Improving Deep Object Detection Backbone with Feature Layers

Weiheng Hong[1,2,3] and Andy Song[1,4]

[1] RMIT University, Melbourne, VIC 3000, Australia
[2] Xiamen Research Center of Urban Planning Digital Technology, Xiamen, Fujian 361015, China
[3] vincent0102hwh@gmail.com
[4] andy.song@rmit.edu.au

Abstract. Deep neural networks are the frontier in object detection, a key modern computing task. The dominant methods involve two-stage deep networks that heavily rely on features extracted by the backbone in the first stage. In this study, we propose an improved model, ResNeXt101S, to improve feature quality for layers that might be too deep. It introduces splits in middle layers for feature extraction and a deep feature pyramid network (DFPN) for feature aggregation. This backbone is neither much larger than the leading model ResNeXt nor increasing computational complexity distinctly. It is applicable to a range of different image resolutions. The evaluation of customized benchmark datasets using various image resolutions shows that the improvement is effective and consistent. In addition, the study shows input resolution does impact detection performance. In short, our proposed backbone can achieve better accuracy under different resolutions comparing to state-of-the-art models.

Keywords: Object Detection · Deep Learning · Deep Neural Networks · Input Resolutions · Feature Extraction · Feature Learning

1 Introduction

As a longstanding and fundamental field in computer vision, object detection remains an active yet challenging area in modern AI [16, 25]. The goal of object detection is to determine whether there are any objects of given categories (such as person, dog, car) in the given images, if present, to return the location and area of each object instance marked by a bounding box [20]. The recent success of deep learning has made significant advancements in object detection [24]. In general, there are two dominating deep network backbones, Faster RCNN (Region Based Convolutional Neural Networks) and Mask RCNN, proposed by Girshick *et al.* [19, 5]. They achieved state-of-the-art performance on various datasets such as the MS COCO (Microsoft Common Objects in Context) dataset.

In this study, we aim to improve object detection by proposing alternative feature extraction layers for the existing backbones. The hypothesis is that features from the multi-resolution feature layers may have different importance towards

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8

the final object objection performance. Features at certain layers may not be well captured if the layer is too deep. Hence redirecting the flow of feature learning may be more beneficial as feature quality may be improved. Consequently, the detection performance can be improved. The existing object detection framework could be enhanced by leveraging these features. In addition, the effect of input resolution is investigated in this study. Input size does not only affect network structure but also connects to detection accuracy. Existing work shows that low resolution may not negatively impact some vision tasks yet can significantly save computational cost [27]. A study by [17] shows that face recognition requires a minimum resolution of 32×32 pixels. However, the input size for object detection is relatively unexplored [21]. Hence the proposed improvement on object detection backbone accommodates that need so the input of different sizes can be used. To evaluating the effectiveness of the proposed improvement, customized benchmark COCO data are used in the following study. The proposed object detection backbone is beneficial as evidenced by the comparison with state-of-the-art. The details are presented in the following sections.

2 Background & Related Work

Object detection is defined as follows, to determine if or not there are instances of objects from given categories on a given image. If objects are present, the locations and areas of the detected instances should be marked. Although there are numerous kinds of objects that exist in our visible world, object detection research mainly studies methods for detecting highly structured objects and articulated objects rather than unstructured scenes. Structured objects such as faces, cars, ships, and airplanes, normally have a consistent shape. Articulated objects are usually living beings such as a person, a dog, and a bird. Different from these two types of objects, unstructured scenes are unpredictable in terms of shape, for example, sky, fire, and water. Four kinds of recognition can be derived

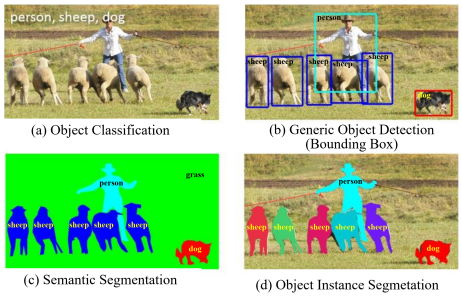

Fig. 1. Examples of four object detection tasks illustrated by Liu *et al.* [13].

from object detection. That includes image-level object classification, bounding box level object detection, pixel-wise semantic segmentation and instance-level semantic segmentation, as illustrated by [13] (**Figure 1**). Surveys indicate the

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8

bounding box object detection is the most widely used and is the basis for evaluating the performance of an object detection algorithm. Some object detection frameworks use a bonding box combined with others. For example, Mask RCNN uses pixel-wise segmentation, which is to assign each pixel in an image to a semantic class label for high-level detection. Object detection tasks can be handled by several types of methods including deep networks models [28], statistical models, and Genetic Programming. The currently best performing methods are deep networks based.

