Region-based Convolutional Networks for Accurate Object Detection and Segmentation

Ross Girshick; Jeff Donahue; Trevor Darrell; Jitendra Malik;

IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 38, Issue: 1, Pages: 142-158

May 25, 2015

Presenter: Junwon Moon
mppn98@g.skku.edu

Contents

- Introduction
- Related Work
- Object Detection with An R-CNN
- Analysis
- The ILSVRC2013 Detection Dataset
- Semantic Segmentation
- Implementation and Design Details
- Conclusion

Introduction

■ Background

- objects and localizing them in images is one of the most fundamental and challenging problems in computer vision.

■ Previous Methods

- traditional computer vision techniques like SIFT (Scale-Invariant Feature Transform) and HOG (Histogram of Oriented Gradients) had limitations in capturing complex patterns, and further significant progress became difficult.

Introduction

■ The core idea of R-CNN

- R-CNN is the combination of region proposals and CNN (Convolutional Neural Network). This method first proposes regions in an image where objects are likely to be present, and then applies CNN to these regions to classify the objects. This approach is much more efficient and accurate than the previous sliding window method.

Related Work

■ Deep CNNs for Object Detection

- Szegedy et al. proposed a method that uses CNNs to predict the whole and parts of objects, and generate bounding boxes based on these predictions. However, their model did not utilize pre-training on ImageNet, which led to lower performance. On the other hand, Agrawal et al. trained R-CNN from random initialization and achieved 40.7% mAP, which outperformed Szegedy's results despite using only half the amount of training data.

Related Work

■ Scalability and Speed

- R-CNN maintains its performance even as the number of object classes increases, as most computations are shared across classes. In contrast, **DPM (Deformable Part Models)** experiences a significant drop in accuracy as the number of objects grows, while R-CNN requires only a small amount of additional time. However, R-CNN takes 10 to 45 seconds per image, which limits its speed. To address this, He et al.'s **SPPnet** improved speed through computation sharing. Additionally, Girshick proposed **Fast R-CNN**, which further reduced detection and training times.

Related Work

■ Localization Methods

- Traditional object detection methods mostly relied on the sliding window technique, which was commonly used for detecting specific objects like faces and pedestrians. An alternative approach involves proposing several candidate regions in an image and filtering them to retain only true objects. Selective Search demonstrated strong performance using this approach.

Related Work

■ Transfer Learning

- R–CNN training is based on **inductive transfer learning**, where a pre–trained model from ImageNet classification is fine–tuned on the PASCAL VOC dataset, leading to significant performance improvements. This is different from the **unsupervised transfer learning** commonly seen in recent neural network research.

■ R-CNN Extensions

- R–CNN has been extended to various new tasks and datasets. It has been widely adopted in many systems, including the **ILSVRC2014 Object Detection Challenge**, where **GoogLeNet** used R–CNN to win the competition.

Object Detection with An R-CNN

■ Basic Concept

- R-CNN addresses the object detection problem by combining region proposals and CNN (Convolutional Neural Networks). First, it extracts regions from the image where objects are likely to be located, and then applies a CNN to classify the objects in those regions.

Object Detection with An R-CNN

■ Step-by-step Process

- Region Proposals: Approximately 2000 candidate regions are proposed where objects might be present in the image. This is typically done using the Selective Search algorithm.

- Feature Extraction: Each proposed region is resized to a fixed size, and a CNN is used to extract the feature vector for that region.

- Classification: The extracted feature vectors are then classified using an SVM to determine which class the object belongs to.

- Bounding Box Regression: Finally, bounding box regression is applied to accurately predict the object's location.

Object Detection with An R-CNN

■ R-CNN Performance

- R-CNN significantly improved mean Average Precision (mAP) on the PASCAL VOC dataset, showing over a 50% performance increase compared to previous methods.

■ Efficiency and Drawbacks:

- While R-CNN provides high accuracy, the processing time is slow because each region proposal is passed through the CNN independently. To address this, later studies introduced more efficient methods like Fast R-CNN and Faster R-CNN.

Analysis

■ Error Analysis

- The detection error analysis tool developed by Hoiem et al. was used to analyze the main types of errors in R-CNN.

– This tool helped break down performance and identify various errors such as localization errors, background confusion, and misclassification.

– R-CNN reduces these errors through fine-tuning, particularly improving the ability to correctly classify objects and predict their exact locations.

