer and a new linear classifier at the end of the last trained convolutional layer, and also train the newly added layers separately. Note that unlike Zhou's model [12], which removed last layers of the original architecture and replaced them with GAP layers, we keep the completeness of the original architecture but add another branch right after a mid-level convolutional layer for object localization and segmentation. Therefore, another advantage of our model is that classification ability of the original model will not decrease at all.

When training a single Conv-GAP-Linear structure, we keep other layers completely fixed, thus weights and biases of those layers will not be updated. This process is illustrated in Fig.3.

4. Experiments

4.1. Dataset

$WhereIsSolar$ is an aerial image dataset developed by Stanford Sustainable Systems Lab for solar panel identification with remote sensing imagery. The target is to make household-level localization and area estimation of distributed solar panels. There are 381,805 samples in training set, of which 47,480 are positive samples (containing solar panels), and 93,500 samples in test set. In training set, only image-level annotations are available, but in test set, the polygon regions of solar panel in positive samples are annotated. Therefore, it is a good benchmark dataset for testing our weakly-supervised object localization and segmentation method.

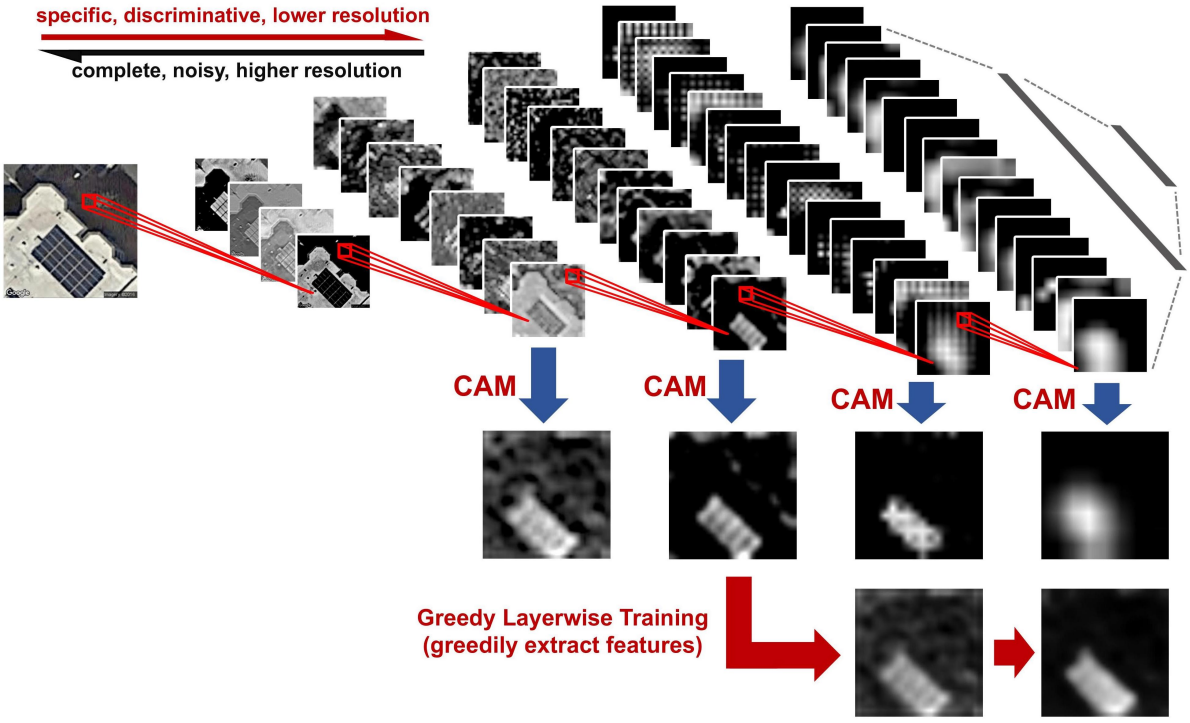

Figure 2. Feature evolution

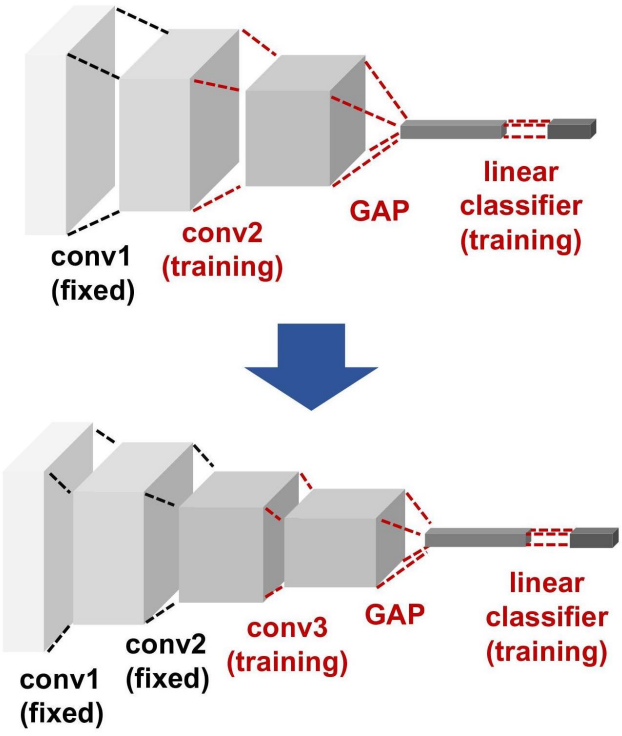

Figure 3. Greedy layerwise training

4.2. Setup

Our model is adaptive to various CNN frameworks including AlexNet, VGG, GoogLeNet, ResNet, *etc*. In this work, we experimented our method on VGG-16 [16] and GoogLeNet Inception v3 [18]. Due to the limit of time and computing resource, we have only experimented up to two convolutional layers in the branch.

4.2.1 VGG-16

Using pre-trained VGG-16 network as the base framework for classification, we added the branch of localization and segmentation after `CONV4_3` layer. And we designed the branch to have the following layers:

1. **`CONV`**: 3x3x512, 512 filters, 1 stride
2. **`ReLU`**: $max(x_i, 0)$
3. **`CONV`**: 3x3x512, 512 filters, 1 stride
4. **`ReLU`**: $max(x_i, 0)$
5. **`GAP`**: $\sum_{x,y} h_i(x, y)$
6. **`Softmax`**: $\frac{exp(S_c)}{\sum_c exp(S_c)}$

