Enriched Feature Guided Refinement Network for Object Detection

Jing Nie[1*†], Rao Muhammad Anwer[2*], Hisham Cholakkal[2], Fahad Shahbaz Khan[2]
Yanwei Pang[1‡], Ling Shao[2]
[1]School of Electrical and Information Engineering, Tianjin University
[2]Inception Institute of Artificial Intelligence (IIAI), UAE
[1]{jingnie,pyw}@tju.edu.cn, [2]{rao.anwer, hisham.cholakkal, fahad.khan, ling.shao}@inceptioniai.org

Abstract

We propose a single-stage detection framework that jointly tackles the problem of multi-scale object detection and class imbalance. Rather than designing deeper networks, we introduce a simple yet effective feature enrichment scheme to produce multi-scale contextual features. We further introduce a cascaded refinement scheme which first instills multi-scale contextual features into the prediction layers of the single-stage detector in order to enrich their discriminative power for multi-scale detection. Second, the cascaded refinement scheme counters the class imbalance problem by refining the anchors and enriched features to improve classification and regression. Experiments are performed on two benchmarks: PASCAL VOC and MS COCO. For a 320×320 input on the MS COCO test-dev, our detector achieves state-of-the-art single-stage detection accuracy with a COCO AP of 33.2 in the case of single-scale inference, while operating at 21 milliseconds on a Titan XP GPU. For a 512×512 input on the MS COCO test-dev, our approach obtains an absolute gain of 1.6% in terms of COCO AP, compared to the best reported single-stage results [5]. Source code and models are available at: https://github.com/Ranchentx/EFGRNet.

1. Introduction

Object detection is an active research problem with numerous real-world applications. Modern object detection methods based on convolutional neural networks (CNNs) can be divided into two categories: (1) the two-stage methods [33, 23], and (2) the single-stage approaches [27, 32]. Two-stage methods first generate object proposals and then these proposals are classified and regressed. Single-stage methods directly localize objects by regular and dense sampling grids on the input image. Generally, two-stage object detectors have the advantage of being more accurate compared to single-stage methods. Single-stage methods, on the other hand, have time computational efficiency but compromise on performance compared to the two-stage detectors [19]. In this work, we investigate the problem of generic object detection in a single-stage framework.

In recent years, a variety of single-stage object detection methods have been introduced [27, 32, 41, 24]. Among existing single-stage object detectors, the single shot multibox detector (SSD) [27] has recently gained popularity due to its combined advantage of improved detection performance and high speed. The standard SSD framework utilizes a base network (*e.g.*, VGG) and adds a series of convolutional layers at the end of the truncated base network. Both the added convolutional layers and some of the earlier base network layers, of varying resolutions, are employed to conduct independent predictions. In the standard SSD, each prediction layer focuses on predicting objects of a specific scale. It adopts a pyramidal feature hierarchy in which shallow or former layers target small objects whereas deep or later layers aim at detecting large objects. While achieving high computational efficiency, SSD still lags behind most modern two-stage detectors in terms of detection accuracy.

In this work, we distinguish two key obstacles impeding the standard SSD detector from achieving state-of-the-art accuracy while maintaining its hallmark speed. First, the standard SSD struggles to handle large *scale variations* [1]. This is likely due to fixed contextual information in the SSD prediction layers. Existing approaches tackle this issue by *e.g.*, adding contextual information along with deeper backbone model [13] and feature pyramid representations [41, 24, 4, 30]. Most approaches [41, 24, 4] adopt a top-down pyramid representation where low-resolution feature maps of deep layers are first up-sampled and then combined with high-resolution feature maps of shallow layers to inject high-level semantic information. While such a feature pyramid representation helps tackle large scale variation, the performance is still far from satisfactory.

The second key issue is the foreground-background *class*

*Equal contribution
†Work done at IIAI during Jing's internship
‡Corresponding author

imbalance problem encountered during the training of the SSD detector. Existing solution [24, 41] to this problem include, *e.g*., training on a sparse set of hard examples while down-weighting well-classified examples and integrating a two-step anchor refinement strategy to reduce the search space for the classifier by removing negative anchors. Though of success, the work of [41] employs a top-down feature pyramid representation and only refines the anchors due to which the features does not align well with the refined anchors. In this work, we look into an alternative way to jointly tackle the problem of multi-scale object detection *and* class imbalance in order to improve the accuracy of SSD without sacrificing its characteristic speed.

Contributions: We re-visit the standard SSD framework to jointly tackle the problem of multi-scale object detection and class imbalance. First, we introduce a feature enrichment scheme to improve the discriminative power of prediction layers in the standard SSD. Instead of deepening the backbone model, our feature enrichment scheme is designed to produce multi-scale contextual features. We further introduce a cascaded refinement scheme with dual objectives. First, it instills the multi-scale contextual features into the standard SSD prediction layers in a bottom-up pyramidal feature hierarchy. The resulting enriched features are more robust to scale variations. Second, it addresses the class imbalance problem by utilizing the enriched features to perform class-agnostic classification and bounding-box regression for accurate localization. Afterwards, the initial box regression and the binary classification are further utilized to refine the associated enriched features for obtaining final classification scores and bounding-box regression.

We perform comprehensive experiments on the two challenging benchmarks: PASCAL VOC 2007 [12] and MS COCO [25]. Our detector achieives superior results compared to existing single-stage methods on both datasets. For 512×512 on MS COCO test set, our detector outperforms RefineDet [41] with the same backbone (VGG) by 4.5% in terms of COCO AP, while operating at inference time of 39 milliseconds (ms) on a Titan XP GPU.

