Scalable, High-Quality Object Detection

Christian Szegedy
Google
szegedy@google.com

Scott Reed
University of Michigan – Ann Arbor
reedscot@umich.edu

Dumitru Erhan
Google
dumitru@google.com

Dragomir Anguelov
Google
dragomir@google.com

Abstract

Most high quality object detection approaches use the same scheme: salience-based object proposal methods followed by post-classification using deep convolutional features. In this work, we demonstrate that fully learnt, data-driven proposal generation methods can effectively match the accuracy of their hand engineered counterparts, while allowing for very efficient runtime-quality trade-offs. This is achieved by making several key improvements to the MultiBox method [3], among which are an improved neural network architecture, use of contextual features and a new loss function that is robust to missing groundtruth labels. We show that our proposal generation method can closely match the performance of Selective Search [20] at a fraction of the cost. We report new state-of-the-art on the ILSVRC 2014 detection challenge data set, with 55.7% *mean average precision when combining both Selective Search and MultiBox proposals with our post-classification model. Finally, our approach allows the training of single class detectors that can process 50 images per second on a Xeon workstation, using CPU only, rivaling the quality of the current best performing methods.*

1. Introduction

After the dramatic improvements in object detection demonstrated by Girshick et al. [7], most of the current state of the art approaches, including the top performing entries [18, 13, 17] of the 2014 Imagenet object detection competition [14], make use of salience-based object localization, in particular Selective Search [20] 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 more efficient way of ranking the proposals is to cut down their number at generation time already. If successful, this can have the additional beneficial side-effect of improving the running time and quality at the same time. In the ideal case, proposals come with some class-agnostic ranking on the probability of being the bounding box of some object. This provides a way to balance recall versus running-time in a simple and 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.

The fact that hand-engineered features tend to get replaced by similar or higher-quality learned features [11, 10, 18] for image recognition, raises the question of whether can we expect our highest quality proposal generation method to be learned from scratch as well. An affirmative answer would suggest that custom made proposal generation and segmentation methods for special use cases like medical or aerial imaging could be replaced by a more general methodology, improving their quality and simplifying their pipelines as well.

Given the recent success of novel sophisticated proposal generation methods [20, 22, 12, 2, 9, 1], the possibility of directly learned methods overtaking engineered methods might seem far fetched at first glance. We still argue that using modern convolutional vision network architectures, the crossover point is almost reached. In section 4.4 we demonstrate that our fully learned proposal method closely rivals salience-based methods in performance at a significantly lower computational cost. Another advantage of the proposed approach is that it can make use of GPU computing[1] without significant engineering effort by leveraging exisiting frameworks for evaluating convolutional networks.

[1] All the results and running times presented in this paper were achieved by utilizing CPUs only.

On the other hand we are also interested in applying fully learned methods in a monolithic setting that does not involve postclassification at all, but produces detection results directly with a single network evaluation. In fact, we argue in Section 4.3 that we can get competitive results for single object types, like persons, even using this extremely fast and direct solution.

Our highest quality result comes from combining both methods: learned and saliency based proposals. We present significant improvements over the state the art [14] that uses the model-free Selective Search for proposal generation, but with modest computational cost increase.

The learning-based proposal method that our present work builds upon is the MultiBox approach presented in Erhan et al. [3], with improved underlying network architecture and training. We demonstrate that switching from classical AlexNet [10] and Zeiler-Fergus [21] networks to the new Inception [18]-style architecture together with improvements in the objective function and increased number of potential proposal boxes leads us to state-of-the-art detection performance.

We will also present object-detection-specific adjustments to the Inception architecture that enhance the quality of our models. Combining this with a simple but powerful 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.

All the models mentioned in this paper are trained only on the ILSVRC 2014 detection challenge training set after being pretrained only on the ILSVRC 2012 recognition challenge training sets. For the latter, we only use the labels, but **not** the localization data, which could have been helpful in this particular setting. All the graphs and numbers will refer to evaluation results on the ILSVRC 2014 detection challenge validation set.

