Scalable High Quality Object Detection

Christian Szegedy
Google Inc.
1600 Amphitheatre Pkwy, Mountain View, CA
szegedy@google.com

Scott Reed
University of Michigan
reedscot@umich.edu

Dumitru Erhan
dumitru@google.com

Dragomir Anguelov
dragomir@google.com

Sergey Ioffe
sioffe@google.com

Abstract

Current high-quality object detection approaches use the same scheme: salience-based object proposal methods followed by post-classification using deep convolutional features. This spurred recent research in improving object proposal methods [18, 32, 15, 11, 2]. However, domain agnostic proposal generation has the principal drawback that the proposals come unranked or with very weak ranking, making it hard to trade-off quality for running time. Also, it raises the more fundamental question of whether high-quality proposal generation requires careful engineering or can be derived just from data alone. We demonstrate that learning-based proposal methods can effectively match the performance of hand-engineered methods while allowing for very efficient runtime-quality trade-offs. Using our new multi-scale convolutional MultiBox (MSC-MultiBox) approach, we substantially advance the state-of-the-art on the ILSVRC 2014 detection challenge data set, with 0.5 mAP for a single model and 0.52 mAP for an ensemble of two models. MSC-Multibox significantly improves the proposal quality over its predecessor Multibox [4] method: AP increases from 0.42 to 0.53 for the ILSVRC detection challenge. Finally, we demonstrate improved bounding-box recall compared to Multiscale Combinatorial Grouping [18] with less proposals on the Microsoft-COCO [14] data set.

1. Introduction

After the dramatic improvements in object detection demonstrated by Girshick et al. [9], most of the current state of the art approaches, including the top performing entries [26, 16, 25] of the 2014 Imagenet object detection competition [22], make use of salience-based object localization, in particular Selective Search [29] followed by some post-classification method using features from a deep convolutional network.

Given the fact that the best salience-based methods can reach up to 95% coverage of all objects at 0.5 overlap threshold on the detection challenge validation set, it is tempting to focus on improving the post-classification ranking alone while considering the proposal generation part to be solved. However, this might be a premature conclusion: a better way of ranking the proposals is to cut down their number at generation time already. In the ideal case, we will be able to achieve high coverage with very few proposals. This can improve not only the running time but also the quality, because the post-classification stage would need to handle fewer potential false positives. Furthermore, a strong proposal ranking function provides a way to balance recall versus running-time in a simple, consistent manner by just selecting appropriate thresholds: use a high threshold for use cases where speed is essential, and a low threshold when quality matters most.

Motivated by the fact that hand-engineered features are getting replaced by higher-quality deep neural network features for image classification [13, 12, 26], we show that the same trend holds for proposal generation. In Section 4.6 we demonstrate that our purely learned proposal method closely rivals salience-based methods in performance, at a significantly lower computational cost. Furthermore, the ability to directly learn region proposal methods is a key advantage as it is easy to adapt the model to new domains such as medical or aerial imaging or to specific use cases, such as recognizing only certain objects. In contrast, hand-engineered proposal methods are typically tuned for natural objects with clear segmentation, but do less well in domains where the distinction between objects needs more subtle cues and cannot return proposals only for objects of interest.

Our work builds upon the MultiBox approach presented in [4], which was an earlier attempt to learn a proposal generation model but was never directly competitive with the best expert-engineered alternatives. We demonstrate that switching to the latest Inception [28]-style architec-

ture and utilizing multi-scale convolutional predictors of bounding box shape and confidence, in combination with an Inception-based post-classification model significantly improves the proposal quality and the final object detection quality. Combining this with a simple but efficient contextual model, we end up with a single system that scales to a variety of use cases from real-time to very high-quality detection and achieves a new state of the art result on the ImageNet detection challenge.

In summary, the main contributions of our approach are:

- Improved network architecture for bounding box generation, including multi-scale convolutional bounding box predictors.

- Integration of a context model during post-classification, which improves performance.

- 200 classes detection at 0.45 mAP with 15 proposals per image generated by our box proposal method.

- 0.50 mAP with a single model and 0.52 with an ensemble of three post-classifiers and two MultiBox proposal generators.

Additionally, in Sec. 4 we analyze the effect of the various components of the MSC-Multibox model.

2. Related Work

The previous state-of-the-art paradigm in detection is to use part-based models [6, 5] such as Deformable Part Models (DPMs). Sadeghi and Forsyth [23] developed a framework with several configurable runtime-quality trade-offs and demonstrate real-time detection using DPMs on the PASCAL 2007 detection data.

Deep neural network architectures with repeated convolution and pooling layers [7, 13] have more recently become the dominant approach for large-scale and high-quality recognition and detection. Szegedy et al. [27] used deep neural networks for object detection formulated as a regression onto bounding box masks. Sermanet et al. [24] developed a multi-scale sliding window approach using deep neural networks, winning the ILSVRC2013 localization competition.

