# R-FCN-3000 at 30fps: Decoupling Detection and Classification

Bharat Singh*[1] Hengduo Li*[2] Abhishek Sharma[3] Larry S. Davis[1]
University of Maryland, College Park [1] Fudan University[2] Gobasco AI Labs[3]
{bharat,lsd}@cs.umd.edu lihd14@fudan.edu.cn abhisharayiya@gmail.com

## Abstract

*We present R-FCN-3000, a large-scale real-time object detector in which objectness detection and classification are decoupled. To obtain the detection score for an RoI, we multiply the objectness score with the fine-grained classification score. Our approach is a modification of the R-FCN architecture in which position-sensitive filters are shared across different object classes for performing localization. For fine-grained classification, these position-sensitive filters are not needed. R-FCN-3000 obtains an mAP of 34.9% on the ImageNet detection dataset and outperforms YOLO-9000 by 18% while processing 30 images per second. We also show that the objectness learned by R-FCN-3000 generalizes to novel classes and the performance increases with the number of training object classes - supporting the hypothesis that it is possible to learn a universal objectness detector. Code will be made available.*

## 1. Introduction

With the advent of Deep CNNs [16, 20], object-detection has witnessed a quantum leap in the performance on benchmark datasets. It is due to the powerful feature learning capabilities of deep CNN architectures. Within the last five years, the mAP scores on PASCAL [9] and COCO [24] have improved from 33% to 88% and 37% to 73% (at 50% overlap), respectively. While there have been massive improvements on standard benchmarks with tens of classes [13, 12, 31, 6, 14], little progress has been made towards real-life object detection that requires real-time detection of thousands of classes. Some recent efforts [30, 17] in this direction have led to large-scale detection systems, but at the cost of accuracy. We propose a solution to the large-scale object detection problem that outperforms YOLO-9000 [30] by 18% and can process 30 images per second while detecting 3000 classes, referred to as R-FCN-3000.

R-FCN-3000 is a result of systematic modifications to some of the recent object-detection architectures [6, 5, 23, 25, 29] to afford real-time large-scale object detection. Recently proposed fully convolutional class of detectors [6, 5, 23, 25, 29] compute per-class objectness score for a given image. They have shown impressive accuracy within limited computational budgets. Although fully-convolutional representations provide an efficient [19] solution for tasks like object detection [6], instance segmentation [22], tracking [10], relationship detection [41] etc., they require class-specific sets of filters for each class that prohibits their application for large number of classes. For example, R-FCN [5]/ Deformable-R-FCN [6] requires 49/197 position-specific filters for each class. Retina-Net [23] requires 9 filters for each class for each convolutional feature map. Therefore, such architectures would need hundreds of thousands of filters for detecting 3000 classes, which will make them extremely slow for practical purposes.

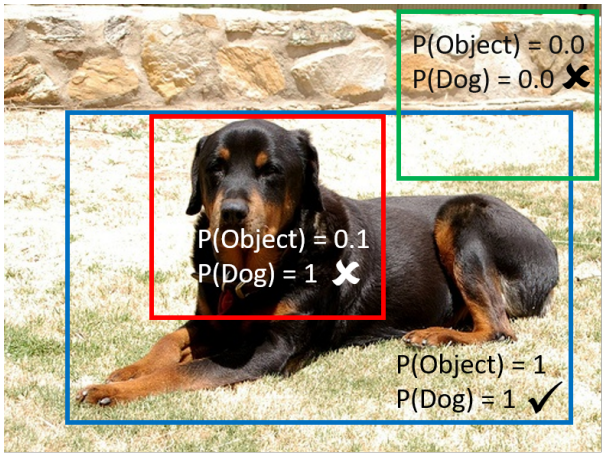


Figure 1. We propose to decouple classification and localization by independently predicting objectness and classification scores. These scores are multiplied to obtain a detector.

The key insight behind the proposed R-FCN-3000 architecture is to decouple *objectness detection* and classification of the detected object so that the computational requirements for localization remain constant as the number of classes increases - see Fig. 1. We leverage the fact that

*Equal Contribution. Work done during H. Li's internship at UMD.

many object categories are visually similar and share parts. For example - different breeds of dogs all have common body parts; therefore, learning a different set of filters for detecting each breed is overkill. So, R-FCN-3000 performs object detection (with position-sensitive filters) for a fixed number of *super-classes* followed by fine-grained classification (without position-sensitive filters) within each super-class. The super-classes are obtained by clustering the deep semantic features of images (2048 dimensional features of ResNet-101 in this case); therefore, we do not require a semantic hierarchy. The fine-grained class probability at a given location is obtained by multiplying the super-class probability with the classification probability of the fine-grained category within the super-class.

