ScaleNet: Guiding Object Proposal Generation in Supermarkets and Beyond

Siyuan Qiao[1]   Wei Shen[1,2]   Weichao Qiu[1]   Chenxi Liu[1]   Alan Yuille[1]
Johns Hopkins University[1]   Shanghai University[2]

{siyuan.qiao, wqiu7, cxliu, alan.yuille}@jhu.edu   wei.shen@t.shu.edu.cn

Abstract

Motivated by product detection in supermarkets, this paper studies the problem of object proposal generation in supermarket images and other natural images. We argue that estimation of object scales in images is helpful for generating object proposals, especially for supermarket images where object scales are usually within a small range. Therefore, we propose to estimate object scales of images before generating object proposals. The proposed method for predicting object scales is called ScaleNet. To validate the effectiveness of ScaleNet, we build three supermarket datasets, two of which are real-world datasets used for testing and the other one is a synthetic dataset used for training. In short, we extend the previous state-of-the-art object proposal methods by adding a scale prediction phase. The resulted method outperforms the previous state-of-the-art on the supermarket datasets by a large margin. We also show that the approach works for object proposal on other natural images and it outperforms the previous state-of-the-art object proposal methods on the MS COCO dataset. The supermarket datasets, the virtual supermarkets, and the tools for creating more synthetic datasets will be made public.

1. Introduction

There is an exciting trend in developing intelligent shopping systems to reduce human intervention and bring convenience to human's life, *e.g.*, *Amazon Go*[1] system, which makes checkout-free shopping experience possible in physical supermarkets. Another way to enhance the shopping experience in supermarkets is setting customer free from finding and fetching products they want to buy, which drives the demand to develop shopping navigation robots. This kind of robots can also help visually impaired people shop in supermarkets. The vision system of such a robot should have the abilities to address two problems sequentially. The first is generating object proposals for products in images captured by the equipped camera (Fig. 1), and the second is

---

[1]https://www.amazon.com/b?node=16008589011

Figure 1: Example Object Annotations in the Supermarket Datasets (Left) and the MS COCO Datasets [26] (Right). Yellow: object scale is between 20% and 30% of the image scale; red: between 10% and 20%; green: less than 10%. The ratio is calculated as the maximum of the width and the height of the object divided by the maximum of the width and the height of the image. No other object scales appear in the examples.

identifying each product proposal. In this paper, we focus on the first problem.

There are many object proposal methods for general natural images [33, 34, 42, 46]. However, scenes of supermarkets are usually very crowded, *e.g.*, one image taken in supermarkets could have over 60 products. More challengingly, products of the same brands and categories are usually placed together, *i.e.*, the appearance similarities between adjacent products are often high, making the boundaries between them hard to detect. Consequently, the current object proposal detection methods, including superpixel grouping based [1, 20, 42], edge or gradient computation based [7, 46] and saliency and attention detection based [2, 4, 5, 24, 28], are less effective and require a large number of proposals to achieve reasonable recall rates.

However, we observe that the products in supermarkets typically occur at a limited range of scales in the image. To demonstrate this, we plot the distribution of the number of object scales in real-world supermarkets (Fig. 2). This suggests a strategy where we estimate object scales and use them to guide proposals rather than exhaustive searching on all scales. The same strategy of reducing search space of scales is also applicable to other natural images in the MS COCO [26], and it becomes very effective especially for those that have sparse object scales (Fig. 2), for which an effective scale prediction can reduce the search space and
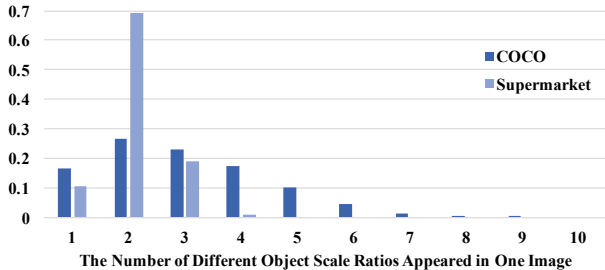
Figure 2: Distributions of the Number of Different Object Scale Ratios of One Image on the MS COCO [26] Dataset and the Real-World Supermarket Dataset. The ratio of the object size (the maximum of width and height) to the image size (the maximum of width and height) is partitioned evenly to 10 bins from 0 to 1. We count the number of different scale ratios appeared in one image on the datasets. The object scales of supermarket images are sparser than that of images in the MS COCO. Since 97.5% supermarket images have neighboring non-zero bins, the scale distributions are within a small range compared to the entire scale space. Moreover, a reasonable number of images in the MS COCO dataset also have fairly sparse object sizes.

eliminate false positives at improper scales.