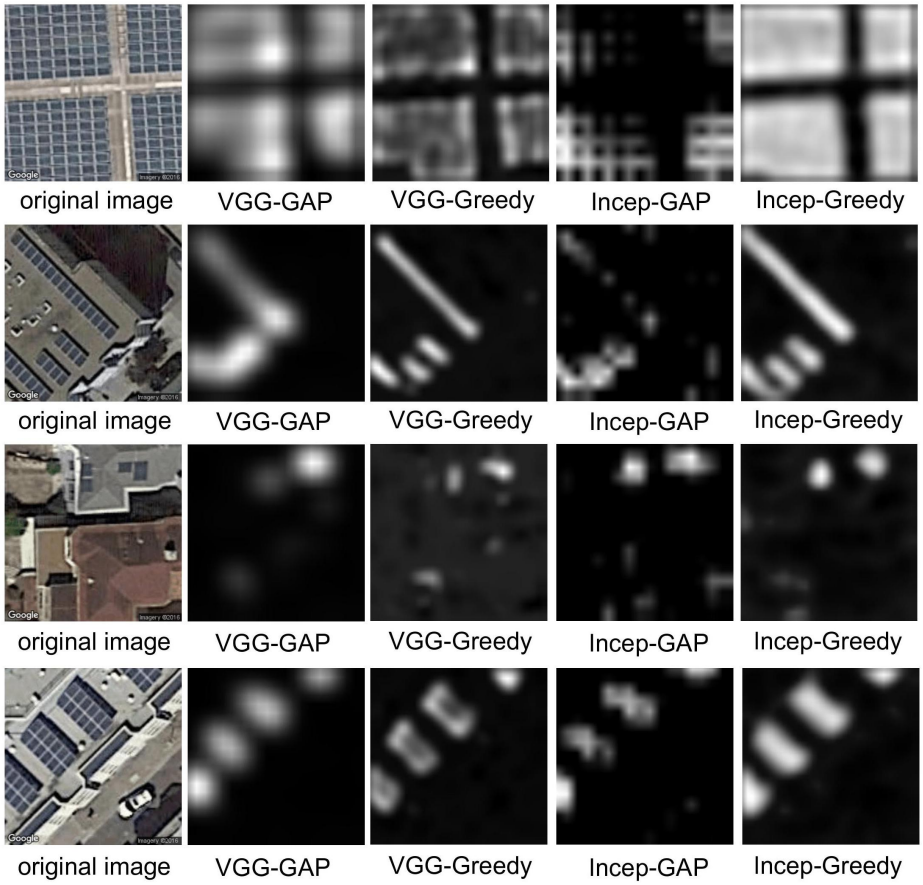

Figure 4. Comparison among input images, CAMs generated by Zhou's model [22] and our greedily layerwise trained models

4.2.2 GoogLeNet Inception v3

We also experimented on GoogLeNet Inception v3. The branch for localization and segmentation was built at the end of the `35×35×288b` layer as follow:

1. **`CONV`**: 3x3x288, 512 filters, 1 stride
2. **`ReLU`**: $max(x_i, 0)$
3. **`CONV`**: 3x3x512, 512 filters, 1 stride
4. **`ReLU`**: $max(x_i, 0)$
5. **`GAP`**: $\sum_{x,y} h_i(x, y)$
6. **`Softmax`**: $\frac{exp(S_c)}{\sum_c exp(S_c)}$

4.3. Training

First, we fine-tuned the model pre-trained on ImageNet for classification with the a subset of training set for 30 epochs. We only used a subset with the size of 42,070 for training due to training time consideration. The initial learning rate is 0.001, and decay with a factor of 0.5 every 7 epochs. Once it is done, we executed the greedy layerwise training process and trained up to 2 convolutional layers in the branch. The learning rate is 0.005 and the total number of epochs is 20 for training the branch.

5. Results and Discussion

We first report results on image classification to prove that our method preserves the best classification performance. Then we demonstrate that our approach is effective and better than Zhou's model for weakly-supervised object localization and segmentation.

Classification: Tbl.1 summarizes the classification performance of both Zhou's GAP networks and our greedily layerwise trained models. In the table, GAP indicates Zhou's model and GAP+Greedy indicates our model. We can find that for both VGG-16 and Inception v3 frameworks, both precision and recall of our models are higher than those of Zhou's model. Among them, Inception-GAP+Greedy has the best performance since Inception v3 itself is a better framework than VGG-16. In fact, it is important for the networks to perform well on classification

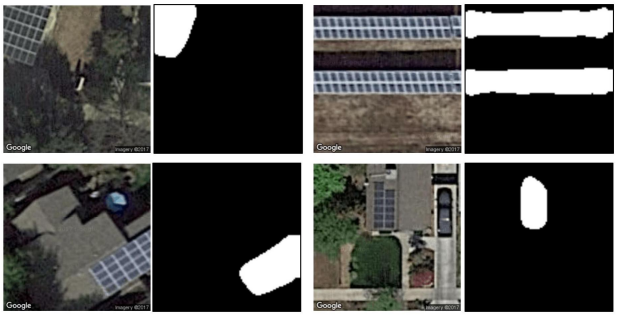

Figure 5. Examples of segmentation results generated with greedy layerwise training

in order to achieve a high performance on localization as it involves identifying both the object category and the bounding box location accurately.

Table 1. Classification results

Model	Precision	Recall
VGG-GAP	78.3%	82.2%
VGG-GAP+Greedy	**86.4%**	**86.5%**
Inception-GAP	87.5%	83.1%
Inception-GAP+Greedy	**93.5%**	**92.2%**

Localization and Segmentation: Fig.4 shows class activation maps generated by Zhou's model and models trained with our greedy layerwise method. It can be observed that our model has much better performance versus GAP networks in localizing and segmenting solar panels.