2.1 Object Detection Frameworks

Many object detection models have been proposed. They can be categorized into two groups: one-stage framework and two-stage framework. The latter is represented by region-based frameworks which in general achieved superior performance than other methods. This approach uses a CNN (Convolutional Neural Networks) backbone to generate category-independent regions from an image. Consequently feature extractors are embedded to find useful features from these regions. Based on these features a classifier then is applied to determine whether instances of a given class are present in region proposals. If present, category labels will be returned. This type of two-stage approach can be found in many object detection frameworks such as RCNN [4], Fast RCNN [3], Faster RCNN and Mask RCNN. On the MS COCO object detection competition, state-of-the-art Mask RCNN framework, Cascade Mask R-CNN (Triple-ResNeXt152, multi-scale), achieved top performance of 71.9% mAP with $IoU = 0.50$ [15].

Two-stage frameworks can achieve superior detection performance. In comparison, one-stage frameworks can often increase detection speed as they use a unified pipeline structure to directly predict class labels with a single neural network. Region proposal network and feature extractor are absent in this category. One-stage frameworks such as YOLO [18], SSD [14], and YOLO9000 have gained popularity in recent years, due to their simplicity and low cost. This approach makes real-time object detection possible especially under circumstances that computational resource is limited, such as droids and other embedded systems.

In summary, two-stage frameworks achieve state-of-the-art detection accuracy with complex neural network architectures, while one-stage framework using a simple elegant structure to achieve high detection speed. This study aims to improve object detection accuracy then focuses on the two-stage approach. In the family of two-stage object detection frameworks, RCNN is the leading model. Originating from the earliest RCNN model, the RCNN family has advanced to Faster RCNN and Mask RCNN, which are mentioned before as state-of-the-art in object detection. Hence the focus of this study is on Faster RCNN and Mask RCNN, improving their feature extraction backbone.

2.2 Feature Extractors

One of the major components in the object detection models is the backbone, which is responsible for extracting features for the subsequent classification stage.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8

Feature extractor plays a crucial role in feature representation as well as in the whole detection task [23].Deep neural networks (DCNNs) have been found capable of generating distinct features from raw images at a different level in multi-resolution pyramid representation. DCNN is present in many well-known deep models such as AlexNet [9], VGGNet [22], ResNet [6], ResNeXt [26], DenseNet [8], and MobileNet [7] *et al.*. They are listed in **Table 1** for comparison in terms of parameter size, the number of layers, and test errors on the benchmark.

Table 1. Comparison of Representative DCNN for Image Classification

DCNN Architecture	#Paras ($\times 10^6$)	#Layers (CONV+FC)	Test Error (Top 5)
AlexNet [9]	57	5 + 2	20.91%
VGGNet19 [22]	134	13 + 2	9.62%
ResNet50 [6]	23	49	7.13%
ResNet101 [6]	42	100	6.44%
ResNeXt50 [26]	23	49	6.30%
ResNeXt-101 [26]	42	100	5.47%
DenseNet201 [8]	18	200	6.43%
MobileNet [7]	3.2	27 + 1	9.71%

From **Table 1** a trend can be seen that in general deeper networks, meaning with more layers, can lead to better feature representation in CNN, hence to lower error rates. Another observation is the relation between model size or the number of parameters and the use of FC (Fully Connected) layers. A model with a '+' sign in **Table 1** means FC is present. For example, AlexNet is 5+2, meaning 2 layers of FC are presented with 5 convolutional layers. From the table, we can see that AlexNet and VGGNet utilize FC layers which result in significantly more parameters onto the model. DenseNet and ResNet have fewer parameters by leaving out the FC layers. Therefore, avoiding the use of FC layers can lead to smaller models without damaging, if not improving, the detection accuracy.