2. Related Work

Object detection [33, 27, 7, 28, 35] is a challenging and active computer vision problem. Convolutional neural networks (CNNs) [36, 18, 9, 38, 29, 37] based object detectors [14, 15, 32, 17, 33, 8, 27, 2] have shown outstanding results in recent years. This work focuses on single-stage object detectors [32, 27] that are generally faster compared to their two-stage counterparts. Among existing single-stage approaches, SSD [27] has shown to provide excellent performance while operating at real-time. It uses a multi-scale representation that detect objects in a pyramidal hierarchical structure. In such a hierarchy, shallow layers contribute to predict smaller objects while deeper layers helps in detecting larger objects. We base our approach on standard SSD due to its superior accuracy and high speed.

Single-stage detectors, such as SSD, struggle to accurately detect objects with significant scale variations. Further, SSD detector also suffers from the class imbalance problem. Existing methods in literature [13, 3, 6, 42] tackle the first issue by exploiting contextual information, better feature extraction or top-down feature pyramid representation. A popular strategy is to build a top-down feature pyramid representation to inject the high-level semantic information from the deeper layers to shallow layers with limited information [24, 4]. The work of [30] proposes an alternative way of constructing feature pyramids based on image pyramids termed as featurized image pyramids. In contrast, our approach does not require any featurized image pyramids or top-down pyramid construction and instead focuses on capturing multi-scale contextual information. Moreover, our approach comprises a dedicated module to address the class imbalance problem. The work of [6] investigates the integration of context via a multi-deformable head and uses box regression (position and scale offsets) for refining features. Instead, we improve the discriminative power of standard SSD prediction layers in two ways. First, we introduce a feature enrichment scheme inspired from the multi-branch ResNeXT architecture [39, 31] that produces multi-scale contextual features to enrich the standard SSD features with contextual information. Second, we introduce a cascaded refinement scheme in which both the box regression and the binary classification are utilized to refine the features. The binary classification (object-category prediction) is used to generate an objectness map that highlights probable object locations. During feature refinement, only the position offsets are utilized for the alignment of features with the refined anchors while scale offsets are ignored.

To address the issue of class imbalance during the training stage, RetinaNet [24] introduces focal loss to down-weight the contribution of easy samples. RefineDet [41] proposes a two-step anchor refinement module to reduce the search space for the classifier by removing several negative anchors. Additionally, the anchor refinement module coarsely adjusts the location of anchors. Different to [41], our cascaded refinement scheme utilizes enriched features by first instilling the multi-scale contextual information into the standard SSD prediction layers. Further, the cascaded refinement removes several negative anchors and not only refines anchor locations, but also the features.

3. Method

Our detection framework consists of three components: the standard SSD layers, feature enrichment (FE) scheme and cascaded refinement scheme. Our FE scheme (sec. 3.1) contains a multi-scale contextual feature module (MSCF) to address scale variations. The FE scheme produces multi-

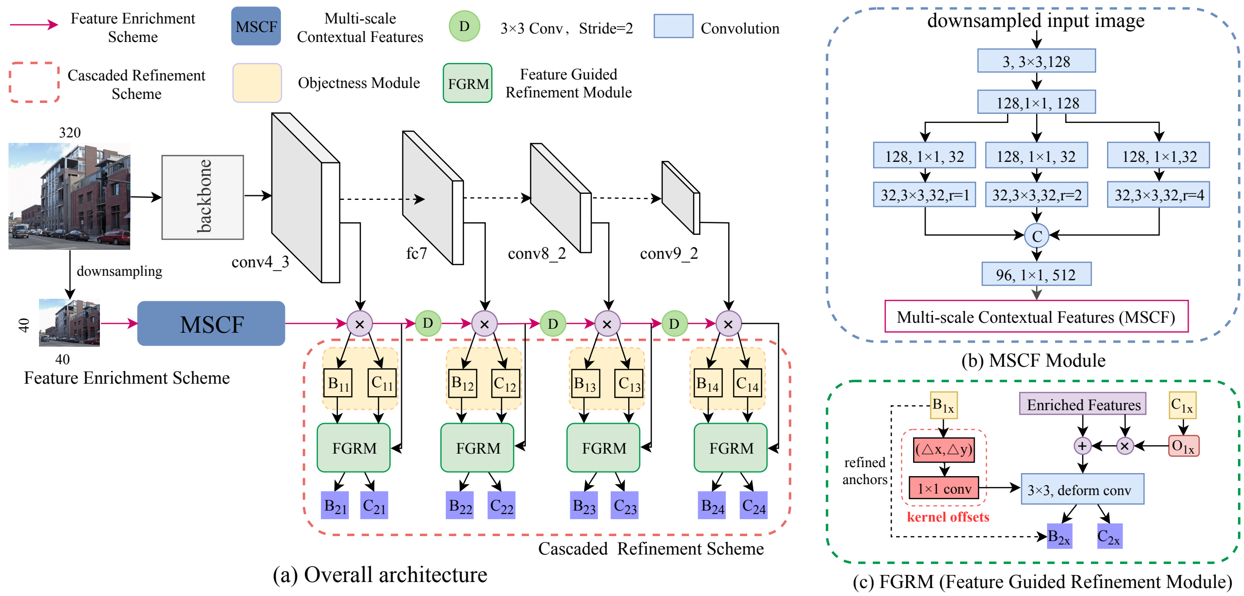

Figure 1. (a) Overall architecture of our single-stage detection approach using VGG backbone. It consists of three components: standard SSD layers, feature enrichment scheme and cascaded refinement scheme. The feature enrichment scheme is designed to extract multi-scale contextual features using a MSCF module shown in (b). These contextual features are then instilled in SSD prediction layer ($conv4\_3$) and propagated further, using a bottom-up feature hierarchy, in the objectness module of cascaded refinement scheme. The objectness module also performs class-agnostic classification (C_{1x}) and initial regression (B_{1x}). Further, the class-agnostic classification provides an objectness map later used in the FGRM module, shown in (c), of our cascaded refinement scheme. The FGRM module generates final refined features used to predict final classification (C_{2x}) and bounding-box regression (B_{2x}).