More precisely, we propose a scale-aware object proposal detection framework to address the problem (Fig. 3). Our framework consists of two sequential parts. The first is a scale estimation network, called ScaleNet, which predicts the scale distribution of the objects appeared in an image. The second is an object proposal detection network, which performs detection on re-scaled images according to the estimated scales. For the second part, we use a deep learning based object proposal detection method Sharp-Mask [34], which predicts objectness confidence scores and object masks at each location of the input image at several pre-defined scales. Since this method can output dense object masks, it fits the supermarket images well.

We evaluate the proposed framework on general natural images and supermarket images. To evaluate our framework on natural images, we test it on the MS COCO dataset. For the supermarket images, we collect two real-world supermarket datasets, in which the bounding boxes of products are annotated by humans. The first dataset is called *Real-Far*, which is composed of 4033 products labeled and has less variation in object scales. The second dataset is called *Real-Near*, which has 3712 products labeled with more variation in scales. The objective of collecting two datasets is to evaluate and compare the performances in different settings of object scales.

Since human labeling for crowded scenes is very time-consuming and expensive, to generate enough training data, we use a Computer Graphics technique [35] to generate a synthetic dataset, which includes 154238 objects labeled for training and 80452 objects for validation. The synthetic

dataset is used for training and validation and the two real-world datasets are used only for testing.

To summarize, the contributions of this paper include

- A scale estimation method ScaleNet to predict the object scales of an image.
- An object proposal framework based on ScaleNet that outperforms the previous state-of-the-arts on the supermarket datasets and MS COCO.
- Two real-world supermarket datasets and a synthetic dataset, where the model trained only on synthetic dataset transfers well to the real-world datasets. The datasets and the tools will be made public.

2. Related Work

In this section, we review the related work in the research topics including object proposal methods and virtual environment constructions.

2.1. Object proposal

The previous work usually falls into two categories: one is bounding box based, and the other is object mask based. Both can generate object proposals in the form of bounding box. In bounding box based methods such as Bing [7] and EdgeBox [46], local features such as edges and gradients are used for assessing objectness of certain regions. Following the success of CNNs in image classification [14, 22, 41], DeepBox [23] re-ranks the object proposals generated by EdgeBox [46], and DeepProposal [13] generates object proposal by an inverse cascade from the final to the initial layer of the CNN. MultiBox [10] and SSD [29] compute object regions by bounding box regression based on CNN feature maps directly. In SSD, YOLO [36] and RPN [37], anchor bounding boxes are used to regress bounding boxes. Jie *et al.* [18] proposed scale-aware pixel-wise proposal framework to handle objects of different scales separately. Although some methods use multi-scales to generate proposals, they do not explicitly estimate the object scales.

Object mask based methods propose object bounding boxes by segmenting the objects of interest from the corresponding background at pixel or region level. This type of methods can detect objects by seed segmentation such as GOP [20] and Learning to Propose Objects [21]. They can also group over-segmented regions to propose objects such as Selective Search [42] and MCG [1]. More recently, DeepMask [33] assesses objectness and predicts object masks in a sliding window fashion based on CNN features, which achieved the state-of-the-art performance on the PASCAL VOC [11] and the MS COCO [26] datasets. SharpMask [34] further refines the mask prediction of DeepMask by adding top-down refinement connection. Our method extends the previous state-of-the-art SharpMask by adding object scale prediction and outperforms them on the supermarket dataset and on the MS COCO.
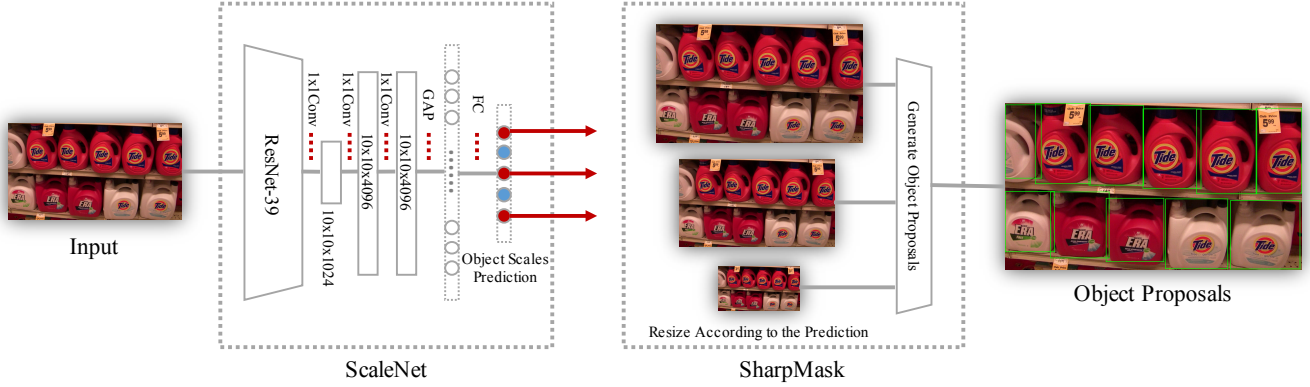
Figure 3: The System Overview of the Proposed Object Proposal Framework. The system has two components: ScaleNet proposed in this paper and SharpMask [34]. ScaleNet outputs a predication of the scale distribution of the input image, according to which the input image is resized and fed to SharpMask. SharpMask then generates object proposals at the predicted scales. The image is best viewed in color.

