CRAFT Objects from Images

Bin Yang[1] Junjie Yan[2] Zhen Lei[1]* Stan Z. Li[1]
[1]National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
[2]Tsinghua University
{bin.yang, zlei, szli}@nlpr.ia.ac.cn yanjunjie@outlook.com

Abstract

Object detection is a fundamental problem in image understanding. One popular solution is the R-CNN framework [15] and its fast versions [14, 27]. They decompose the object detection problem into two cascaded easier tasks: 1) generating object proposals from images, 2) classifying proposals into various object categories. Despite that we are handling with two relatively easier tasks, they are not solved perfectly and there's still room for improvement.

In this paper, we push the "divide and conquer" solution even further by dividing each task into two sub-tasks. We call the proposed method "CRAFT" (Cascade Region-proposal-network And FasT-rcnn), which tackles each task with a carefully designed network cascade. We show that the cascade structure helps in both tasks: in proposal generation, it provides more compact and better localized object proposals; in object classification, it reduces false positives (mainly between ambiguous categories) by capturing both inter- and intra-category variances. CRAFT achieves consistent and considerable improvement over the state-of-the-art on object detection benchmarks like PASCAL VOC 07/12 and ILSVRC.

1. Introduction

The problem definition of object detection is to determine where in the image the objects are and which category each object belongs to. The above definition gives us a clue of how to solve such a problem: by generating object proposals from an image (where they are), and then classifying each proposal into different object categories (which category it belongs to). This two-step solution matches to some extent with the attentional mechanism of humans seeing things, which is to first give a coarse scan of the whole scenario and then focus on regions of our interest.

As a matter of fact, the above intuitive solution is where the research community is moving forward for years. Recently, the two steps (proposal generation and object classification) have been solved quite satisfactorily by two advances in computer vision: first is the introduction of general object proposals, second is the revival of the Convolutional Neural Networks (CNN). The general object proposal algorithm (e.g., Selective Search [34] and EdgeBox [38]) can provide around 2000 proposals per image to cover most of the objects and make the employment of more complex classifier for each proposal possible. The prosperity of the Convolutional Neural Networks (CNN) comes from its rich representation capacity and powerful generalization ability in image recognition, which is proved in challenging ImageNet classification task [20, 31, 29]. With the off-the-shelf methods available, the seminal work R-CNN [15] shows that Selective Search based region proposals plus the CNN based object classifier can achieve very promising performance in object detection. The R-CNN framework is further improved by Fast R-CNN [14] and Faster R-CNN [27], while the former enables end-to-end learning of the whole pipeline, and the latter introduces the Region Proposal Network (RPN) to get object proposals of higher quality.

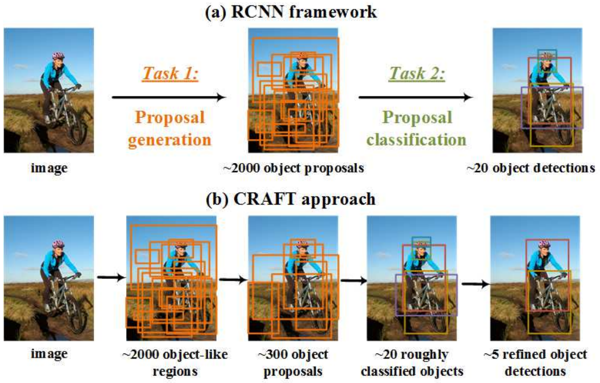

Figure 1. Overview of the widely used two-step framework in object detection, and the proposed CRAFT pipeline.

*Corresponding author.

Although the R-CNN framework achieves superior performance on benchmarks like PASCAL VOC, we discover quite large room for improvement after a detailed analysis of the result on each task (proposal generation and classification). We claim that there exists an offset between current solution and the task requirement, which is the core problem of the popular two-step framework. Specifically, in proposal generation, the task demands for proposals for only objects, but the output of general object proposal algorithms still contains a large proportion of background regions. In object classification, the task requires classification among objects, while practically in R-CNN it becomes classification among object categories plus background. The existence of many background samples makes the feature representation capture less intra-category variance and more inter-category variance (ie, mostly between the object category and background), causing many false positives between ambiguous object categories (eg, classify tree as potted plant).

Inspired by the "divide and conquer" strategy, we propose to further divide each task via a network cascade to alleviate the above issues (see Figure 1 for an illustration). Practically, in proposal generation task, we add another CNN based classifier to distinguish objects from background given the output of off-the-shelf proposal algorithm (eg, Region Proposal Network); and in object classification task, since the N+1 class (N object categories plus background) cross-entropy objective leads the feature representation to learn inter-category variance mainly, we add a binary classifier for each object category in order to focus more on intra-category variance. Through delicate design of the cascade structure in each task, we discover that it helps a lot: object proposals are more compact and better localized, while the detections are more accurate with fewer false positives between ambiguous object categories.