scale contextual features to improve the discriminative power of the standard SSD prediction layers. The cascaded refinement scheme (sec. 3.2) utilizes both multi-scale contextual and standard SSD features and tackles the class imbalance problem. The cascaded refinement scheme refines both anchors and the features, by performing box regression and classification in two cascaded modules, namely objectness module (OM) and feature guided refinement module (FGRM), respectively. The objectness module (OM) performs a binary classification of object vs. background along with an initial box regression. The FGRM module then refines the features and anchor locations to predict the final multi-class classification and bounding box localization.

Fig. 1 illustrates the overall architecture of our framework when using VGG as the backbone network, as in [27]. Following [41], we only utilize four prediction layers ($conv4\_3$, $fc7$, $conv8\_2$, $conv9\_2$) for detection, instead of six layers as used in original SSD. Increasing the prediction layers beyond four does not improve our performance.

3.1. Feature Enrichment Scheme

In the standard SSD framework, the feature extraction from a deep convolutional network backbone, *e.g.* either VGG16 or ResNet, is performed by a repeated process of convolutional and max-pooling operations. Despite preserving a certain degree of semantic information, they still lose the low-level feature information that is likely to aid in discriminating object regions from the background regions. Moreover, the constant receptive field at each prediction layer captures only a fixed contextual information. In this work, we introduce a feature enrichment (FE) scheme to capture multi-scale contextual information. We start by downsampling an input image with a simple pooling operation to match its size with that of first SSD prediction layer. Then, the downsampled image is passed through our Multi-Scale Contextual Feature (MSCF) module.

Multi-scale Contextual Features Module: The proposed MSCF module is highlighted with dotted blue-box in Fig. 1(b). It is a simple module comprising several convolution operations and produces multi-scale contextual features. The structure of MSCF module is inspired from the multi-branch ResNeXT architecture [39, 31] and is an operation of splitting, transformation and aggregation strategy. The MSCF module takes a downsampled image as input, and outputs contextually enhanced multi-scale features. The downsampled image is first passed through two consecutive convolutional layers of size 3×3 and 1×1, resulting in an initial feature projection. Then, these feature projections are sliced into three low-dimensional branches through a 1×1 convolutional layer. To capture the multi-scale contextual information, we employ three dilated convolutions [40] with dilation rates set to 1, 2 and 4, respectively for different branches. The dilated convolutional operation transformed the initial feature projection into a contextually enhanced feature set. Then, these transformed features are aggregated through a concatenation operation and pass to a 1×1 convolution operation. The output of MSCF is used in the objectness module (OM) of our cascaded refinement scheme.

3.2. Cascaded Refinement Scheme

Our refinement scheme consists of two cascaded modules: objectness module and feature guided refinement module (FGRM), as shown in Fig.1(a). The objectness module enriches the SSD features with multi-scale contextual information and identifies possible object locations (objectness). Enriching the features with multi-scale contextual information improves the performance on small objects whereas the objectness predictions are used in the FGRM to address the class imbalance problem.

Objectness Module: The objectness module first enriches the SSD features by instilling the multi-scale contextual features from the MCSF module at $conv4\_3$, through element-wise multiplication operation. Then, we introduce a bottom-up pyramidal feature hierarchy to propagate the enriched features to the subsequent SSD prediction layers, as shown in Fig. 1 (a). The objectness module uses a 3×3 convolution operation with stride two (D), and projects the features from previous layer to match with the spatial resolution and number of channels at the current layer. Enriched features are then obtained by performing an element-wise multiplication between projected features and SSD features at each prediction layer. Finally, the enriched features are used to perform a binary classification (C_{1x}) and an initial box regression (B_{1x}) at each prediction layer x. Here x = 1, 2, 3, and 4 corresponds to four prediction layers.

Fig. 2 shows example images from PASCAL VOC dataset and the corresponding $fc7$ feature maps from the standard SSD (second column), multi-scale contextual features after D (third column) and the enriched features (fourth column). The examples show that enriching standard SSD features with multi-scale contextual information helps to pay more attention to regions containing object instances. The binary classification C_{1x} output from the objectness module is further used in the FGRM to reduce the class imbalance between positive and negative anchors by filtering-out a large number of negative anchors. In addition, C_{1x} output is used to generate an attention map to guide the enriched features to pay more attention to the objects while suppressing the background. The box regression B_{1x} outputs are also used in the FGRM to refine both the features and the anchors locations.

Feature Guided Refinement Module: Our FGRM consists of three steps: objectness map generation, kernel offsets extraction and local contextual information extraction (see Fig.1(c)). Next, we describe these three steps.