2.2. Virtual environment construction

Using synthetic data for Computer Vision research has attracted a lot of attention in recent work. Examples include using synthetic data on semantic segmentation [38, 39], optical flow [3, 8], stereo [31, 45], *etc*. To get virtual environments, the first way is by taking advantages of the existing virtual environments [9, 19, 30, 38]. The second way is to use open source platform such as UnrealCV [35] to construct virtual worlds from scratch. We adopt the second approach and use UnrealCV to build virtual supermarkets. When constructing virtual environment from scratch, spatial modeling is important for creating realistic environments [12, 44]. The synthetic dataset introduced in this paper builds the virtual environments from scratch with randomness considered in spatial modeling, material and lighting conditions to create realistic images.

3. System Overview

This section presents the system overview of the object proposal framework proposed in this paper, as shown in Fig. 3. The system is composed of two sequential components: the ScaleNet proposed in this paper and Sharp-Mask [34]. The function of ScaleNet is to predict the scales that best describe the statistics of the image so that Sharp-Mask can utilize the predicted scales to find objects better in the image and outputs proposals. ScaleNet looks at the input image only once to predict the distribution of the object scales while SharpMask looks at the input image multiple times at the scales that are predicted by ScaleNet.

The main difference between the proposed framework and SharpMask alone is the way they handle scales. Sharp-Mask exhaustively searches a pre-defined scale set and generates object proposals from that. By contrast, this paper refines the scale set so that SharpMask can take the image at a finer range of scales for object proposal generation.

4. Scale Distribution Prediction

This section formulates the problem of scale distribution prediction, presents the architecture of the proposed method ScaleNet, and connects ScaleNet to SharpMask.

4.1. Problem formalization

Given an image I, we denote the objects of interest in the image I as $O = \{o_1, o_2, ..., o_n\}$. Let m_i denote the maximum of the width and the height of the bounding box of object o_i, for $i = 1, ..., n$. Suppose the object o_i can be best detected when the image is resized such that m_i is equal to an ideal size denoted as D. This is aiming at work in which there is a set of object sizes that models are trained at [6, 16, 27, 33, 34, 43]. Then the scale that image I needs to be resized to favor detecting object o_i is $g_i = D/m_i$. Note that g_i is continuous, and finding scales for every object o_i is inefficient. Therefore, instead of formulating the problem as a regression problem, we discretize the scales into several integer bins and model the problem as a distribution prediction problem.

Suppose for scale distribution we have integer bins $B = \{b_1, b_2, ..., b_l\}$ with discretization precision $\sigma \in \mathbb{Z}^+$, where $b_{i+1} = b_i + 1$, $i = 1, ..., l-1$, and for every possible scale g_i in the dataset $b_1 < -\sigma \log_2 g_i < b_l$. Then, the ground truth scale distribution $P = \{p_1, p_2, ..., p_l\}$ over the integer bins $B = \{b_1, b_2, ..., b_l\}$ is defined by

$$p_i = \frac{\sum_{1 \leq j \leq n} \max\left(0, 1 - |b_i + \sigma \log_2 g_j|\right)}{\sum_{1 \leq k \leq l} \sum_{1 \leq j \leq n} \max\left(0, 1 - |b_k + \sigma \log_2 g_j|\right)} \tag{1}$$

Let $Q = \{q_1, q_2, ..., q_l\}$ denote the predicted distribution. We formulate the problem of scale prediction as minimizing Kullback-Leibler divergence (cross entropy) from

Q to P defined by

$$D(Q, P) = \sum_{1 \le i \le l} p_i \cdot (\log p_i - \log q_i) \qquad (2)$$

We now justify Eq. 1 in details. SharpMask [34] is a scale-sensitive method, which can generate correct object proposals only if the image is properly resized. For each object size, there is a narrow range of image sizes within which the object can be detected. This is where g_i comes from. The rest of Eq. 1 comes naturally.