The original work on MultiBox [4] also used deep networks, but focused on increasing efficiency and scalability. Instead of producing bounding box masks, the MultiBox approach directly produces bounding box coordinates, and avoids linear scaling in the number of classes by making class-agnostic region proposals. In our current work (detailing improvements to MultiBox) we demonstrate greatly increased recall of object locations by increasing the number of *potential* proposals with a fixed budget of evaluated

proposals. We also demonstrate improvements to the training strategy and underlying network architecture that yield state-of-the-art performance.

Other recent works have also attempted to improve the scalability of the now-predominant R-CNN detection framework [9]. He et al. proposed Spatial Pyramid Pooling [10] (SPP), which engineers robustness to aspect-ratio variation into the network. They also improve the speed of evaluating Selective Search proposals by classifying mid-level CNN features (generated from a single feed-forward pass) rather than pushing all image crops through a full CNN. They report roughly two orders of magnitude ($\sim$ 100x) speedup over R-CNN using their method.

Compared to the SPP approach, we show a comparable efficiency improvement by drastically reducing the number and improving the quality of region proposals via our Multi-Box network, which also associates a confidence score to each proposal. Architectural changes to the underlying network and contextual post-classification were the main factors in reaching high quality. We emphasize that MultiBox and SPP are complementary in the sense that spatial pyramid pooling can be added to the underlying ConvNet if desired, and post-classification of proposals can be sped up in the same way with no change to the MultiBox objective.

Another way in which efficiency of detection methods can be improved is by unifying the detection and classification models, reusing as much computation as possible and in the process abandoning the idea of data-independent region proposals. An example of such an approach is Pinheiro et al. [17]'s work, who propose a convolutional neural network model with two branches: one that can generate class-agnostic *segmentation* masks, and second branch predicting the likelihood of a given patch being centered on an object. Inference is efficient since the model is applied convolutionally on an image and one can get the class scores and segmentation masks using a single model.

The YOLO approach by Redmon et al. [19] is similar to it, in that it uses a single network to predict bounding boxes and class probabilities, in an end to end network. The difference is that it divides the input image into a grid of cells and predicts the coordinates and confidences of objects contained in the cells. This approach is fast, but limited in that each grid cell can only contain one object by construction, with the grid being quite coarse. It is also unclear to which extent these results can translate to good performance on data sets with significantly more objects, such as the ILSVRC detection challenge.

Faster R-CNN [21] is a technique that merges the convolutional features of the full-image network with the detection network, thereby simultaneously predicting boxes and objectness scores. The detection network–called the Region Proposal Network (RPN)–is trained end to end in an alternating fashion with the Fast R-CNN network; its objective

is to produce good region proposals. The RPN is thus quite similar to the Multibox approach described in this paper: the two approaches have been co-developed at the same time. The biggest similarity is the usage of priors (called "anchors in the Fast R-CNN work [8]) that are designed to be translation invariant and that are predicted from the top layer feature map. Our multiscale priors are different in that we use multiple tapering layers, while the Fast R-CNN approach is predicting boxes of many scales from a single feature map. The other differences include the fact that in our approach the confidences are class-agnostic, and we used different box regression and classification losses. Notably, we also use radically different network architectures, with parts designed specifically to overcome the shortcomings of networks designed for classification. Finally, we argue that our two-stage setup scales to a higher number of classes well: the Faster R-CNN work uses many thousands of priors and scaling that approach to thousands of classes is not obvious. It would ultimately be interesting to disentangle which of these differences are important, by comparing the two methods on the same evaluation set.

3. Model

3.1. Background: MultiBox objective

In order to describe the changes to [4], let us revisit the basic tenets of the MultiBox method. The fundamental idea is to train a convolutional network that outputs the coordinates of the object bounding boxes directly. However, this is just half of the story, since we would also like to rank the proposals by their likelihood of being an accurate bounding box for an object of interest. In order to achieve this, the MultiBox loss is the weighted sum of the following two losses:

- **Confidence**: a logistic loss on the estimates of a proposal corresponding to an object of interest.

- **Location**: a loss corresponding to some similarity measure between the objects and the closest matching object box predictions. By default we used L2 distance.

The network is an improved Inception-style [28] convolutional network, followed by a structured output module producing a set of bounding box coordinates and confidence scores. In the original MultiBox solution, the predictors were fully connected to the top layer of the network. Here we propose a multi-scale convolutional architecture described below.