As a result, the object detection performance gets improved by a large margin. We show consistent and considerable gain over the Faster R-CNN baseline in object detection benchmark PASCAL VOC 07/12 as well as the more challenging ILSVRC benchmark.

The remainder of the paper is organized as follows. We review and analyze related works in Section 2. Our CRAFT approach is illustrated in Section 3 and validated in Section 4 respectively. Section 5 concludes the paper.

2. Related work

CRAFT can be seen as an incremental work built upon the state-of-the-art two-step object detection framework. In order to give readers a full understanding of our work and the underlying motivation, in this section we first review the development of the two-step framework from the "divide and conquer" perspective. We introduce in turn the significant advances in proposal generation and object classification respectively. After a summary of the building stones, we briefly introduce some related works that also try to improve upon the state-of-the-art two-step framework and also show our connection with them.

2.1. Development of the two-step framework

Proposals are quite important for object detection and diverse methods for object proposal generation are proposed. In case of detecting one particular category of near rigid objects (like faces or pedestrians) with fixed aspect ratio, sliding window mechanism is often used [23, 28, 35]. The main disadvantage is that the number of candidate windows can be about $O(10^6)$ for an image, therefore limiting the complexity of classifier due to efficiency issues. When it comes to generating proposals covering general objects of various categories and in various shapes, sliding window approach becomes more computationally expensive.

Many works are proposed to get more compact proposals, which can be divided into two types: the unsupervised grouping style and the supervised classification style. The most popular method in grouping style is the Selective Search [34], which hierarchically groups super-pixels generated through [10] to form general object proposals. Other typical grouping style proposal methods include the EdgeBox [38] which is faster and MCG [1] which is more compact. With around 2000 proposals kept for each image, a recall rate of 98% on Pascal VOC and 92% on ImageNet can be achieved. Besides the smaller number of proposals, another advantage of grouping style over sliding window is that proposals at arbitrary scale and aspect ratio can be generated, which provides much more flexibility. Many works have been proposed for further improvement and an evaluation can be found in [16].

In the supervised camp, the proposal generation problem is defined as a classification and/or regression problem. Typical methods include the BING [4] and Multi-box [32, 8]. The BING uses the binary feature and SVM to efficiently classify objects from background. The Multi-box uses CNN to regress the object location in an end-to-end manner. A recently proposed promising solution is the Region Proposal Network (RPN) [27], where a multi-task fully convolutional network is used to jointly estimate proposal location and assign each proposal with a confidence score. The number of proposals is also reduced to be less than 300 with higher recall rate. We use the RPN as the baseline proposal algorithm in CRAFT.

Given object proposals, detection problem becomes an object classification task, which involves representation and classification. Browsing the history of computer vision, the feature representation is becoming more and more sophisticated, from hand-craft Haar [35] and HOG [7] to learning based CNN [15]. Built on top of these feature representations, carefully designed models can be incorporated. The two popular models are the Deformable Part Model (DPM [9]) and the Bag of Words (BOW [25, 3]). Given the feature representation, classifiers such as Boosting [11] and SVM [5] are commonly used. Structural SVM [33, 18] and its latent version [37] are widely used when the problem has a structural loss.

In recent three years, with the revival of CNN [20], CNN based representation achieves excellent performance in various computer vision tasks, including object recognition and detection. Current state-of-the-art is the R-CNN approach. The Region-CNN (R-CNN) [15] is the first to show that Selective Search region proposal and the CNN together can produce a large performance gain, where the CNN is pre-trained on large-scale datasets such as ImageNet to get robust feature representation and fine-tuned on target detection dataset. Fast R-CNN [14] improves the speed by sharing convolutions among different proposals [19] and boosts the performance by multi-task loss (region classification and box regression). [27] uses Region Proposal Network to directly predict the proposals and makes the whole pipeline even faster by sharing full-image convolutional features with the detection network. We use the Fast R-CNN as the baseline object classification model in CRAFT.

2.2. Improvements on the two-step framework

Based on the two-step object detection framework, many works have been proposed to improve it. Some of them focus on the proposal part. [24, 36] find that using the CNN to shrink the proposals generated by grouping style proposals leads to performance gain. [12, 21] use CNN cascade to rank sliding windows or re-rank object proposals. CRAFT shares both similarities and differences with these methods. The common part is that we both the "cascade" strategy to further shrink the number of proposals and improve the proposal quality. The discrepancy is that those methods are based on sliding window or grouping style proposals, while ours is based on RPN which already has proposals of much better quality. We also show that RPN proposals and grouping style proposals are somewhat complementary to each other and they can be combined through our cascade structure.

