ME R-CNN: Multi-Expert R-CNN for Object Detection

Hyungtae Lee[†‡],   Sungmin Eum[†‡],   Heesung Kwon[†]

[†]U.S. Army Research Laboratory, Adelphi, MD, USA

[‡]Booz Allen Hamilton Inc., McLean, VA, USA

lee_hyungtae@bah.com   eum_sungmin@bah.com   heesung.kwon.civ@mail.mil

Abstract

We introduce Multi-Expert Region-based CNN (ME R-CNN) which is equipped with multiple experts and built on top of the R-CNN framework known to be one of the state-of-the-art object detection methods. ME R-CNN focuses in better capturing the appearance variations caused by different shapes, poses, and viewing angles. The proposed approach consists of three experts each responsible for objects with particular shapes: horizontally elongated, square-like, and vertically elongated.

On top of using selective search which provides a compact, yet effective set of region of interests (RoIs) for object detection, we augmented the set by also employing the exhaustive search for training only. Incorporating the exhaustive search can provide complementary advantages: i) it captures the multitude of neighboring RoIs missed by the selective search, and thus ii) provide significantly larger amount of training examples. We show that the ME R-CNN architecture provides considerable performance increase over the baselines on PASCAL VOC 07, 12, and MS COCO datasets.

1. Introduction

In general, object detection uses distinctive shape patterns as evidence to find the object-of-interest in an image. Object detection models are trained on these shape patterns that are commonly shown within the same object categories yet discriminative among the different categories. However, it is quite burdensome for a single model to accurately identify all the appearances since how objects are seen in the images greatly vary according to their fundamental shapes (e.g., airplane vs. person) as well as different poses and viewing angles (e.g., a person lying down vs. standing upright). Therefore, conventional object detection methods often use mixture of experts, each expert associated only with the corresponding shape patterns, in order to better capture large variations of object appearance [12, 22, 27].

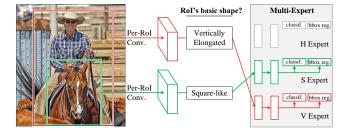In this paper, we introduce a novel CNN-based approach

Figure 1: **ME R-CNN.** ME R-CNN adopts "multi-expert" to allow different streamlines for processing different RoIs. The optimal streamline for each RoI is selected by analyzing the RoI's basic shape category.

for object detection, referred to as ME R-CNN, which adopts multiple experts. The ME R-CNN inherits the architecture of the region-based CNN (R-CNN) [7, 13, 14, 16, 28] which uses a single stream pipeline for processing each region-of-interest (RoI). However, unlike these approaches, the ME R-CNN is equipped with multiple stream pipelines, where one of the pipelines becomes an "expert" for processing certain type of RoIs.

Within the ME R-CNN architecture, the regions-of-interest (RoIs) are first categorized into three fundamental object shape categories according to their aspect ratios: horizontally elongated, square-like, and vertically elongated. Then each RoI is processed by the appropriate expert which specializes in handling the corresponding shape category. Each expert is constructed by connecting several fully connected layers, and all the experts are preceded by a single RoI pooling layer and a set of shared convolutional layers. Figure 1 depicts the conceptual mechanism of the ME R-CNN.

We also focused on augmenting the training data for learning ME R-CNN. It is a notion widely agreed upon, that more training data serves as a better source to learn a model with an enhanced accuracy. One way is to augment the training data by combining multiple datasets, for instance, using both PASCAL VOC and Microsoft COCO for training. Instead of bringing more examples from more
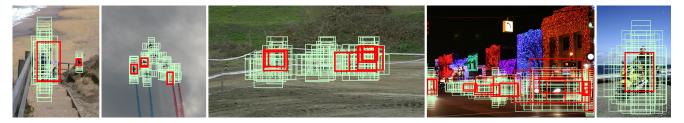
Figure 2: **Complementary roles of the selective and exhaustive search.** RoIs by using the selective (red) and the exhaustive (green) search are depicted.

datasets, we use a method to acquire more training examples by generating more RoIs for each image, equivalent to providing more training examples within the same dataset.

The R-CNN and several of its descendants [13, 14, 17], which we share the structure with, use approximately 2000 RoIs per image generated by selective search. For ME R-CNN, we have employed a multi-scale sliding window searching strategy (exhaustive search) along with the selective search in order to acquire an augmented set of RoIs. The augmented set of RoIs provided by the exhaustive search has its own complementary roles in training ME R-CNN as follows.