Overall, this paper presents a case study of the various points on the runtime-quality trade-off-curve of our system, including the following highlights:

- Detection of single categories in the ILSVRC recognition challenge data, with 0.44 average precision for person and 0.704 for dogs using one network evaluation/image. This means a rate of 50 images/second on a current Xeon workstation[2], in batches of 64, without the use of GPU.

- 200 classes detection at 0.41 mAP with 15 proposals per image on average, without any hand-engineered proposal generation method.

- 0.557 mAP with an ensemble of three postclassifiers and utilizing both MultiBox and Selective Search proposals at a computational cost of 5916 network evaluations per image.

In what follows, we analyze the contribution of the various factors that lead to this performance.

2. Related Work

The previous state-of-the-art paradigm in detection is to use part-based models [5, 4] such as Deformable Part Models (DPMs). Sadeghi and Forsyth [15] developed a framework with several configurable runtime-quality tradeoffs and demonstrate real-time detection using DPMs on the PASCAL 2007 detection data.

Deep neural network architectures with repeated convolution and pooling layers [6, 11] have more recently become the dominant approach for large-scale and high-quality recognition and detection. Szegedy et al. [19] demonstrated effective use of deep neural networks for object detection formulated as a regression onto bounding box masks. Sermanet et al. [16] developed a multi-scale sliding window approach using deep neural networks, winning the ILSVRC2013 localization competition.

The original work on MultiBox [3] 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 improved detection recall by increasing the number of predicted proposals, while continuing to evaluate a fixed-size subset of those 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 [7]. He et al. proposed Spatial Pyramid Pooling [8] (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 MultiBox network, which also associates a confidence score to each proposal. Architectural changes to the underlying network and contextual postclassification 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 de-

[2]Intel(R) Xeon(R) CPU E5-1650 0 @ 3.20GHz, memory 32G.

sired, and post-classification of proposals can be sped up in the same way with no change to the MultiBox objective.

Related to our work is the Bing approach proposed by Cheng et al. [1] which employs a hybrid methodology of learning linear classifier over hand-engineered features. Their proposals come also scored, but the scoring is relatively weak and the computational effinciency gains are accompanied by a major quality loss when used a proposal generation component in a detection system.

3. Model

3.1. Background: MultiBox objective

In order to describe the changes to [3], 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 the bounding box of an object. In order to achieve this, the MultiBox loss is the weighted sum of the following two losses:

- **Confidence**: a logistic loss on the confidence that each proposal corresponds to an object of interest.
- **Location**: loss corresponding to some similarity measure between the ground-truth bounding boxes and the closest matching box predictions. By default we used L2 distance.

Fig. 1 provides a schematic description for the top layers of the network used for MultiBox. The network is an Inception-style [18] convolutional network, followed by a structured output layer producing a set of bounding box coordinates and confidence scores. Let c_i be the confidence of the i-th box containing an object of interest, $l_i \in \mathbb{R}^4$ be the i-th set of predicted box coordinates, and let $g_j \in \mathbb{R}^4$ denote the j-th ground-truth box coordinates. At training time, for each image, we perform a bipartite matching between the predictions and the ground-truth boxes. We denote $x_{ij} = 1$ to indicate that the i-th prediction is matched to the j-th groundtruth, 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

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

the confidence loss term can be written as follows:

