Novel Single Stage Detectors for Object Detection

Jian Huang
Stanford University
jhuang33@stanford.edu

Danyang Wang
Stanford University
danyangw@stanford.edu

Xiaoshi Wang
Stanford University
xiaoshiw@stanford.edu

Abstract

Most of the recent successful methods in accurate object detection utilized some variants of R-CNN style two stage Convolutional Neural Networks (CNN) in which plausible regions were proposed in the first stage followed by a second stage for decision refinement. These methods are accurate but hard and slow to train. Single stage detection methods, on the other hand, enjoy the high speed of training and the efficiency in deployment. But they have not been as competitive as two stage methods in terms of accuracy such as mAP for high IoU threshold. Recently, Recurrent Rolling Convolution (RRC) architecture, a novel single stage end-to-end object detection network over multi-scale feature maps to construct object classifiers and bounding box regressors, was proposed. The RRC model has achieved state-of-the-art performance in some tasks. In our project, we introduce Backward Recurrent Rolling Convolution (BRRC) based on RRC, and show that BRRC is able to produce better results and meanwhile faster than original RRC. We also investigate SSD with more bounding boxes and introduce an encoder-decoder structure, Detection SegNet, for object detection. We evaluate and compare all these models based on IoU scores.

1. Introduction

Object detection is a crucial task for computer vision. In many vision applications, robustly detecting objects with high localization accuracy is important to the quality of service. For instance, in advanced driver assistance systems(ADAS), accurately detecting cars and pedestrians plays a crucial rule on the safety of the autonomous actions.

For objection detection, there are two types of methods applying convolutional neural nets(CNN) that are popular in recent years. First type is two stage CNNs where first stage is region proposal and second stage is decision refinement. Some well-known methods include the R-CNN[5], fast R-CNN[4] and faster R-CNN[16] sequence and their variations. These models have relatively high accuracy on detection accuracy but are slow. Even for faster-RCNN, it can only process 7 frames per second(FPS), which is not enough for real-time application. The other type of methods is based on single stage neural nets without the procedure of region proposal. Some examples are SSD[12] and YOLO[13]. These methods are faster than two stage methods but prone to low accuracy when the requirement of Intersection over Union (IoU) is high.

In order to create a model with both high accuracy and high speed, it is important to analyze the reason behind the low accuracy of single stage models. After experiments, it is shown that most of the low quality bounding boxes come from the failure localization of either small objects or overlapping objects. The idea here is single model usually use high resolution feature maps to detect small objects. However, high resolution feature maps may not be deep enough to include information needed for detection. A recent paper by Ren et al. [14] applied a method called "Recurrent Rolling Convolution" (RRC) to merge information from upper layer and lower layer feature maps to current feature map based on SSD architecture and achieved good performance, which is interesting.

In our project, we modify RRC with only backward rolling, and show that our model achieves better accuracy and runs moderately faster than original RRC model. We also investigate SSD model with more bounding boxes and introduce the encoder-decoder structure for object detection. Although performance of the latter two is not as good as BRRC, they are much more efficient than BRRC in terms of computation speed.

2. Related Work

Convolutional neural network approaches with a region proposal stage have recently been popular in the area of object detection and have achieved very successful results. R-CNN[5] used selective search[22] to generate object proposals , and CNN to extract and feed features to the

classifier. Fast R-CNN[4] and faster R-CNN[16] were later proposed to accelerate R-CNN. In [4], RoI pooling was used to efficiently generate features for object proposals, whereas in [16], CNN was used to perform region proposal instead of selective search. A number of variants of [16] were proposed and performs well in benchmarks considering mAP for high IoU threshold.

However, one problem with R-CNN based methods[5, 23, 26] is the heavy computation in the second stage due to the process of a large number of proposals. Various single stage methods[17, 12, 13] which do not rely on region proposals have been proposed to solve this problem. For example, SSD[12] is a single stage model in which the feature maps with different resolutions in the feed-forward process were directly used to detect objects with sizes of a specified range. It performed much faster than [16] and achieved good results. YOLO[13] is another fast single stage method which achieved promising results.