After the introduction of Mask RCNN, a more effective feature extraction method is proposed by Lin *et al.*, call Feature Pyramid Network (FPN). FPN uses a top-down architecture with lateral connections in each layer of the backbone to build a feature pyramid and made predictions independently at all levels. These RoI (Region of Interest) features extracted from different layers contribute to the feature maps in various aspects. Mask RCNN produces excellent accuracy and efficiency largely due to the use of ResNet-FPN backbone [11].

2.3 Performance Evaluation

Most of the current work uses mAP for evaluation, especially after the introduction of the MS COCO dataset. Instead of using a fixed IoU (Intersection over Union) threshold, the MS COCO introduces various metrics to better measure the performance of a given object detection model. That includes AP ($IoU = 0.50 : 0.95$), AP ($IoU = 0.75$), and AP ($IoU = 0.5$). AP ($IoU = 0.50 : 0.95$) metric is primarily used in the MS COCO Dataset challenge. AP ($IoU = 0.75$) represents a more strict metric for evaluation. In this study, we use the most commonly used AP ($IoU = 0.5$) metric to evaluate our object detection model.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8

2.4 Datasets

For generic object detection, four famous datasets are utilized around the community, include PASCAL VOC [2], ImageNet [1], MS COCO [12] and Open Images [10]. In this research, in order to study the relationship between various image resolution and object detection models, MS COCO dataset with the highest image resolution and well organized could be the preference around four image datasets. In addition, object segmentation data make it possible to use in the Mask R-CNN model, which is designed for object detection and segmentation using segmentation-level detection technology.

2.5 Resolutions

The impact of resolution is relatively under-explored in machine vision, in particular object detection. Shivanthan *et al.* [27] report that using low-resolution grayscale (LG) images for saliency detection can lead to speedups in model training and detection time. Region Proposal Network (RPN), based on this novel saliency-guided selective attention theory, separates the objects' regions and background regions. Therefore, using LG images for object detection can greatly improve the efficiency of object detection and keep the object detection model in a small size. But experiments indicate this model usually fails to detect the main object when the size of the image is smaller than 64×64 pixels. A study on face recognition requires a minimum input of 32×32 pixels [17].

3 Methodology

The main methodology of this study is presented in this section. That includes dataset preparation for evaluation, the preliminary experiment on the chosen data, and the design of deep models.

3.1 Data Preparation

COCO 2017 data set is a leading benchmark for object detection [5]. Three types of datasets are included Training dataset, Validation dataset, and Testing dataset. The testing dataset is used for COCO competition that does not provide annotations for evaluation locally, so we use the Training dataset for model learning, and the Validation dataset to evaluate the model by computing the bounding box AP ($IoU = 0.50$) value. COCO data sets contain 200,000 images of 80 object categories. In this study, we group selected categories into three categories: rectangle object class (such as buses, vehicles), convex-polygon object class (such as dogs), and round objects (such as apples). These include the most representative classes of the COCO dataset. Such alternation is to facilitate the study especially the analysis as a benefit on backbone improvement should be independent of how to categorize target objects.

[5] Available on the MS COCO dataset website http://cocodataset.org

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8

3.2 Preliminaries

As the RPN object detection model proposed by Shivanthan *et al.* failed to detect the object when the image is resized to 64×64 pixels, we set 64×64 as the definition of low resolution in our preliminary work to test the performance of different object detection model include Faster RCNN and Mask RCNN with various popular backbones. Model is trained with a training dataset of 64×64 resolution, and then tested on 64×64 validation dataset. The metric is the bounding box mAP ($IoU = 0.50$) resulting from our three-class COCO datasets. **Table 2** shows the results from the preliminary study which involves a range of widely used deep models including AlexNet, MobileNet, DenseNet, VGG, ResNet models, and ResNeXt models. The suffix number after a model name is the depth of the model. For example, ResNet34 means that is a ResNet model with 34 layers. As can be seen from the table, with FPN, ResNet models and ResNeXt models

Table 2. Bounding box mAP (IoU=0.50) value of object detection results on validation dataset under 64×64 resolution. The number in the backbone represents the number of the layers.