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

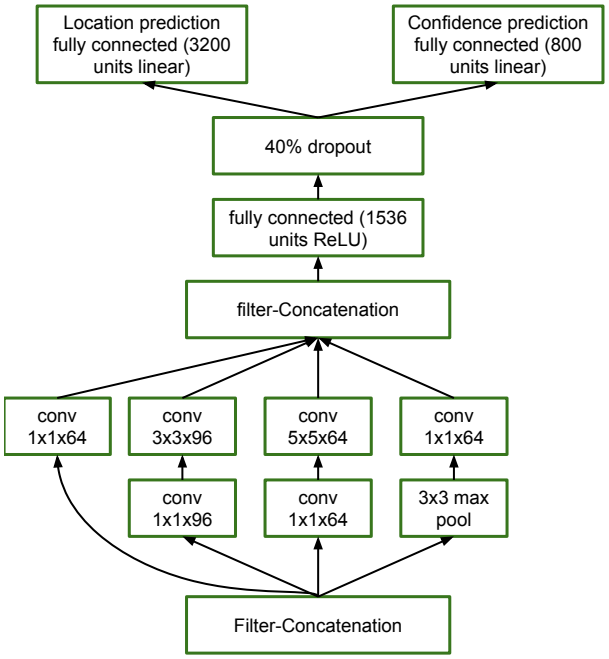

Figure 1. Head of the inception model tailored for MultiBox.

The overall objective is a weighted sum of both terms

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

We train the network with SGD. For each training example (c, l, g) we compute the best matching

$$\begin{aligned} x^* &= \arg\min_x F(x, c, l, g) \\ \text{such that} \quad & x_{ij} \in \{0, 1\}, \sum_i x_{ij} = 1 \end{aligned} \tag{4}$$

and update the network parameters after evaluating the gradient given x^*.

A crucial detail of the MultiBox method is that it does not let the proposals free-float. Instead, it imposes diversity by introducing a prior for each box output by the network. These prior boxes are computed before training to minimize the expected maximum overlap for each bounding box in the training set. The predicted coordinates are computed with respect to those priors, meaning that the network output is the delta to to the coordinates of the corresponding prior box, not the actual coordinate with respect to the whole image. However, it is a highly non-convex objective to minimize the expected maximum overlap, so we resort to the heuristic of performing k-means clustering of training bounding box coordinates, and taking the k-means centroids as priors. We learn the priors after performing the following transformation on box coordinates:

$$[x_{\text{center}}, y_{\text{center}}, c/(\epsilon + \text{width}), c/(\epsilon + \text{height})] \tag{5}$$

with $\epsilon = 0.01$ and $c = 0.2$ (found by cross-validation on the maximum overlap of priors with the groundtruth boxes), and where the image region is normalized to the unit box.

3.2. 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 missing or noisy training data may encourage the model to be overly conservative in its predictions and thereby reduce the recall of MultiBox proposals.

One way to overcome this incomplete labeling is to take the best available ensembled detection system, and use its highest-scoring detections to generate additional positive training labels for "bootstrapping" MultiBox training. A better MultiBox could be trained, leading to a better ensemble, and this long-loop bootstrapping procedure could even be repeated. However, such a procedure is slow due to repeated training and data generation, and it requires significant engineering effort for large data sets.

Motivated by the problem of missing positives, but desiring a less cumbersome solution, we introduced a novel training heuristic that empirically increases recall: At the start of each training batch, drop the top-L most confident predictions in each image from the confidence objective. L is chosen by cross-validation. This can also be viewed as using the model state at the start of a batch to add a limited number of true positives to the training data.

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 $F_{bootstrap}$.

3.3. MultiBox network architecture

For both the MultiBox localizer model and the postclassifier, we have been using variants of the Inception architecture as described in [18]. This is a 25 layers deep convolutional architecture, containing about 100 layers. Here we describe additional considerations that led to the architecture changes to fit it to the MultiBox use-case.

First, the extra side heads are removed for simplicity and also because their effect is poorly understood and hard to predict. The GoogLeNet [18] network has a big average pooling layer on top of the network that averages the whole preceding 7x7 convolutional layer spatially into a 1024 long feature vector which is fed into the classifier. The localizer and confidence prediction layers of the MultiBox cannot be put on top of the average pooling layer, since they both need to rely heavily on the spatial information. However, the output of each of the 7×7 units of the last Inception module is a 1024 vector, so even just putting a 400-box predictor on top would add $(4\times400 + 400)\times1024\times7\times7 = 100$ million extra parameters and the corresponding amount of computation to the model, effectively blowing up the amount of parameters by 20 times.