In order to study the effect of using super-classes instead of individual object categories, we varied the number of super-classes from 1 to 100 and evaluated the performance on the ImageNet detection dataset. Surprisingly, the detector performs well even with one super-class! This observation indicates that position-sensitive filters can potentially learn to detect universal objectness. It also reaffirms a well-researched concept from the past [1, 2, 39] that objectness is a generic concept and a universal objectness detector can be learned. Thus, for performing object detection, it suffices to multiply the objectness score of an RoI with the classifiation probability for a given class. This results in a fast detector for thousands of classes, as per-class position sensitive filters are no longer needed. On the PASCAL-VOC dataset, with only our objectness based detector, we observe a 1.5% drop in mAP compared to the deformable R-FCN [6] detector with class-specific filters for all 20 object classes. R-FCN-3000, trained for 3000 classes, obtains an 18% improvement in mAP over the current state-of-the-art large scale object detector (YOLO-9000) on the ImageNet detection dataset. Finally, we also evaluate the generalizability of our objectness detector on *unseen* classes (a zero-shot setting for localization) and observe that the generalization error decreases as we train the objectness detector on larger numbers of classes.

## 2. Related Work

Large scale localization using deep convolutional networks was first performed in [33, 35] which used regression for predicting the location of bounding boxes. Later, RPN [31] was used for localization in ImageNet classification [15]. However, no evaluations were performed to determine if these networks generalize when applied on detection datasets without specifically training on them. Weakly-supervised detection has been a major focus over the past few years for solving large-scale object detection. In [17], knowledge of detectors trained with bounding boxes was transferred to classes for which no bounding boxes are available. The assumption is that it is possible to train object detectors on a fixed number of classes. For a class for which supervision is not available, transformations are learned to adapt the classifier to a detector. Multiple-instance learning based approaches have also been proposed which can leverage weakly supervised data for adapting classifiers to detectors [18]. Recently, YOLO-9000 [30] jointly trains on classification and detection data. When it sees a classification image, classification loss is back-propagated on the bounding box which has the highest probability. It assumes that the predicted box is the ground truth box and uses the difference between other anchors and the predicted box as the objectness loss. YOLO-9000 is fast, as it uses a lightweight network and uses 3 filters per class for performing localization. For performing good localization, just 3 priors are not sufficient.

For classifying and localizing a large number of classes, some methods leverage the fact that parts can be shared across objects categories [27, 32, 37, 28]. Sharing filters for object parts reduces model complexity and also reduces the amount of training data required for learning part-based filters. Even in traditional methods, it has been shown that when filters are shared, they are more generic [37]. However, current detectors like Deformable-R-FCN [6], R-FCN [5], RetinaNet [23] do not share filters (in the final classification layer) across object categories: because of this, inference is slow when they are applied on thousands of categories. Taking motivation from prior work on sharing filters across object categories, we propose an architecture where filters can be shared across some object categories for large scale object detection.

The extreme version of sharing parts is objectness, where we assume that all objects have something in common. Early in this decade (if not before), it was proposed that objectness is a generic concept and it was demonstrated that only a very few category agnostic proposals were sufficient to obtain high recall [39, 3, 2, 1]. With a bag-of-words feature-representation [21] for these proposals, better performance was shown compared to a sliding-window based part-based-model [11] for object detection. R-CNN [13] used the same proposals for object detection but also applied per-class bounding-box regression to refine the location of these proposals. Subsequently, it was observed that per-class regression was not necessary and a class-agnostic regression step is sufficient to refine the proposal position [5]. Therefore, if the regression step is class agnostic, and it is possible to obtain a reasonable objectness score, a simple classification layer should be sufficient to perform detection. We can simply multiply the objectness probability with the classification probability to make a detector! Therefore, in the extreme case, we set the number of super-classes to one and show that we can train a detector which obtains an mAP which is very close to state-of-the-art object detection architectures [5].

## 3. Background

This section provides a brief introduction of Deformable R-FCN [6] which is used in R-FCN-3000. In R-FCN [5], *Atrous* convolution [4] is used in the conv5 layer to increase the resolution of the feature map while still utilizing the pre-trained weights from the ImageNet classification network. In Deformable-R-FCN [6], the atrous convolution is replaced by a deformable convolution structure in which a separate branch predicts offsets for each pixel in the feature map, and the convolution kernel is applied after the offsets have been applied to the feature-map. A region proposal network (RPN) is used for generating object proposals, which is a two layer CNN on top of the conv4 features. Efficiently implemented local convolutions, referred to as position sensitive filters, are used to classify these proposals.

## 4. Large Scale Fully-Convolutional Detector

This section describes the process of training a large-scale object detector. We first explain the training data requirements followed by discussions of some of the challenges involved in training such a system - design decisions for making training and inference efficient, appropriate loss functions for a large number of classes, mitigating the domain-shift which arises when training on classification data.