Let $l_i \in \mathbb{R}^4$ be the i-th set of predicted box coordinates for an image, and let $g_j \in \mathbb{R}^4$ be the j-th ground-truth box coordinates. At training time, for each image, we perform a bipartite matching between predictions and ground-truth

boxes. We denote $x_{ij} = 1$ to indicate that the i-th prediction is matched to the j-th ground-truth, and $x_{ij} = 0$ otherwise. Note that x is constrained so that $\sum_i x_{ij} = 1$. Given a matching between predictions and groundtruth, the location loss term can be written as

$$F_{loc}(x, l, g) = \frac{1}{2} \sum_{i,j} x_{ij} ||l_i - g_j||_2^2. \qquad (1)$$

Given the predicted scores c_i, the confidence loss term can be written as follows:

$$F_{conf}(x, c) = -\sum_{i,j} x_{ij} \log(c_i) - \qquad (2)$$
$$\sum_i (1 - \sum_j x_{ij}) \log(1 - c_i)$$

The overall objective is a weighted sum of both terms

$$F(x, c, l, g) = F_{conf}(x, c) + \alpha F_{loc}(x, l, g) \qquad (3)$$

We train the network with stochastic gradient descent. For each training example with ground truth g and network output (c, l) we compute the matching x^* by picking the minimizer of the loss:

$$x^* = \arg\min_x F(x, c, l, g) \qquad (4)$$
$$\text{such that} \qquad x_{ij} \in \{0, 1\}, \sum_i x_{ij} = 1$$

and update the network parameters following the gradient evaluted at the matching x^* that was found.

3.2. Convolutional Priors

The MultiBox [4] setup is to predict locations (the five coordinates) and confidences for a constant number of boxes. We call the associated outputs of the network "slots": each slot corresponds to one predicted proposal. However, these proposals might be low confidence, in which case the network predicts that the associated box does not correspond to any object on the image. Our goal is to maximize the coverage of the high-confidence predictions. Our network is an "objectness" detector, but our notion of what constitutes an object depends on the task we try to tackle.

A crucial detail of our approach is that we do not let the proposals free-float, but impose diversity by introducing a prior for each box output slot of the network. Let us assume the our network predicts k boxes, together with their confidences, then each of those output slots will be associated with a prior rectangle p_i. These rectangles are computed before training the network in a way that matches the distribution of object boxes in the training set. Our goal is to maximize the expected coverage of this constant set of
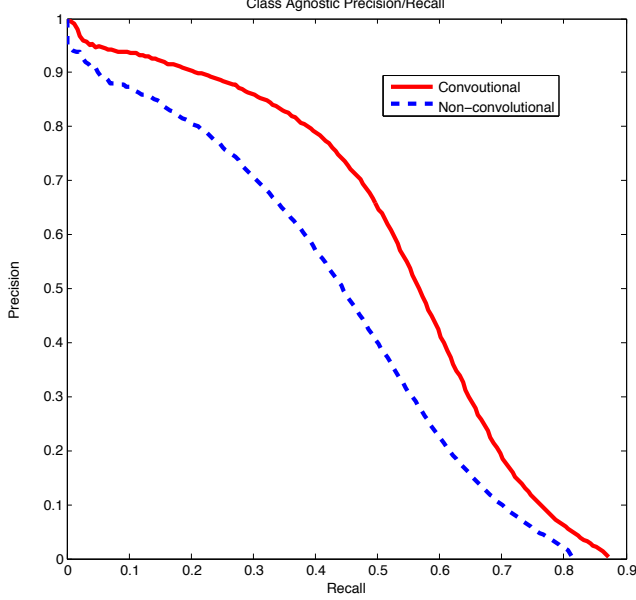
Figure 1. Class-agnostic Precision-Recall at IOU threshold 0.5 of the MultiBox trained with convolutional vs. non-convolutional priors. The (class-agnostic) average precision goes up from 0.417 to 0.529.

priors at a given Jaccard (IOU) overlap threshold $t = 0.5$. In [4], the goal was to maximize the expected overlap between each ground-truth object box and the best matching prior. Here we try to find a set of priors to optimize $E([min_{\{p_i\}}(IOU(p_i, b_i)) > t])$, where b_i are matching groundtruth bounding boxes. Intuitively, we can say that we want the best proposal generation method that is independent of the image pixels and has the maximum coverage at a given overlap threshold t (0.5 in our case).

As in Multibox [4], the bounding boxes l_i predicted by slot i of the network will be interpreted with respect to prior p_i. That is, we are regressing toward $g_j - p_i$ where g is the groundtruth box minimizing $\|g_j - p_i\|$ and at inference time if the network outputs l'_i for slot i, the predicted box l_i will be set to $l'_i + p_i$. Erhan *et al.* [4] took a similar approach, but they tried to maximize the expected overlap as opposed to the coverage. However, it is a highly non-convex objective function, so they needed to resort to the heuristic of performing k-means clustering of the ground-truth object boxes of the training set objects and took the k-means centroids as priors. Here, we are taking a different approach that is closely related to the approach of Faster R-CNN [21] and exploits the expected translation invariance of the object locations in the data set. The priors are assumed to lie on grids with grid lines parallel to the image boundary. Formally, we assume that our set P of prior boxes is the union

of boxes placed regularly on those grids.