Instead, we connect the location and confidence predictors to the output of a newly added bottleneck layer fully connected to the last convolutional layers as shown in fig 1.

3.4. Postclassification

MultiBox can be used in two ways: as a fast, monolithic detector that produces object locations and confidences, or as a class-agnostic proposal generator producing region proposals in conjunction a post-classifier. Monolitic detection without the extra postclassification step is demonstrated in section 4.3. However, in the high-quality regime, it is essential to zoom into the actual object proposals and perform an extra postclassification step to maximize performance. When used in this setting, an additional postclassification step is necessary. Again, for this use case we utilize the Inception architecture from [18].

3.5. Postclassifier architecture improvements

As motivation for designing a new network architecture, we noted that the postclassifier 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 cropping methodology of the R-CNN [7] 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 [18] architecture – might be detrimental for accurately predicting spatial information. This leads to the construction of the following variant of the Inception network, in which the large pooling layers are completely omitted and instead stride-2 convolutions are used in the Inception modules to reduce the grid size.

Table 1 shows the actual layer dimensions of our best performing model. After the first two max-pooling layers, the only nonlinearities are rectified linear units, as all the pooling is average pooling. However, these are embedded within the filter concatenation module, in parallel

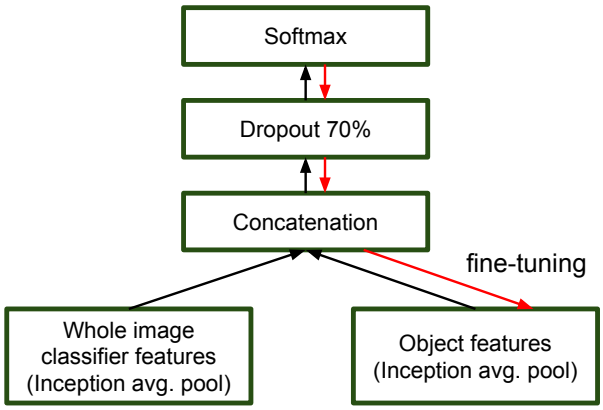

Figure 2. Scheme of the combiner architecture. Note, that in our setting fine-tuning was performed only for the object-feature network, shown in red above.

with convolutions. For the $28\times28 \longrightarrow 14\times14$ (3c) and $14\times14 \longrightarrow 7\times7$ (4e) grid-size reduction modules, there are therefore also paths that go through convolutional units with learned parameters.

Another interesting aspect is that this architecture allows for reducing the computational cost in the modules preceeding the grid-size reductions. This savings was spent on bumping up filter bank sizes in the 14×14 modules to 640, as well as on adding the new (3c) module. Given the success of this network, it is hypothesized that replacing the last average pooling layer, or the early max-pooling layers could be beneficial too. The computational cost of this network is roughly the same at 1.5Bn multiply-add operations as the original Inception network in [18].

3.6. Context Modelling

It is known that whole-image features can be useful when making predictions within local image regions. Most high-performing detectors use elaborate schemes to update scores or take whole-image classification into account. Instead 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. 2.

Note, however that two separate models are used for the context and object features and they dont share weights.

The context classification network is trained first with the logistic objective (a separate logistic classifier for each class). We used logistic loss rather than softmax because several different objects can be present in each image.

We do not use the classifier output of the context network at object proposal evaluation time. The combiner network in Fig. 2 is trained in a second step after the whole image features have been extracted. The combiner is uses a softmax classifier, since each bounding box encloses tightly only a single object. A designated "background" class is used for crops that don't overlap any of the objects with at least 0.5 in Jaccard similarity.

3.7. 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 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 c_i for each of those $k = 6$ features extracted, the final score will be given by $\sum C(c_i, N(p))/k$, 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, which is typically the case.