1. The exhaustive search can localize a multitude of closely neighboring regions of objects which carries valuable visual contents for training, most of which are missed by the selective search.

2. An incomparably large number of regions can be acquired as positive examples as well as negative examples, providing rich source of information to train the CNN.

Figure 2 shows these complementary roles of the exhaustive search. Note that, for testing, we only use a sparse set of RoIs generated by the selective search to maintain the efficiency of the testing process.

To show that the proposed architecture can be effectively implanted into various types of object detection CNN architectures, we have built ME R-CNN on top of two widely used object detection CNN architectures which are Fast R-CNN and Faster R-CNN. For both cases, we verified that ME R-CNN can provide constant performance boost over the baseline approaches in PASCAL VOC 07, 12, and MS COCO datasets.

The contributions of the proposed ME R-CNN can be summarized as follows.

1. Introduction of ME R-CNN adopting multiple experts to better capture variations of the object appearance in terms of aspect ratio.

2. Exploitation of exhaustive search to augment the RoI set for training.

3. Considerable performance boost over the baselines on benchmark datasets.

2. Related Works

Object Detection. Object detection is one of the most challenging tasks in computer vision. Prior to the introduction of CNNs, non-CNN based object detection approaches, such as HOG-SVM, DPM, etc., were widely used for classifying RoIs into corresponding object categories [8, 12, 22, 27]. Within the past several years, multiple attempts have been made to use CNNs for object detection. Prominent methods among them are R-CNN [14] and its descendants [7, 13, 16, 17, 28] that provided the state-of-the-art performance.

Although having achieved the top-notch performance, R-CNNs have not yet exploited some of the effective strategies which conventional object detection methods commonly use for boosting the performance. While the R-CNNs rely on heuristics to select hard negative examples, Shrivastava et al. [30] and Wang et al. [36] used the online hard example mining (OHEM) to automatically select hard examples with high optimization loss in every iteration of training. These approaches were motivated by the offline bootstrapping idea for training a classical object detection method [8].

Motivated by their successful practice, two conventional performance boosting strategies have been used in our method which enhance the network in two different aspects: i) introducing a multi-expert strategy associated with shape categories and ii) using a complementary combination of the exhaustive and selective search. We incorporate the two components into the Fast/Faster R-CNN architectures to construct ME R-CNN.

Mixture-of-Experts Models. Multiple experts embedded in the proposed ME R-CNN is based on the concept of mixture-of-experts models. The mixture-of-experts model is used to better estimate the probability distribution of a composite data with large variation (e.g., Gaussian mixture model [37]). In the image domain, object appearances can also show large variations according to their shapes, poses, and viewing angles. Felzenswalb et al. [12] nicely illustrates the importance of using a mixture of models

by presenting two models, each of which captures the appearance of the front and the side view of a bicycle. Accordingly, many recent approaches [3, 12, 29] have shown that using the mixture-of-experts model for advanced object detection is very effective. However, to date to the best of our knowledge, none of the CNN-based object detection methods have incorporated the mixture-of-experts model into their architectures.

RoI Generation. One of the conventional ways to generate RoIs is to use multi-scale sliding windows [2, 8, 12, 15, 22, 27, 35] which can be considered as a 'dense' search. To avoid impractical computational complexity, the search space is confined to a regular grid and a fixed set of scales and aspect ratios. The branch and bound strategy was found to reduce the search space even more by using optimal windows within an image [23, 34].

Instead of going 'dense', some methods employed relatively 'sparse' searching approaches by introducing the concept of objectness. Lampert et al. [26] used an objectness quality function to discard sub-search spaces whose objectness scores are under a certain threshold, where object detector becomes an objectness quality function. Instead of using the object detector, Alexe et al. [1] introduces a generic objectness measure, to estimate how likely it is for a region to contain object of any category using saliency, color contrast, edge density, and boundary information. Several more approaches [4, 5, 19, 20, 33, 38] to generate RoIs based on objectness characteristics have been introduced afterwards. Recently, Ren et al.[28] introduced a region proposal network (RPN) incorporated into the CNN which also generates RoIs based on the objectness.

To garner the advantages from both of the searching approaches, ME R-CNN utilizes multi-scale sliding window (exhaustive search) along with the objectness-based (selective search) RoI generators.