Backbone	Faster RCNN	Mask RCNN
AlexNet	9.7	9.9
MobileNet	14.6	14.2
Densenet201	15.7	16
VGGNet16	21.2	21.5
VGGNet19	19.8	20.7
ResNet34 + FPN	30	30.9
ResNet50 + FPN	30.8	31.7
ResNet101 + FPN	30.6	32.1
ResNet152 + FPN	31.7	31.6
ResNeXt50 + FPN	**32.5**	**33.4**
ResNeXt101 + FPN	31.4	32.2

achieved mAP values higher than other models in both Faster RCNN and Mask RCNN. ResNeXts performs slightly better than ResNets. The subsequent study is therefore based on ResNeXt models. In addition, Mask RCNN models, in general, perform better than their Faster RCNN counterparts. While ResNeXt50 with FPN gets the best result of mAP on both Faster RCNN and Mask RCNN frameworks, a natural question is that why ResNeXt101 with a deeper convolutional neural network was inferior to ResNeXt50. A similar phenomenon happens in VGGNet16 and VGGNet19. While VGGNet19 contains more layers and parameters than VGGNet16, it does not achieve a better result than VGGNet16. The analysis is that the deeper networks may result in better features but also may treat features indifferently through the deep layers. So deeper net may not be as helpful if features are not utilized well. These features may not represent some small and unnoticeable objects, especially from low-resolution images. If these objects are indeed targets, the generated features may not capture them leading to a slight decline in terms of mAP measure.

Preliminary work shows ResNeXt50 with FPN performs better than ResNeXt101 with FPN in object detection of low-resolution images. The detail of our model

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8

Table 3. ResNeXt50 structure, ResNeXt101 structure and ResNeXt101S structure. "C=32" suggests grouped convolutions with 32 groups.

layer name	ResNeXt50	ResNeXt101	ResNeXt101S **(ours)**
conv1	7×7, 64, stride 2		
	3×3 max pool, stride 2		
conv2_x	$\begin{bmatrix} 1 \times 1, & 128, \\ 3 \times 3, & 128 \\ 1 \times 1, & 256, \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, & 128, \\ 3 \times 3, & 128 \\ 1 \times 1, & 256, \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, & 128, \\ 3 \times 3, & 128 \\ 1 \times 1, & 256, \end{bmatrix} \times 3$
conv3_x	$\begin{bmatrix} 1 \times 1, & 256, \\ 3 \times 3, & 256 \\ 1 \times 1, & 512, \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, & 256, \\ 3 \times 3, & 256 \\ 1 \times 1, & 512, \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, & 256, \\ 3 \times 3, & 256 \\ 1 \times 1, & 512, \end{bmatrix} \times 4$
conv4_x	$\begin{bmatrix} 1 \times 1, & 512, \\ 3 \times 3, & 512 \\ 1 \times 1, & 1024, \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, & 512, \\ 3 \times 3, & 512 \\ 1 \times 1, & 1024, \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, & 512, \\ 3 \times 3, & 512 \\ 1 \times 1, & 1024, \end{bmatrix} \times 6$ $\begin{bmatrix} 1 \times 1, & 512, \\ 3 \times 3, & 512 \\ 1 \times 1, & 1024, \end{bmatrix} \times 17$
conv5_x	$\begin{bmatrix} 1 \times 1, & 1024, \\ 3 \times 3, & 1024 \\ 1 \times 1, & 2048, \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, & 1024, \\ 3 \times 3, & 1024 \\ 1 \times 1, & 2048, \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, & 1024, \\ 3 \times 3, & 1024 \\ 1 \times 1, & 2048, \end{bmatrix} \times 3$

design will be introduced in the first part. The experiment is set to evaluate our model in the second part.

3.3 Backbone Model Design

In this study, we propose an improved object detection backbone by using the deep feature pyramid network (DFPN) method which can enhance the expression of feature maps. There are a number of powerful backbones based on Faster RCNN and Mask RCNN with high performance, such as ResNet, RTesNeXt, and VGGNet. In this section, we first describe our backbone design and the two frameworks that are used, Faster RCNN and Mask RCNN respectively. Our backbone construction is based on ResNeXt that uses a parallel structure with 32 groups of the identical blocks of ResNet to construct its block.