3.7.1 Training Methodology

All three models: the MultiBox, the context model and the postclassifier were trained with the Google DistBelief machine learning system using stochastic gradient descent. The context and postclassifier 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, but did not use any available bounding box information. The pretraining was done according to the prescriptions of [18]. 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 postclassifier. 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 [21]) to an Inception-style network greatly improved the quality of the MultiBox proposals. A thorough ablative study of the underlying network is not the focus of this paper, but we observed that for a given

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	avg Pool proj
convolution	7×7/2	112×112×64	1						
max pool	3×3/2	56×56×64	0						
convolution	3×3/1	56×56×192	2		64	192			
max pool	3×3/2	28×28×192	0						
inception (3a)		28×28×256	2	64	96	128	16	32	32
inception (3b)		28×28×320	2	64	96	128	32	64	64
inception (3c)	stride 2	28×28×640	2	0	128	256	32	64	pass through
inception (4a)		14×14×640	2	256	96	192	32	64	128
inception (4b)		14×14×640	2	224	112	224	32	64	128
inception (4c)		14×14×640	2	192	128	256	32	64	128
inception (4d)		14×14×640	2	160	144	288	32	64	128
inception (4e)	stride 2	14×14×1024	2	0	160	256	64	128	pass through
inception (5a)		7×7×1024	2	384	192	384	48	128	128
inception (5b)		7×7×1024	2	384	192	384	48	128	128
avg pool	7×7/1	1×1×1024	0						

Table 1. Inception type network for detection postclassification, shown up to the object feature layer in the bottommost row.

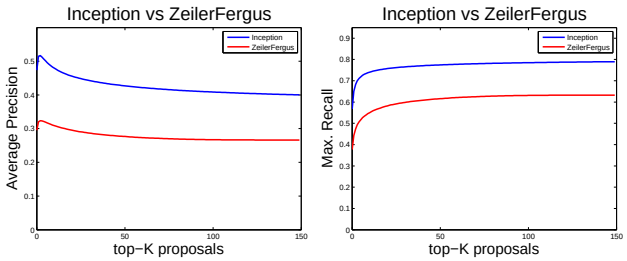

Figure 3. MultiBox proposal improvement due to switching from the Zeiler-Fergus to the Inception architecture.

budget K, both the (class-agnostic) AP and maximum recall increased substantially by the change.

With the Inception-style convolutional networks, we found that increasing the number of priors from around 150 (used in the original MultiBox paper [3]) to 800 also provided a large benefit. Beyond 800, we did not notice a significant improvement.

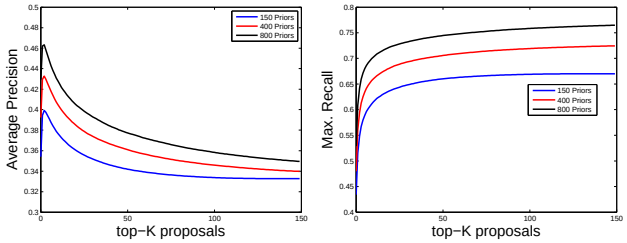

Figure 4. Maximum recall and average precision tends to increase as the number of priors is increased.

4.2. Runtime-quality tradeoff

In this section we present an analysis of the runtime-quality trade-off of our method. The detection runtime is determined mostly by the number of network evaluations, which scales linearly with the number of evaluated proposal boxes. We use the Multibox confidence estimate to prune the number of evaluated boxes. On aggregate, we found that post-classifying all boxes whose scores are above a fixed threshold is more effective than evaluating the top-K most confident boxes per image.