Going Wider with CNN. One of the major innovations introduced into ME R-CNN is that the network has expanded in width, where the network width refers to the number of nodes in each layer. This is to equip the network with multiple number of specialized experts to better capture variations of object appearance. There have already been several attempts where the width of CNN architecture was expanded. Krizhevsky et al. [25] splits each layer into two parallel layers in order to fully use two GPUs in a parallel fashion. Szegedy et al. [32] uses the inception module which employs multiple parallel layers in order to make use of dense sets of different sized convolutional filters. Several other approaches [6, 10, 24] also introduced widened networks for the task of co-learning multiple tasks in a single framework.

3. The Proposed Approach

3.1. Architecture

As ME R-CNN shares the structural backbone of the Fast R-CNN [13] architecture, we briefly introduce how Fast R-CNN works to help the readers better understand the proposed architecture. Fast R-CNN consists of the per-image convolutional network and the per-RoI network. The per-image convolutional network takes an input image and computes the convolutional per-image feature map which is the output of the last convolutional layer. Meanwhile, a sparse set of RoIs is generated by the selective search. For each RoI, the per-RoI network generates a per-RoI feature map by cropping the corresponding RoI from the per-image feature map. This is then max pooled to have a fixed size output. The output size is set to match the input size of the first fully-connected layer of the predefined CNN (e.g., 7×7 for VGG16 [31]). The per-RoI feature map is then fed into a single stream of fully connected layers which is followed by two sibling fully connected layers. Two sibling layers are for object classification and bounding box regression.

In ME R-CNN, we remodel the two major modules of Fast R-CNN: RoI generation module and the per-RoI network module. In terms of RoI generation, ME R-CNN acquires a combined set of RoIs generated by both the selective and exhaustive search. Instead of using a single stream per-RoI network, it adopts per-RoI multi-expert network which consists of three streams each of which is called an 'expert.' Each expert has the same form of the fully connected layers of Fast R-CNN. Appropriate expert assignment is carried out by matching the shape of the given RoI to one of the predefined distinctive shape categories: horizontally elongated, square-like, or vertically elongated. For each RoI, its associated per-RoI feature map, which is the output of the RoI pooling layer, is fed into the assigned expert. Figure 3 illustrates the proposed ME R-CNN structure.

RoI Augmentation. As ME R-CNN contains larger number of parameters compared to Fast R-CNN, more examples are required to train the network. Therefore, along with the relatively sparse set of RoIs generated by the selective search [33], the proposed network also intakes a dense set of RoIs produced by multi-scale sliding windows in an exhaustive manner.

The exhaustive search looks for regions with various aspect ratios. We have used the height and width ratios of [4:1, 2:1, 1:1, 1:2, 1:4] in our experiments which are intended to cover square-like objects as well as elongated objects. For a particular aspect ratio r, the multi-scale search begins with the initial window size of width (w) and height (h), such that $w/h = r$ and $\max{(w/W, h/H)} = 1$, where W and H are the width and the height of the input
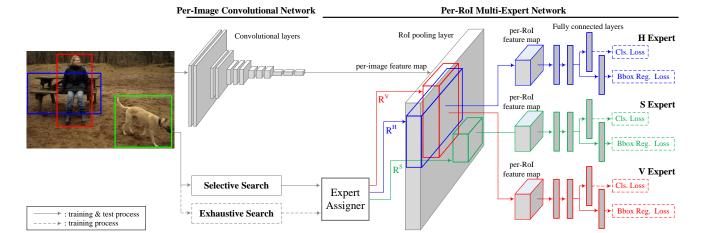
Figure 3: **ME R-CNN architecture.** Per-image convolutional network computes per-image feature map while the selective and the exhaustive search generates the RoIs (R). Then, an appropriate expert is assigned to each R. (R^H, R^S, and R^V for H, S, and V expert, respectively.) The associated per-RoI feature maps which are the outputs of the RoI pooling layer are fed into the assigned experts. Exhaustive search is only used for the training process.

image, respectively. The stride is set as $0.25 \times \min(w, h)$. After sliding the window with a particular scale over the entire image, window size is divided by $2^{(1/4)}$ and the window is slid again. This is to decrease the size of the window by half for every four iterations. The process is iterated until $\min(h, w)$ is less than 25 pixels.

Expert Assignment. After RoIs are generated, each RoI is fed into one of the three experts according to its shape category. Each RoI is labeled with a shape category chosen among horizontally elongated (**H**), square-like (**S**), or vertically elongated (**V**) according to its aspect ratio. We express the aspect ratios for the RoIs in a logarithmic form (θ) as:

$$\theta = \log_2(w/h), \tag{1}$$

where w and h are the width and the height of an RoI, respectively.