Analysis

■ Visualizing Learned Features

- To better understand the network's learning process, the pool5 layer of the TorontoNet network was visualized.

– The first layer is easy to visualize and interpret directly, but deeper layers become harder to understand.

– To address this, a non-parametric method was proposed, isolating specific features within the network and using them as object detectors.

Analysis

■ Visualizing Learned Features

- Fig. 4 shows the top 16 activated units in the pool5 layer, which demonstrate what the network has learned. For instance, some units focus on dog faces, which is visible in the visualization.

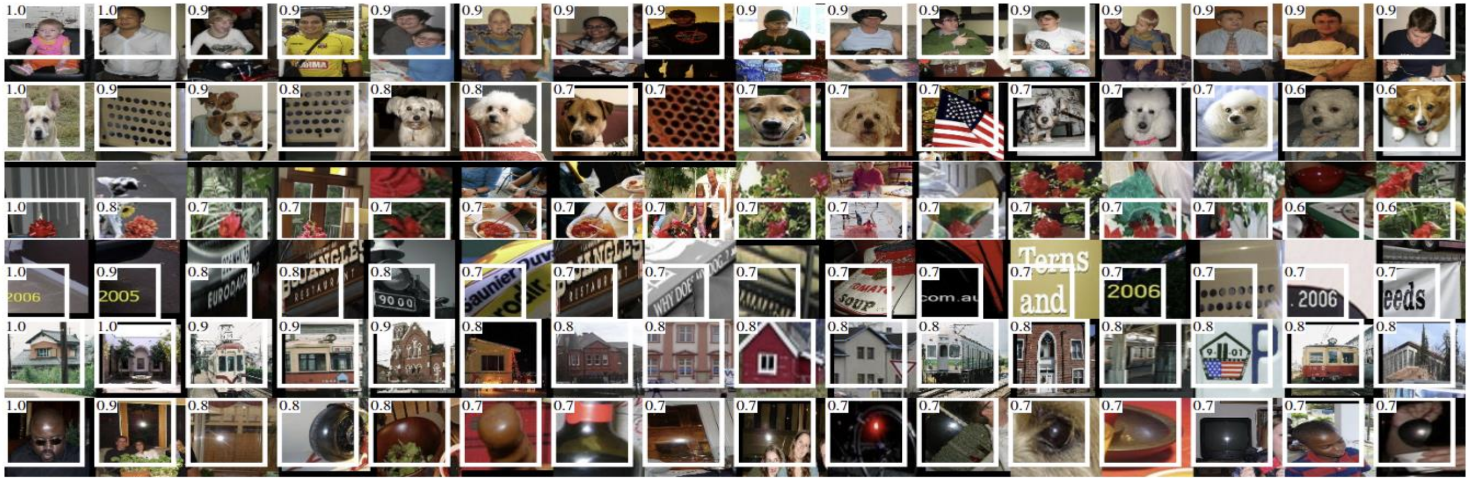


Fig. 4. Top regions for six pool_5 units. Receptive fields and activation values are drawn in white. Some units are aligned to concepts, such as people (row 1) or text (4). Other units capture texture and material properties, such as dot arrays (2) and specular reflections (6).

Analysis

■ Performance Comparison between OxfordNet and TorontoNet

- Although most of the results were based on the TorontoNet network architecture, it was found that the choice of architecture significantly affects R-CNN's performance.

- Table 3 compares the performance of TorontoNet and OxfordNet on the VOC 2007 dataset. OxfordNet, which performed well in the ILSVRC 2014 Challenge, consists of 13 convolutional layers with 3x3 kernels, 5 max-pooling layers, and 3 fully connected layers (FCN).