Some other works put the efforts in improving the detection network (R-CNN and Fast R-CNN are popular choices). [13] proposes the multi-region pipeline to capture fine-grained object representation. [2] introduces the Inside-Outside Net, which captures multi-scale representation by skip connections and incorporates image context via spatial recurrent units. These works can be regarded as learning better representation, while the learning objective is not changed. In CRAFT, we identify that current objective function in Fast R-CNN leads to flaws in the final detections, and address this by cascading another complementary objective function. In other words, works like [13, 2] that aim to learn better representation are orthogonal to our work.

In a word, guided by the "divide and conquer" philosophy, we propose to further divide the two steps in current state-of-the-art object detection framework, and both tasks are improved considerably via a delicate design of network cascade. Our work is complementary to many other related works as well. Besides these improvements built on the two-step framework, there are also some works [22, 30, 26] on end-to-end detection framework that drops the proposal step. However, these methods work well under some constrained scenarios but the performance drops notably in general object detection in unconstrained environment.

3. The CRAFT approach

In this section we explain why we propose CRAFT, how we design it and how it works. Following the proposal generation and classification framework, we elaborate in turn how we design the cascade structure based on the state-of-the-art solutions to solve each task better. Implementation details are presented as well.

3.1. Cascade proposal generation

3.1.1 Baseline RPN

An ideal proposal generator should generate as few proposals as possible while covering almost all object instances. With the help of strong abstraction ability of CNN deep feature hierarchies, RPN is able to capture similarities among diverse objects. However, when classifying regions, it is actually learning the appearance pattern of an object that distinguishes it from non-object (such patterns may be colorful segments, sharp and closed edges). Therefore its outputs are actually object-like regions. The gap between object-like regions and the demanded output – object instances – makes room for improvement. In addition, due to the resolution loss caused by CNN pooling operation and the fixed aspect ratio of sliding window, RPN is weak at covering objects with extreme scales or shapes. On the contrast, the grouping style methods are complementary in this aspect.

To analyze the performance of the RPN method, we train a RPN model based on the VGG_M model (defined in [29]) using PASCAL VOC 2007 train+val and show its performance in Table 1. The recall rates in the table are calculated with 0.5 IoU (intersection of union) criterion and 300 proposals per image on the PASCAL VOC 2007 test set. The overall recall rate of all object categories is 94.87%, but the recall rate on each object category varies a lot. In accordance with our assumption, objects with extreme aspect ratio and scale are hard to be detected, such as boat and bottle. What's more, objects with less appearance complexity, or

aero	bike	**bird**	**boat**	**bottle**
95.44	98.81	**93.90**	**92.78**	**80.38**
bus	car	cat	**chair**	cow
98.12	96.00	99.16	**91.80**	99.18
table	dog	horse	mbike	persn
95.15	99.59	97.70	96.31	95.49
plant	sheep	sofa	train	**tv**
86.87	98.76	98.74	97.52	**90.58**

Table 1. Recall rates (%) of different classes of objects on VOC2007 test set, using 300 proposals from a Region Proposal Network for each image. The overall recall rate is 94.87%, and categories that get lower recall rates are highlighted. VGG_M model is used as network initialization.

those usually immersed in object clutters, are also difficult to be distinguished from background by RPN, like plant, tv and chair.

3.1.2 Cascade structure

In order to make a bridge between the object-like regions provided by RPN and the object proposals demanded by the detection task, we introduce an additional classification network that comes after the RPN. According to definition, what we need here is to classify the object-like regions between real object instances and background/badly located proposals. Therefore we take the additional network as a 2-class detection network (denoted as FRCN net in Figure 2) which uses the output of RPN as training data. In such a cascade structure, the RPN net takes universal image patches as input and is responsible to capture general patterns like texture, while the FRCN net takes input as object-like regions, and plays the role of learning patterns of finer details.

The advantages of the cascade structure are two-fold: First, the additional FRCN net further improves the quality of the object proposals and shrinks more background regions, making the proposals fit better with the task requirement. Second, proposals from multiple sources can be merged as the input of FRCN net so that complementary information can be used.

3.1.3 Implementation

We train the RPN and FRCN nets consecutively. The RPN net is trained regularly in a sliding window manner to classify all regions at various scales and aspect ratios in the image, with the same parameters as in [27]. After the RPN net is trained, we test it on the whole training set to produce 2000 primitive proposals of each training image. These proposals are used as training data to train the binary classifier FRCN net. Note that when training the second FRCN net, we use the same criterion of positive and negative sampling as in RPN (above 0.7 IoU for positives and below 0.3 IoU for negatives).