Objectness Map Generation: The binary classifier (C_{1x}) output in the objectness module predicts each anchor as object/background, which is used to generate an objectness map O_{1x} that highlights probable object locations. We perform a max-pooling operation along channel axis on the object-category prediction of all anchors at a given spatial location, followed by a sigmoid activation. As a result, a spatial objectness map O_{1x} is produced which is used to improve the enriched features F_{in} obtained from the objectness module by,

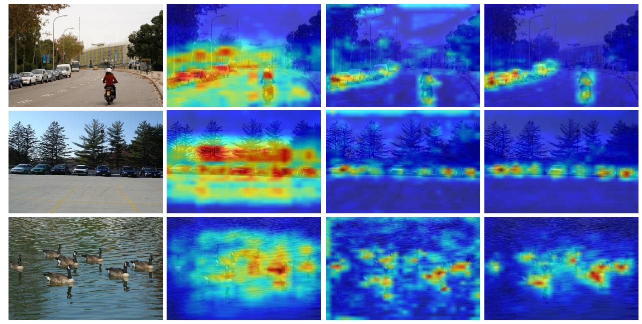

Figure 2. Example images from PASCAL VOC dataset and the corresponding $fc7$ feature maps from the standard SSD (second column), multi-scale contextual features (third column) and the enriched features (fourth column). The examples show that the enriched features obtained as a result of instilling multi-scale contextual features into the standard SSD features helps in better discriminating object regions from the background.

$$F_m = F_{in} \odot O_{1x} + F_{in}, \tag{1}$$

where $\odot$ is element-wise multiplication and F_m is enriched feature after improvement.

Kernel Offsets Extraction: The box regressions at objectness and FGRM modules predict four outputs: Δx, Δy, Δh, and Δw. The former two (Δx , Δy) correspond to the spatial offsets and the latter two (Δw , Δh) correspond to scale offsets in spatial dimensions. Here, we use the spatial offsets (Δx , Δy) from the objectness module to guide the feature refinement in FGRM by estimating the kernel offsets Δp_k as,

$$\Delta p_k = f^{1\times 1}({B_{1x}}^{\Delta x, \Delta y}), \tag{2}$$

where, $f^{1\times 1}$ denotes the convolutional layer whose kernel size is 1×1 and ${B_{1x}}^{\Delta x, \Delta y}$ denotes the spatial offsets (Δx , Δy) predicted by the objectness module. Finally, the kernel offsets are used as an input to the deformable convolution [11] in order to guide the feature sampling and align with the refined anchors.

Local Contextual Information: To further enhance the contextual information at a given spatial location, we utilize dilated convolutions [40] in our FGRM. We set the dilation rates as 5, 4, 3, and 2 at SSD prediction layers having stride 8, 16, 32, 64, respectively.

In summary, the final refined features F_{rf}, obtained after all operations within the FGRM, is formulated as:

$$F_{rf}(p_0) = \sum\nolimits_{p_k \epsilon R} w(p_k) \cdot F_m(p_0 + p_k \cdot d + \Delta p_k) \quad (3)$$

where p_0 denotes each spatial location in the final refined feature map F_{rf} and d is the dilation rate. R is a regular grid to sample the input features (*i.e.* If the kernel is 3×3, dilation 1, $R = (-1,-1), (-1,0), ..., (0,1), (1,1))$. The final refined feature F_{rf} is the summation of the sampling values weighted by w. Δp_k is the kernel offset to augment the regular sampling grid enhancing the capability of CNN to model geometric transformations. Generally, in deformble convolution, the offsets are obtained by applying a convolutional layer over the same input feature map. In our FGRM, the offsets are generated by the first box regressions from the objectness module. To obtain the refined anchor locations, we follow a similar strategy as in [41]. We utilize the offsets (B_{1x}) predicted from the objectness module to refine the original anchor locations. Consequently, the refined locations and refined feature F_{rf} are used to perform multi-class classification (C_{2x}) and box regression (B_{2x}).

4. Experiments

4.1. Datasets and Evaluation Metrics

Datasets: We perform experiments on two benchmarks: PASCAL VOC 2007 [12] and MS COCO [25]. The PASCAL VOC 2007 dataset consists of 20 different object categories. We perform training on the combined set of VOC 2007 trainval with 5k images and VOC 2012 trainval with 11k images where the evaluation is performed on the VOC 2007 test set with 5k images. MS COCO is a more challenging dataset with 80 object categories and is divided into 80k training, 40k validation and 20k test-dev images. The training is performed on the trainval35k set and the evaluation is done on minival set and test-dev2015.

Evaluation Metrics: We follow standard protocols for evaluation originally defined with both datasets. For Pascal VOC, the results are reported, in terms of mean Average Precision (mAP), which measure detection accuracy at an intersection-over-union(IOU) overlap exceeding a threshold of 0.5. The evaluation metric for MS COCO is different from Pascal VOC, where the overall performance, average precision (AP), is measured by averaging over multiple IOU thresholds, varying from 0.5 to 0.95.

4.2. Implementation Details

Our framework employs VGG-16, pretrained on ImageNet [34] as backbone architecture. We use the same setting for model initialization and optimization for both datasets. The warming up strategy is adopted for setting the initial learning rate from 10^{-6} to 4×10^{-3} for the first 5 epochs. Then, we gradually decrease the learning rate by a factor of 10, for PASCAL VOC 2007 dataset at 150 and 200 epoch, and for MS COCO dataset at 90, 120 and 140 epoch, respectively. For both datasets, the weight decay is set to 0.0005, the momentum to 0.9 and the batch size is 32. In our experiments, a total number of 250 and 160 epoch are performed for PASCAL VOC 2007 and MS COCO dataset, respectively. In addition to VGG-16, we also perform experiments using the stronger ResNet-101 backbone on MS COCO dataset. For ResNet-101, two extra convolution layers (*i.e.* $res6\_1$, $res6\_2$) are added at the end of the truncated ResNet-101 backbone. We utilize four prediction layers ($res3$, $res4$, $res5$, $res6\_2$) for detection.