For training, we assigned all RoIs to **H** category when θ is equal or larger than 0. When θ is not greater than +1 and not less than -1, **S** category is assigned. Lastly, when θ is equal or smaller than 0, **V** is chosen. Note that, under this RoI assignment criteria, RoIs can be categorized into more than one categories. For training, this is done to have multiple experts responsible for the RoIs which can be shared across the different categories.

To enforce a RoI to be assigned to only one expert for testing, expert assignment criteria are set to have non-overlapping regions and defined as:

$$\text{RoI's shape category} = \begin{cases} \mathbf{H} & \text{if } \theta > 0.5 \\ \mathbf{V} & \text{if } \theta < -0.5 \\ \mathbf{S} & \text{otherwise,} \end{cases}$$

For instance, according to this rule, all RoIs whose aspect ratios are closer to 2:1 than 1:1 or 1:2, are assigned to **H** expert, where the ratios indicate $w : h$.

3.2. Learning the Network

The network whose weights are denoted as W is optimized by minimizing the loss function $L(W)$ which is a sum of a regularization function $R(W)$ and three pairs of loss functions, each pair being connected to one of the experts in the network. For each expert e, a softmax loss $L_{softmax}$ and L1 smooth loss L_{smooth} are used for object classification and bounding box regression, respectively. (Details of two loss functions are described in [13].) The loss function is formularized as follows:

$$L(W) = R(W) + \sum_{e \in \{\mathbf{V}, \mathbf{H}, \mathbf{S}\}} L_{softmax}^{(e)}(W) + L_{smooth}^{(e)}(W). \tag{2}$$

As in Fast R-CNN, our network is trained using stochastic gradient descent (SGD) with three batches of 128.

Multi-batch Preparation. Three batches are prepared for every iteration to optimize the three experts. Each batch is built from two images, and each image contributes 64 randomly chosen RoIs. For each expert, only the RoIs that match its associated shape category are selected for training. Each RoI is labeled as a positive or negative example according to an intersection over union (IoU) overlap criteria between the RoI and the groundtruth bounding box. The RoIs having IoU overlap equal to or bigger than 0.5 are labeled as positive examples and the ones with IoU between 0.1 and 0.5 are labeled as negative.

For each batch, the ratio between the number of positive and negative examples is fixed as 1:3.

Finetuning. The proposed network is finetuned from the image classification CNN pretrained over a large scale ImageNet dataset [9]. All layers of the pretrained network, except the last fully connected layer, are used to set the initial weights of the new network. Instead of the last fully connected layer, two sibling fully connected layers (classification layer and bounding box regression layer) are appended at the end of each expert. The classification layer weights are initialized by randomly selecting them according to Gaussian distribution with the mean of 0 and the standard deviation of 0.01. For the bounding box regression layer, we initialized the weights randomly selected from Gaussian distribution with the mean and the standard deviation of 0 and 0.001, respectively.

Our network does not finetune the first two convolutional layers because those layers tend to capture general image characteristics while other layers are more correlated with the training purpose, which is to perform the object detection. The single stream of fully connected layers of the pretrained CNN is used to finetune all three streams of fully connected layers in ME R-CNN. When finetuning the shared convolutional layers, we multiply 1/3 to the base learning rate because optimizing these layers are affected by all three streams for each training iteration when backpropagation takes place.

3.3. Object Detection

In testing, ME R-CNN outputs three sets of detection results, bounding boxes and their scores, from three different experts. The bounding boxes are refined by incorporating the output of bounding box regression layers. We combine these three sets of detection results and apply non-maximum suppression (NMS) with overlap criteria of 0.3 for each object category.

4. Analyses of ME R-CNN

We present experimental results which demonstrate the effectiveness of adopting the multi-experts into the architecture and the RoI augmentation using the exhaustive search.

4.1. Experimental Setup

For all the analyses in Section 4, we use the ME R-CNN built on top of Fast R-CNN with VGG16 [31]. All the experiments are conducted on PASCAL VOC07 [11]. According to VOC's general protocol for object detection, trainval and test sets are used for training and testing the network, respectively. We train all methods with 80k iterations. A base learning rate is set as 0.001 and dropped to 0.0001 after 60k iterations.

Table 1: Effects of two major modules of ME R-CNN (SS and ES are the selective search and the exhaustive search, respectively.)