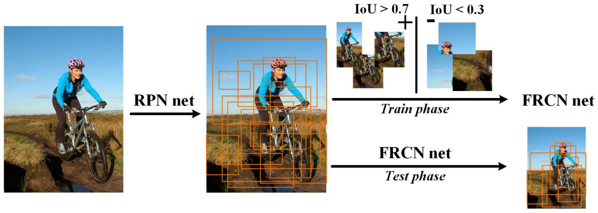

Figure 2. The pipeline of the cascade proposal generator. We first train a standard Region Proposal Network (RPN net) and then use its output to train another two-class Fast-RCNN network (FRCN net). During testing phase, the RPN net and the FRCN net are concatenated together. The two nets do not share weights and are trained separately from the same pre-trained model.

At testing phase, we first run the RPN net on the image to produce 2000 primitive proposals and then run FRCN net on the same image along with 2000 RPN proposals as the input to get the final proposals. After proper suppression or thresholding, we can get fewer than 300 proposals of higher quality.

We use the FRCN net rather than RPN net as the second binary classifier for that FRCN net has more parameters in its higher-level connections, making it more capable to handle with the more difficult classification problem. If we use the model definition of RPN net as the second classifier, the performance degrades. In our current implementation, we do not share full-image convolutional features between RPN net and FRCN net. If we share them, we expect little performance gain as in [27].

3.2. Cascade object classification

3.2.1 Baseline Fast R-CNN

A good object classifier is supposed to classify each object proposal correctly into certain number of categories. Due to the imperfection of the proposal generator, there exists quite a large number of background regions and badly located proposals in the proposals. Therefore when training the object classifier, an additional object category is often added as "background". In the successful solution Fast R-CNN, the classifier is learned with a multi-class cross-entropy loss through softmax layer. Aided by the auxiliary loss of bounding box regression, the detection performance is superior to "softmax + SVM" paradigm in R-CNN approach. In order to get an end-to-end system, Fast R-CNN drops the one-vs-rest SVM in R-CNN, which creates the gap between the resulting solution and the task demand.

Given object proposals as input and final object detections as output, the task demands for not only further distinguishing objects of interested categories from non-objects, but also classifying objects into different classes, especially

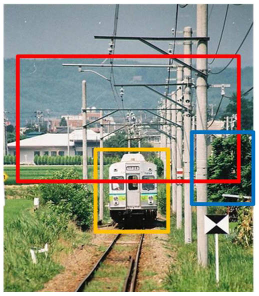

Figure 3. Example detections from a Fast-RCNN model. Different colors indicate different object categories. Specifically, orange color denotes “train”, red denotes “boat” and blue denotes “potted plant”.

those with similar appearance and/or belong to semantically related genres (car and bus, plant and tree). This calls for a feature representation that captures both the inter-category and intra-category variances. In the case of Fast R-CNN, the multi-class cross-entropy loss is responsible for helping the learned feature hierarchies capture inter-category variance, while it is weak at capturing intra-category variance as the “background” class usually occupies a large proportion of training samples. Example detection results of Fast R-CNN are shown in Figure 3, where the mis-classification error is a major problem in the final detections.

3.2.2 Cascade structure

To ameliorate the problem of too many false positives caused by mis-classification, we bring the one-vs-rest classifier back in the form of an additional two-class cross-entropy loss for each object category (shown in Figure 4). In essence, the added classifier is playing the role of SVM in R-CNN framework. We find it important to train each one-vs-rest classifier using the detection output of that specific category (meaning the detection should have highest score on that specific category). In this way, each one-vs-rest classifier sees proposals specific to one particular object category (also containing some false positives), making it focused at capturing intra-category variance.

For example, in PASCAL VOC dataset, the training samples for the additional classifier of class “potted plants” are usually trees, grass, potted plants and some other green things. After the training, it is able to capture the minute difference between various types of plants, so as to reduce false positives related to this class. This effect can hardly be achieved through a multi-class cross-entropy loss.

3.2.3 Implementation

During the training phase, a standard FRCN net (FRCN-1) is first trained using object proposals from the cascade proposal generator. Thereafter, we train another FRCN net (FRCN-2) based on the output of FRCN-1 (which we call primitive detections). Since we are now dealing with classification task among objects, we discard the primitive detections which are classified as “background”. The objective function of the FRCN-2 is the sum of N 2-class cross-entropy losses (N equals the number of object categories), with each 2-class classifier depends only on primitive detections assigned with the corresponding class label. The criterion of positive and negative sampling for the one-vs-rest classifier is the same as RPN. Practically, there are roughly 20 primitive detections per image used for FRCN-2 training, which is quite limited.