### 4.1. Weakly Supervised vs. Supervised?

Obtaining an annotated dataset of thousands of classes is a major challenge for large scale detection. Ideally, a system that can learn to detect object instances using partial image level tags (class labels) for the objects present in training images would be preferable because large-scale training data is readily available on the internet in this format. Since the setting with partial annotations is very challenging, it is commonly assumed that labels are available for all the objects present in the image. This is referred to as the *weakly supervised* setting. Unfortunately, explicit boundaries of objects or atleast bounding-boxes are required as supervision signal for training accurate object detectors. This is the *supervised* setting. The performance gap between supervised and weakly supervised detectors is large - even 2015 object detectors [15] were better by 40% on the PASCAL VOC 2007 dataset compared to recent weakly supervised detectors [8]. This gap is a direct result of insufficient learning signal coming from weak supervision and can be further explained with the help of an example. For classifying a dog among 1000 categories, only body texture or facial features of a dog may be sufficient and the network need not learn the visual properties of its tail or legs for correct classification. Therefore, it may never learn that legs or tail are parts of the dog category, which are essential to obtain accurate boundaries.

On one hand, the huge cost of annotating bounding boxes for thousands of classes under settings similar to popular detection datasets such as PASCAL or COCO makes it prohibitively expensive to collect and annotate a large-scale detection dataset. On the other hand, the poor performance of weakly supervised detectors impedes their deployment in real-life applications. Therefore, we ask - is there a middle ground that can alleviate the cost of annotation while yielding accurate detectors? Fortunately, the ImageNet database contains around 1-2 objects per image; therefore, the cost of annotating the bounding boxes for the objects is only a few seconds compared to several minutes in COCO [24]. It is because of this reason that the bounding boxes were also collected while annotating ImageNet! A potential downside of using ImageNet for training object detectors is the loss of variation in scale and context around objects available in detection datasets, but we do have access to the bounding-boxes of the objects. Therefore, a natural question to ask is, how would an object detector perform on "detection" datasets if it were trained on classification datasets with bounding-box supervision? We show that careful design choices with respect to the CNN architecture, loss function and training protocol can yield a large-scale detector trained on the ImageNet classification set with significantly better accuracy compared to weakly supervised detectors.

### 4.2. Super-class Discovery

Fully convolutional object detectors learn class-specific filters based on scale & aspect-ratio [23] or in the form of position sensitive filters [5, 6] for each class. Therefore, when the number of classes become large, it becomes computationally in-feasible to apply these detectors. Hence, we ask is it necessary to have sets of filters for each class or can they be shared across visually similar classes? In the extreme case - can detection be performed using just a foreground/background detector and a classification network? To obtain visually similar sets of objects for which position-sensitive filters can be shared, objects should have similar visual appearances. We obtain the $j^{th}$ object-class representation, $x_j$, by taking the average of 2048-dimensional feature-vectors ($x_j^i$), from the final layer of ResNet-101, for the all the samples belonging to the $j^{th}$ object-class in the ImageNet classification dataset (validation set). Super-classes are then obtained by applying K-means clustering on $\{x_j : j \in \{1, 2, \dots C\}\}$, where $C$ is the number of object-classes, to obtain $K$ super-class clusters.

### 4.3. Architecture

First, RPN is used for generating proposals, as in [6]. Let the set of individual object-classes the detector is being trained on be $\mathcal{C}, |\mathcal{C}| = C$, and the set of super-classes (SC) be $\mathcal{K}, |\mathcal{K}| = K$. For each super-class $k$, suppose we