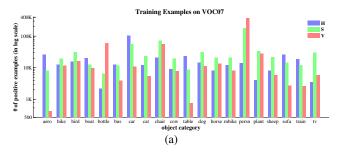| Method	RoIs	No. of Experts	mAP (%)
Fast R-CNN [13]	SS	1	66.9
Fast R-CNN w/ Augmented RoIs	SS+ES	1	68.1
ME R-CNN w/o Augmented RoIs	SS	3	68.3
ME R-CNN	SS+ES	3	**69.0**

4.2. Multiple Experts

Multiple Experts vs. Single Expert. As Table 1 shows, ME R-CNN outperforms the single expert network (Fast R-CNN w/ Augmented RoIs) by 0.9%. Note that these methods use the RoIs generated by both the exhaustive and the selective search. Even without the augmented set of RoIs, ME R-CNN outperforms Fast R-CNN by 1.4%.

Layer Sharing in Multi-Experts. Multi-experts in ME R-CNN contain three times more number of fully connected layers compared to Fast R-CNN, which brings up the memory efficiency issues. Table 2 shows the detection performances acquired by employing three different structural variations on which fully connected layers are shared across the multiple experts. We observe that sharing fc6 layer across the experts is a reasonable compromise between the memory efficiency and the detection performance. Similar to optimizing the convolutional layers, shared fully connected layers are optimized by multiplying 1/3 to the base learning rate.

Table 2: Effects of layer sharing across multi-experts

| | Fast R-CNN | layers shared in ME R-CNN | | |
		none	fc6	fc6 & fc7
mAP (%)	66.9	68.9	**69.0**	68.6

Comparison among Three Experts. Table 3 shows the detection performance achieved by separate experts. Each of the result is obtained by employing only one of the three experts into the network. This is to analyze the comprehensive capability of each expert regardless of the RoI shape category. The **S** expert (67.1%) performs better than the **H** and **V** experts, which is comparable to the performance obtained by Fast R-CNN with augmented RoIs (68.1%). The **S** expert not only covers the square-like objects, but it seems to have the ability to detect the object instances that fall into
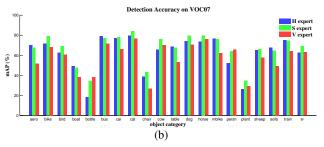
Figure 4: **Comparison of Multiple Experts.** (a) The number of training examples to be used for training three experts. (b) Detection accuracy achieved by three experts.

Table 3: Detection accuracy of multiple experts

Expert	**H**	**S**	**V**
mAP (%)	61.4	67.1	58.7

the other shapes (**H** or **V**). When these three experts are used in a unified network setting of ME R-CNN, the performance is boosted up to 69.0% (Table 1).

Figure 4a shows the distribution of training examples of different shape categories (i.e., **H**, **S**, and **V**) for each object category. Figure 4b depicts the detection accuracies achieved by different experts for each object category. For most object categories, the accuracies for **H**, **S**, and **V** follow the distribution trend shown in 4a. That is, when an object category mostly contains examples with certain shape, the best detection accuracy was obtained by the expert responsible for that shape. For instance, **H** expert performs the best in detecting horizontally elongated objects such as *aeroplane*, *car*, and *sofa* while **V** expert shows the best performance in *bottle* and *person* object categories which mostly show vertically elongated shapes.

4.3. RoI Augmentation

Effects of RoI Augmentation. In Table 1, we show that the performance of the original Fast R-CNN can be increased by 1.2% just by using the combined RoIs (the exhaustive and selective search) instead of using the RoIs generated by the selective search only. Using the combined RoIs for ME R-CNN also brings a boosted performance by 0.7% than
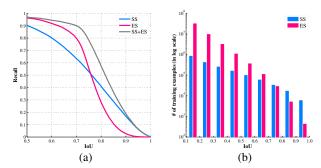

Figure 5: **Exhaustive search vs. selective search.** (a) Recall achieved by RoIs whose IoU overlaps with groundtruth are over varying thresholds. (b) The histogram of RoIs w.r.t. their IoU overlaps with groundtruth.

Table 4: **Computation time** for training and testing Fast R-CNN and ME R-CNN (using one Nvidia Titan XP).