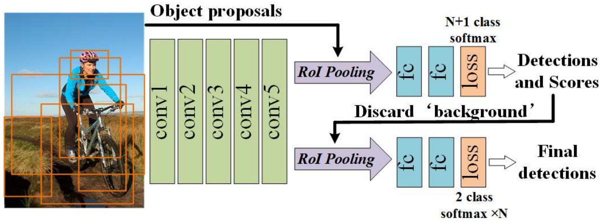

Figure 4. The work flow of the cascade proposal classifier. We first train a standard Fast-RCNN network (FRCN-1) and use its output scores to assign each detection with a class label. Then detections belonging to “background” are discarded and the rest are used to train another Fast-RCNN network (FRCN-2) whose loss is the sum of N two-class softmax losses. Note that the auxiliary bounding box regression loss is also used in both FRCN nets but left out in the figure for better presentation. The two FRCN nets are optimized consecutively with shared convolution weights so that the image feature maps are computed only once during testing phase.

To effectively train FRCN-2 and efficiently detect objects from proposals, we share the convolution weights of FRCN-1 and FRCN-2 so that the full-image feature maps need only be computed once. That is to say, the convolution weights of FRCN-2 are initialized from FRCN-1 and keep fixed during FRCN-2 training. The fully-connected layers of FRCN-2 are initialized from FRCN-1 as well, and new layers to produce 2N scores and 4N bounding box regression targets are initialized from a gaussian distribution.

At test time, with 300 object proposals as input, FRCN-1 outputs around 20 primitive detections, each with N primitive scores. Then each primitive detection is again classified by FRCN-2 and the output scores (N categories) is multiplied with the primitive scores (N categories) in a category-by-category way to get the final N scores for this detection.

4. Experiments

We first validate that the proposed cascade structure does improves the performance of each task in the two-step object detection framework through a delicate design, then we show the overall performance gain in object detection by evaluating CRAFT on benchmarks like PASCAL VOC 07/12 and ILSVRC. Note that we do not share full-image convolutional features between the proposal generation and classification tasks, therefore the proper baseline would be the unshared version of Faster R-CNN [27].

4.1. Proposal generation

Firstly we justify the design choice of the cascade proposal generator. We answer two questions: 1) do we really need a more complex network in the second stage? 2) do we need the strict sampling criterion during training? We show evaluation of different parameterization in Table 2. Since RPN already performs quite well on PASCAL VOC benchmarks, we show parameterization evaluation on the more challenging ILSVRC dataset, and then present a thorough evaluation of the final design of the cascade proposal generator on PASCAL VOC.

We show comparison of different choices of the sampling criterion and network definition of the cascade binary classifier in Table 2. All models in the table are initialized from a pre-trained VGG19 model, trained on the ILSVRC DET train+val1 sets and tested on val2 set[1], with a evaluation metric of 0.5IoU threshold. To handle with small objects in ILSVRC, we add two additional anchor scales (64 and 32) in RPN and change the batch-size to 2 images. However, the RPN's performance (89.94%) is still inferior to Selective Search (92.09%). When cascaded with an additional binary classifier ("+FRCN"), the recall rate increases by over 2%.

We show that the strict sampling criterion (0.7IoU threshold for positives, 0.3IoU threshold for negatives) leads to slightly better performance. When we replace the FRCN with a RPN-like network definition which uses a 512-d feature representation rather than 4096-d, the recall rate degrades. With the best design choice, we can achieve higher recall rate (92.37%) than Selective Search with only 300 proposals.

Next we throughly evaluate our cascade proposal generator on PASCAL VOC in Table 3. Baselines are Selective Search ("SS"), RPN (VGG16 net with 512-d feature representation), and RPN_L (VGG16 net with 4096-4096-d feature representation, meaning larger RPN). We evaluate with regard to not only recall rates at different IoU thresholds (from 0.5 to 0.9), but also the mAP of a Fast R-CNN detector trained on different proposal algorithms, which makes the comparison more meaningful because the proposals are eventually used for object detection. Note than all the methods in Table 3 are purely for proposal task without joint optimization with object detection.