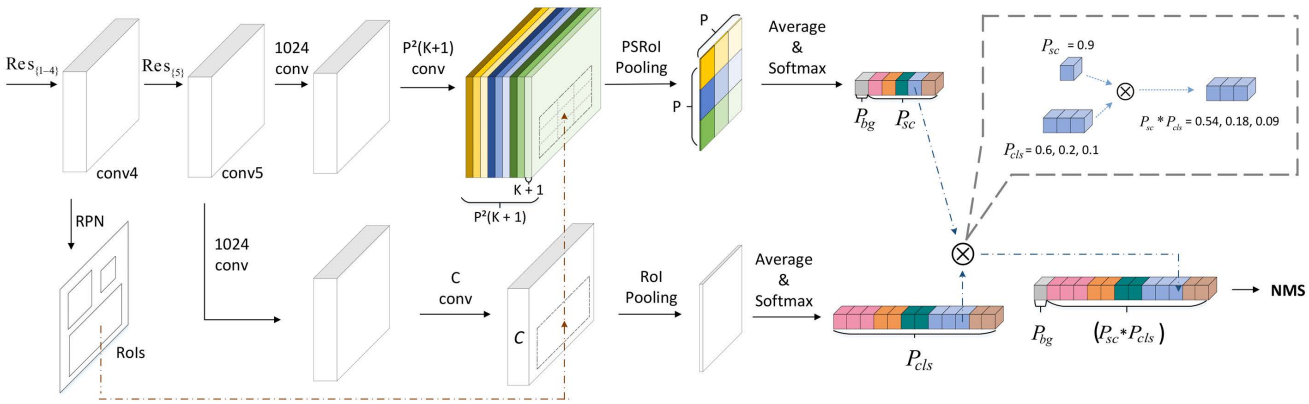


Figure 2. R-FCN-3000 first generates region proposals which are provided as input to a super-class detection branch (like R-FCN) which jointly predicts the detection scores for each super-class (sc). A class-agnostic bounding-box regression step refines the position of each RoI (not shown). To obtain the semantic class, we do not use position-sensitive filters but predict per class scores in a fully convolutional fashion. Finally, we average pool the per-class scores inside the RoI to get the classification probability. The classification probability is multiplied with the super-class detection probability for detecting 3000 classes. When K is 1, the super-class detector predicts objectness.

have $P \times P$ position-sensitive filters, as shown in Fig 2. On the conv5 feature, we first apply two independent convolution layers as in R-FCN for obtaining detection scores and bounding-box regression offsets. On each of these branches, after a non-linearity function, we apply position sensitive filters for classification and bounding-box regression. Since we have $K$ super-classes and $P \times P$ filters per super-class, there are $(K+1) \times P \times P$ filters in the classification branch (1 more for background) and $P \times P$ filters in the bounding-box regression branch as this branch is class-agnostic. After performing position-sensitive RoI pooling and averaging the predictions in each bin, we obtain predictions of the network for classification and localization. To get the super-class probability, softmax function over $K$ super-classes is used and predictions from the localization branch are directly added to get the final position of the detection. These two branches help detect the super-classes which are represented by each cluster $k$. For obtaining fine-grained class information, we employ a two layer CNN on the conv5 feature map, as shown in Fig . 2. A softmax function is used on the output of this layer for obtaining the final class probability. The detection and classification probabilities are multiplied to obtain the final detection score for each object-class. This architecture is shown in Fig. 2. Even though there are other challenges such as entailment, cover, equivalence etc. [26, 7] which are not correctly modelled with the softmax function, the Top-1 accuracy even on the ImageNet-5000 classification dataset is greater than 67% [40]. So, we believe these are not the bottlenecks for detecting a few thousand classes.

### 4.4. Label Assignment

Labels are assigned exactly the same way as fast-RCNN [12] for the $K$ super-classes on which detection is performed. Let $C$ be the total number of object-classes and let $k_i$ and $c_j$ denote the $i^{th}$ super-class and $j^{th}$ sub-class in $k_i$, then $k_i = \{c_1, c_2, ..., c_j\}$ and $\sum_{i=1}^{K} |k_i| = C$. For detecting super-class $k_i$, an RoI is assigned as positive for super-class $k_i$ if it has an intersection over union (IoU) greater than 0.5 with any of the ground truth boxes in $k_i$, otherwise it is marked as background (label for background class $K+1$ is set to one). For the classification branch (to get the final 3000 classes), only positive RoIs are used for training, i.e. only those which have an IoU greater than 0.5 with a ground truth bounding box. The number of labels for classification is $C$ instead of $K+1$ in detection.

### 4.5. Loss Function

For training the detector, we use online hard example mining (OHEM) [34] as done in [6] and smooth L1 loss for bounding box localization [12]. For fine-grained classification we only use a softmax loss function over $C$ object-classes for classifying the positive bounding boxes. Since the number of positive RoIs are typically small compared to the number of proposals, the loss from this branch is weighted by a factor of 0.05, so that these gradients do not dominate network training. This is important as we train RPN layers, R-FCN classification and localization layers, and fine-grained layers together in a multi-task fashion, so balancing the loss from each branch is important.

## 5. Experiments

In this section, we describe the implementation details of the proposed large-scale object detector and compare against some of the weakly supervised large-scale object detectors in terms of speed and accuracy.

| Dataset (ImageNet) | Images | Object Instances |
|---|---|---|
| Detection | 400,000 | 764,910 |
| CLS 194 | 87,577 | 100,724 |
| CLS 500 | 121,450 | 141,801 |
| CLS 1000 | 191,463 | 223,222 |
| CLS 2000 | 403,398 | 462,795 |
| CLS 3000 | 925,327 | 1,061,647 |

Table 1. The number of images and object instances in the ImageNet Detection and different versions of our ImageNet classification (CLS) training set.

### 5.1. Training Data

We train on the ImageNet classification dataset which contains bounding boxes for 3,130 classes. Each class contains at least 100 images in the training set. In the complete dataset, there are 1.2 million images. The detection test set (ILSVRC 2014) of ImageNet contains 194 classes out of the 3,130 classes present in the classification set. Therefore, we present our results on the 194 classes in our experiments (6 classes did not have bounding boxes in the ImageNet classification dataset which were present in the ImageNet detection test set). We also perform experiments on the PASCAL VOC 2007+2012 object detection dataset. The PASCAL dataset contains 20 object classes and 15,000 images for training. We evaluate our models on the VOC 2007 test set.