The structure of ResNeXt50 and ResNeXt101 are presented in **Table 3**. The main difference between these two CNNs is the *conv4_x* layer. While ResNeXt50 uses 6 sequential blocks to extract features, ResNeXt101 uses a deeper block of 23 layers in the *conv4_x* layer. As we discussed in the preliminary work, that leads to the relative lower mAP as some small and unnoticed objects may not be captured by the extracted features.

The structure of our proposed backbone is also presented in **Table 3** alongside with ResNeXt50 and ResNeXt101. We name it as ResNeXt101S. The major difference is the splitting of the *conv4_x* layer into two sub-layers. They are

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8

conv4_x_6 layer and *conv4_x_17* layer respectively. The first sub-layer consists of 6 blocks of ResNeXt, while the second sub-layer is composed of 17 blocks of ResNeXt. As for the structure of each block, the same set of output channels is still maintained. They all behave as blocks of *conv4_x* of ResNeXt101 and are grouped by 32 parallel paths. A diagram of the proposed ResNeXt101 backbone

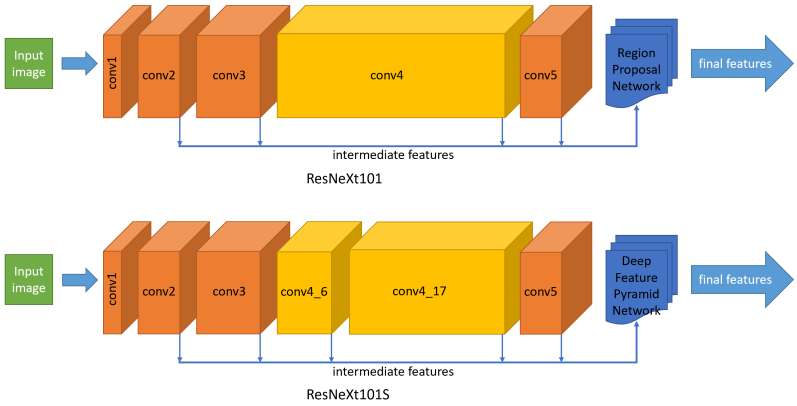

Fig. 2. Two Backbones: Top: ResNeXt101 with FPN. Bottom: ResNeXt101S with DFPN.

can be seen in **Figure 2**. Other than the splitting middle layer, a new Deep Feature Pyramid Network (DFPN) is also proposed to replace the region proposal network. It not only takes features from the basic layers of ResNeXt101S, but also takes features from the inner layer of *conv4_x* as shown in the figure. By using a 5-layer top-down architecture through the entire ResNeXt101S CNN, it extracts and generates a feature pyramid from basic layers and inner layers include *conv2_x*, *conv3_x*, *conv4_x_6* layer, *conv4_x_17* layer, and *conv5_x*. The features are extracted from each level of the feature pyramid to contribute towards the feature maps in which predictions are made at all levels. The aim is to compensate for lost features and to enable more prominent features to be captured. Based on ResNeXt101S with DFPN, our backbone is adapted into both Faster RCNN and Mask RCNN frameworks.

With Faster RCNN: while the original Faster RCNN model using the RoIPool method and applying VGG16 as the backbone, we introduce our backbone based on the original Faster RCNN framework, but applying RoIAlign for pooling, which has been shown of being able to increase mAp in Mask RCNN.

The Anchor Generator as the top component of region proposal network (RPN) is used to generate a number of boxes (Anchors) to detect target objects in the image that the Anchor Box size in RPN can be associated with the input image size, thus we set the size with this equation:

$$area(x) = \left\{ \left(\frac{0.7x}{2^i}\right)^2 \; i = 0, 1, 2, 3, 4 \right. \tag{1}$$

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8

For the different input image sizes, we design different object detection with size-specific anchor boxes respectively, then apply for training and testing.

With Mask RCNN: while the original Mask RCNN model using ResNeXt-101 with FPN as a backbone to achieve state-of-the-art mAP performance, we introduce our proposed backbone on the original Mask RCNN Framework, but set the Anchor Box size also using the Equation (1) shown above.