Zhou's model is able to detect salient parts of solar panels, such as panel edges, but activations are low in non-salient parts, while our greedy layerwise method can extract nearly complete regions of solar panels and boundaries are much clearer. This is due to our greedy feature extraction at the mid-level hierarchy of the network, which keeps both completeness and specificity. Moreover, the CAM generated with our model has higher resolution than that of Zhou's model. For VGG-16, CAM generated with Zhou's model has resolution of 14×14 but ours is 28×28. For Inception v3, CAM generated with Zhou's model has resolution of 17×17 but ours is 35×35.

In order to perform localization and segmentation, we need to generate a bounding box around the object with its associated category. To draw a bounding box from the CAMs, a simple thresholding technique is used to segment the feature map. First, we segment the regions of which the value is above 0.5 of the max value of the CAM. Then we take the bounding box that covers the largest connected component in the segmentation map. Fig.5 shows some segmentation examples and Fig.6 shows some bounding boxes example generated using this technique (Inception-GAP+Greedy). The localization performance on the test set

Figure 6. Examples of bounding boxes generated with greedy layerwise training

is shown in Tbl.2. The metric we used for evaluation is intersection over union (IoU) between ground truth bounding box and predicted bounding box.

Table 2. Localization results

Model	IoU
VGG-GAP	0.654
VGG-GAP+Greedy	**0.679**
Inception-GAP	0.607
Inception-GAP+Greedy	**0.728**

We observe that our models outperform all the baseline approaches (Zhou's model) in object localization tasks. Among them, Inception v3-based model with greedy layerwise training has the highest IoU of 0.728, which is remarkable given that this network has not been trained with even a single annotated bounding box. Further, we observe that the performance of Inception-GAP is the poorest despite that Inception is a better framework than VGG-16. The reason may be that Inception is a deeper network and a large part of features have already been diminished after the filtering effect of many layers, thus the high-level features are not suitable for segmenting a complete object boundary. By contrast, greedily extracting features from low-level hierarchy can overcome this problem.

6. Conclusion

In this work, we propose the greedy layerwise training method for weakly-supervised object localization and segmentation. Armed with global average pooling (GAP) and class activation map (CAM), CNNs trained for classification are endowed with the ability to draw the discriminative object boundaries. With greedy layerwise training at mid-level hierarchy of the network, features can be greedily extracted to keep both completeness and specificity, contributing to an accurate and clear boundary for an object. The experience results on *WhereIsSolar* dataset shows that our models can preserve the best classification ability and also yield better results on localization and segmentation tasks. We hope our work can draw attention to the benefit of greedy layerwise training, which is popular at the initial stage of deep learning but rarely raised in current deep learning community.

7. Suggested Future Work

Through our experiments, we have obtained promising results. However, we still need to quantify our results other metric such as mAP. Moreover, we need to find out at which hierarchy to add a branch for object localization and segmentation can achieve the best results, and how many convolutional layers need to be greedily trained in that branch. Also, if more computing resource is available, we will test the performance of our model on more general computer vision benchmark datasets, such as ILSVRC 2014, PASCAL-VOC, and compare it with previous model under the same base frameworks.

References

[1] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[2] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. 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] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[4] R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[5] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[6] G. E. Hinton. Rectified linear units improve restricted boltzmann machines vinod nair, 2010.

[7] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. 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 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[9] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. *SSD: Single Shot MultiBox Detector*, pages 21–37. Springer International Publishing, Cham, 2016.

[11] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[12] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[13] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[14] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[15] 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*, 115(3):211–252, 2015.

[16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[19] M. D. Zeiler and R. Fergus. *Visualizing and Understanding Convolutional Networks*, pages 818–833. Springer International Publishing, Cham, 2014.

[20] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. *Part-Based R-CNNs for Fine-Grained Category Detection*, pages 834–849. Springer International Publishing, Cham, 2014.

[21] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *CoRR*, abs/1412.6856, 2014.

[22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.