4.2. ScaleNet architecture

To devise a model that outputs Q which minimizes Eq. 2, we propose a deep neural network called ScaleNet. This section presents the architecture of ScaleNet and discusses the motivations behind the design.

The input size of ScaleNet is 192×192 with RGB channels. Given input image I of size $w \times h$, we first resize the image to fit the input of ScaleNet I'. More specifically, we compute $d = \max(w, h)$, then resize the image such that $d = 192$. Next, we copy the resized I to the center of I', and pad I' with a constant value. I' is then fed into ResNet [14] to extract image features. Here, the fully connected layers and the last convolutional stage have been removed from ResNet. After extraction, the features from ResNet go through two 1×1 convolutional stages which serve as local fully connected layers to further process the features separately at each location on the feature map. ReLU [32] and batch normalization [17] are used in the two stages to stabilize and speed up training. At the end, a global average pooling layer [25] collects features at each location of the feature map from the two convolutional stages, then outputs scale distribution by a SoftMax operation.

The intuition is to learn the object scales at each location of the image then combine them into one image property. The global average pooling applied at the end of ScaleNet distributes this learning problem to different locations of the image. The distributed tasks can be learned separately by fully connected layers on top of each location of feature map from the last convolutional stage of ResNet. 1×1 convolutional operation then serves as a local fully connected layer to process the features. Similar to the fully connected layers of VGGNet [41], we deploy two 4096 dimension feature extractors. The main difference is that the extracted features in ScaleNet have 4096 features for each location of feature map instead of the whole image.

4.3. Connecting ScaleNet to SharpMask

For an image I, ScaleNet is able to predict a scale distribution $Q = \{q_1, ..., q_l\}$. This is a probability density function, which we denote as $q(x)$. We assume that the optimal number of scales needed by SharpMask is h (usually $h \sim 8$). To exploit Q for SharpMask, the task is to

choose a set of scales $S = \{s_1, ..., s_h\}$ to resize I as the input of SharpMask. The intuition is to densely sample scales around the scales b_i that have high probability q_i. To achieve this, we consider the cumulative distribution function of q, *i.e.*,

$$F(s) = \int_{-\infty}^{s} q(x) \, dx \qquad (3)$$

Then we sample scales in the space of $F(s)$ such that

$$F(s_i) = \frac{i}{h+1}, \text{ for } i = 1, ..., h \qquad (4)$$

Before sampling, the distribution q can be smoothed by

$$q'(x) = \frac{q(x)^\lambda}{\int q(x)^\lambda \, dx} \qquad (5)$$

where λ is the smoothing parameter.

5. Supermarket Datasets

5.1. Real-world datasets

We aim to study the importance of the scales to the existing object proposal methods; therefore, we prepared two real-world datasets, each of which focuses on one setting of object scales. The first dataset, which we call *Real-Far*, is composed of 4033 products labeled in bounding boxes. The images in this dataset were taken from a far distance with less variation in scales, thus usually having more objects within one image. On average, one image contains 58 objects. The second dataset is called *Real-Near*, which contains 3712 products annotated. For this dataset, we took the images from a near distance and the images have more variation in object scales. The images in *Real-Near* have 27 products for each on average. Two professional labelers worked on the datasets during collection. In total, we have 7745 products labeled for testing.

5.2. Synthetic dataset

Labeling images in supermarkets can be very time-consuming since there are usually 30 to 60 objects in one typical image. Although for SharpMask the number of training examples grows linearly with respect to the number of the annotated objects, ScaleNet considers one image labeled as one example, thus requiring more data for training; what's more, SharpMask is a mask-based proposal method, which needs objects annotated in object masks, making annotation much harder for humans. Our solution is to build a virtual supermarket to let models learn in this virtual environment. The training and the validation of models are all done in the virtual supermarket. The models are then tested directly on the real-world datasets without fine-tuning. By doing this, we can significantly reduce human labeling, but we need to be very careful when designing the virtual environments so that the models can transfer well to the real-world data from the synthetic data.