4 Experiments and Results

4.1 Experiment Settings

Experiments are set to verify and evaluate the performance of our proposed ResNeXt + DFPN backbone. During the training, each batch has 2 images in one GPU. The Adadelta algorithm for optimization is used with the learning rate of 0.3, which is decreased by 5 at the 30 iterations [29]. A coefficient of 0.9 is used. For each model, we train with one NVIDIA Tesla V100 GPU and select the results with the best bounding box AP value.

Firstly we compare the performances of object detection models on three-class datasets under 64×64 resolution, similar to the experiments in the preliminary work. Our ResNeXt101S + DFPN backbone with the backbones which perform quite well in the preliminary work include VGGNet16, ResNet + FPN and ResNeXt + FPN. All backbones are combined with Fater RCNN and Mask RCNN in the experiments by training with 64×64 training dataset and testing in 64×64 validation dataset. The detection accuracy is measured via the bonding box mAP ($IoU = 0.50$) value over three class datasets. As the Mask RCNN framework can give better detection ability than Faster RCNN due to its more informed learning style, we adopt the Mask RCNN framework to evaluate various backbones in terms of model size and detection accuracy.

In the second stage of experiments we investigated the effectiveness of our ResNeXt101S + DFPN backbone with Mask RCNN on three class datasets under various image resolutions. That includes 64×64, 128×128, 256×256, 512×512, and 1024×1024. Since the maximum resolution of original images is 640, we also test with the size of 640×640. We compare our backbone with ResNeXt50 + FPN and ResNeXt101 + FPN backbones. The performances are measured via bonding box mAP ($IoU = 0.50$) value over three class datasets.

The results are presented in the following two subsections. To compare the performance of our model with others, VGGNet16, ResNet50 models, ResNeXt models are also included.

4.2 Object Detection with low-resolution images

While testing with "rectangle" class dataset, "convex polygon" class dataset, and "round" class dataset under 64×64 image resolution, **Table 4** presents the boding box mAP ($IoU = 0.50$) value of the object detection results.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8

Table 4. Bounding box mAP ($IoU = 0.50$) value (%) of object detection results on dataset under 64×64 resolution. The experiment result of VGGNet16, ResNet + FPN, ResNeXt + FPN, and ResNeXt101S + DFPN backbones with Faster RCNN and Mask RCNN frameworks.

Backbone	Faster RCNN	Mask RCNN
VGGNet16	*21.2*	21.5
ResNet50 + FPN	30.8	31.7
ResNet101 + FPN	30.6	32.1
ResNet152 + FPN	31.7	31.6
ResNeXt50 + FPN	32.5	33.4
ResNeXt101 + FPN	31.4	*32.2*
ResNeXt101S + DFPN (ours)	**33**	**33.9**

As the Faster RCNN first proposed in 2015 using VGGNet16 achieves 21.2 mAP testing accuracy, Mask RCNN proposed in 2017 using ResNeXt101 with FPN with the detection accuracy increases to 32.2. Rather than testing on high-resolution images as in previous papers, we apply low-resolution images. It can be seen that ResNeXt50 with FPN performs better than ResNeXt101 with FPN in both RCNN frameworks. While ResNeXt50 with FPN achieves 32.5 mAP in Faster RCNN and 33.4 mAP in Mask RCNN. The proposed new backbone, ResNeXt101S with DFPN, actually increases 0.5 mAP value higher than them in both Faster RCNN and Mask RCNN frameworks.

The experiment results of bounding box mAP value of object detection on the dataset is presented in **Table 5**. That result shows DFPN used in ResNeXt101S is beneficial in terms of raising detection accuracy in low-resolution images. That observation verifies the analysis that ResNeXt101 does not give better performance than ResNeXt50 in low-resolution images because its deeper convolutional neural network structure is less effective in capturing features.

By splitting In this way, it increases 1.7 mAp point of detection accuracy to 33.9 compare with the original Mask RCNN model which uses ResNeXt101 with FPN as the backbone. While ResNeXt101S with DFPN in the Mask RCNN framework obtains the best mAP result in bounding box object detection in low-resolution images, the model size of it is very close to the ResNeXt101 with FPN. Compared with the model size of 430 MB of ResNeXt101 with FPN, our ResNeXt101S with DFPN is just 10 MB larger.