Though Recurrent Neural Networks (RNN) has been widely adopted in many areas such as machine translation[20], image captioning[8, 25] and multimedia[15, 2], the idea of applying sequence modeling to improve object detection accuracy has not been actively explored by researchers. RRC architecture was one of the first models that explored sequence modeling on object detection in which every object was efficiently detected by a network which is deep in context and achieved state-of-the-art performance under a high IoU threshold. Our proposed model utilizes RRC method but only considers backward rolling, and we can show that our model is able to achieve comparable(better) results and run reasonably faster.

3. Approaches

In this section we would like to introduce the three models we come up with to achieve high accuracy and speed for image detection in advanced driver assistance systems domain.

3.1. SSD with More Bounding Boxes

SSD[12] model is introduced by Liu et al in 2016. As a detection model, the inputs are preprocessed images and outputs are a set of purposed bounding boxes. Each bounding box contains position of the bounding boxes(4) and classification result for the bounding box(one-hot coding, for KITTI dataset we only care Car class). The loss function is the following:

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

Where N is the number of matched default boxes, x is a indicator to match predicted bounding boxes with ground truth boxes, c are classes confidences in classification, l and g are predicted and ground truth bounding boxes respectively. L_{loc} is L1 loss on bounding box regression and L_{conf} is cross-entropy loss on classification. Figure1 shows the architecture of the SSD network. SSD contains a reduced VGG-16[19] network, on top of which are several added convolution layers. These newly added convolution layers, as well as the last layer in VGG net are served as the source of bounding box generation for object detection.

Labels and losses for SSD are generated in the following manner: for a feature map in detection group, e.g. a $4 * 4$ map, we split the original image into $4 * 4$ cells and for each cell we create a set of default bounding boxes centered around the cell. This set of default bounding boxes corresponds to one data point in the feature map. Then we compare the default bounding boxes and ground truth bounding boxes, if the Intersection over Union(IoU) between them is larger than 0.7, we let the default box be a positive example. For its labeling, the class of it is Car and the 4 positions are the offset between it and the corresponding ground truth box. The other bounding boxes are negative examples, with class be Background. In training, for the feature map, we apply 6*(num class + 4) 3*3 filters to convolve the feature map and get all the outputs(dimension is 4(position offsets) + num classes) for all bounding boxes and use these outputs to compute loss. During testing, we use the trained set of detection feature maps to detect bounding boxes.

One straightforward idea to improve detection accuracy with less sacrifice on speed is to add more default bounding boxes on original SSD model. The intuition is with more default bounding boxes, the probability that one of them being closer to a ground truth bounding box is larger, making the positive examples easier to train. The default number of bounding boxes per feature map in SSD is 4 or 6. We decide to expand the number to 8 for our new model.

3.2. Backward Recurrent Rolling Convolution

Backward Recurrent Rolling Convolution (BRRC) is based on RRC model. First we would like to give a brief introduction of the original RRC model and then highlight the difference between our model and RRC.

Ren et al's RRC model is based on SSD[12] with the input, output and loss function setting are all the same. Following SSD, the backbone of RRC network architecture is reduced VGG-16[19] network. However, different from SSD, on top of the VGG net is a set of layers for detection connected in the manner of "The Recurrent Rolling Convolution". This set of feature maps are connected by