Figure 4: Comparison of Product Arrangements with Different Proximities. Left: an example of product arrangement result with proximity set to 0; right: an example of product arrangement result with proximity set to 1. Setting proximity to a lower value makes the arrangement look more random while setting to a higher value will get a more organized arrangement. The valid range of proximity is within 0 to 1.

Realism   The first aspect we consider is the realism of the rendered images. Although some work suggested that realism might not be critical for some vision tasks [8], it is a high priority in this paper since we do not fine-tune on the real-world data. The rendering engine we chose is Unreal Engine[2] for its flexibility of object manipulation and high rendering quality. UnrealCV [35] is used to extract the ground truth of object masks. To fully exploit the power of Unreal Engine, all the objects in the virtual supermarket are set to be static and the lighting is baked (*i.e.* pre-computed) before the game is run.

Randomness of placement   The products in a real supermarket are usually placed according to certain rules. However, since the generalizability must be taken care of when generating a virtual dataset, the randomness of placement is introduced into the rules that guide the construction of the virtual environment.

Similar to some 3D object arrangement methods [12, 44], we specify a stochastic grammar of spatial relationship between products and shelves. First, the products are initially located at a position that is not in the rendering range. Next, given a shelf that products can be placed on, the products will be moved to fill the shelf one by one. Note that similar products are usually placed together in supermarkets. Therefore, before placing the products, for a group of the products, we first find an anchor point on the shelf. Then we specify a parameter, which we call *proximity*, to denote the probability that the next product will be placed near that anchor point or will be placed randomly somewhere on the shelf. Fig. 4 demonstrate the placing arrangements with different proximities.

Product overlapping   Product arrangement must prevent overlapping. Motivated by reject sampling, we first randomly create arrangements then reject those that have overlapping products. To efficiently detect overlapping while

---

[2]https://www.unrealengine.com/

preserving concave surfaces, convex decomposition is applied to the 3D models before calculating overlapping.

Figure 5: A Zoom-In Example of the Ground Truth Extracted by UnrealCV [35] with Heavily Occluded Objects Ignored. The virtual dataset is compatible with the MS COCO dataset [26]. The visualization result shown here uses the COCO API. The occlusion threshold is set to 0.9.

Occlusion   A problem of using synthetic dataset is that all objects will be labeled, including extremely occluded objects that are usually ignored in building real-world datasets. Our solution to this problem is to calculate the ratio of occlusion for each object, then ignore the objects of occlusion under threshold μ when extracting the ground truth. To achieve this, we implement a standard rendering pipeline of vertex shader and fragment shader for computing occlusion. To gather data at high speed, we approximate the occlusion calculation by projecting the objects to the surface parallel to the shelf and calculating them only once.

Object scales   The object scales can be controlled by modifying the distance between the camera and the shelf. We set the camera to be at distance $\nu \cdot d_{max}$, where d_{max} is the distance at which the camera can exactly take in one shelf completely. Then we can modify ν to generate data with different object scales.

Lighting and material randomness   To augment the virtual dataset, lighting and materials for objects are changed

randomly during data gathering.

Summary   This section presents how the synthetic dataset is constructed with the above aspects taken into account. We develop a plugin for Unreal Engine to construct virtual supermarket stochastically by only one click. We also modify the COCO API to integrate the virtual supermarket dataset into the MS COCO dataset [26]. Fig. 5 demonstrates the visualization of the mask annotations using the COCO API with the occlusion threshold set to 0.9.

6. Implementation Details

This section presents the implementation details of ScaleNet, the object proposal system, the generation of the virtual supermarket dataset, and the data sampling strategy.

6.1. Virtual supermarket

We bought 1438 3D models[3] for products and shelves to construct the virtual supermarket. During the data collection, two parameters are manually controlled while others are drawn randomly from a uniform distribution. The two parameters are the occlusion threshold μ and the distance ratio ν. The range of μ is $\{0.9, 0.8, 0.7, 0.6, 0.5\}$, and the range of ν is $\{1, 1/1.5, 1/2, 1/2.5, 1/3\}$. Combining different μ and different ν results in 25 configurations, for each we use different product arrangements, and random lighting/material settings at each frame to generate 200 images. The above process generates 5000 synthetic images and 234690 objects labeled in total. We denote this virtual dataset as dataset *V*. We split dataset *V* into *Vtrain* and *Vval* for training and validation, respectively. The dataset *Vtrain* has 3307 images and 154238 objects while the dataset *Vval* has 1693 images and 80452 objects.