$$P = \bigcup_q (G_q + t_q), \quad (5)$$

where $G_q = \delta_q\{1, \ldots m_q\} \times \delta_q\{1 \ldots m_q\}$ is a regular two dimensional grid and t_q is the template box displaced by the grid and m_q denotes the grid resolution. In our setup we have set $\delta_q = \frac{1}{m_q+1}$. In addition to the 8×8 top layer of our base network, we add a prediction tree to our network as depicted in Fig. 2. We have a dedicated layer for producing prediction locations and scores for each of the 8×8, 6×6, a 4×4, a 3×3 a 2×2 and a 1×1 grids (the 1×1 grid is created by applying average pooling on the 8×8 top base network layer). Each tile of each grid but the 1×1 is responsible for predicting 11 outputs with priors of different aspect ratios. The top 1×1 grid is used for predicting the single largest prior. This way we end up using

$$1 + 11 \times (8 \times 8 + 6 \times 6 + 4 \times 4 + 3 \times 3 + 2 \times 2)) = 1420$$

priors. Each of these priors is associated with one location output slot and its associated confidence output slot of the network. The outputs are emitted by the LOC and $CONF$ layers as shown in Fig. 2.

3.3. Training with missing positive labels

In large-scale data sets such as ImageNet, there are many missing true-positive labels. In the confidence term of the MultiBox training objective, a large loss will be incurred if the model assigns a high confidence to a true positive object in the image that is missing a label. We hypothesize that the dissonance caused by missing or noisy training data may encourage the model to be overly conservative in its predictions and thereby reduce the recall of MultiBox proposals. To deal with the issue, we adopted the "hard bootstrapping" approach of Reed et al. [20].

Training with this method is equivalent to reformulating the confidence objective as follows:

$$F_{bootstrap}(x, c) = -\sum_i 1_{\{i \notin topL(c)\}} \quad (6)$$
$$(\sum_j x_{ij} \log c_i + (1 - \sum_j x_{ij}) \log(1 - c_i)),$$

where $topL(c)$ is the set of indices into the top-L most confident predictions. In practice, we precompute $topL(c)$ for every image within a batch before computing the gradients. The learning iterates between "generating data" according to the previous model state, and then updating the model based on the augmented data. In our experiments we initialized the network with networks pre-trained with no bootstrapping, and then fine-tuned on $F_{bootstrap}$.
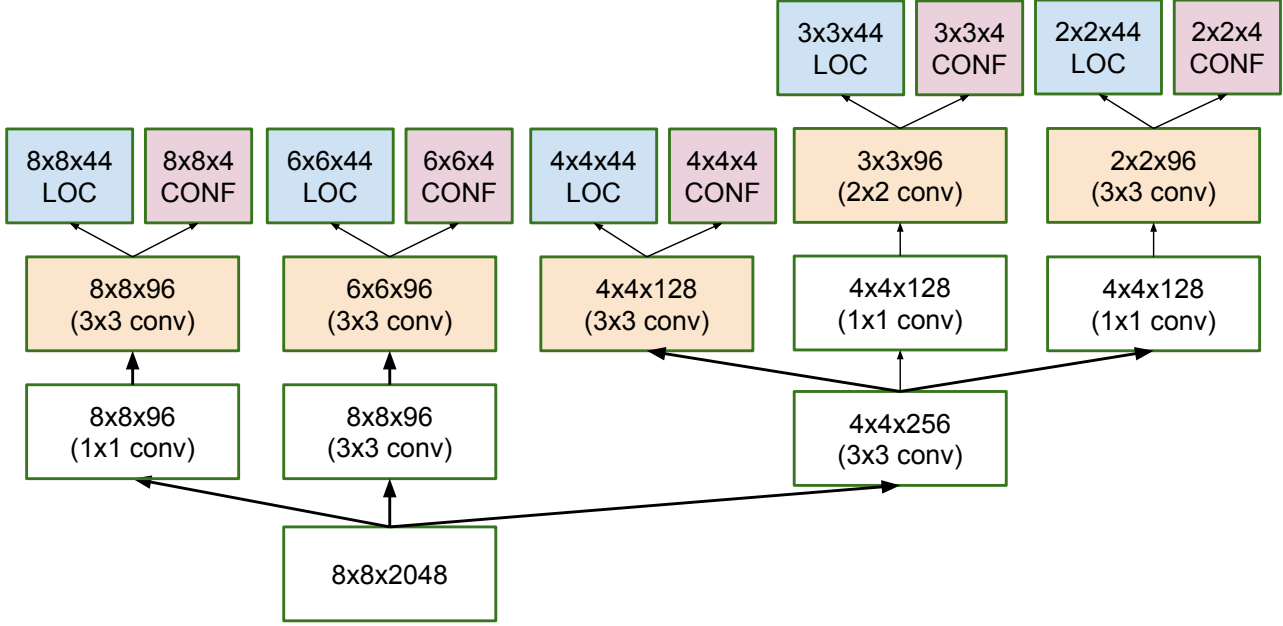
Figure 2. An illustration of the multi-scale convolutional prediction of the locations and confidences for MultiBox.

3.4. MultiBox network architecture

For both the MultiBox localizer model and the post-classifier, we have been using new variants of the Inception architecture as described in [28]. This is a 42 layers deep convolutional network over a 299×299 receptive field, containing over 130 layers. We are using the top $8 \times 8 \times 2048$ convolutional layer as described earlier. The extra side heads are removed for simplicity. The exact architecture topology is given in the supplementary `model.txt` file that can be downloaded together with the source file of this paper. Also we employ the spatially sensitive grid-size reduction technique as depicted in Figure 3.