	Fast R-CNN	ME R-CNN
train time (hr)	3.3	5.6
test time (sec/image)	0.060	0.068

Table 5: **Minibatch iterations and step sizes** for different architectures and trainset. † We follow the 4-step alternating training approach [28] when using Faster R-CNN as backbone.

backbone	train set	ME R-CNN			
Fast R-CNN	07	80k/60k			
	12	80k/60k			
	07+12	200k/150k			
	07++12	240k/180k			

backbone	train set	RPN	ME	RPN	ME
Faster R-CNN†	07	80k/60k	40k/30k	80k/60k	40k/30k
	12	80k/60k	40k/30k	80k/60k	40k/30k
	07+12	200k/150k	100k/75k	200k/150k	100k/75k
	07++12	240k/180k	120k/90k	240k/180k	120k/90k
	COCO train	320k/240k	320k/240k	320k/240k	320k/240k

the case when only the selective search is used.

Exhaustive Search vs. Selective Search. The exhaustive search has two complementary roles compared to the selective search: (i) it provides RoIs which capture objects missed by the selective search, and (ii) it can provide more positive and negative training examples.

Figure 5a shows the recall achieved by the RoIs whose IoU overlaps with the groundtruth are over varying thresholds on PASCAL VOC07 dataset [11]. The graph indicates that when we set the IoU threshold conservatively, the selective search provides better recall over the exhaustive search. On the other hand, when considering the positive example selection criteria (i.e., at 0.5 IoU threshold), the exhaus-

Table 6: **VOC 2007 test** detection average precision. All methods use VGG16. Training set key: **07**: VOC07 trainval, **07+12**: union of VOC07 trainval and VOC12 trainval.

method	train set	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
Fast R-CNN [13]	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
ME R-CNN	07	**69.0**	70.6	78.9	68.2	55.8	44.3	80.9	78.2	84.6	44.4	76.5	70.4	80.6	81.5	76.6	70.8	35.1	66.4	69.9	76.8	69.5
Faster R-CNN [28]	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
ME R-CNN	07	**70.4**	69.4	78.0	68.0	58.3	51.3	77.4	80.4	84.9	52.5	78.4	67.1	80.8	83.5	74.4	76.8	38.4	71.1	66.5	76.9	74.5
Fast R-CNN [13]	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
ME R-CNN	07+12	**72.2**	**78.1**	78.9	69.4	61.3	44.5	84.8	81.7	87.6	50.7	80.1	**70.6**	85.8	**84.8**	78.7	72.3	35.0	71.9	**75.3**	79.9	72.7
Faster R-CNN [28]	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
ME R-CNN	07+12	**75.8**	77.2	**79.7**	**76.3**	**67.0**	**60.3**	**86.0**	**87.1**	**88.6**	**58.3**	**83.8**	70.3	**86.4**	84.7	**78.4**	**78.4**	**45.1**	**76.0**	73.8	**83.6**	**74.6**

Table 7: **VOC 2007 test** detection average precision. All methods use ResNet-101.

method	train set	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
Faster R-CNN [18]	07+12	76.4	79.8	80.7	76.2	68.3	55.9	85.1	85.3	89.8	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	**85.3**	72.0
ME R-CNN	07+12	**78.7**	**81.2**	**81.9**	**78.0**	**71.8**	**65.0**	**86.0**	**87.5**	**91.3**	**61.0**	**89.2**	**69.9**	**88.4**	**90.1**	**83.9**	**81.4**	**45.2**	**81.0**	**81.7**	85.3	**73.9**

tive search achieves better recall by 7.4% than the selective search. This supports the first complementary role of the exhaustive search. Based on this observation, we use the combined set of RoIs generated by both searching strategies.

Figure 5b show a histogram of RoIs with respect to their IoU overlaps with the groundtruth. In the entire training examples, the exhaustive search generates significantly larger number of RoIs than the selective search. This verifies the second role of the exhaustive search.

4.4. Computational Cost

To analyze the computational overhead of exploiting "multiple experts", we compare the train/test time of ME R-CNN with the Fast R-CNN as shown in Table 4. Although ME R-CNN requires more time than the Fast R-CNN for training, using multi-experts brings almost no overhead in terms of test time.

5. Evaluation on PASCAL VOC and MS COCO

5.1. Experimental Setup

We use either VGG16 [31] or ResNet-101 [18] as a predefined CNN for all experiments. As the backbone architectures for ME R-CNN, Fast R-CNN or Faster R-CNN have been used. For all evaluations in Section 5, we use single-scale training/testing as in [13], by setting the shorter side of the images to be 600 pixels.

For all the methods we have tested, we used stochastic gradient descent with a base learning rate of 0.001 (0.003 when using MS COCO) and the weight decay of 0.1. As reported in Table 5, the minibatch iterations and the step sizes were varied for the two different ME R-CNN architectures and variations of the train dataset.