6.2. ScaleNet

We use Torch7 to build and test ScaleNet. Before training ScaleNet, the ResNet component is pre-trained on ImageNet [40]. The discretization precision σ is set to 1, while the discrete scale bins are set to $B = \{-32, -31, ..., 0, ..., 31, 32\}$. To accommodate the parameters used in SharpMask [34], D is set to $640/7$.

During training, we resize the image to fit the input of ScaleNet, and calculate the scale distribution P as the ground truth. The mean pixel calculated on ImageNet is subtracted from input image before feeding into ScaleNet. All layers are trained, including the ResNet component. We train two ScaleNet models for the supermarket datasets and the MS COCO [26] dataset, individually. We use the corresponding models when evaluating the performances on different datasets. The training dataset for ScaleNet for supermarket datasets is *COCOtrain* + *Vtrain* while the validation dataset is *COCOval* + *Vval*. For the MS COCO, the datasets

---

[3]https://www.turbosquid.com/

Methods	Real-Far	Real-Near
EdgeBox@100 [46]	0.006	0.015
Selective Search@100 [42]	0.019	0.043
DeepMask@100 [33]	0.183	0.198
SharpMask@100 [34]	0.191	0.205
DeepMask-ft@100	0.209	0.231
SharpMask-ft@100	0.224	0.249
ScaleNet+DeepMask@100	0.256	0.342
ScaleNet+DeepMask-ft@100	0.278	0.373
ScaleNet+SharpMask@100	0.269	0.361
ScaleNet+SharpMask-ft@100	**0.298**	**0.396**
EdgeBox@1000	0.203	0.324
Selective Search@1000	0.225	0.328
DeepMask@1000	0.472	0.488
SharpMask@1000	0.499	0.518
DeepMask-ft@1000	0.497	0.533
SharpMask-ft@1000	0.526	0.567
ScaleNet+DeepMask@1000	0.542	0.593
ScaleNet+DeepMask-ft@1000	0.561	0.621
ScaleNet+SharpMask@1000	0.570	0.625
ScaleNet+SharpMask-ft@1000	**0.589**	**0.651**

Table 1: The Comparison of the Average Recalls [15] of Object Proposal Methods Tested on the Real-World Supermarket Datasets *Real-Far* and *Real-Near*. The method name indicates what method is used and how many proposals are considered in computing recall rates, *e.g.*, EdgeBox@100 means EdgeBox with the number of object proposals limited to 100. Methods that have suffix *-ft* are trained on the MS COCO and the synthetic supermarket dataset.

used for training and validation include only the MS COCO itself. Here, *COCOtrain* and *COCOval* are the training and the validation set of the MS COCO, respectively. To connect ScaleNet to SharpMask, h is set to 6 for the supermarket datasets, and 10 for the MS COCO. The smoothing factor λ is set to 0.9 for the supermarket datasets, and 0.25 for the MS COCO.

6.3. Data sampling

In the original data sampling strategy adopted in both DeepMask and SharpMask, each image has the same probability for objectness score training and each category has the same probability for object mask training. Instead, we propose to train both the objectness score and object mask so that each annotation has the same probability of being sampled. Following this strategy, the performance can be slightly improved. We denote SharpMask trained in this way as SharpMask-Ours.

7. Experimental Results

7.1. Object proposal on supermarket datasets

We first present the performance of our model on the supermarket datasets while only trained on the combination of

Figure 6: Proposals Generated by Our Method ScaleNet+SharpMask-ft with Highest IoU to the Ground Truth on the Selected Real-World Supermarket Images. Top images are selected from dataset *Real-Far* while bottom images are selected from dataset *Real-Near*. Green bounding boxes are from top 100 proposals. Blue bounding boxes are from proposals ranked between 101 and 1000. Red bounding boxes are ground truth of objects not found by our method within 1000 proposals. The IoU threshold is set to 0.7.

the MS COCO training dataset and the virtual supermarket training dataset. We evaluated the methods on the dataset *Real-Near* and *Real-Far*. Qualitative results of our method are shown in Fig. 6.

Metrics The metric used to evaluate the performance of the object proposal methods is the Average Recalls (AR) [15] over 10 intersection over union thresholds from 0.5 to 0.95 with 0.05 as step length.

Methods We compare the performance of the proposed method with the top methods of proposing bounding boxes for objects: DeepMask [33], SharpMask [34], Selective Search [42], and EdgeBox [46].

transferability Table 1 demonstrates the improvements of

performances of the model trained using virtual supermarket dataset. Methods that have suffix *-ft* are trained on the MS COCO and the synthetic supermarket dataset. It's worth noting that the models trained solely on the combination of the general purpose dataset and the task specific synthetic dataset exhibit consistent improvements on the task specific real-world datasets even none of them has a look at the real-world data.