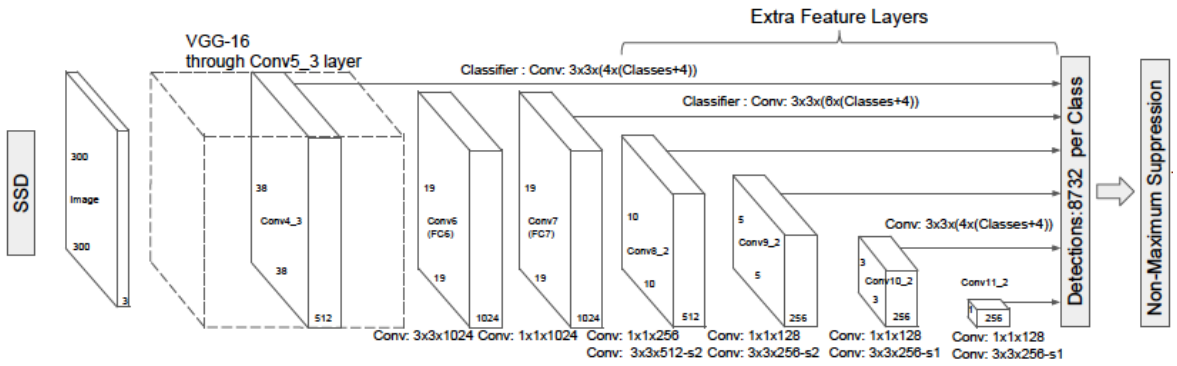

Figure 1. A visualization of SSD network, figure from[12].

convolution and the resolutions are decreasing. During training, for each feature map in the "Recurrent Rolling Convolution" set, the information of its previous and next feature map will be conveyed to it using convolution plus pooling (previous) or convolution plus deconvolution (next) in a iterative manner. Through this mechanism lower level feature maps are able to get information from higher level feature maps, which is necessary for detection small objects. Each feature map is in charge of detecting objects of a certain scale range. Figure2 shows the visualization of the "Recurrent Rolling Convolution" set and information convey mechanism.

RRC used bi-directional rolling convolution, namely padding lower layers to upper layers and vice versa. Its idea of backward padding is to add "abstract" information to lower level feature maps to solve the problem that SSD is not able to provide good detection on small objects. The idea for forward padding is to add "context" info from lower level feature maps to upper level maps. We argue that backward padding is useful but the forward padding is redundant because the context information should be conveyed to upper layer from lower layers in the normal forward convolution path if we correctly train the model. The forward padding also makes the net harder and slower to train. Thus we introduce Backward Recurrent Rolling Convolution, only keep the backward path and throw away the forward path. Figure 3 shows the net structure of BRRC.

3.3. Encoder-Decoder Structure for Image Detection

After finishing the BRRC experiment, we find although it is faster than the original RRC, the speed is still a lot slower than SSD. In order to further speed up the network and keep relative high accuracy in detection, we come up with the idea to refer to some works in image segmentation. As mentioned in previous sections, one important problem of SSD we would like to solve is the lack of abstraction information when using lower level feature maps to detect small objects. The encoder-decoder structure in image segmentation structures can provide a solution for this. The reason is that the deeper the layer, the larger the resolution and more abstracted the information, making it suitable for detecting small objects.

One of the state of art works in image segmentation with encoder-decoder system is SegNet[1]. SegNet was introduced by Badrinarayanan et al in 2015 and achieved state of art results in image segmentation. Figure4 is a visualization of SegNet, the encoder part of SegNet architecture is based on VGG-16[19] network and the decoder part is the mirror of encoder.

For our project, we created Detection SegNet by changing the decoding part of original SegNet to have 3 feature map sizes instead of 5 and applied the last feature map for each map size to perform detection task. The setting for input, output and detection part is similar to SSD and RRC.

4. Experiments

4.1. Datasets

We use the challenging KITTI Object Detection Evaluation dataset [3] to evaluate our model. The dataset consists of 7,481 training images and 7,518 test images, comprising a total of 80,256 labeled objects. All images are colored with png format. They are taken in various real world road settings with multiple objects of different sizes and scales, occlusions and different lighting conditions. The training images are labeled out as images with cars, pedestrians, cyclists or just background with bounding boxes. The dataset and benchmarks is widely used in autonomous driving object detection researches.