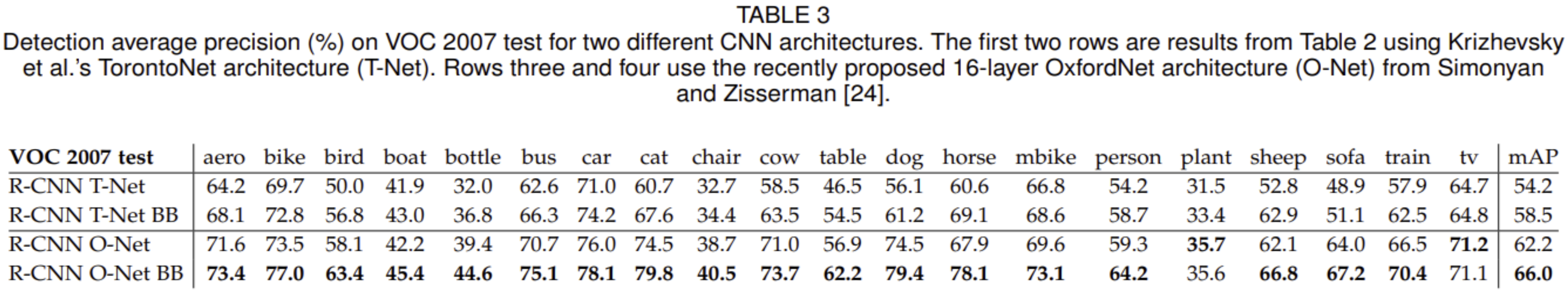


TABLE 3
Detection average precision (%) on VOC 2007 test for two different CNN architectures. The first two rows are results from Table 2 using Krizhevsky et al.'s TorontoNet architecture (T-Net). Rows three and four use the recently proposed 16-layer OxfordNet architecture (O-Net) from Simonyan and Zisserman [24].

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN T-Net	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN T-Net BB	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8	58.5
R-CNN O-Net	71.6	73.5	58.1	42.2	39.4	70.7	76.0	74.5	38.7	71.0	56.9	74.5	67.9	69.6	59.3	**35.7**	62.1	64.0	66.5	**71.2**	62.2
R-CNN O-Net BB	**73.4**	**77.0**	**63.4**	**45.4**	**44.6**	**75.1**	**78.1**	**79.8**	**40.5**	**73.7**	**62.2**	**79.4**	**78.1**	**73.1**	**64.2**	35.6	**66.8**	**67.2**	**70.4**	71.1	**66.0**

Analysis

■ Performance Gap

- Table 3 shows that OxfordNet achieved 7.5% higher mAP than TorontoNet (58.5% → 66.0%).

- However, OxfordNet's computation time is about 7 times slower than TorontoNet due to its deeper network structure.

- While performance improved, the high computational cost suggests that further research is needed to overcome R-CNN's limitations.

Analysis

■ Summary Analysis

- R–CNN achieved high performance, but there are still localization errors that need to be addressed, and fine–tuning plays a crucial role in improving performance.

– Furthermore, visualizing the network's learning patterns helps provide a clearer analysis of what CNNs have learned, and comparing OxfordNet with TorontoNet highlights how the choice of architecture impacts R–CNN's performance.

The ILSVRC2013 Detection Dataset

■ ILSVRC2013 Dataset Overview

- The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) provides a massive dataset for object detection and classification, containing images with a wide variety of objects and complex scenes. This dataset includes numerous object classes and diverse backgrounds, making it well-suited for accurately evaluating the performance of object detection systems.

The ILSVRC2013 Detection Dataset

■ R-CNN Performance on ILSVRC2013

- R-CNN demonstrated outstanding performance on this dataset, particularly in more complex images compared to datasets like PASCAL VOC. mAP (Mean Average Precision), the primary performance metric used in VOC 2007, was also used here. R-CNN showed a significant performance improvement over previous methods.

The ILSVRC2013 Detection Dataset

■ Network Architecture and Performance

- Architectures such as TorontoNet and OxfordNet were used on this dataset, with deeper and more complex networks contributing to improved performance. Specifically, R-CNN using OxfordNet achieved remarkable results, demonstrating that deeper network structures are more advantageous for detecting complex objects.

The ILSVRC2013 Detection Dataset

■ Summary of Results

- The ILSVRC2013 dataset proved that R-CNN can effectively perform object detection even in complex images, indicating that the model is well-suited for a wide range of real-world applications.