3.5. Post-classification

MSC-MultiBox can be used in two ways: as a one-shot detector that produces object locations and confidences, or as a class-agnostic localizer providing region proposals to a post-classifier. However, in the high-quality regime, it is essential to zoom into the actual object proposals and perform an extra post-classification step to maximize performance. When used in this setting, an additional post-classification step is necessary. Again, for this use case we utilize the Inception architecture from [28].

3.6. Post-classifier architecture improvements

As a motivation for designing a new network architecture, we noted that the post-classifier network not only needs to produce the correct label for each class, but it also needs to decide whether the object overlaps the crop occupying the center part of the receptive field. (We follow the
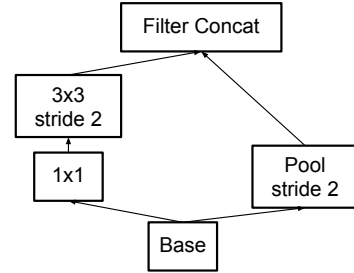

Figure 3. Inception module that avoids using a pooling layer alone to do the grid reduction. The stride 2 convolution can preserve the geometric information with less overhead.

cropping methodology of the R-CNN [9] paper.) This requires the network to be spatially sensitive.

We hypothesized that the large pooling layers of traditional network architectures – which are also inherited by the Inception [26] architecture – might be detrimental for accurately predicting spatial information. This leads to the construction of a variant of the Inception network, where in parallel to the large pooling layers [30] stride-2 convolutions are used in the Inception modules when reducing the grid size. This is depicted in Fig. 3.

3.7. Context Modeling

It is known that the global context can be useful when making predictions for local image regions. Most high-performing detectors use elaborate schemes to update scores or take whole-image classification into account. In-
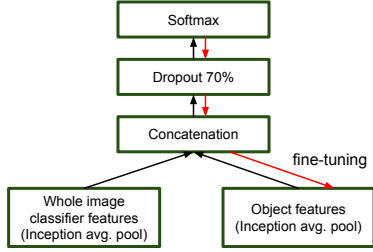
Figure 4. Scheme of the combiner architecture. Note, that in our setting fine-tuning was performed only for the object-feature network, shown in red above.

stead of working with scores, we just concatenate the whole image features with the object features, where the feature vector is taken from the topmost layer before the classifier. See Fig. 4.

Note, however that two separate models are used for the context and object features and they don't share weights.

The context classification network is trained first with the logistic objective, meaning that we have a separate logistic classifier for each class and the sum of their losses is used as the total objective of the whole network. We do not use the classifier output of the context network at object proposal evaluation time. The combiner network in fig 4 is trained in a second step after whole image features have been extracted. The combiner is uses a softmax classifier, since each bounding box can only have a single class. A designated "background" class is used for crops that don't overlap any of the objects with at least 0.5 intersection over union (IOU) similarity.

3.8. In-Model Context Ensembling

Another interesting feature of our approach that it allows for a computationally efficient form of ensembling at evaluation time. First we extract context features $\{f_i\}$ for k large crops in the image. In our case we used the whole image, 80% size squares in each corner and one same sized square at the center of the image. After context features f_i for each of those $k = 6$ features extracted, the final score will be given by $\sum C(f_i, N(p))/k$, which is the average of the combiner classifier C scores evaluated for each pair of context and object. This results in a modest (0.005-0.01 mAP), but consistent improvement at a relatively small additional cost, if there are a lot of proposals for each image and the combiner classifier is much cheaper to evaluate than extracting the features.

3.8.1 Training Methodology

All three models: the MultiBox, the context and post-classifier were trained with the Google DistBelief [3] machine learning system using stochastic gradient descent. The context and post-classifier networks reported in this paper had been pretrained on the 1.28 million images of the ILSVRC classification challenge task. We used only the classification labels during pretraining and ignored any available location information. The pretraining was done according to the prescriptions of [28]. All other models were trained with AdaGrad. There were two major factors that affected the performance of our models:

- The ratio of positives versus negatives during the training of the post-classifier. A ratio of $7 : 1$ negatives versus positive samples gave good results.

- Geometric distortions like random size and aspect ratio distortions proved to be crucial, especially for the MultiBox model. We have employed random aspect ratio distortions of up to $1.4\times$ in random (either horizontal or vertical) directions.