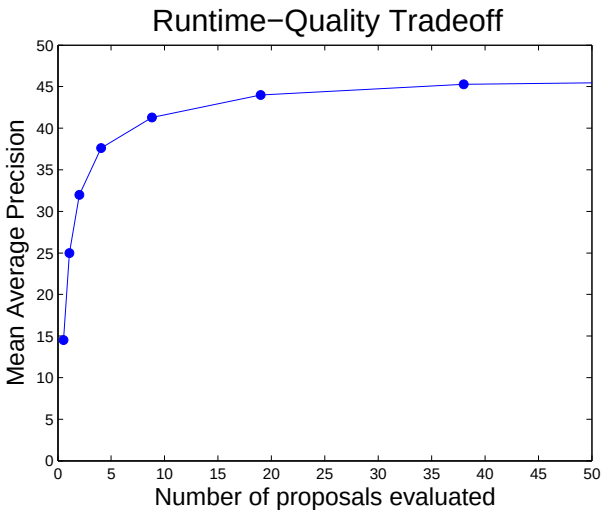

Figure 5. Trade-off between evaluated proposal boxes and quality (mAP) for a single-crop MultiBox model (which linearly reshapes the whole image to match the network input size) followed by a single post-classifier model. Only boxes whose MultiBox confidence exceeds a given threshold are evaluated – the x-axis shows the average number of evaluated boxes per image in the test set.

Fig. 5 shows that performance degrades very gracefully with the decrease of computational budget. Compared to

the highest-quality operating point[3], very competitive performance (e.g. maintaining $> 90\%$ of the mAP) can be achieved with an order of magnitude fewer network evaluations. These results validate the quality of our Multibox confidence estimates. Also worth noting is that the curve eventually flattens – any potential recall gains from extra boxes seem to be offset by the increased likelihood of post-classification errors.

4.3. Fast monolithic detection in MultiBox

In this section we demonstrate that pure MultiBox, without the extra postclassification step, can be used for very fast detection of a particular object type, such as person, using a single network evaluation.

Detector	AP	Recall @ 60prec
Person	0.309	17.8 %
Person (bootstrap)	0.44	43.9 %
Dog	0.704	77.0 %
Dog (bootstrap)	0.694	76.6 %
Person (full)	0.514	56.2 %
Dog (full)	0.895	90.5 %

Table 2. A comparison of monolithic and full-pipeline detection within the ILSVRC 2014 detection challenge validation set. Top: Rows with "bootstrap" indicate that we use bootstrapping to fine-tune the MultiBox network. Bottom: Rows with "full" show the performance of the conventional (> 3x more expensive) detection pipeline with single-crop MultiBox proposals, deep context features and a postclassifier network.

Table 2 shows that bootstrapping gives a big improvement for the person single-shot detector, but for dogs the performance is slightly worse. We hypothesize that this is because there are far more unlabeled humans appearing in the images compared to unlabeled dogs; typically dog photos contain one or a small number of dogs that are the subject of the image, and they are well-covered by labels.

Detector	mAP	# Proposals
MultiBox	0.455	71
MultiBox (bootstrap)	0.459	71

Table 3. Full 200-way detection on the ILSVRC2014 detection challenge data, with our best single-crop MultiBox detector, with and without bootstrap training. Both models evaluate on the same set of proposals per image at threshold -7.0 on the logits.

In the full detection dataset with 200 categories, we also observe a significant improvement from bootstrapping (see Table 3), suggesting that many categories with missing labels can benefit from bootstrap training.

[3]The mAP leveled off at around 45.8%.

4.4. Contextual features

We observed some of our biggest gains from using contextual models (second only to switching Multibox to the Inception architecture). Given the huge gain provided by the contextual models and the very different pipelines required to run their non-contextual counterparts, we stopped generating non-contextual models relatively early and focused our attention on further improving our contextual system. As a result, the comparison we present in Table 6 represents the relative gain from switching from the non-contextual to the contextual model. The numbers in this control experiment do not include some of the most recent improvements: better networks, training methodology and improved MultiBox loss function.