Scales Table 1 compares the different object proposal methods on the two real-world dataset *Real-Near* and *Real-Far*. Without the help of ScaleNet to narrow down the search space of scales, DeepMask and SharpMask actually have similar performances on them. Instead, our pro-
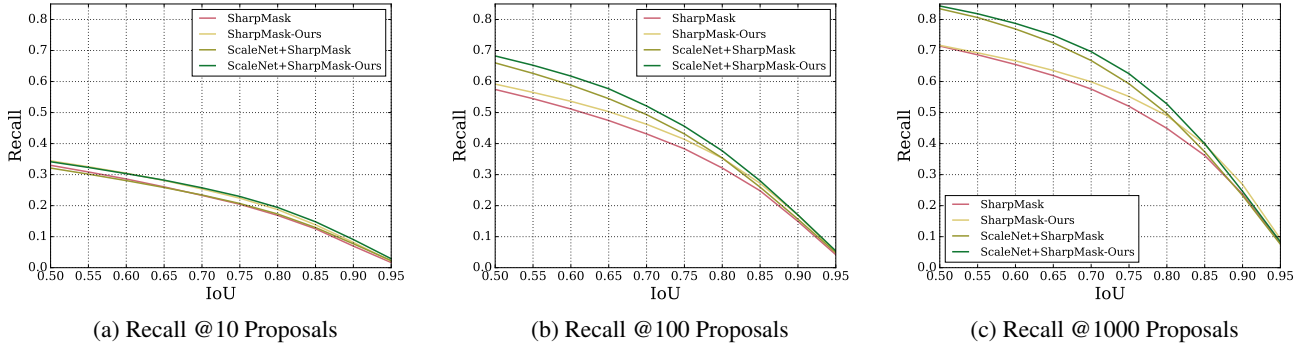
(a) Recall @10 Proposals   (b) Recall @100 Proposals   (c) Recall @1000 Proposals

Figure 7: Recall versus IoU Threshold for Different Number of Bounding Box Proposals on the MS COCO Dataset.

Methods	AR@10	AR@100	AR@1k
DeepMask-VGG [33]	0.153	0.313	0.446
DeepMaskZoom-VGG [33]	0.150	0.326	0.482
DeepMask-Res39 [34]	0.180	0.348	0.470
SharpMask [34]	0.197	0.364	0.482
SharpMaskZoom [34]	0.201	0.394	0.528
SharpMask-Ours	0.216	0.392	0.510
ScaleNet+SharpMask	0.201	0.416	0.557
ScaleNet+SharpMask-Ours	**0.220**	**0.439**	**0.578**

Table 2: Comparison of Our Framework to DeepMask [33] and SharpMask [34] on Bounding Box Object Proposals on the MS COCO validation dataset [26].

posed method exhibit stronger improvements on *Real-Near* in which the image has fewer objects, thanks to the accurate prediction by ScaleNet of the scales to resize images.

In short, Table 1 demonstrates the significant performance improvements by using our proposed framework.

7.2. Object proposal on the MS COCO dataset

Next, we evaluate our method on the MS COCO dataset. Following the evaluations done in DeepMask [33] and SharpMask [34], the recall rates are evaluated on the first 5000 images on the validation set.

Methods We compare the performance of the proposed method with the state-of-the-art methods of proposing bounding boxes for objects: DeepMask-VGG [33], DeepMaskZoom-VGG [33], DeepMask-Res39 [34], Sharp-Mask [34], SharpMaskZoom [34].

Metrics We adopt the same metrics used for evaluating performances on the supermarket datasets. The performances are evaluated when the number of proposals is limited to 10, 100 and 1000.

Results Table 2 summarizes the performance comparisons on the MS COCO dataset. Since the object scales in these natural images are not always sparse, we do not ex-

pect significant improvements as shown in the supermarket datasets. However, consistent improvements can be observed at all number of proposals. More notably, our method demonstrates stronger performance improvements compared with that between SharpMask and DeepMask.

Fig. 7 shows the additional performance plots comparing our methods with the previous state-of-the-art. Our framework improves the recall rates significantly at 1000 proposals, *e.g.*, the recall rate increases from 0.714 to 0.843 when IoU threshold is set to 0.5, and from 0.575 to 0.696 at 0.7 IoU threshold. We also observe strong performance increases at 100 proposals: the recall rate at 0.5 IoU threshold increases from 0.574 to 0.682, and from 0.431 to 0.521 at 0.7 IoU threshold.