Method	Backbone	Input Size	mAP
Two-Stage Detectors:			
Faster RCNN [18]	ResNet101	1000×600	76.4
R-FCN [10]	ResNet101	1000×600	80.5
CoupleNet[45]	ResNet101	1000×600	82.7
Single-Stage Detectors:			
SSD300 [27]	VGG16	300×300	77.2
RON320++ [21]	VGG16	320×320	76.6
DSSD321 [13]	ResNet101	321×321	78.6
RefineDet320 [41]	VGG16	320×320	80.0
DES300 [42]	VGG16	300×300	79.7
DFPR300 [20]	VGG16	300×300	79.6
RFBNet300 [26]	VGG16	300×300	80.5
EFIPNet[30]	VGG16	300×300	80.4
EFGRNet(Ours)	VGG16	320×320	**81.4**
SSD512 [27]	VGG16	512×512	79.5
DSSD513 [13]	ResNet101	513×513	81.5
DES512 [42]	VGG16	512×512	81.7
RefineDet512 [41]	VGG16	512×512	81.8
DFPR512 [20]	VGG16	512×512	81.1
EFIPNet512 [30]	VGG16	512×512	81.8
RFBNet512 [26]	VGG16	512×512	82.1
EFGRNet(Ours)	VGG16	512×512	**82.7**

Table 1. State-of-the-art comparison of our method with existing detectors on PASCAL VOC 2007 test set. Our detector outperforms existing single-stage methods for both 300×300 and 512×512 inputs.

4.3. State-of-the-art Comparison

PASCAL VOC 2007: Here, we perform a comparison of our approach with state-of-the-art single and two-stage object detection methods in literature. Tab. 1 shows the results on PASCAL VOC 2007 test set. Note that most existing two-stage methods rely on a larger input image size (typically 1000×800) for improved performance. Among existing two-stage object detectors, CoupleNet [45] obtains a detection score of 82.7 mAP. In case of single-stage methods, we perform a comparison with two input variants: 300×300 and $\sim 500 \times 500$ range. With an input image size of 300×300, the baseline SSD method obtains a detection accuracy of 77.2 mAP. Our detector provides a significant absolute gain of 4.1% in terms of mAP, over the baseline

Methods	Backbone	Input size	Time	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Two-Stage Detector									
Faster RCNN [33]	VGG16	1000×600	147ms	21.9	42.7	-	-	-	-
CoupleNet [45]	ResNet101	1000×600	121ms	34.4	54.8	37.2	13.4	38.1	50.8
Mask-RCNN[16]	ResNetXt-101-FPN	1280×800	210ms	39.8	62.3	43.4	22.1	43.2	51.2
Single-Stage Detector									
SSD [27]	VGG16	300×300	20ms$*$	25.1	43.1	25.8	6.6	25.9	41.4
DSSD [13]	ResNet101	321×321	-	28.0	46.1	29.2	7.4	28.1	47.6
RefineDet [41]	VGG16	320×320	20ms$*$	29.4	49.2	31.3	10.0	32.0	44.4
DES [42]	VGG16	300×300	-	28.3	47.3	29.4	8.5	29.9	45.2
RFBNet [26]	VGG16	300×300	15ms	30.3	49.3	31.8	11.8	31.9	45.9
EFIPNet [30]	VGG16	300×300	14ms	30.0	48.8	31.7	10.9	32.8	46.3
EFGRNet (Ours)	VGG16	320×320	21ms$*$	**33.2**	**53.4**	**35.4**	**13.4**	**37.1**	**47.9**
SSD [27]	VGG16	512×512	45ms	28.8	48.5	30.3	10.9	31.8	43.5
DSSD [13]	ResNet101	513×513	182ms	33.2	53.3	35.2	13.0	35.4	51.1
RefineDet [41]	VGG16	512×512	39ms$*$	33.0	54.5	35.5	16.3	36.3	44.3
DES [42]	VGG16	512×512	-	32.8	53.2	34.6	13.9	36.0	47.6
DRN[6]	VGG16	512×512	-	34.3	57.1	36.4	17.9	38.1	44.8
RFBNet-E [26]	VGG16	512×512	33ms	34.4	55.7	36.4	17.6	37.0	47.6
EFIPNet [30]	VGG16	512×512	29ms	34.6	55.8	36.8	18.3	38.2	47.1
RetinaNet [24]	ResNet101-FPN	500×832	90ms	34.4	53.1	36.8	14.7	38.5	49.1
RefineDet [41]	ResNet101	512×512	-	36.4	57.5	39.5	16.6	39.9	51.4
TripleNet [4]	ResNet101	512×512	-	37.4	**59.3**	39.6	18.5	39.0	52.7
RetinaNet+AP-Loss [5]	ResNet-101-FPN	512×512	90ms	37.4	58.6	40.5	17.3	40.8	51.9
ExtremeNet [43]	Hourglass104	511×511	348ms$*$	40.2	55.5	**43.2**	**20.4**	43.2	53.1
CornerNet [22]	Hourglass104	511×511	227ms$*$	**40.5**	56.5	43.1	19.4	42.7	53.9
EFGRNet (Ours)	VGG16	512×512	38.9ms$*$	37.5	58.8	40.4	19.7	41.6	49.4
EFGRNet (Ours)	ResNet101	512×512	46ms$*$	39.0	58.8	42.3	17.8	**43.6**	**54.5**
RefineDet (MS) [41]	ResNet101	512×512	-	41.8	62.9	45.7	25.6	45.1	54.1
CornerNet (MS) [22]	Hourglass104	511×511	-	42.1	57.8	45.3	20.8	44.8	56.7
ExtremeNet (MS)[43]	Hourglass104	511×511	-	**43.7**	60.5	47.0	24.1	46.9	**57.6**
FSAF (MS)[44]	ResNet101	800×1333	-	42.8	63.1	46.5	**27.8**	45.5	53.2
EFGRNet (Ours)(MS)	ResNet101	512×512	-	43.4	**63.8**	**48.2**	26.8	**47.2**	55.9

$*$: Tested in Pytorch041 with a single NVIDIA Titan X PASCAL and the batchsize 1 for fair comparison

Table 2. State-of-the-art comparison on MS COCO test-dev2015. For 300×300 input, our approach outperforms existing single-stage methods without a significant reduction in speed. For 512×512 input, CornerNet provides the best overall detection accuracy. However, our detector provides a 5-fold speedup over CornerNet, while being superior in accuracy at IoU threshold of 0.5. We also compare the multi-scale inference (MS) variant of our approach with recent methods (numbers reported from respective papers).