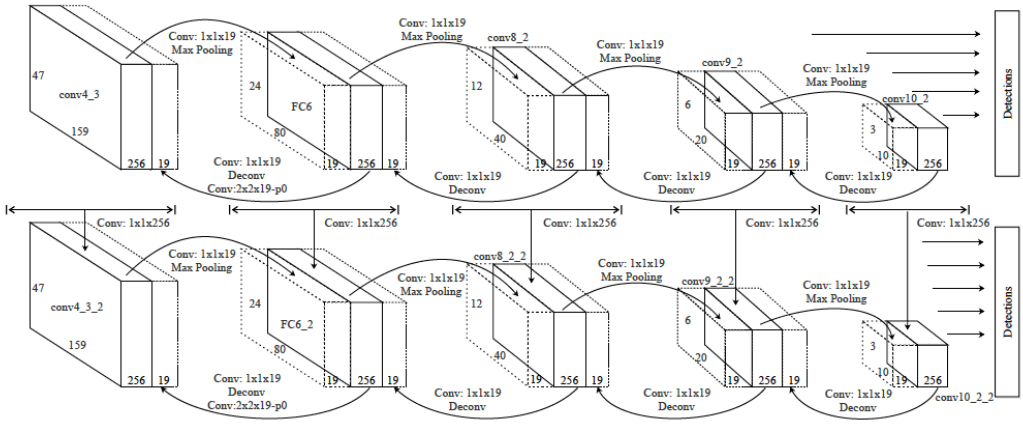

Figure 2. Visualization of Recurrent Rolling Convolution mechanism and detection feature maps. Figure from Ren et al's paper

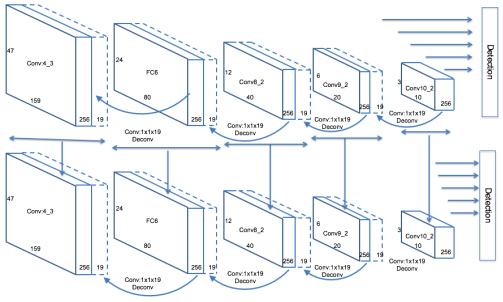

Figure 3. Visualization of Backward Recurrent Rolling Convolution net structure

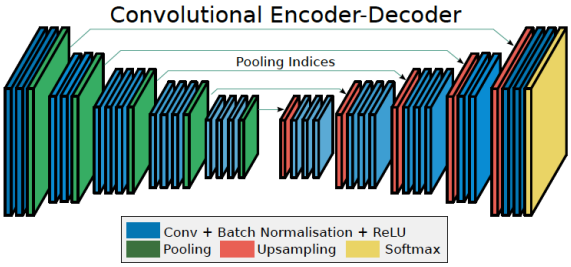

Figure 4. A visualization of SegNet, figure from[1].

4.2. Training and Testing

We implemented the three models in Caffe framework[7] based on the structures and source code in the SSD[11], RRC[24] and SegNet[9] papers. We applied a pre-trained model of the fully convolutional reduced (atrous) VGGNet[10] in order to make the whole networks easier to train and converge. First we over-fit each model on a small training dataset for about 250 iterations to achieve a detection IoU score of more than 0.7 to show the correctness of models, followed by formal training and testing. The KITTI dataset images are of resolution 2560 by 768 which can take a lot of computation time and memory to train (For reference, the dataset images used in the SSD[12] paper are 300 by 300). Due to hardware limitation of only one Nviadia Tesla K80 GPU with around 12 GB of memory, we trained each of our models and the RRC baseline model with a training set of 200 images, validation set of 50 images for 1000 iterations and batch size of 1. After training we generate bounding boxes on a test image set of size 500. Training with a larger dataset for more iterations is likely to result in better test results. However, given the available computation resources and budget it will take more than a week to train the whole KITTI dataset of more than 7000 images for 60000 iterations as the original RRC[14] paper did. We believe our experiments with smaller dataset and less iterations is enough to demonstrate the characteristics of each models and make comparisons with the RRC model under the same training and testing conditions as presented in the next section.

The following settings are used throughout the experiments. For BRRC network architecture, we do BRRC for 5 times in training and assign 5 separate regressors for each corresponding feature map. For learning, stochastic gradient descent (SGD) with momentum of 0.9 is used for optimization. Weight decay is set to 0.0005 and we set the initial learning rate to 0.0005. For evaluation, we use Intersection over Union (IoU) to evaluate the performance of each model:

$$IoU = \frac{S_I}{S_U}$$

where S_I is the area of overlap between ground truth bounding box and predicted bounding box and S_U is the area of union between ground truth bounding box and predicted bounding box.