Semantic Segmentation

■ Importance of Semantic Segmentation

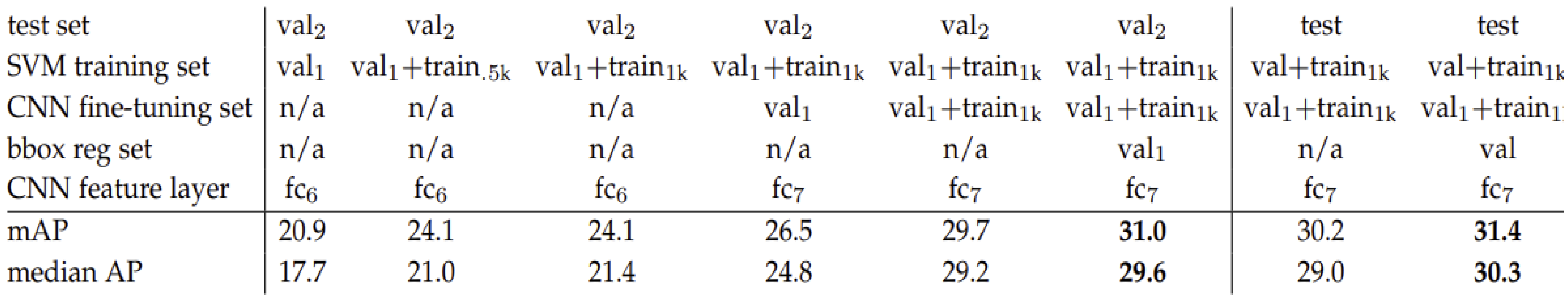
- Semantic Segmentation is the process of assigning each pixel in an image to a specific class. This allows for more accurate identification of the shape and location of objects, making it highly valuable in fields such as autonomous driving and medical image analysis.

Semantic Segmentation

■ Segmentation using R-CNN

- As shown in Table 5, R-CNN-based Semantic Segmentation demonstrated excellent performance on the VOC 2011 validation dataset. In particular, among various network structures (e.g., fg R-CNN, full R-CNN), full+fg R-CNN achieved the highest accuracy.

TABLE 4
ILSVRC2013 ablation study of data usage choices, fine-tuning, and bounding-box regression. All experiments use TorontoNet.



test set	val$_2$	val$_2$	val$_2$	val$_2$	val$_2$	val$_2$	test	test
SVM training set	val$_1$	val$_1$+train$_{.5k}$	val$_1$+train$_{1k}$	val$_1$+train$_{1k}$	val$_1$+train$_{1k}$	val$_1$+train$_{1k}$	val+train$_{1k}$	val+train$_{1k}$
CNN fine-tuning set	n/a	n/a	n/a	val$_1$	val$_1$+train$_{1k}$	val$_1$+train$_{1k}$	val$_1$+train$_{1k}$	val$_1$+train$_1$
bbox reg set	n/a	n/a	n/a	n/a	n/a	val$_1$	n/a	val
CNN feature layer	fc$_6$	fc$_6$	fc$_6$	fc$_7$	fc$_7$	fc$_7$	fc$_7$	fc$_7$
mAP	20.9	24.1	24.1	26.5	29.7	**31.0**	30.2	**31.4**
median AP	17.7	21.0	21.4	24.8	29.2	**29.6**	29.0	**30.3**

Semantic Segmentation

■ Results

- The ILSVRC2013 dataset proved that R-CNN can effectively perform object detection even in complex images, indicating that the model is well-suited for a wide range of real-world applications.

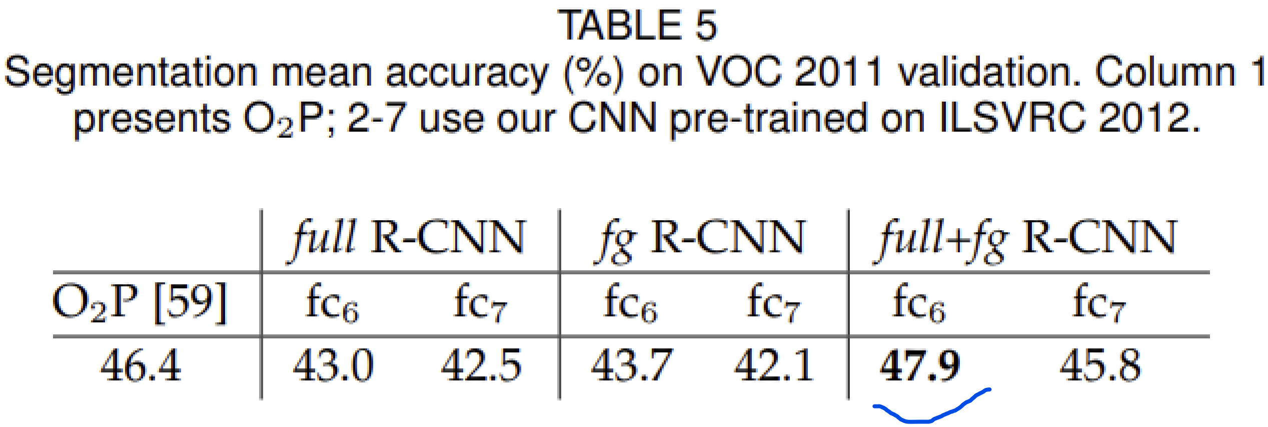


TABLE 5
Segmentation mean accuracy (%) on VOC 2011 validation. Column 1 presents O_2P; 2-7 use our CNN pre-trained on ILSVRC 2012.