We used the same networks to generate both the contextual and non-contextual models, but the non-contextual model employed SVM with hard negative mining as a classifier while the contextual model was trained with a softmax layer on top of the concatenated feature-vector as depicted in fig 2. We have tried to train various similar non-contextual model with softmax layer as well, but they turned out to produce significantly inferior results to the SVM based version (see Table 4). Note that despite this potential drawback, the Contextual model reaches 0.04 higher mean average precision without employing hard negative mining, as opposed to the SVM based version.

Model	mAP
Noncontextual + SVM + HNM	0.388
Contextual (using softmax as in fig 2)	0.429

Table 4. A comparison of using contextual versus non-contextual post-classification models without all the other improvements described in this paper. Both experiments were performed using the same mixture of Selective Search and MultiBox-based proposals. An earlier version of the MultiBox model postclassifier networks were used here.

All the following experiments below were done with our latest, fully optimized models.

4.5. Comparison to Selective Search

In this section we compare various MultiBox detection pipelines to Selective search in terms of quality and efficiency. The most efficient MultiBox results are obtained by generating 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 $(0.1, 0.1) - (0.9, 0.9)$ subwindow of the crop. However this implies that MultiBox should be applied on highly overlapping windows. We have run two experiments in which a 224×224 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.

Model	mAP	# Proposals
MultiBox single-crop	0.459	71
MultiBox multi-crop with 0.5 overlap.	0.502	353
MultiBox multi-crop with 0.625 overlap.	0.507	643

Table 5. Higher-quality MultiBox using multiple image crops to generate proposals.

Table 5 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 significantly, but is still significantly lower than that used by Selective Search.

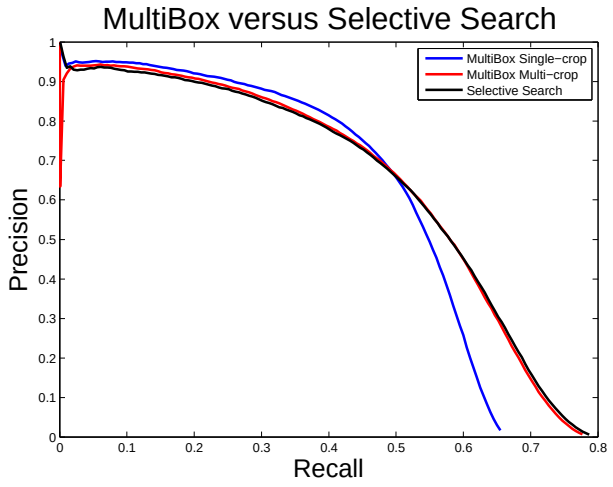

Figure 6. Precision-recall curves overlaid for MultiBox single crop, multi-crop and Selective Search.

Fig. 6 shows the relative strengths of MultiBox and Selective search. At the high-precision part of the precision-recall curve, single-crop MultiBox performs the best. Selective Search still performs very slightly better at the high-recall part of the curve.

4.6. ILSVRC2014 detection challenge

In this section we combine MultiBox proposals with context features and a post-classifier network on the full

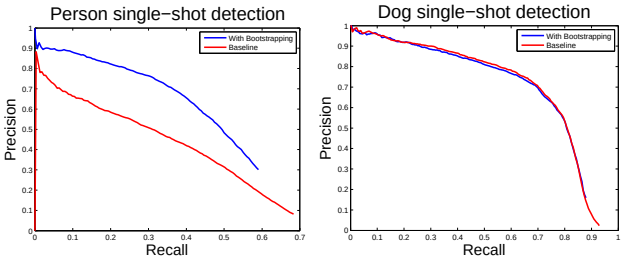

Figure 7. Single-shot person detection. The bootstrapping trick gives much better performance for the person class, and slightly worse for dogs.

Model	mAP	#Proposals
MultiBox single-crop	0.459	71
MultiBox multi-crop (0.625)	0.507	643
Selective Search	0.513	1126

Table 6. Control experiments using our models on the ILSVRC 2014 detection challenge validation set with a single model.