### 5.2. Implementation Details

For fast training and inference, we train on images of resolution (375x500), where the smaller side is at least 375 pixels and the larger side is a maximum of 500 pixels. Three anchor scales of (64,128,256) pixels are used. At each anchor scale, there are 3 aspect ratios of (1:2), (1:1) and (2:1) for the anchor boxes, hence there are a total of 9 anchors in RPN. We train for 7 epochs. A warm-up learning rate of 0.00002 is used for first 1000 iterations and then it is increased to 0.0002. The learning rate is dropped by a factor of 10 after 5.33 epochs. Training is performed on 2 Nvidia P6000 GPUs. End-to-end training on 3,130 classes takes 2 weeks. When increasing the number of classes beyond 194, we first select classes with the least number of samples (each class still has at least 100 samples) from the classification set. This is done for accelerating our ablation experiments on 500, 1000 and 2000 classes. Statistics of the detection set and classification set with different numbers of classes are shown in Table 1. For faster training and inference, we do not use deformable position-sensitive RoI pooling and only use bi-linear interpolation. For our analysis, we first train a region proposal network on 3,130 classes separately and extract proposals on the training and test set. Then, deformable R-FCN is trained like fast-RCNN with different numbers of clusters and classes. This is also done to accelerate training. It takes 2 days to train on 1000 classes after making these changes. During training, horizontal flipping is used as a data augmentation technique. Multi-scale inference is performed at two scales, (375,500) and (750,1000) and predictions of the two scales are combined using NMS. In all our experiments, a ResNet-50 backbone network is used. On the PASCAL VOC dataset we train under the same settings as [6].

| Method | LSDA [17] | SKT [36] | KDT [39] | Ours |
|---|---|---|---|---|
| mAP | 18.1 | 20.0 | 34.3 | 43.3 |

Table 2. Comparison of our decoupled R-FCN trained on classification data with bounding-box supervision vs. weakly-supervised methods that use a knowledge transfer approach to exploit information from detectors pre-trained on 100 classes on the ImageNet detection set.

### 5.3. Comparison with Weakly Supervised Detectors

First, to calibrate our results with existing methods and to highlight the improvement by training on classification data with bounding-box supervision, we compare our method with knowledge transfer based weakly supervised methods. Methods like LSDA [17] and Semantic Knowledge Transfer (SKT) [36] assume that detectors for 100 classes (trained on the ImageNet detection dataset) are available and use semantic similarity between weakly supervised classes and strongly supervised classes to leverage information learned from pre-trained detectors. They evaluate on the remaining 100 classes in the ImageNet detection set. Contemporary work [38] (KDT) also employs a knowledge transfer based approach, albeit with a modern Inception-ResNet-v2 based Faster-RCNN detector. Since these methods leverage classification data and also detection data for other classes, these can be considered as a very loose upper-bound on what a *true weakly-supervised detector*, which does not have any access to bounding boxes, would achieve. Our single scale ResNet-50 based model trained on 194 classes obtains an mAP of 40.5% [1] and after multi-scale testing (2 scales), we obtain an mAP of 43.3%. So, with bounding-box supervision on classification data, we obtain a 9% improvement. The performance of our detector can be further improved with a stronger backbone architecture, with deformable PSRoI layers etc. Note that we compare our method trained on 194 classes because a detector trained on larger number of classes is likely to perform worse if the new classes do not frequently occur in the test set.

We also provide some statistics on the number of images and object instances in the ImageNet detection and ImageNet classification set in Table 1. Weakly supervised methods like LSDA [17], SKT [36], KDT [39] use detectors trained on 400,000 instances present in 200,000 im-

[1] 2 classes did not have bounding box annotations in the ImageNet classification training set, so results are on 98 out of 100 classes


Figure 3. The mAP on the 194 classes in the ImageNet detection set is shown as we vary the number of clusters (super-classes). This is shown for 194 class and 1000 class detectors. We also plot the mAP for different number of classes for an objectness based detector.

| Clusters | 1 | 5 | 25 | 100 | 1000 |
|---|---|---|---|---|---|
| mAP | 36 | 36.7 | 37.1 | 37.3 | - |
| Time(ms) | 33 | 33 | 34 | 51 | 231 |

Table 3. The mAP scores for different number of clusters for the 1000 class detector and run-time(in milli-seconds)/image.

| Clusters | 20 | 50 | 100 | 200 | 1000 |
|---|---|---|---|---|---|
| mAP | 35.6 | 35.6 | 35.6 | 35.7 | 36.0 |
| Time(ms) | 1 | 1.5 | 1.8 | 2.6 | 10.1 |

Table 4. The mAP for different number of super-classes in NMS for the 1000 class objectness based detector and the NMS run-time (in milli-seconds).

ages from the detection dataset. This acts as a prior which is used as a basis for adapting the classification network. We compare with these methods as we could not find methods with superior performance which are trained in a completely weakly supervised way on the ImageNet classification/detection dataset and are evaluated on the ImageNet detection set.