	full R-CNN		*fg* R-CNN		*full+fg* R-CNN	
O_2P [59]	fc_6	fc_7	fc_6	fc_7	fc_6	fc_7
46.4	43.0	42.5	43.7	42.1	**47.9**	45.8

Implementation and Design Details

■ Object Proposal Transformations

- R-CNN resizes images to a fixed size of 227 x 227 pixels for object detection input into the CNN. Since object proposals can vary in shape and size, two methods were evaluated to convert proposals into fixed-size inputs:

1. Tightest square with context: Each object proposal is enclosed in the smallest square, maintaining its aspect ratio and including surrounding context, then resized to a fixed size.

2. Isotropic scaling: The image's aspect ratio is preserved while resizing the object to a fixed-size CNN input.

Implementation and Design Details

■ CNN Architecture

- R-CNN was implemented using both TorontoNet and OxfordNet architectures. OxfordNet, being deeper and more complex, contributed to better performance. The CNN extracts features from the fixed-size input, and those features are classified using an SVM.

Implementation and Design Details

■ Bounding Box Regression

- To improve localization performance, a simple BBox regression is used.

– After calculating the Selective Search Proposal with an SVM, a BBox regressor is applied to predict new BBoxes for detection.

– This BBox regressor is similar to the one used in DPM.

Algorithm Input: (Proposal_box, GT_box)

– Algorithm Goal: To map P to G.

– d(P): The mapping function, modeled as a linear function of the features from the 5th pooling layer of proposal P.

$$\begin{aligned}
\hat{G}_x &= P_w d_x(P) + P_x \\
\hat{G}_y &= P_h d_y(P) + P_y \\
\hat{G}_w &= P_w \exp(d_w(P)) \\
\hat{G}_h &= P_h \exp(d_h(P)).
\end{aligned}$$

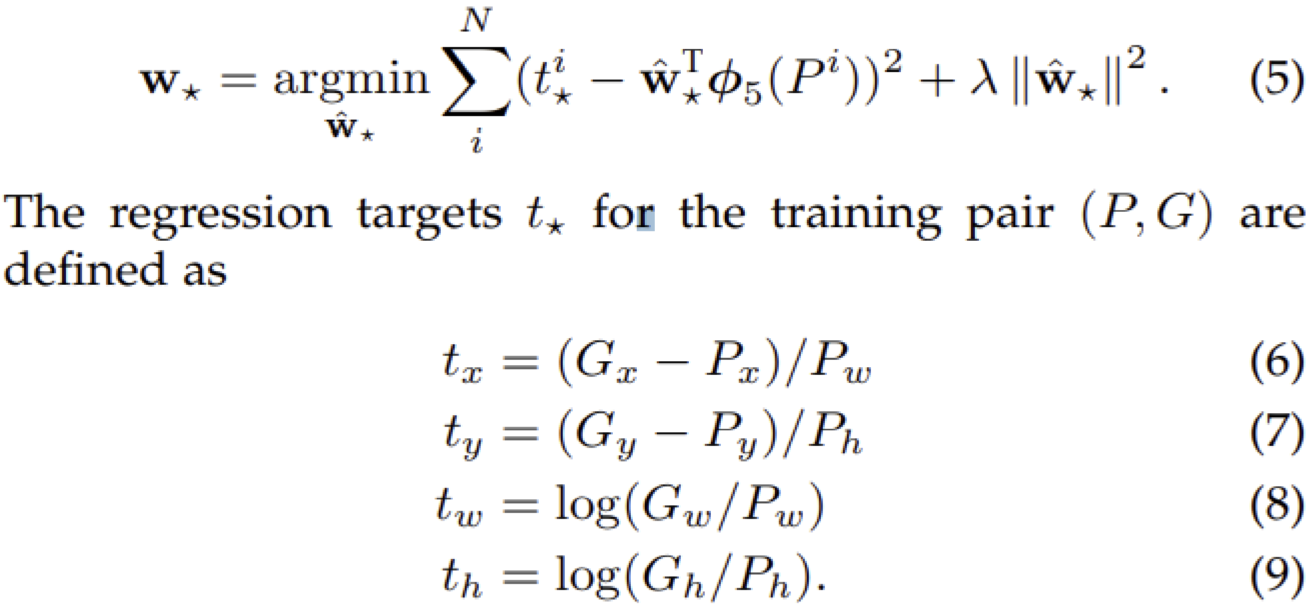
Implementation and Design Details

■ Bounding Box Regression

- The error between the predicted and GT values is calculated using MSE (Mean Squared Error).