4.3 Object detection with various resolutions

As ResNeXt101S with DFPN performed quite well in low-resolution images, we here evaluate whether it is still superior to others in different resolutions of images. The resolutions for test include 64×64, 128×128, 256×256, 512×512, 640×640, and 1024×1024.

Table 5 shows the results. Under the resolution of 64×64 and 128×128, ResNeXt101 has the worst result among three backbones, while ResNeXt101 performs slightly better than ResNeXt50, which means by using DFPN with ResNeXt101S in Mask RCNN, the model does improve feature quality in low-resolution images. For the other resolutions that are higher than 128×128,

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8

Table 5. Bounding box mAP ($IoU = 0.50$) value (%) of object detection results on three class datasets under dataset-specific resolutions. The experiment result of ResNeXt50 + FPN, ResNeXt101 + FPN, and ResNeXt101S + DFPN backbones with Mask RCNN frameworks.

Resolution	ResNeXt50 with FPN	ResNeXt101 with FPN	ResNeXt101S with DFPN (ours)
64×64	33.4	32.2	**33.9**
128×128	42.3	41.5	**42.7**
256×256	54.1	55.4	**55.9**
512×512	58.9	60.2	**61**
640×640	61.1	63.6	**63.9**
1024×1024	63.8	65.1	**65.9**

ResNeXt50 does not have better detection accuracy than ResNeXt101, which shows the advantage of a deep convolutional neural network. Since under the sufficient resolution condition, a deeper CNN extracts the shape of the feature with more distinct expression than others. In this case, using ResNeXt101 + FPN as the backbone is possible to achieve a better object detection performance than using ResNeXt50 + FPN. While ResNeXt101 + FPN performs quite well in high resolution, ResNeXt101S + DFPN can still achieve higher mAP result with slightly improvement in 256×256, 512×512, 640×640, and 1024×1024. Such an outcome indicates using ResNeXt101S with DFPN can still improve feature quality at high-resolution input.

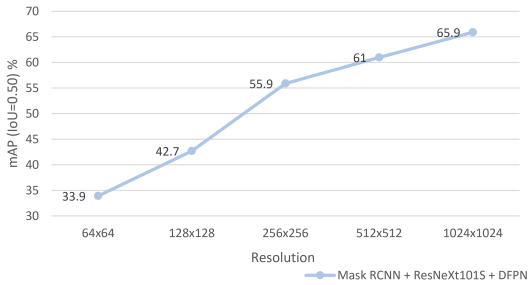

Fig. 3. Bounding box mAP ($IoU = 0.50$) value (%) vs. Dataset-specific resolutions. The experiment result of ResNeXt101S + DFPN backbone with Mask RCNN frameworks.

Overall by using ResNeXt101S with DFPN in Mask RCNN, our model achieved the best object detection performance for various resolutions of input images. An interesting observation that worth mentioning is comparing the performance under 640×640 with the performance under 1024×1024. While we upscale the original image with a maximum image size of 640 to a higher 1024, the object detection accuracy still increases with 2.0 mAP points rather than remain the same as 640×640 images despite the fact that no extra information was added in the up-scaling process.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8

Table 6. Bounding box AP ($IoU = 0.50$) value (%) of object detection results under 1024×1024 resolution on Rectangle class, Convex-polygon class and Round object class. The experiment result of ResNeXt50 + FPN, ResNeXt101 + FPN, and ResNeXt101S + DFPN backbones with Mask RCNN frameworks.

Backbone	Rectangle Class	Convex-polygon Class	Round Class	Average mAP
ResNeXt50 + FPN	80.9	80	30.5	63.8
ResNeXt101 + FPN	81.7	81.2	32.4	65.1
ResNeXt101S + DFPN (ours)	**82.6**	**82.1**	**33.1**	**65.9**

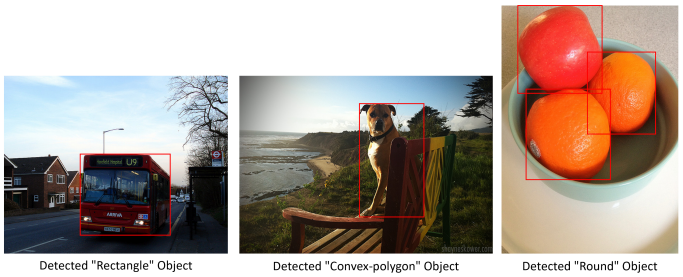

Fig. 4. Example of prediction result.