SSD. With a input image size of 512×512, RefineDet [41] and RFBNet [26] achieve accuracies of 81.8 and 82.1 in terms of mAP, respectively. Our approach with the same input size and backbone outperforms RFBNet [26] with an accuracy of 82.7 mAP on this dataset. Fig.3 shows results on PASCAL VOC 2007 test set with our detector.

MS COCO: Tab. 2 shows the state-of-the-art comparison. With an input size of 320×320, the baseline SSD achieves an overall detection score of 25.1. Our approach obtains a significant improvement of 8.1% in terms of overall detection score, over the baseline SSD when using same backbone. Notably, large gains of 11.2% and 6.8% are achieved on medium and small sized objects, over the baseline SSD. Among existing single-stage methods, RFBNet [26] and EFIPNet [30] provide overall detection accuracies of 30.3 and 30.0, respectively with 300×300 input. Our approach

Figure 3. Qualitative results of our approach on VOC 2007 testset (corresponding to 82.7 mAP). Each color belongs to an object class.

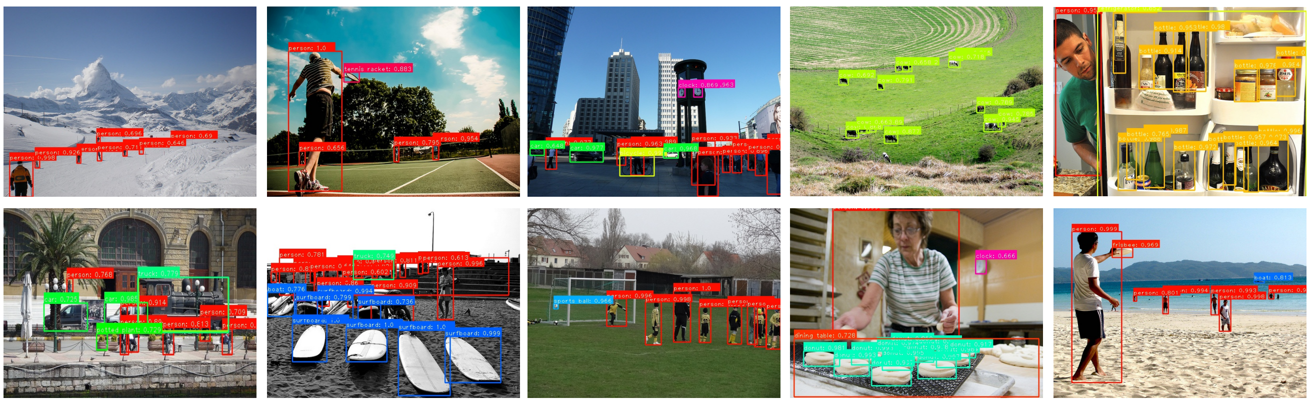

Figure 4. Qualitative detection results of our detector on the MS COCO 2015 test-dev. The detection results corresponds to 37.5 AP.

sets a new state-of-the-art with an overall detection score of 33.2 using approximately similar input scale (320 × 320) and the same backbone network.

With an input size of 512 × 512 and VGG backbone, the baseline SSD achieves an overall detection score of 28.8. Our approach significantly outperforms the baseline SSD with an overall detection accuracy of 37.5 with the same input size and backbone. Our detector provides a further improvement in performance when using the more powerful ResNet-101 backbone with an overall detection score of 39.0. When using a 512 × 512 input, CornerNet [22] achieves the best overall detection accuracy with AP score of 40.6. Our method provides a 5-fold speedup over CornerNet [22], while being superior in accuracy at IoU threshold of 0.5. Both ExtremeNet [43] and CornerNet [22] are superior on higher IoU (reflected in total AP), likely due to computationally expensive multi-scale Hourglass architecture. Fig.4 shows detection results on coco test-dev.

We conduct an error analysis on MS COCO using the analysis tool provided by [25]. Fig.5 shows the comparison for RefineDet [41] (on the left) and our approach (on the right) with 320×320 input across all COCO categories. The overall performance of RefineDet at IoU=.75 is .309 and perfect localization is likely to boost the AP to .583. Similarly, eliminating background false positives would increase

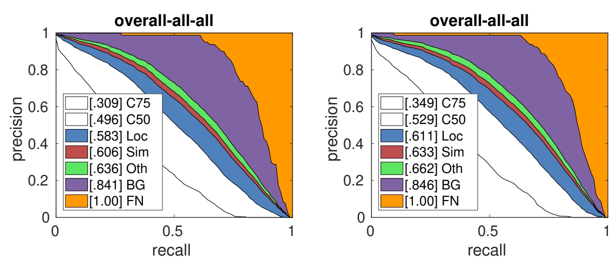

Figure 5. Error analysis between the RefineDet [41] (on the left) and our detector (on the right) for all 80 COCO object categories. For fair comparison, the analysis is performed using the same backbone (VGG) and input size (320 × 320) for both approaches. Here, the plots in each sub-image presents a series of precision recall curves. These curves are computed using different settings [25]. Additionally, the AUC curve is presented in the legend.

the result to .841 AP. The overall performance of our detector at IoU=.75 is .349 and perfect localization is likely to increase the AP to .611. Likewise, eliminating background false positives would increase the performance to .846 AP. Our approach shows superior performance over RefineDet.