– The argmin function is used to find the minimum value for optimization.

– Lambda (λ) is used as a regularization term.



$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_{i}^{N} (t_\star^i - \hat{\mathbf{w}}_\star^{\mathrm{T}} \phi_5(P^i))^2 + \lambda \left\| \hat{\mathbf{w}}_\star \right\|^2 . \qquad (5)$$

The regression targets $t_\star$ for the training pair (P, G) are defined as

$$t_x = (G_x - P_x)/P_w \qquad (6)$$

$$t_y = (G_y - P_y)/P_h \qquad (7)$$

$$t_w = \log(G_w/P_w) \qquad (8)$$

$$t_h = \log(G_h/P_h). \qquad (9)$$

Implementation and Design Details

■ Comparison of Segmentation Performance between R-CNN, O2P, and R&P Methods

- In Table 7, the R–CNN–based full+fg R–CNN fc6 shows segmentation performance similar to O2P while outperforming the R&P method. When comparing the performance across categories, ours (full+fg R–CNN fc6) achieved an average accuracy of 47.9%, which is very close to O2P's 47.6%, and significantly higher than R&P's 40.8%.

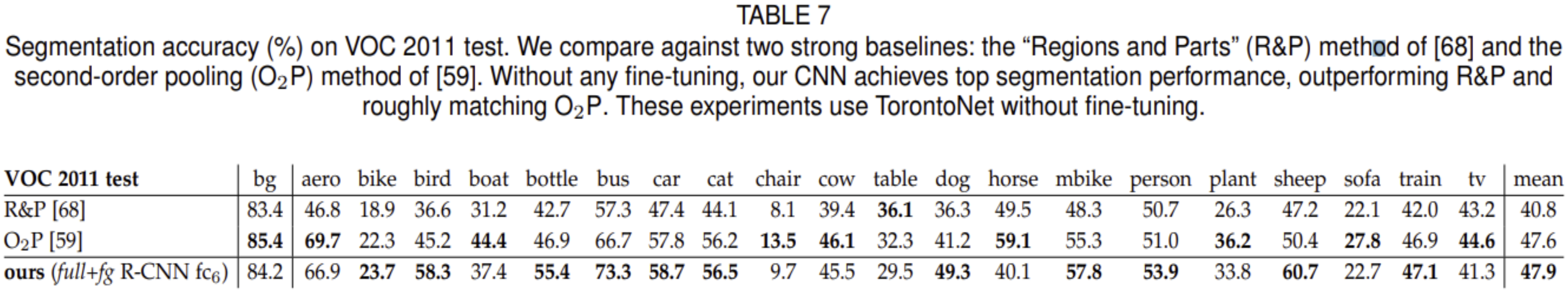


TABLE 7

Segmentation accuracy (%) on VOC 2011 test. We compare against two strong baselines: the "Regions and Parts" (R&P) method of [68] and the second-order pooling (O_2P) method of [59]. Without any fine-tuning, our CNN achieves top segmentation performance, outperforming R&P and roughly matching O_2P. These experiments use TorontoNet without fine-tuning.

VOC 2011 test	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
R&P [68]	83.4	46.8	18.9	36.6	31.2	42.7	57.3	47.4	44.1	8.1	39.4	**36.1**	36.3	49.5	48.3	50.7	26.3	47.2	22.1	42.0	43.2	40.8
O_2P [59]	**85.4**	**69.7**	22.3	45.2	**44.4**	46.9	66.7	57.8	56.2	**13.5**	**46.1**	32.3	41.2	**59.1**	55.3	51.0	**36.2**	50.4	**27.8**	46.9	**44.6**	47.6
ours (*full+fg* R-CNN fc_6)	84.2	66.9	**23.7**	**58.3**	37.4	**55.4**	**73.3**	**58.7**	**56.5**	9.7	45.5	29.5	**49.3**	40.1	**57.8**	**53.9**	33.8	**60.7**	22.7	**47.1**	41.3	**47.9**

Implementation and Design Details

■ Comparison of Segmentation Performance between R-CNN, O2P, and R&P Methods

- Specifically, in categories such as aeroplane, bottle, bus, car, cow, train, and TV monitor, ours (full+fg R-CNN fc6) outperformed both O2P and R&P methods.