Figure 3 shows the tendency graph of detection performance in relation with various resolutions. We collect the data with the sequence of the multiple of image resolution which contains the resolution include 64×64, 128×128, 256×256, 512×512, and 1024×1024. With increasing resolution, the accuracy of object detection increases. We can observe that the accuracy increasing rate is fastest from 128 × 128 to 256 × 256, and then become slow down in the 512 × 512 and 1024 × 1024. As a larger image needs more time-consuming and computational-consuming, an image resolution as small as possible is desirable. Therefore, 256×256 could be selected as the optimal resolution for object detection with good object detection accuracy and high efficiency.

For further investigation, we present the analysis of detection performance per class in **Table 6**. The table shows the mAPs of all three models tested under images of 1024 × 1024 in this study. As can be seen, our proposed ResNeXt101S + DFPN backbone performed best in all three classes. Its good performance is independent of class type. In terms of the classes, "Rectangle" objects and "Convex-polygon" objects can be much more accurately detected than "Round" objects. That is the case for all three models. The possible explanation is that round objects may be confused with non-target round objects. Our further study will address that to improve performance. Nevertheless, such results confirm that the good performance of our proposed backbone is not random. Examples of detected objects are illustrated in **Figure 4**. They represent three categories of objects from the COCO dataset. The bounding boxes (in red) are the output from the Mask RCNN model using our proposed backbone. These boxes fit with the target object tightly showing the good performance of the detection model.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8

5 Conclusions

In this paper, we proposed an improved backbone for object detection with an innovative method that combines advantages from both backbones. The aim is to improve feature quality for deep layers. From experiments, it can be seen that deep models may not extract high-quality features if the layers are deep. The improved backbone split the feature layers and re-direct intermediate features to a proposed deep feature pyramid network (DFPN) for feature aggregation. This backbone can be integrated into leading frameworks including Faster RCNN and Mask RCNN and is applicable for handling a range of different image resolutions. With the improved backbone, better detection performance can be achieved on different resolutions comparing to state-of-the-art models. In conclusion, our method improves the object detection performance without increasing the number of parameters and computational complexity distinctly. The proposed backbone is beneficial in improving feature quality for object detection.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
2. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International journal of computer vision **111**(1), 98–136 (2015)
3. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
4. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
5. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
7. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
8. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
9. Krizhevsky, A.: One weird trick for parallelizing convolutional neural networks. arXiv preprint arXiv:1404.5997 (2014)
10. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T., et al.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. arXiv preprint arXiv:1811.00982 (2018)
11. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8

12. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
13. Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M.: Deep learning for generic object detection: A survey. arXiv preprint arXiv:1809.02165 (2018)
14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
15. Liu, Y., Wang, Y., Wang, S., Liang, T., Zhao, Q., Tang, Z., Ling, H.: Cbnet: A novel composite backbone network architecture for object detection. arXiv preprint arXiv:1909.03625 (2019)
16. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
17. Lui, Y.M., Bolme, D., Draper, B.A., Beveridge, J.R., Givens, G., Phillips, P.J.: A meta-analysis of face recognition covariates. In: 2009 IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems. pp. 1–8. IEEE (2009)
18. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
19. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
20. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
21. Shekhar, S., Patel, V.M., Chellappa, R.: Synthesis-based robust low resolution face recognition. arXiv preprint arXiv:1707.02733 (2017)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
23. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781–10790 (2020)
24. Tang, P., Wang, X., Wang, A., Yan, Y., Liu, W., Huang, J., Yuille, A.: Weakly supervised region proposal network and object detection. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 352–368 (2018)
25. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Scaled-yolov4: Scaling cross stage partial network. arXiv preprint arXiv:2011.08036 (2020)
26. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017)
27. Yohanandan, S., Song, A., Dyer, A.G., Tao, D.: Saliency preservation in low-resolution grayscale images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 235–251 (2018)
28. Zangeneh, E., Rahmati, M., Mohsenzadeh, Y.: Low resolution face recognition using a two-branch deep convolutional neural network architecture. arXiv preprint arXiv:1706.06247 (2017)
29. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_8