### 5.4. Speed and Performance

In Table 3, we present the speed accuracy trade-off when increasing the number of clusters for the 1000 class detector. The 100 class clustering based detector is 66% slower than the objectness based detector. It was infeasible to train the original detector with 1000 classes, so we only present the run time for this detector. All the speed results are on a P6000 GPU. We also present results when we use different numbers of clusters during NMS. In this process, NMS is performed for a group of visually similar classes together, instead of each class separately. We use the same clustering based grouping of classes. The clusters used during NMS can be different from those which are used when grouping classes for R-FCN as this is only done for accelerating the post-processing step. We present the runtime for NMS (on GPU) for different numbers of clusters in Table 4. Note that 10 ms is 33% of the runtime of our detector, and this is only for 1000 classes. Therefore, performing NMS on visually similar classes is a simple way to speed up inference without taking a significant hit in average precision. As mentioned in the title, our 3000 class detector can be applied to more than 30 images per second (on a resolution of 375x500 pixels, minimum side 375, maximum side 500) on an Nvidia P6000 GPU.

## 6. Discussion

In order to better understand the behaviour of the proposed object detection system, we evaluate it while varying the number of clusters and classes under different training and testing dataset conditions. Lastly, we also conduct experiments with unseen classes during training to assess the generalization performance of the proposed detector beyond the training classes.

### 6.1. Impact of Number of Classes and Clusters

We present results as we increase the number of classes on the ImageNet detection test set which contains 194 classes in Fig. 3 (c). In this experiment, we only use one cluster, hence the position sensitive RoI filters only predict objectness and perform bounding-box regression. The drop in performance typically reduces as we increase the number of classes. For example, there is a drop of 2% as the number of classes is increased from 200 to 500, but from 1000 to 2000, the performance drop is only 0.3%. Even with 3,000 classes, we obtain an mAP of 34.9% which is 15% better than YOLO-9000 which obtains an mAP of 19.9%. Performance of YOLO-9000 drops to 16% when it is evaluated on classes which are not part of the detection set (these are majority of the classes which it detects). Therefore, we perform better by 19% on classes which are not part of the detection set compared to YOLO-9000. Although we detect 3,000 instead of 9,000 classes, our performance is more than 2 times better than YOLO-9000. Qualitative results for the R-FCN-3000 detector are also shown in Fig. 5.

To assess the effect of the number of super-classes on performance, we varied the number of super-classes and report the results. All results use a single-scale inference. Fig. 3 (a) reports mAP for training/testing on 194 classes from


Figure 4. The objectness, classification and final detection scores against various transformations such as combinations of scaling and translation are shown. These scores are generated by forward propagating an ideal bounding-box RoI (in green) and a transformed bounding-box RoI (in red) through the R-FCN (objectness) and classification branch of the network. The selectiveness of the detector in terms of objectness is clearly visible against the various transformations that lead to poor detection.

the ImageNet detection dataset and Fig. 3 (b) reports mAP for the same 194 classes while training with 1,000 object classes. The drop in performance is merely 1.7% for a detector with only one super-class as compared to 100 super-classes for 194 class training. More interestingly, as the number of training classes are increased to 1,000, the drop is only 1.3%, which is counter-intuitive because one would expect that using more super-classes would be helpful as we increase the number of object classes. In light of these observations, we can conclude that more crucial to R-FCN is learning an *objectness* measure instead of class-specific objectness.

### 6.2. Are Possition-Sensitive Filters Per Class Necessary?

To further verify our claim that detection can be modelled as a product of objectness and classification probability, we conduct more experiments on the PASCAL VOC dataset. We train a deformable R-FCN detector, as the baseline, with a ResNet-50 backbone that uses deformable position sensitive filters and obtains an mAP of 79.5%. After training a decoupled network which predicts objectness and performs classification on RoIs, we observe a similar pattern even on this dataset. At a 0.5 overlap, the performance only drops by 1.9% and at 0.7 by 1.5%, Table 6. This confirms that our proposed design changes to R-FCN are effective and only marginally deteriorate the mAP of the detector. We show a few visual examples of objectness and classification scores predicted by our class-agnostic detector in Fig 4.