The reason R-CNN outperforms both R&P and O2P is due to its superior feature extraction and CNN structure, which allow for more precise segmentation of fine-grained object areas.

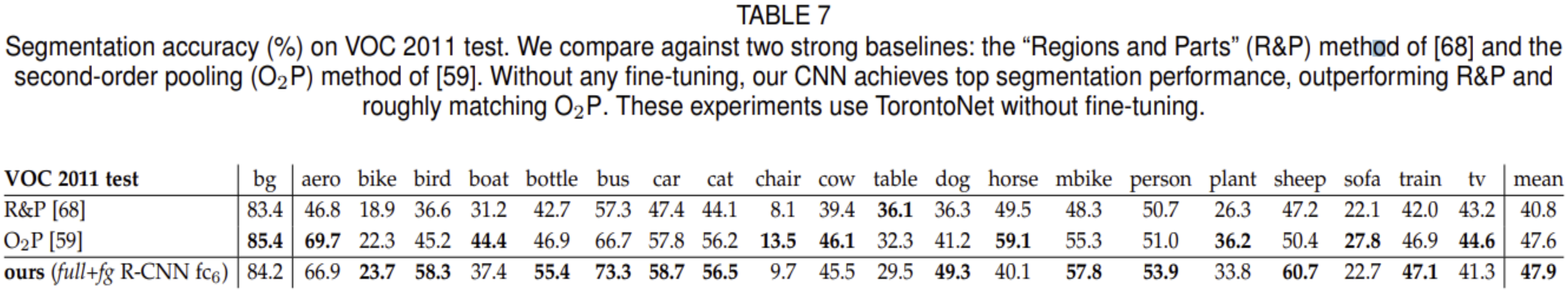


TABLE 7
Segmentation accuracy (%) on VOC 2011 test. We compare against two strong baselines: the "Regions and Parts" (R&P) method of [68] and the second-order pooling (O_2P) method of [59]. Without any fine-tuning, our CNN achieves top segmentation performance, outperforming R&P and roughly matching O_2P. These experiments use TorontoNet without fine-tuning.

VOC 2011 test	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
R&P [68]	83.4	46.8	18.9	36.6	31.2	42.7	57.3	47.4	44.1	8.1	39.4	**36.1**	36.3	49.5	48.3	50.7	26.3	47.2	22.1	42.0	43.2	40.8
O_2P [59]	**85.4**	**69.7**	22.3	45.2	**44.4**	46.9	66.7	57.8	56.2	**13.5**	**46.1**	32.3	41.2	**59.1**	55.3	51.0	**36.2**	50.4	**27.8**	46.9	**44.6**	47.6
ours (*full+fg* R-CNN fc_6)	84.2	66.9	**23.7**	**58.3**	37.4	**55.4**	**73.3**	**58.7**	**56.5**	9.7	45.5	29.5	**49.3**	40.1	**57.8**	**53.9**	33.8	**60.7**	22.7	**47.1**	41.3	**47.9**

Conclusion

■ Introduction of R-CNN and Performance Improvements

- Previous models that achieved the best performance in object detection were complex ensemble models, which combined low-level image features with high-level contextual information.

– However, the R-CNN introduced in this paper is a simpler and more scalable object detection model, achieving approximately a 50% performance improvement over the previous best models on the PASCAL VOC 2012 benchmark.

Conclusion

■ Key Ideas of R-CNN

- Combining CNN and Selective Search to use bottom–up region proposals for efficient object localization.

– Supervised pre–training and domain–specific fine–tuning: Utilizing large amounts of data (image classification) to pre–train the CNN, followed by fine–tuning the network for object detection tasks where labeled data is scarce.

Conclusion

■ Limitations of R-CNN and Future Research Directions

- Although R-CNN offers high accuracy, it suffers from limitations in processing time. To address this, further research introduced Fast R-CNN and Faster R-CNN. Future work should focus on balancing speed and accuracy, and exploring deeper networks to further enhance performance.

■ Conclusion

- This conclusion highlights R-CNN's revolutionary contributions to object detection, and suggests that this model will serve as a foundation for future research in the field.