4.4. Baseline Comparison

We first evaluate the impact of our feature enrichment (sec. 3.1) and cascaded refinement (sec. 3.2) schemes by

Methods	VOC 2007	MS COCO			
	mAP	AP	AP_s	AP_m	AP_l
Baseline SSD	77.2	24.4	6.8	27.5	40.9
SSD + FE scheme 3.1	79.4	29.1	9.4	34.1	45.3
SSD + Cascaded refinement 3.2	81.0	31.1	13.0	34.5	47.4
EFGRNet (Ours)	**81.4**	**33.0**	**14.5**	**37.4**	**49.5**

Table 3. Comparison of integrating our proposed feature enrichment and cascaded refinement schemes into the baseline SSD framework on the PASCAL VOC 2007 and MS COCO minival set datasets. For all the experiments, the backbone is VGG16 and the input is 320×320. Our final approach provides a large gain in performance over the baseline SSD on both datasets.

integrating them in the baseline SSD. Tab. 3 shows the results on both PASCAL VOC 2007 and MS COCO datasets. For a fair comparison, we utilize the same settings for all the experiments. On the PASCAL VOC 2007 dataset, the baseline SSD achieves 77.2 mAP. The introduction of feature enrichment scheme leads to an improvement of 2.2% in mAP over the baseline SSD. Note that the feature enrichment scheme is integrated into the baseline SSD via objectness module. The detection performance is improved from 77.2 to 81.0 mAP by the integration of cascaded refinement scheme. For a fair evaluation of our cascaded refinement, we exclude both the feature enrichment and bottom-up feature hierarchy of objectness module. Both feature enrichment and cascaded refinement schemes provide a combined gain of 4.2% in mAP over baseline SSD.

On the MS COCO dataset, the baseline SSD obtains an overall accuracy of 24.4 AP. The introduction of our feature enrichment scheme significantly improves the overall performance from 24.4 to 29.1 in AP. A notable gain in accuracy is achieved on medium sized objects. Integrating our cascaded refinement scheme boosts the overall accuracy of baseline SSD from 24.4 to 31.1 in AP. A notable performance gain is achieved on small sized objects. Our final framework combining both feature enrichment and cascaded refinement schemes provides an overall accuracy of 33.0 AP which is 8.6% higher than the baseline SSD.

Ablation Study on PASCAL VOC 2007: We try three different designs of MSCF module in our feature enrichment scheme. Tab.4 shows the results when using three different branches with varying dilation rates (*i.e.* 1, 2, 4). The best results of 79.4 mAP are obtained when using three branches in our MSCF highlighting the importance of capturing multi-scale contextual information. We further investigated adding additional branches with different dilation rates. However, this does not result in any performance improvement. Next, we analyze the effect of the kernel offset in the feature guided refinement module (FGRM) in our cascaded refinement scheme. Tab.5 shows the comparison when using different types of offsets generation used in deformable convolution operator of our FGRM. We also report standard dilated convolutional result (80.2 mAP). In case of standard deformable convolution (second row), a convolutional layer is used to learn the offsets [11]. A straigtforward way is to learn offsets by applying it directly on standard features F_m. This shows a slight improvement in performance compared to standard dilated convolution. The initial box regression from the objectness module B_{1x} predicts both position and scale offsets $(\Delta x, \Delta y, \Delta h, \Delta w)$ that can be used to learn the offsets through a 1×1 convolution. Only using the scale offsets $(\Delta h, \Delta w)$ deteriorates the performance. The best results of 81.0 mAP are obtained when using the position offsets $(\Delta x, \Delta y)$ to generate the offsets for deformable convolution. Throughout experiments, we use same dilation rates as in sec. 3.2.

Method	r1=1	r2 = 2	r3 = 4	mAP
Baseline SSD				77.2
(a)	✓			78.7
(b)	✓	✓		79.0
(c)	✓	✓	✓	**79.4**

Table 4. Ablation experiments regarding the design of MSCF module in the feature enrichment scheme on the Pascal VOC2007 test set. The results show that using multi-scale contextual information improves the detection performance.

Convolution Type	Offsets Generation	mAP
Dilated Convolution	-	80.2
Deformble Convolution	Offsets generated as in [11]	80.5
	B_{1x} $(\Delta x, \Delta y, \Delta h, \Delta w)$	80.7
	B_{1x} $(\Delta h, \Delta w)$	80.3
	B_{1x} $(\Delta x, \Delta y)$	**81.0**

Table 5. Performance comparison on PASCAL VOC 2007 when using different types of offsets generation used in deformable convolution operator of our FGRM. Offsets generated as in [11] provides only a slight improvement in performance over dilated convolution. The initial box regression from the objectness module B_{1x} predicts both position and scale offsets $(\Delta x, \Delta y, \Delta h, \Delta w)$. The best results are obtained when using the position offsets $(\Delta x, \Delta y)$ to generate the offsets for deformable convolutions.