4. Results

4.1. Network architecture improvements

In this section we discuss aspects of the underlying convolutional network that benefited the detection performance. First, we found that switching from a Zeiler-Fergus-style network (detailed in [31]) to an Inception-style network greatly improved the quality of the MultiBox proposals (see Fig. 5). A thorough ablative study of the underlying network is not the focus of this paper, but we observed that for a given budget K, both the (class-agnostic) AP and maximum recall increased substantially by the change, as shown in Fig. 6.
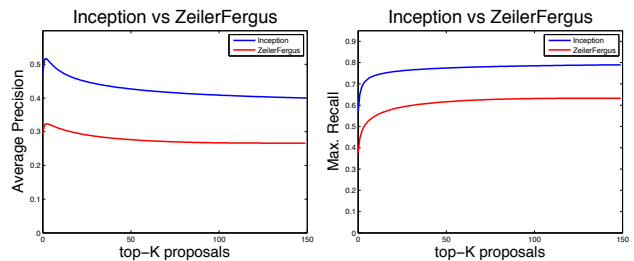

Figure 5. The inception architecture is particularly well-suited to localization, drastically improving over the Zeiler-Fergus architecture for MultiBox training.

Figure 6 also shows that with the Inception-style convolutional networks, increasing the number of priors from around 150 (used in the original MultiBox paper [4]) to 800 provided a large benefit. Beyond 800, we did not notice a significant improvement.

4.2. Runtime-quality trade-off

In this section we present an analysis of the runtime-quality trade-off in our proposed method. The detection runtime is determined mostly by the number of network
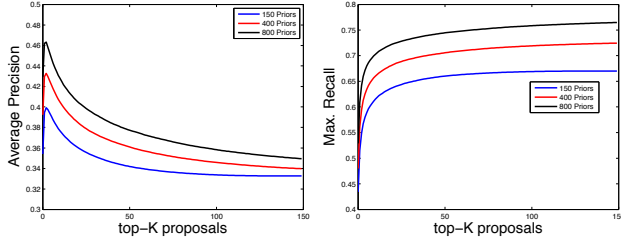
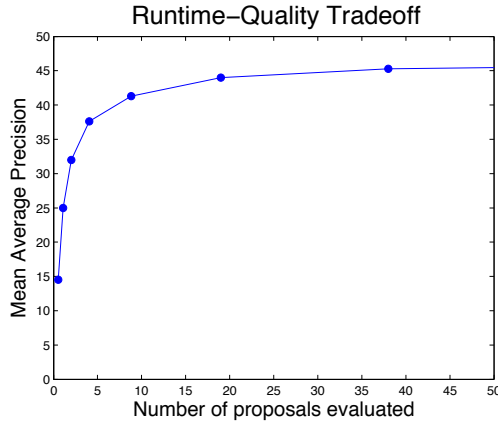Figure 6. Maximum recall and average precision tends to increase as the number of priors is increased.

Figure 7. A visualization of the trade-off between runtime (scaling with the number of proposal boxes) and quality (mAP) for a single model, single-crop MultiBox model. Note that the network was evaluated once on the image mapped affinely to the network input to generate all proposals using a single network evaluation.

evaluations, which scales linearly with the number of proposal boxes. Since MultiBox scores the region proposal boxes, we can achieve the maximum quality with the number of network evaluations we can afford by only evaluating the top-K most confident ones.

Figure 7 shows that performance degrades very gracefully with computational budget. Compared to the highest-quality operating point[1], very competitive performance (e.g. maintaining $> 90\%$ of the mAP) can be achieved with an order of magnitude fewer network evaluations. Also worth noting is that quality does not increase indefinitely with the number of proposals; swamping the post-classifier with low-quality proposals actually reduces the quality.

4.3. Contextual features

We used the same networks to generate both the contextual and non-contextual features, but the non-contextual network was trained without the extra context features. Both have a softmax classifier at the top and neither of them used hard negative mining, they were both pre-trained on the ImageNet classification challenge and used the same 42 lay-

---
[1]The mAP leveled off at around 45.8%.

ers deep Inception variant as the MultiBox proposal generation model. Table 1 shows that adding contextual features greatly improves results.

Model	mAP
Non-contextual MSC-MultiBox	0.473
Contextual MSC-MultiBox as in fig 4	0.5

Table 1. Control experiments for post-classification using contextual versus non-contextual models. Both models were trained as on multi-scale convolutional MultiBox-based proposals and with the multi-crop methodology described below.

4.4. Multibox on many image crops

The most efficient MultiBox solution generates proposals from a single network evaluation on a single image crop. We can increase the quality at the cost of a few more network evaluations by taking multiple crops of the image at multiple scales and locations, and combining all of the generated proposals and applying non-maximal suppression.

In the MultiBox case, one needs to be cautious: if the proposals are kept indiscriminately, then the system will produce high confidence boxes from partial objects that overlap the crop. This naive implementation ends up with a loss of quality. Our solution was to drop all the proposals that are not completely contained in the $(0.1, 0.1) - (0.9, 0.9)$ sub-window of the crop. However this implies that MultiBox should be applied on highly overlapping windows. We have run two experiments in which a 299×299 crop was slid over the image such that each window overlaps at least 50% (or 62.5%) each of its neighboring window in the dimension they are adjacent, respectively. This allows enough room for small object to be picked up by at least one of the crops evaluated with MultiBox.