Model	IoUthr_pos	IoUthr_neg	Recall (%)
SS	-	-	92.09
RPN	0.7	0.3	89.94
+ FRCN	0.5	0.5	92.13
+ FRCN	0.5	0.3	92.24
+ FRCN	0.7	0.3	**92.37**
+ RPN-2	0.7	0.3	91.83

Table 2. Evaluation results of different design choices of the cascaded binary classifier in cascade proposal generator on ILSVRC DET val2 set. All recall rates except that of Selective Search ("SS") are reported with 300 proposals per image. For Selective Search baseline, there are roughly 2000 proposals per image.

method	#box	0.5	0.6	0.7	0.8	0.9	mAP
SS	2000	92.1	85.2	72.5	52.9	**26.6**	70.0
RPN	2000	**98.5**	**95.8**	84.1	40.7	4.1	-
RPN	300	96.3	92.5	78.8	37.9	3.9	71.6
RPN_L	300	95.4	90.3	76.5	37.4	3.8	-
Ours	300	97.9	95.5	**89.6**	**63.7**	13.0	72.2
Ours_S	**87**	96.8	94.1	87.8	62.4	12.9	**72.5**

Table 3. Proposal evaluation by recall rate (%) with regard to different IoUs and detection mAP (%) on PASCAL VOC 07 test set. All CNN based methods use VGG16 net as initialization and are trained on PASCAL VOC 07+12 trainval set. "Ours" keeps fixed number of proposals per image (same as RPN), while "Ours_S" keeps proposals whose scores (output of the cascaded FRCN classifier) are above a fixed threshold.

From the table we can see that: (1) RPN proposals aren't so well localized compared with bottom-up methods (low recall rates at high IoU thresholds). (2) This cannot be ameliorated by using a larger network because it is caused by fixed anchors. (3) Our cascaded proposal generator not only further eliminates background proposals, but also brings better localization, both help in detection AP.

4.2. Object classification

In this part we justify the use of the one-vs-rest classifier as well as explain how many layers to fine-tune for the one-vs-rest classifier. We show the evaluation results in Table 4.

The cascade object classifier can be regarded as a concatenation of two FRCN nets (FRCN-1 + FRCN-2). In the top table in Table 4 we compare several strategies concerning training the FRCN-2 net, which are no fine-tuning ("the same"), fine-tuning the additional one-vs-rest classifier weights ("cls"), fine-tuning layers above the last convolution maps ("fc + clf"), and fine-tuning all layers except for conv1 ("conv + fc + clf"). In fact, "the same" is simply running the FRCN-1 net twice, and hopefully the iterative bounding box regression would help improve the result. Another three fine-tuning strategies are trying to improve

[1] The splits of "val1" and "val2" are the same as [15]

FRCN-1	FRCN-2	FT layers	mAP (%)
VGG_M	-	-	65.0
VGG_M	the same	-	65.2
VGG_M	VGG_M	clf	66.3
VGG_M	VGG_M	fc + clf	**68.0**
VGG_M	VGG_M	conv + fc + clf	67.7

classifier objective	mAP (%)
FRCN (one-shot)	65.0
one-vs-rest	46.1
Ours (one-shot + one-vs-rest)	**68.0**

Table 4. **Top:** Evaluation of how many layers to fine-tune for the one-vs-rest classifier. **Bottom:** Evaluation of different classifier objectives. All models use VGG_M as network initialization. All results are evaluated on PASCAL VOC 07 (trainval for training, test for testing) with the same object proposals from a trained RPN model.

the performance by introducing additional class-specific one-vs-rest classifiers to capture intra-category variance. The difference between these three is different level of feature sharing with FRCN-1 net: "clf" uses exactly the same feature representation as FRCN-1, and "conv + fc + clf" trains totally new feature representation for itself.

From the results in the table we can see that iteratively detecting twice improves the results a little, which mainly comes from iterative bounding box regression. As for fine-tuning the net with a binary softmax loss, different settings vary in performance. In a word, through sharing convolutional features but fine-tuning high-level connections we get best result. There are two possible reasons that account for it: 1) the training samples for the FRCN-2 are limited and biased; 2) we just want to learn another classifier rather than learn total different feature representation. What's more, the improvement gained by the cascade approach is more than that by iteratively detecting twice, proving that the one-vs-rest softmax loss does play part of the role of hard negative mining and helps reduce the mis-classification error.

We additionally justify the cascade structure in the bottom table in Table 4. One-vs-rest classifier alone performs poorly because each binary classifier has to handle with objects of various classes but binary label provides limited information, while in our case each binary classifier only handles with detections of one class (ie, detection output of the FRCN-1 net), making it more specialized.

4.3. Object detection

After showing the superiority of CRAFT on both tasks in the two-step object detection framework, we now evaluate CRAFT on object detection benchmarks. We first evaluate CRAFT on PASCAL VOC 07&12 in comparison with the state-of-the-art detectors Fast R-CNN and Faster R-CNN, and then show our results on the more challenging ILSVRC benchmark.