Based on these results, we explore some other alternatives of R-FCN for estimating objectness. First, we just use RPN scores as the objectness measure and classify the proposals with our network (which is a linear classifier). Then, we add a bounding box regression step on the proposals, as they are already class agnostic. These two baselines are significantly worse than R-FCN. The mAP of only RPN is very poor at an overlap of 0.7. Although bounding-box regression provides a boost of 35% at 0.7 overlap, the performance is still 15% worse than R-FCN. Since RPN uses an overlap of 0.7 for assigning positives and 0.3 for assigning negatives, we decided to change these two thresholds to 0.5 and 0.4 respectively, like [23]. We train two versions of RPN, on conv4 and conv5 and present the results. These results show that performance with RPN also improves after changing the overlap criterion and with better features, so other objectness measures could also be an alternative for R-FCN. Results for these experiments are presented in Table 5.

| $Ov^{0.5}$ | $Ov^{0.7}$ | C4 | C5 | BBR | $mAP^{0.5}$ | $mAP^{0.7}$ |
|---|---|---|---|---|---|---|
| × | ✓ | ✓ | × | × | 47.3 | 12.7 |
| × | ✓ | ✓ | × | ✓ | 65.1 | 47.8 |
| ✓ | × | ✓ | × | × | 52.0 | 16.0 |
| ✓ | × | ✓ | × | ✓ | 66.8 | 49.7 |
| ✓ | × | × | ✓ | × | 70.6 | 44.1 |
| ✓ | × | × | ✓ | ✓ | 74.1 | 56.9 |

Table 5. Results for different versions of RPN scores used for objectness are reported. C4 and C5 denote if RPN is applied on Conv4 or Conv5 feature-map. $Ov^{0.5}$, $Ov^{0.7}$ denotes if the overlap for assigning positives in RPN is 0.5 or 0.7. BBR denotes if bounding box regression of deformable R-FCN is used or not.

| Method | $mAP^{0.5}$ | $mAP^{0.7}$ |
|---|---|---|
| D-R-FCN (decoupled) | 77.6 | 63.8 |
| D-R-FCN | 79.5 | 65.3 |

Table 6. Results of D-R-FCN (ResNet-50) and our decoupled version where the R-FCN classification branch only predicts objectness.

### 6.3. Generalization of Objectness on Unseen Classes

We evaluate the generalization performance of our objectness detector on a held out set of 20 classes. In this experimental setting, we train two objectness detectors - **OB** (objectness baseline), which includes the 20 object classes during training and **GO** (generalized objectness), which does not. For both the settings, the same classifier is used with different objectness detectors. OB and the classifier are trained on 194, 500, 1000, 2000 and 3130 classes and GO on 174, 480, 980, 1980 and 3110 classes. While going from 194 to 500 classes, the number of classes increase significantly but the number of samples do not (see Table

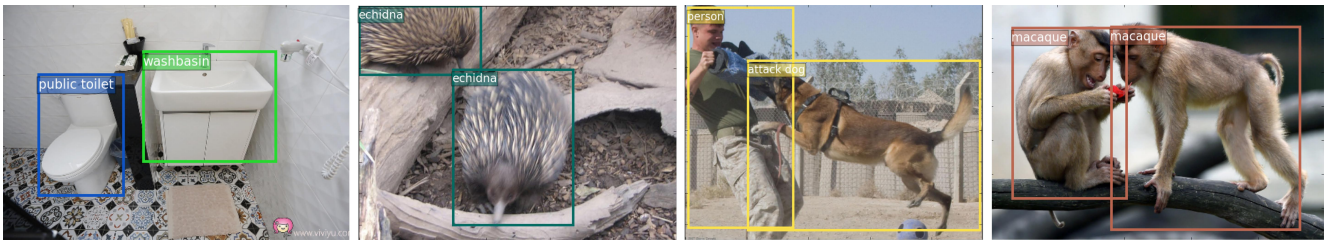


Figure 5. Detections for classes in the ImageNet3K dataset which are typically not found in common object detection datasets are shown.


Figure 6. Objectness scores on images containing unseen object-classes from the ImageNet detection dataset.

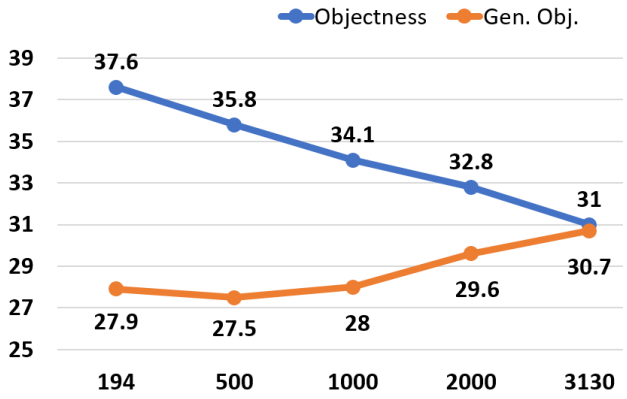


Figure 7. The mAP scores on a held out set of 20 classes for Generalized Objectness and Objectness baseline.