When using Faster R-CNN as the backbone of the ME R-CNN, we do not use our exhaustive search strategy since the region proposal network (RPN) embedded into the Faster R-CNN shares the similar concept of generating the RoIs in an exhaustive manner. We have carried out all the experiments on Caffe framework [21] with a Titan XP GPU.

Adopting Multi-Expert into ResNet-101. For object detection, He et al. [18] assigns the last 10 convolutional layers of ResNet-101 to function as the per-RoI network. We only use the last 6 convolutional layers as the per-RoI multi-expert network and insert the first 4 convolutional layers into the per-image convolutional network due to the GPU memory limitation. We also reduce the batch size from 128 to 64 for training.

5.2. PASCAL VOC 07 and 12 Results

Table 6 shows that, on VOC07, ME R-CNN provides improved detection accuracy in mAP than Fast/Faster R-CNN when using VOC07 trainval set for training (69.0% vs. 66.9% and 70.4% vs. 69.9%, respectively). When using **07+12**, ME R-CNN outperforms both Fast R-CNN and Faster R-CNN by 2.2% and 2.6%, respectively (72.2% vs. 70.0% and 75.8% vs. 73.2%). VOC12 results are shown in Table 8 where we observe consistent performance

Table 8: **VOC 2012 test** detection average precision. All methods use VGG16. Training set key: **12**: VOC12 trainval, **07++12**: union of VOC07 trainval, VOC07 test, and VOC12 trainval.

method	train set	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
Fast R-CNN [13]	12	65.7	80.3	74.7	66.9	46.9	37.7	73.9	68.6	87.7	41.7	71.1	51.1	86.0	77.8	79.8	69.8	32.1	65.5	63.8	76.4	61.7
ME R-CNN[1]	12	**67.8**	82.6	76.4	69.9	50.3	41.8	75.5	71.1	87.0	42.0	74.3	56.0	86.3	81.5	78.9	72.4	34.1	68.5	62.6	79.6	64.7
Faster R-CNN [28]	12	67.0	82.3	76.4	71.0	48.4	45.2	72.1	72.3	87.3	42.2	73.7	50.0	86.8	78.7	78.4	77.4	34.5	70.1	57.1	77.1	58.9
ME R-CNN[2]	12	**69.2**	81.2	75.7	71.2	51.1	47.8	73.3	74.6	88.1	46.9	76.4	52.9	87.1	81.7	81.4	78.8	38.4	72.9	60.0	78.4	66.9
Fast R-CNN [13]	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	**89.3**	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
ME R-CNN[3]	07++12	**70.7**	84.0	79.8	72.4	54.9	43.3	78.4	74.7	**89.3**	46.6	76.1	**60.6**	**87.8**	**83.6**	82.1	74.8	39.4	70.6	**65.7**	**82.5**	67.9
Faster R-CNN [28]	07++12	70.4	84.9	79.8	**74.3**	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
ME R-CNN[4]	07++12	**73.3**	**85.4**	**80.7**	74.0	**58.3**	**55.0**	**79.7**	**78.5**	88.6	**52.9**	**78.2**	57.8	87.7	83.3	**83.7**	**81.9**	**50.6**	**74.8**	62.4	81.8	**69.8**

Table 9: **VOC 2012 test** detection average precision. All methods use ResNet-101.

method	train set	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
Faster R-CNN [18]	07++12	73.8	86.5	81.6	**77.2**	58.0	51.0	78.6	76.6	**93.2**	48.6	**80.4**	59.0	**92.1**	**85.3**	84.8	80.7	48.1	77.3	66.5	**84.7**	65.6
ME R-CNN[5]	07++12	**76.1**	**87.1**	**82.7**	76.3	**62.5**	**62.6**	**81.7**	**80.8**	90.6	**54.8**	79.1	**63.1**	89.6	84.4	**85.4**	**84.1**	**55.0**	**77.9**	**67.1**	84.3	**71.9**

boost for ME R-CNN. In both cases of VOC12 trainval and **07++12**, ME R-CNN outperforms both Fast/Faster R-CNN by at least 2.1% mAP (2.9% at most).

In table 7 and 9, ME R-CNN shows a consistent performance boost when compared with ResNet-101 with Faster R-CNN on both VOC07 (78.7% vs. 76.4%) and VOC12 (76.1% vs. 73.8%). For this result, ME R-CNN was built on top of the ResNet-101 with Faster R-CNN architecture for fair comparison, also showing that the proposed architecture can effectively be combined with various types of object detection CNNs.