4.3.1 PASCAL VOC 2007 & 2012

We compare CRAFT with state-of-the-art detectors under the two-step detection framework, which are Fast R-CNN [14] and Faster R-CNN [27]. The comparative results on PASCAL VOC 2007 & 2012 are shown in Table 5. Qualitative results on PASCAL VOC 2007 test set are shown in Figure 5. All methods use VGG16 model, and "RPN_un" represents the unshared version of Faster R-CNN. All baseline results are got from original papers or by running the original open source codes. On PASCAL VOC 2007, all methods use 07+12 trainval as training data. CRAFT outperforms the baseline "RPN_un" by 4.1% absolute value in mAP (from 71.6% to 75.7%). On PASCAL VOC 2012, all methods use 12 trainval as training data, and this time CRAFT achieves an edge of 5.8% absolute value in mAP (from 65.5% to 71.3%).

We do not compare with many other detectors which also improve over the basic two-step detection framework like [13, 2] because we believe that their contributions are orthogonal to ours. If we incorporate their methods in CRAFT, as well as using end-to-end multi-task network cascade training [6], we expect notable further improvement.

method	proposal	classifier	voc07	voc12
FRCN [14]	SS	FRCN	70.0	65.7
RPN_un [27]	RPN	FRCN	71.6	65.5†
RPN [27]	RPN	FRCN	73.2	67.0
CRAFT	cascade	FRCN	72.5	-
CRAFT	cascade	cascade	**75.7**	**71.3**‡

Table 5. Object detection mAP (%) on PASCAL VOC 07+12. "voc07": 07+12 trainval for training, VGG16 net. "voc12": 12 trainval for training, VGG16 net. "FRCN" and "RPN" results are from original paper and report. "RPN_un" are Faster R-CNN with unshared feature, whose results are got from open source codes (the proper baseline). Joint optimization is not used in "CRAFT", which would otherwise bring some gain. †: http://host.robots.ox.ac.uk:8080/anonymous/AITNWY.html, ‡: http://host.robots.ox.ac.uk:8080/anonymous/FFJGZH.html

4.3.2 ILSVRC object detection task

We validate that CRAFT generalizes well to large-scale problems like 200-class ILSVRC object detection task.

As shown in Table 2, RPN does not generalize very well to large-scale object detection tasks even if more scales are added to the anchors. However, with the help of our cascade structure, the recall rates boosts to be over Selective Search. However, the performance is still inferior to that on

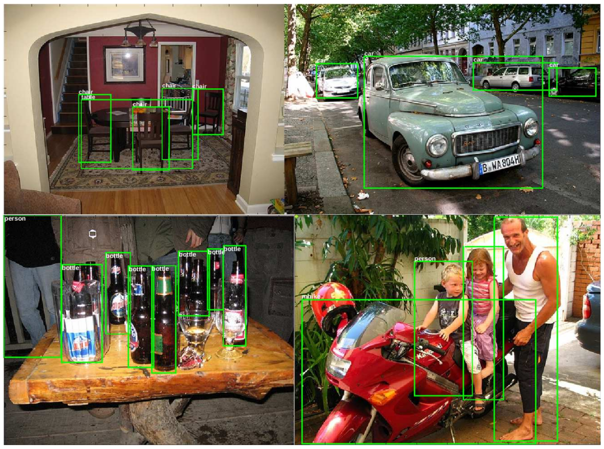

Figure 5. Example detections of CRAFT on PASCAL VOC 2007 test set.

PASCAL VOC. Therefore, we add additional some additional modules to the cascade proposal generator to further improve its performance.

As shown in Table 6 top, using a stricter NMS policy (0.6 IoU threshold) increases the recall rate a bit because the localization accuracy of proposals has already been improved after the cascade structure. Re-scoring each proposal by considering both scores from two stages of cascade structure also helps. Finally, fusion of multiple proposal sources boosts the recall rate to be over 94%. We combine proposals output from "DeepBox" [21] or "SS" (Selective Search) with the RPN proposals as the fusion input to the FRCN net in the cascade structure. Results show that "DeepBox" is better than "SS".

Basic	0.6 NMS	re-score	+DeepBox	+SS
92.37	93.61	93.75	**94.13**	93.04

method	proposal	classifier	ilsvrc
Ouyang *et al.* [24]	SS+EB	RCNN	45.0
Yan *et al.* [36]	SS+EB	RCNN	45.4
RPN_un [27]	RPN	FRCN	45.4
CRAFT	cascade	FRCN	47.0
CRAFT	cascade	cascade	**48.5**

Table 6. **Top:** Recall rate (%) of cascade proposal generator on ILSVRC detection val2 set with regard to 0.5 IoU evaluation metric. **Bottom:** Detection mAP (%) of CRAFT on ILSVRC detection val2 set in comparison with other state-of-the-art detectors.