Table 2 demonstrates that we can get almost 5% mAP improvement by taking multiple image crops in the proposal generating step. The resulting number of proposals increases from 13 per image to 51 per image on average, but is still significantly lower than that used by Selective Search.

4.5. ILSVRC2014 detection challenge

In this section we combine multi-scale convolutional MultiBox proposals with context features and a post-classifier network on the full 200-category ILSVRC2014 detection challenge data set.

Table 1 shows several rows, each of which lies on a different point along the runtime-quality trade-off. Note that our improved MultiBox pipeline with a *single* crop yields 0.45 mAP, which exceeds last year's GoogLeNet ensemble validation performance in the ILSVRC2014 competition, and is even higher than the latest and best known result published with Deep-ID-Net [16]. In addition, we attain superior performance at the high-precision operating point.

Model	mAP	boxes
MSC-MultiBox single-crop	0.45	13
MSC-MultiBox multi-crop (0.625)	0.5	51

Table 2. Control experiments using our models on the ILSVRC 2014 detection challenge validation set with a single model.

category	AP	Recall at 60% precision
person	0.6	63.1%
bird	0.91	93.1%
dog	0.94	95.7%
can opener	0.55	56.3%
table	0.36	33%
horizontal bar	0.097	0%

Table 4. Performance of the two-model ensemble on a few selected classes of ILSVRC-2015.

Given a single Multibox region proposal network and a single post-classifier model, we obtain 0.499 mAP.

We obtain even better results by using an ensemble of models. Naive ensembling, such as the one done by the GoogLeNet team on the ILSVRC 2014 detection challenge [26], uses a single Multibox network to propose boxes and then averages the result of several post-classifier models on the boxes. When we tried this with 3 post-classifier models, we got a mAP of 0.506 – a slight improvement.

We wanted to leverage the results of several different Multibox models, as well. Intuition suggests that box proposals that are consistent across several different Multibox models are more likely to be high-quality proposals. To capture this, we designed the following ensembling approach for N Multibox models. For the boxes of each Multibox model $j \in [1, N]$, we can use either a single post-classifier model, or average the scores of several post-classifier models, obtaining a set of bounding boxes l_i^j and class scores $c_{i,k}^j$, for each class k, and post-classifier model i. For each class score $c_{i,k}^j$, we aggregate scores from the other Multibox models as follows:

$$s_{i,k}^j = \frac{1}{N} \cdot (c_{i,k}^j + \sum_{n \neq j} \max_m (J(l_i^j, l_m^n) \cdot c_{m,k}^n), \quad (7)$$

where $J(\cdot)$ is the Jaccard overlap between the bounding boxes. Put in words, the objective above reinforces detections that have consistent matches in the other Multibox results both in terms of location (high Jaccard overlap) and high score. After computing these scores for all detections and scores of all Multibox models, we apply non-max suppression to keep only the best ones. This ensembling approach yielded 0.52 mAP with two Multibox models, a substantial improvement over the naive version. Table 3

Model	mAP (%)
Deep Insight ensemble	0.41
GoogLeNet ensemble	0.44
DeepID-Net ensemble	0.44
MSC-MultiBox single-crop	0.45
MSC-MultiBox multi-crop, one model	0.5
Ensemble of two models of MSC-MultiBox	0.52

Table 3. Comparison to the existing state-of-the-art results [22].

demonstrates that multi-scale convolutional MultiBox establishes a new state-of-the-art by a healthy margin.
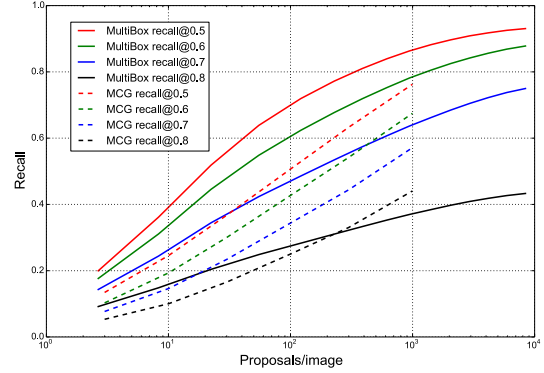

Figure 8. Per-class Average Recall of Multi-scale convolutional MultiBox at Jaccard ranging between 0.5 and 0.8 on the Microsoft-COCO [14] data set compared with proposals generated with MCG[18] (Multiscale Combinatorial Grouping). The corresponding MultiBox recall numbers are reported in Table 5.

4.6. Quality of MSC-MultiBox Proposals

proposals	Recall at Jaccard overlap			
	0.5	0.6	0.7	0.8
2.7	0.2	0.18	0.14	0.09
8.3	0.36	0.31	0.24	0.15
22	0.52	0.44	0.34	0.2
55	0.64	0.55	0.42	0.25
228	0.77	0.68	0.53	0.31
616	0.84	0.75	0.6	0.35
947	0.86	0.78	0.64	0.37
2056	0.9	0.83	0.68	0.4
4168	0.92	0.86	0.72	0.42
8409	0.93	0.88	0.75	0.43

Table 5. Multi-scale convolutional MultiBox Per-class average recall at various Jaccard thresholds on the Microsoft-COCO [14] validation set. Please refer to the corresponding Figure 8 which shows that MultiBox outperforms MCG at overlap thresholds up to 0.75. It still surpasses the recall of MCG for 0.8 when the budget is below 200 proposals per image.