5.3. MS COCO Results

We evaluate ME R-CNN on MS COCO 2014 val dataset and show the results in Table 10. In this experiment, all methods are trained on MS COCO 2014 train dataset. We compare ME R-CNN with the Faster R-CNN using two different standard metrics, which are mAP@.5 (PASCAL VOC metric) and mAP@[.5,.95] (MS COCO metric). The MS COCO metric (mAP@[.5,.95]) indicates the mAPs averaged for IoU$\in$[0.5:0.05:0.95]. For both metrics, the ME R-CNN gained consistent performance gain over the Faster R-CNN.

6. Conclusion

We introduced ME R-CNN which uses multiple experts in place of a conventional single classifier incorporated in CNN-based object detection architectures. Compared to the single model, multiple experts is known to better capture

Table 10: **MS COCO 2014 val** detection average precision. All methods use VGG16.

	train set	test set	mAP@.5	mAP@[.5,.95]
Faster R-CNN [28]	14 train	14 val	41.5	21.2
ME R-CNN	14 train	14 val	**43.0**	**22.8**

variations in basic shape categories as well as object appearance caused by different poses and viewing angles. We categorized given regions-of-interest (RoIs) into three predefined distinctive shape categories: horizontally elongated, square-like, and vertically elongated. Then an appropriate expert is assigned to each RoI according to the shape category of the RoI.

To provide an augmented set of RoIs, we use two methods: the selective search and the exhaustive search. While the selective search produces a sparse set of RoIs, which results in reducing computational complexity for object detection, the exhaustive search provides two complementary roles: 1) the exhaustive search is able to search regions missed by the selective search and 2) provides an incomparably large number of RoIs. For testing, we only use a sparse set of RoIs generated by the selective search to maintain the efficiency of the testing process. With benefits of these two major modules, ME R-CNN proves its effectiveness in enhancing the detection accuracy in PASCAL VOC 07, 12, and MS COCO datasets over the baseline methods.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010. 3

[2] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 3

[3] E. Bernstein and Y. Amit. Part-based statistical models for object classification and detection. In *CVPR*, 2015. 3

[4] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *TPAMI*, 2012. 3

[5] N. Chavali, H. Agrawal, A. Mahendru, and D. Batra. Object-proposal evaluation protocol is 'gameable'. In *CVPR*, 2016. 3

[6] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 3

[7] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. 1, 2

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2, 3

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[10] S. Eum, H. Lee, H. Kwon, and D. Doermann. IOD-CNN: Integrating object detection networks for event recognition. In *ICIP*, 2017. 3

[11] M. Everingham, L. Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 2015. 5, 6

[12] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2010. 1, 2, 3

[13] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 1, 2, 3, 4, 5, 7, 8

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2

[15] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *CVPR*, 2009. 3

[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 2

[17] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 2

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7, 8

[19] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *TPAMI*, 2016. 3

[20] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In *BMVC*, 2014. 3

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACMMM*, 2014. 7

[22] F. Khan, R. Anwer, J. van de Weijer, A. Bagdanov, M. Vanrell, and A. Lopez. Color attributes for object detection. In *CVPR*, 2012. 1, 2, 3

[23] I. Kokkinos. Rapid deformable object detection using dual-tree branch-and-bound. In *NIPS*, 2011. 3

[24] I. Kokkinos. UberNet : Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017. 3

[25] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 3

[26] C. Lampert, M. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *TPAMI*, 2009. 3

[27] T. Malisiewicz, A. Gupta, and A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *ICCV*, 2011. 1, 2, 3

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 2, 3, 6, 7, 8

[29] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *CVPR*, 2000. 3

[30] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016. 2

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3, 5, 7

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 3

[33] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 2013. 3

[34] S. Vijayanarasimhan and K. Grauman. Efficient region search for object detection. In *CVPR*, 2011. 3

[35] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 3

[36] X. Wang, A. Shrivastava, and A. Gupta. A-Fast-RCNN: Hard positive generation via adversary for object detection. In *CVPR*, 2017. 2

[37] K. Yi, K. Yun, S. Kim, H. Chang, and J. Choi. Detection of moving objects with non-stationary cameras in 5.8ms: Bringing motion detection to your mobile device. In *CVPR Workshop*, 2013. 2

[38] L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 3