Given high-quality object proposals, we train a regular Fast R-CNN detector and a cascade object classifier upon it. We use a GoogLeNet model with batch normalization [17] (8.4% top-5 validation error on ILSVRC image classification task) as network initialization. We use ILSVRC 2013train + 2014train + val1 as training set, and evaluate it on val2 set. Since 2013train set does not align well with detection task, we adopt the following batch sampling strategy: each batch is made up of 12 images, with 8 from fully annotated sets (2014train + val1) and 4 from partially annotated sets (2013train). For each fully annotated image, we sample 32 proposals with 8 positives and 24 negatives, and the IoU threshold for distinguishing positives and negatives is 0.5. For each partially annotated image, we sample 8 proposals with 2 positives and 6 negatives, and the IoU range for positives is larger than 0.7 and for negatives it is smaller than 0.5. Due to large batch-size, we train the detector on a 4-GPU implementation.

In Table 6 bottom, a regular Fast R-CNN detector achieves 47.0% mAP on val2, which already surpasses the ensemble result of previous state-of-the-art systems like Superpixel Labeling [36] and Deepid-Net [24]. This edge is basically from better proposals. With cascade object classifier added, the mAP gets additional 1.5% absolute gain.

5. Conclusion

In this paper, we propose the CRAFT (Cascade Region-proposal-network And FasT-rcnn) for general object detection following the "divide and conquer" philosophy. It improves both the proposal generation and classification tasks through carefully designed convolutional neural network cascades. For the proposal task, CRAFT outputs more compact and better localized object proposals. For detection task, CRAFT helps the network learn both inter- and intra-category variances so that false positives among ambiguous categories are largely eliminated. CRAFT achieves consistent and considerable improvements over state-of-the-art methods on PASCAL VOC and ILSVRC benchmarks, while being complementary to many other advances in object detection.

Acknowledgements The authors were supported by Chinese National Natural Science Foundation Projects #61375037, #61473291, #61572501, #61502491, #61572536, by National Science and Technology Support Program Project #2013BAK02B01, by Chinese Academy of Sciences Project No. KGZD-EW-102-2, and by AuthenMetric R&D Funds. We thank NVIDIA gratefully for GPU hardware donation and the reviewers for their many constructive comments.

References

[1] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. CVPR, 2014. 2

[2] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. *arXiv preprint arXiv:1512.04143*, 2015. 3, 7

[3] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011. 3

[4] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *IEEE CVPR*, 2014. 2

[5] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 1995. 3

[6] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. *arXiv preprint arXiv:1512.04412*, 2015. 7

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*. IEEE, 2005. 2

[8] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *CVPR*. IEEE, 2014. 2

[9] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010. 3

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004. 2

[11] J. Friedman, T. Hastie, R. Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 2000. 3

[12] A. Ghodrati, M. Pedersoli, T. Tuytelaars, A. Diba, and L. Van Gool. Deepboxes: Hunting objects by cascading deep convolutional layers. In *Proceedings ICCV 2015*, 2015. 3

[13] S. Gidaris and N. Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1134–1142, 2015. 3, 7

[14] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 1, 3, 7

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2, 3, 6

[16] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In *BMVC*, 2014. 2

[17] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 448–456, 2015. 8

[18] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural svms. *Machine Learning*, 2009. 3

[19] H. Kaiming, Z. Xiangyu, R. Shaoqing, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, 2014. 3

[20] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 3

[21] W. Kuo, B. Hariharan, and J. Malik. Deepbox: Learning objectness with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2479–2487, 2015. 3, 8

[22] K. Lenc and A. Vedaldi. R-cnn minus r. *arXiv preprint arXiv:1506.06981*, 2015. 3

[23] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *CVPR*. IEEE, 1997. 2

[24] W. Ouyang, P. Luo, X. Zeng, S. Qiu, Y. Tian, H. Li, S. Yang, Z. Wang, Y. Xiong, C. Qian, et al. Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2015. 3, 8

[25] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*. Springer, 2010. 3

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*, 2015. 3

[27] 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. 1, 2, 3, 4, 6, 7, 8

[28] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *PAMI*, 1998. 2

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1, 3

[30] R. Stewart and M. Andriluka. End-to-end people detection in crowded scenes. *arXiv preprint arXiv:1506.04878*, 2015. 3

[31] 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. 1

[32] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *NIPS*, 2013. 2

[33] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. In *JMLR*, 2005. 3

[34] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 2013. 1, 2

[35] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 2004. 2

[36] J. Yan, Y. Yu, X. Zhu, Z. Lei, and S. Z. Li. Object detection by labeling superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5107–5116, 2015. 3, 8

[37] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*. ACM, 2009. 3

[38] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*. Springer, 2014. 1, 2