Model	IoU	Speed(Sec/iter)
SSD+Box	0.186	8
SegNet	0.349	18
RRC	0.510	68
BRRC	0.548	62

Table 1. Performance of different models

4.3. Results and Analysis

We evaluate performance of SSD, Detection SegNet, original RRC and BRRC models in terms of IoU scores and speed. The results are shown in Table 1.

As we can see from Table 1, there is in general a trade-off between speed and performance. However as we expected, the speed and accuracy for BRRC are both better than RRC. This proves our hypothesis that the forward path of recurrent rolling convolution is redundant to some extent.

Figure 5 shows two set of results from testing set. From top to bottom are results of SSD+bounding box, Detection SegNet, RRC and BRRC. These are typical results we get after we checked all testing results. For SSD results, we find that it is hard for the net to learn positions and sizes of the ground truth bounding boxes. The boxes it generated are similar to default bounding boxes. In addition, it cannot detect relatively small objects, which support the claim that it is hard for lower level feature maps to have enough abstarct information to detect small objects. For Detection SegNet results, as we can see, the net is able to fairly and precisely locate the position of both small and large objects. This shows the addition of decoder layers does solve the problem of "lower level features maps cannot provide good detection results of small objects" to some extent. But the network seems to have some difficulties learning the correct sizes of the objects. This might due to the fact that we changed the default setting to delete all batch normalization layers in the network in order to be comparable to the architectures of SSD, RRC and BRRC. It could be the case that the lack of batch normalization prevents the network from good convergence. We are interested to see the results after adding batch normalization in this setting. For RRC and BRRC results, both the two networks are able to locate the objects and predict the sizes of the objects. In general BRRC is able to predict more accurate bounding box sizes than original RRC. There exists possibility that the lower accuracy is due to the fact that RRC converges slower because of more complex network structure and more parameters. However, the more than three percentage of advantage in accuracy and faster in speed do show that the BRRC model is a competitive model for image detection and at least comparable to some state of art models.

5. Conclusion

In this project, we designed three models, namely SSD+bounding boxes, BRRC and Detection SegNet for object detection, especially in the domain of advanced driver assistance systems(ADAS). We find that RRC based model outperforms SSD and Detection SegNet methods in terms of IoU accuracy but is computationally more expensive. The BRRC model proposed by us is able to beat the original RRC model, a state of art model in object detection, in both detection accuracy and speed.

There are several things we would like to do in future. Firstly, since all three of our models use pretrained VGGNet[18] as the base model, we would like to try other base architectures, for example GoogLeNet[21] and ResNet[6]. Secondly, the computational capacity limitation and the high resolution of KITTI dataset for now restricts us for running large iterations on whole dataset. We really want to have a chance to run our models on whole datasets for around 60,000 iterations(the number of iterations in RRC paper) with more GPUS(4) to see the complete results. In addition, for speeding up network and keep all nets with the same setting, we have not include batch normalization in all three models, especially Detection SegNet. In future we would like to add batch normalization and compare the difference in training procedure and testing results.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[4] R. Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 580–587, Washington, DC, USA, 2014. IEEE Computer Society.

[6] 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.

[7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[8] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings*