200-category ILSVRC2014 detection challenge data set.

Table 6 shows several rows, each of which lies on a different point along the runtime-quality tradeoff. Note that our improved MultiBox pipeline with a *single* crop yields 0.46 mAP, which is significantly higher than the best GoogLeNet ensemble validation performance in the ILSVRC2014 competition, and even higher than the latest and best known result published with Deep-ID-Net [13]. By feeding multiple crops of the input image to MultiBox, our best model can further improve to 0.507 mAP with a single model. Note that this performs nearly as well as our best pure Selective Search pipeline with 0.513 mAP, at a substantially reduced computational cost. In addition, we attain superior performance at the high-precision operating point.

We achieve best results, with mAP over 0.1 better than the best previously-reported result, by combining Selective Search and MultiBox proposals. This strategy was also used by the GoogLeNet team to win the ILSVRC2014 detection challenge [18], but in this case the MultiBox and postclassifier models are both far more accurate due to our recent improvements described in the previous sections.

Table 7 demonstrates that MultiBox proposals alone outperform the best previous state-of-the-art results, and combined with Selective Search proposals and ensembled, we establish a new state-of-the-art by a large margin.

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 tradeoff curve. Even using single-crop MultiBox with only several dozen

Model	mAP	#Proposals
Deep Insight ensemble	0.405	unknown
GoogLeNet ensemble	0.442	7200
DeepID-Net ensemble	0.45	unknown
MultiBox single-crop	0.459	71
MultiBox multi-crop, one model	0.507	643
Selective search, one model	0.513	1126
Selective search, four models	0.54	4504
Our best ensemble of four models: MultiBox + sel. search	0.557	5916

Table 7. Comparison to the existing state-of-the-art results [14].

proposals per image on average, we exceed the previously-reported state-of-the-art performance on the ILSVRC2014 detection challenge, outperforming even highly-tuned ensembles using costly Selective Search proposal generation. This result demonstrates that learning-based proposal generation has nearly closed the performance gap with Selective Search while reducing the computational cost of detection. Our main advances are due to improved underlying network architecture and of introducing context features to the postclassifier. Advances in training methodology and evaluation-time improvements like multi-crop evaluation and in-model ensembling resulted in additional gains on ILSVRC 2014. Last but not least, we developed a modification to the MultiBox objective to handle the common case of missing positive labels in the data, especially for commonly-appearing categories such as person. We showed that this bootstrapping objective for confidence prediction substantially improves single-shot person detection, and also improves the overall 200-category detection performance.

Finally, we would like to point out that MultiBox is not just a computationally more efficient replacement for static proposal generating algorithms; it can be used in conjunction with existing bounding box proposal methods to obtain even higher-quality results. By combining MultiBox and Selective Search proposals, we were able to obtain a mAP of 0.557 on the ILSVRC 2014 detection challenge. This result improves over 0.1 over the current state-of-the-art and further validates the quality of our approach.

References

[1] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *IEEE CVPR*, 2014. 1, 3

[2] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. 2013. 1

[3] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. *arXiv preprint arXiv:1312.2249*, 2013. 1, 2, 3, 6

[4] 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. 2

[5] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973. 2

[6] 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. 2

[7] 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. 1, 2, 4

[8] 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. 2

[9] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? *arXiv preprint arXiv:1406.6962*, 2014. 1

[10] 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. 1, 2

[11] 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. 1, 2

[12] 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. 1

[13] 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. 1, 8

[14] 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. 1, 2, 9

[15] M. A. Sadeghi and D. Forsyth. 30hz object detection with dpm v5. In *Computer Vision–ECCV 2014*, pages 65–79. Springer, 2014. 2

[16] 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. 2

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 1, 2, 3, 4, 5, 8

[19] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems*, pages 2553–2561, 2013. 2

[20] 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. 1

[21] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014. 2, 5

[22] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014. 1