1); therefore, the mAP of OB drops by 1.8%. Since more samples help in improving the objectness measure for GO, the performance drop is only marginal (Fig 7). Increasing the number of classes to 1000 and 2000 improves the mAP of GO, implying that the improvement in objectness can overshadow the performance drop caused by increasing the number of classes. Fig 7 clearly shows that the initial gap of 9.7% in the performance drops to 0.3% as the number of classes increase. We also compared **OB** with **GO** when we remove all the 194 classes in ImageNet detection set and present the results in Table 7. Note that the performance drop is 3% even after removing 10% of the instances in the dataset (all of which belong to the classes in the test set). It strongly indicates that objectness learned on thousands of classes generalizes to novel unseen classes as well. A few

| Classes | Objectness | Generalized Objectness |
|---|---|---|
| 20 | 31 | 30.7 |
| 194 | 34.9 | 32 |

Table 7. mAP of Objectness and Generalized Objectness on held out classes in the ImageNet detection set.

qualitative results for such cases are shown in Fig. 6. We also show some qualitative results on a few COCO images in Fig 8. Note that we did not train on any detection images!

## 7. Conclusion and Future Directions

We demonstrate that it is possible to predict a universal objectness score by using only one set of filters for object vs. background detection. This objectness score can simply be multiplied with the classification score for detecting objects with only a marginal drop in performance. Finally, we show that the objectness learned generalizes to unseen classes and the performance increases with the number of training object classes. It bolsters the hypothesis of the universality of objectness.

This paper presents significant improvements for large-scale object detection but many questions still remain unanswered. Some promising research questions are - How can we accelerate the classification stage of R-FCN-3000 for detecting 100,000 classes? A typical image contains a limited number object categories - how to use this prior to accelerate inference? What changes are needed in this architecture if we also need to detect objects and their parts? Since it is expensive to label each object instance with all valid classes in every image, can we learn robust object detectors if some objects are not labelled in the dataset?


Figure 8. Qualitative results on COCO. We miss some objects, but the figure depicts the diversity of classes detected by R-FCN-3000!

## References


[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2189–2202, 2012. 2

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011. 2

[3] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3241–3248. IEEE, 2010. 2

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 3

[5] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. 1, 2, 3

[6] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *arXiv preprint arXiv:1703.06211*, 2017. 1, 2, 3, 4, 5

[7] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *European Conference on Computer Vision*, pages 48–64. Springer, 2014. 4

[8] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool. Weakly supervised cascaded convolutional networks. *arXiv preprint arXiv:1611.08258*, 2016. 3

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1

[10] C. Feichtenhofer, A. Pinz, and A. Zisserman. Detect to track and track to detect. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3038–3046, 2017. 1

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010. 2

[12] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1, 4

[13] 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*, pages 580–587, 2014. 1, 2

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017. 1

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 3

[16] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012. 1

[17] J. Hoffman, S. Guadarrama, E. S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko. Lsda: Large scale detection through adaptation. In *Advances in Neural Information Processing Systems*, pages 3536–3544, 2014. 1, 2, 5

[18] J. Hoffman, D. Pathak, E. Tzeng, J. Long, S. Guadarrama, T. Darrell, and K. Saenko. Large scale visual recognition through adaptation using joint representation and multiple instance learning. *The Journal of Machine Learning Research*, 17(1):4954–4984, 2016. 2

[19] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv preprint arXiv:1611.10012*, 2016. 1

[20] 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

[21] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2169–2178. IEEE, 2006. 2

[22] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017. 1, 2, 3, 7

[24] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 3

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1

[26] B. MacCartney and C. D. Manning. An extended model of natural logic. In *Proceedings of the eighth international conference on computational semantics*, pages 140–156. Association for Computational Linguistics, 2009. 4

[27] D. Novotny, D. Larlus, and A. Vedaldi. I have seen enough: Transferring parts across categories. 2016. 2

[28] P. Ott and M. Everingham. Shared parts for deformable part-based models. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1513–1520. IEEE, 2011. 2

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016. 1

[30] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016. 1, 2

[31] 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

[32] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1481–1488. IEEE, 2011. 2

[33] 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

[34] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–769, 2016. 4

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[36] Y. Tang, J. Wang, B. Gao, E. Dellandréa, R. Gaizauskas, and L. Chen. Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2119–2128, 2016. 5

[37] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2004. 2

[38] J. Uijlings, S. Popov, and V. Ferrari. Revisiting knowledge transfer for training object class detectors. *arXiv preprint arXiv:1708.06128*, 2017. 5

[39] 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. 2, 5

[40] H. X. X. J. S. Y. J. F. Yunpeng Chen, Jianan Li. Dual path networks. *arXiv preprint arXiv:1707.01629*, 2017. 4

[41] H. Zhang, Z. Kyaw, J. Yu, and S.-F. Chang. Ppr-fcn: Weakly supervised visual relation detection via parallel pairwise r-fcn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4233–4241, 2017. 1