In this section, we are comparing the coverage of our class-agnostic proposal generation method with the state-of-the-art Multiscale Combinatorial Grouping [18] ap-

proach on the Microsoft-COCO [14] validation set.

For this purpose, we have trained a class-agnostic Multi-Box model on top of the Inception-v3 network [28] using the TensorFlow [1] large scale distributed system with asynchronous gradient descent with 30 model replicas for 2 million batches, each of size 32.

For MultiBox, we have evaluated the crops from each image at three scales:

- The whole image was warped to the 299×299 receptive field of the network.

- A 299×299 square crop was slid on the image such that the minimum overlap between adjacent crops is at least 0.5. Only those proposals are kept that are completely contained in the center square covering 0.8×0.8 of the crop.

- A 185×185 square crop was slid on the image such that the adjacent crops have at least 0.5 overlap. This crop is scaled up to the 299×299 receptive field. Again all predicted proposals not fully contained in the the center 0.8×0.8 square are ignored.

Finally, for each image, we took the union of all proposals from each crop and ran non-maximum-suppression with Jaccard threshold 0.85.

To compute recall, the proposals are ranked by their confidence scores. We have took 15 different pre-sigmoid score thresholds ranging from 2 to -12. which gave rise to various average numbers of proposals per image. The results are reported in Table 5 and the corresponding Figure 8.

As one can see, MultiBox significantly outperforms MCG below 2000 proposals, especially for lower overlap threshold. MCG only outperforms MultiBox at 0.8 or higher thresholds with over 300 proposals. However, we expect that MultiBox might do better if pre-processed with less aggressive Non-Maximum-Suppression threshold (exceeding the currently used 0.85 threshold) when optimizing for recall at tight thresholds (above 0.75).

5. Conclusions

In this work we demonstrated a method for high-quality object detection that is simple, efficient and practical to use at scale.

The proposed framework flexibly allows the choice of operating point along the runtime-quality trade-off curve. Even using single-crop multi-scale convolutional Multi-Box with only several dozen proposals per image on average, we exceed the previously-reported state-of-the-art ILSVRC2014 detection performance, outperforming even highly-tuned ensembles using costly Selective Search proposal generation. At the high-quality end of the curve, we outperform the nearest reported mAP by over 10% relative.

We conclude that learning-based proposal generation has closed the performance gap with state-of-the-art engineered proposal generation methods, MCG [18] in our study, while reducing the computational cost of detection. This is mostly the result of improved underlying network architecture especially the use of multi-scale convolutional proposal generation. Improvements in training methodology, context modeling and inference-time tricks like multi-crop evaluation and in-model ensembling resulted in modest, but significant cumulative gains on ILSVRC Detection 2014. Multi-scale convolutional MultiBox is not just a computationally more efficient replacement for static proposal generating algorithms; by providing a smaller number of higher-quality proposals, multi-scale convolutional MultiBox improves the overall object detection performance.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

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

[3] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012.

[4] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2155–2162. IEEE, 2014.

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[6] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.

[7] K. Fukushima. Neural network model for a mechanism of pattern recognition unaffected by shift in position- neocognitron. *ELECTRON. & COMMUN. JAPAN*, 62(10):11–18, 1979.

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

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Computer Vision–ECCV 2014*, pages 346–361. Springer, 2014.

[11] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? *arXiv preprint arXiv:1406.6962*, 2014.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[13] Y. LeCun, L. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261:276, 1995.

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer, 2014.

[15] S. Manen, M. Guillaumin, and L. V. Gool. Prime object proposals with randomized prim's algorithm. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2536–2543. IEEE, 2013.

[16] 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. *arXiv preprint arXiv:1409.3505*, 2014.

[17] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. *arXiv preprint arXiv:1506.06204*, 2015.

[18] J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. In *arXiv:1503.00848*, March 2015.

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

[20] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training Deep Neural Networks on Noisy Labels with Bootstrapping. *arXiv preprint arXiv:1512.00567*, Dec. 2014.

[21] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. 2014.

[23] M. A. Sadeghi and D. Forsyth. 30hz object detection with dpm v5. In *Computer Vision–ECCV 2014*, pages 65–79. Springer, 2014.

[24] 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.

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[26] 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.

[27] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems*, pages 2553–2561, 2013.

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, S. Jonathon, and Z. Wojna. Rethinking the incption architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015.

[29] 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.

[30] J. Weng, N. Ahuja, and T. S. Huang. Cresceptron: a self-organizing neural network which grows adaptively. In *Neural Networks, 1992. IJCNN., International Joint Conference on*, volume 1, pages 576–581. IEEE, 1992.

[31] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.

[32] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014.