5. Conclusion

We propose a single-stage method that tackles jointly the problem of multi-scale detection and class imbalance. We introduce a feature enrichment scheme to produce multi-scale contextual features. Further, we propose a cascaded refinement scheme that first instills these contextual features into SSD features. Second, it utilizes the enriched features to perform class-agnostic classification and bounding-box regression. Afterwards, initial box regression and binary classification are utilized to refine the features which are then used to obtain final classification scores and bounding-box regression. Experiments on two datasets show that our approach outperforms existing single-stage methods.

Acknowledgments: The work is supported by the National Natural Science Foundation of China (Grant # 61632018).

References

[1] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. Sod-mtgan: Small object detection via multi-task generative adversarial network. In *Proc. European Conference on Computer Vision*, 2018. 1

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[3] Guimei Cao, Xuemei Xie, Wenzhe Yang, Quan Liao, Guangming Shi, and Jinjian Wu. Feature-fused SSD: Fast detection for small objects. *arXiv preprint arXiv:1709.05054*, 2017. 2

[4] Jiale Cao, Yanwei Pang, and Xuelong Li. Triply supervised decoder networks for joint detection and segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2019. 1, 2, 6

[5] Kean Chen, Jianguo Li, Weiyao Lin, John See, Ji Wang, Lingyu Duan, Zhibo Chen, Changwei He, and Junni Zou. Towards accurate one-stage object detection with ap-loss. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2019. 1, 6

[6] Xingyu Chen, Junzhi Yu, Shihan Kong, Zhengxing Wu, and Li Wen. Dual refinement networks for accurate and fast object detection in real-world scenes. *arXiv preprint arXiv:1807.08638*, 2018. 2, 6

[7] Hisham Cholakkal, Jubin Johnson, and Deepu Rajan. Backtracking scspm image classifier for weakly supervised top-down saliency. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2016. 2

[8] Hisham Cholakkal, Jubin Johnson, and Deepu Rajan. Backtracking spatial pyramid pooling-based image classifier for weakly supervised top-down salient object detection. *IEEE Transactions on Image Processing*, 27(12):6064–6078, 2018. 2

[9] Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao. Object counting and instance segmentation with image-level supervision. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2019. 2

[10] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Proc. Advances in Neural Information Processing Systems*, 2016. 5

[11] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proc. IEEE international conference on computer vision*, pages 764–773, 2017. 4, 8

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 2, 5

[13] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv:1701.06659*, 2017. 1, 2, 5, 6

[14] Ross Girshick. Fast R-CNN. In *Proc. IEEE International Conference on Computer Vision*, 2015. 2

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. IEEE International Conference on Computer Vision*, 2017. 6

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 5

[19] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[20] Tao Kong, Fuchun Sun, Chuanqi Tan, Huaping Liu, and Wenbing Huang. Deep feature pyramid reconfiguration for object detection. In *Proc. European Conference on Computer Vision*, 2018. 5

[21] Tao Kong, Fuchun Sun, Anbang Yao, Huaping Liu, Ming Lu, and Yurong Chen. Ron: Reverse connection with objectness prior networks for object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5

[22] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *The European Conference on Computer Vision*, September 2018. 6, 7

[23] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proc. IEEE International Conference on Computer Vision*, 2017. 1, 2, 6

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. European Conference on Computer Vision*, 2014. 2, 5, 7

[26] Songtao Liu, Di Huang, and andYunhong Wang. Receptive field block net for accurate and fast object detection. In *Proc. European Conference on Computer Vision*, 2018. 5, 6

[27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proc. European Conference on Computer Vision*, pages 21–37, 2016. 1, 2, 3, 5, 6

[28] Yanwei Pang, Jiale Cao, and Xuelong Li. Cascade learning by optimally partitioning. *IEEE Transactions on Cybernetics*, 47(12):4148–4161, 2017. 2

[29] Yanwei Pang, Manli Sun, Xiaoheng Jiang, and Xuelong Li. Convolution in convolution for network in network. *IEEE Transactions on Neural Networks Learning Systems*, 29(5):1587–1597, 2018. 2

[30] Yanwei Pang, Tiancai Wang, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Ling Shao. Efficient featurized image pyramid network for single shot detector. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2019. 1, 2, 5, 6

[31] Yanwei Pang, Jin Xie, and Xuelong Li. Visual haze removal by a unified generative adversarial network. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018. 2, 3

[32] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. Advances in Neural Information Processing Systems*, 2015. 1, 2, 6

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 5

[35] Fahad Shahbaz Khan, Jiaolong Xu, Joost van de Weijer, Andrew Bagdanov, Rao Muhammad Anwer, and Antonio Lopez. Recognizing actions through action-specific person detection. *IEEE Transactions on Image Processing*, 24(11):4422–4432, 2015. 2

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. Advances in Neural Information Processing Systems*, 2014. 2

[37] Hanqing Sun and Yanwei Pang. Glancenets: efficient convolutional neural networks with adaptive hard example mining. *Science China Information Sciences*, 61(10):109–101, 2018. 2

[38] Wenguan Wang, Shuyang Zhao, Jianbing Shen, Steven C. H. Hoi, and Ali Borji. Salient object detection with pyramid attention and salient edges. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2019. 2

[39] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 3

[40] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 3, 4

[41] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. Single-shot refinement neural network for object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 3, 5, 6, 7

[42] Zhishuai Zhang, Siyuan Qiao, Cihang Xie, Wei Shen, Bo Wang, and Alan L. Yuille. Single-shot object detection with enriched semantics. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 5, 6

[43] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 6, 7

[44] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2019. 6

[45] Yousong Zhu, Chaoyang Zhao, Jinqiao Wang, Xu Zhao, Yi Wu, and Hanqing Lu. Couplenet: Coupling global structure with local parts for object detection. In *Proc. IEEE International Conference on Computer Vision*, Oct 2017. 5, 6