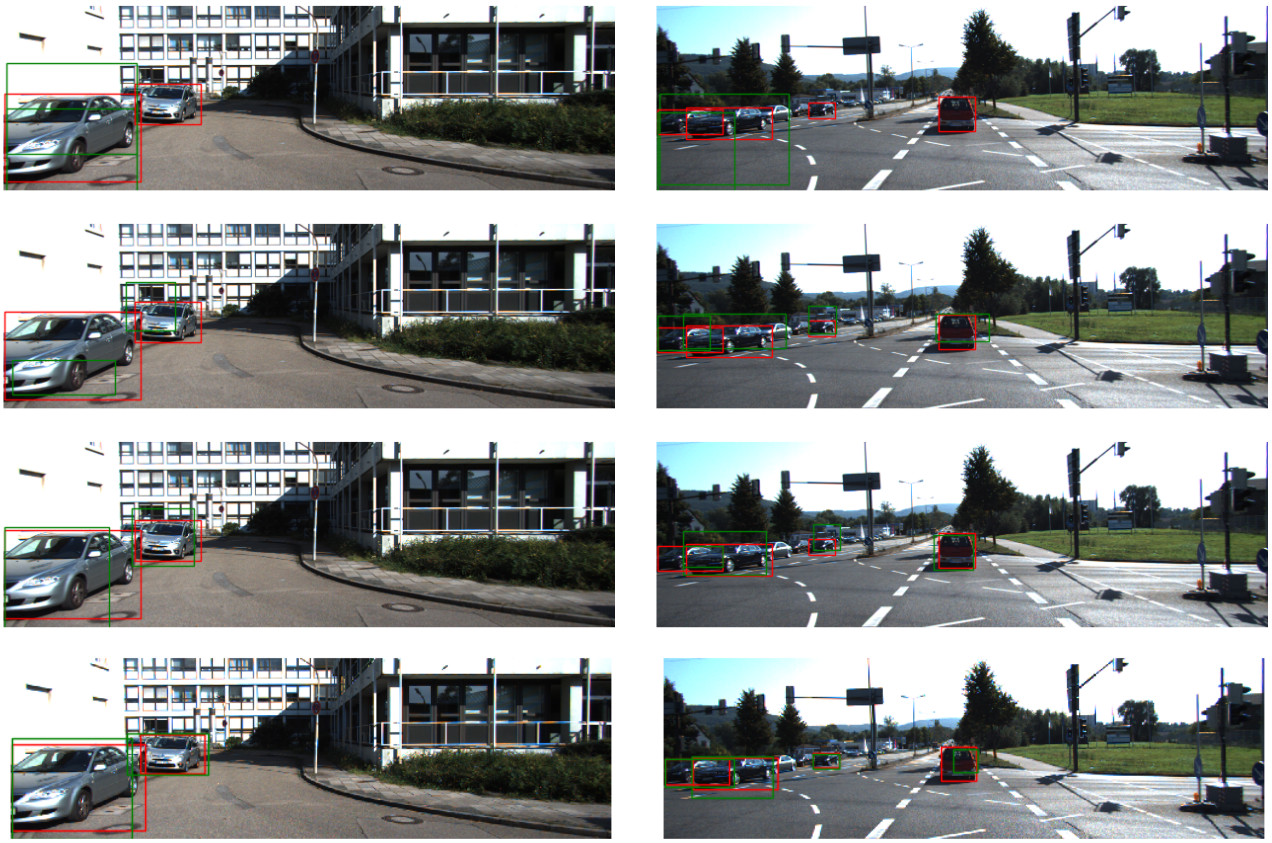

Figure 5. Detection result of test images. The red bounding boxes are ground truth labels and the green bounding boxes are the predicted bounding boxes with highest scores. Images from top to bottom correspond to SSD+bounding box, Detection SegNet, RRC and BRRC result

of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3128–3137, 2015.

[9] A. Kendall. Implementation of segnet: A deep convolutional encoder-decoder architecture for semantic pixel-wise labelling. `https://github.com/alexgkendall/caffe-segnet`.

[10] W. Liu. Fully convolutional reduced (atrous) vggnet. `https://gist.github.com/weiliu89/2ed6e13bfd5b57cf81d6`.

[11] W. Liu. Ssd: Single shot multibox detector. `https://github.com/weiliu89/caffe/tree/ssd`.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. 2016. To appear.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu. Accurate single stage detector using recurrent rolling convolution. *arXiv.org, 1704.05776*, 2017.

[15] J. Ren, Y. Hu, Y.-W. Tai, C. Wang, L. Xu, W. Sun, and Q. Yan. Look, listen and learn-a multimodal lstm for speaker identification. *arXiv preprint arXiv:1602.04364*, 2016.

[16] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[17] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

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

[19] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2015.

[20] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[22] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[23] Y. Xiang, W. Choi, Y. Lin, and S. Savarese. Subcategory-aware convolutional neural networks for object proposals and detection. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 924–933. IEEE, 2017.

[24] C. Xiaohao. Accurate single stage detector using recurrent rolling convolution. `https://github.com/xiaohaoChen/rrc_detection`.

[25] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

[26] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2016.