8. Conclusion

In this paper, we study the problem of object proposal generation in supermarket images and other natural images. We introduce three supermarket datasets – two real-world datasets and one synthetic dataset. We present an innovative object proposal framework, in which the object scales are first predicted by the proposed scale prediction method ScaleNet. The experimental results demonstrate that the model trained solely on the combination of the MS COCO dataset and the synthetic supermarket dataset transfers well to the two real-world supermarket datasets. The proposed scale-aware object proposal method is evaluated on the real-world supermarket datasets and the MS COCO dataset. Our proposed method outperforms the previous state-of-the-art by a large margin on these datasets for the task of object detection in the form of bounding box.

References

[1] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 1, 2

[2] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *TIP*, 24(12):5706–5722, 2015. 1

[3] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 3

[4] K. Chang, T. Liu, H. Chen, and S. Lai. Fusing generic objectness and visual saliency for salient object detection. In *ICCV*, 2011. 1

[5] K. Chang, T. Liu, and S. Lai. From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model. In *CVPR*, 2011. 1

[6] L. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. *CoRR*, abs/1511.03339, 2015. 3

[7] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014. 1, 2

[8] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 3, 5

[9] A. Dosovitskiy and V. Koltun. Learning to act by predicting the future. *CoRR*, abs/1611.01779, 2016. 3

[10] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *CVPR*, 2014. 2

[11] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 2

[12] M. Fisher, D. Ritchie, M. Savva, T. Funkhouser, and P. Hanrahan. Example-based synthesis of 3d object arrangements. *ACM Trans. Graph.*, 31(6):135:1–135:11, Nov. 2012. 3, 5

[13] A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, and L. Van Gool. Deepproposal: Hunting objects by cascading deep convolutional layers. In *ICCV*, 2015. 2

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 4

[15] J. H. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *CoRR*, abs/1502.05082, 2015. 6, 7

[16] P. Hu and D. Ramanan. Finding tiny faces. *CoRR*, abs/1612.04402, 2016. 3

[17] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. 4

[18] Z. Jie, X. Liang, J. Feng, W. F. Lu, F. E. H. Tay, and S. Yan. Scale-aware pixelwise object proposal networks. *TIP*, 25(10):4525–4539, 2016. 2

[19] M. Johnson, K. Hofmann, T. Hutton, and D. Bignell. The malmo platform for artificial intelligence experimentation. In *IJCAI*, 2016. 3

[20] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *ECCV*, 2014. 1, 2

[21] P. Krahenbuhl and V. Koltun. Learning to propose objects. In *CVPR*, 2015. 2

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[23] W. Kuo, B. Hariharan, and J. Malik. Deepbox: Learning objectness with convolutional networks. In *ICCV*, 2015. 2

[24] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014. 1

[25] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013. 4

[26] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 1, 2, 5, 6, 8

[27] J. Liu and Y. Liu. Grasp recurring patterns from a single view. In *CVPR*, 2013. 3

[28] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *TPAMI*, 33(2):353–367, 2011. 1

[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *ECCV*, 2016. 2

[30] A. Mahendran, H. Bilen, J. F. Henriques, and A. Vedaldi. Researchdoom and cocodoom: Learning computer vision with games. *CoRR*, abs/1610.02431, 2016. 3

[31] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 3

[32] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 4

[33] P. H. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. In *NIPS*, 2015. 1, 2, 3, 6, 7, 8

[34] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *ECCV*, 2016. 1, 2, 3, 4, 6, 7, 8

[35] W. Qiu and A. L. Yuille. Unrealcv: Connecting computer vision to unreal engine. *CoRR*, abs/1609.01326, 2016. 2, 3, 5

[36] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[37] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[38] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 3

[39] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 3

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 6

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 2, 4

[42] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 1, 2, 6, 7

[43] F. Xia, P. Wang, L. Chen, and A. L. Yuille. Zoom better to see clearer: Human part segmentation with auto zoom net. *CoRR*, abs/1511.06881, 2015. 3

[44] L.-F. Yu, S.-K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. J. Osher. Make it home: Automatic optimization of furniture arrangement. *ACM Trans. Graph.*, 30(4):86:1–86:12, July 2011. 3, 5

[45] Y. Zhang, W. Qiu, Q. Chen, X. Hu, and A. L. Yuille. Unrealstereo: A synthetic dataset for analyzing stereo vision. *CoRR*, abs/1612.04647, 2016. 3

[46] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 1, 2, 6, 7