Inter-dependent CNNs for Joint Scene and Object Recognition

Jawadul Hasan Bappy and Amit K. Roy-Chowdhury
University of California, Riverside, CA 92521
Email:mbapp001@ucr.edu, amitrc@ece.ucr.edu

***Abstract*—In this paper, we consider two inter-dependent deep networks, where one network taps into the other, to perform two challenging cognitive vision tasks - scene classification and object recognition jointly. Recently, convolutional neural networks have shown promising results in each of these tasks. However, as scene and objects are interrelated, the performance of both of these recognition tasks can be further improved by exploiting dependencies between scene and object deep networks. The advantages of considering the inter-dependency between these networks are the following: 1. improvement of accuracy in both scene and object classification, and 2. significant reduction of computational cost in object detection. In order to formulate our framework, we employ two convolutional neural networks (CNNs), scene-CNN and object-CNN. We utilize scene-CNN to generate object proposals which indicate the probable object locations in an image. Object proposals found in the process are semantically relevant to the object. More importantly, the number of object proposals is fewer in amount when compared to other existing methods which reduces the computational cost significantly. Thereafter, in scene classification, we train three hidden layers in order to combine the global (image as a whole) and local features (object information in an image). Features extracted from CNN architecture along with the features processed from object-CNN are combined to perform efficient classification. We perform rigorous experiments on five datasets to demonstrate that our proposed framework outperforms other state-of-the-art methods in classifying scenes as well as recognizing objects.**

I. Introduction

Scene classification is a challenging problem due to the severe differences in intra-class and inter-class scene categories [1]. Most of the feature-based object recognition algorithms perform poorly in the face of variability of illumination, deformation, background clutter and occlusion. In recent years, the study of deep learning has been a growing interest due to its superior performance in several recognition tasks, for instance, object detection [2], and scene classification [3]. One of the common architectures used in deep learning is convolutional neural network (CNN) to perform aforementioned tasks. In this paper, we consider two inter-dependent neural networks - where one network taps into the other - to perform *joint scene and object recognition*.

In computer vision, most of the existing approaches focus on individual classification of scenes or objects [3], [2], [4]. These approaches perform feature extraction, then classification. However, there are certain objects that co-occur in a scene. In this circumstance, it can be very useful to represent the inter-relationship between scenes and objects in order to classify both. In [5], [6], context-based approaches are presented using graphical model where inter-relationships between scene and objects are taken into account to recognize them. However,

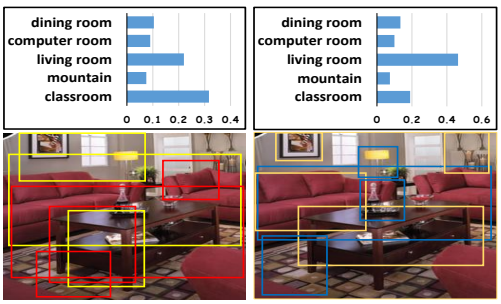


Fig. 1. In this figure, we compare recognition performance with and without considering inter-dependency between scene and object CNNs. In top row, left figure shows the scene prediction of an image using [3] without having CNN inter-link and right side is the prediction of our approach when the interdependency of CNNs is utilized. In the bottom row, our region proposal technique exploiting scene-CNN (right side) provides more accurate candidate windows with small number of proposals (~ 250) whereas selective search [7] (left side) used in current state-of-the-art methods, provides proposal windows (~ 2400) without having tight bounding boxes around the objects. Please note that only few of the proposals are shown.

the performance of these approaches highly depends on classification probabilities and detection scores obtained from scene classifiers and object detectors respectively. Current state-of-the-art [3], [2], [4] methods show that CNN based visual recognition task demonstrates outstanding performance in terms of accuracy. So, we pose a question, '*can CNN based scene and object recognition benefit each other by exploiting the inter-connections between them in order to improve performance?*' The answer to this question is investigated in this paper.

The current best-performing detectors are based on the technique of finding region proposals to localize objects. For instance, R-CNN [2] uses [7] to identify a large number of regions from an image which are then considered to perform object classification. In [8], it is observed that through the activation of receptive fields of a convolutional layer, semantic regions can be localized which is then used in object recognition. These semantic regions are very useful as they are related to objects in an image. Furthermore, as objects comprise a scene, detection scores from the detectors can also be useful to identify robust features to classify a scene. Fig. 1 represents the motivation of this work.

Towards this goal, we use two CNN architectures that mutually interact to predict both scene and object labels. We call these two deep networks - scene-CNN (S-CNN) and Object-CNN (O-CNN). The inter-dependence between S-CNN and O-CNN exploits the fact that one network taps into a layer of the other in order to perform efficient classification and vice versa. S-CNN is used to generate the semantic regions for

object proposals that are taken as input to the O-CNN to detect objects. The information flow from S-CNN to O-CNN helps us to build better object detector. More importantly, the whole framework performs very efficiently in terms of computational time ($10 - 240\times$ faster) when compared to the R-CNN [2]. Similarly, flow from object to S-CNN helps to build informative features to improve scene classification.

Framework Overview and Main Contributions. In this paper, our goal is to design a *bidirectional information flow* framework for jointly classifying scenes and recognizing objects. In our joint scene and object classification model, we have two CNNs, S-CNN and O-CNN. Our framework is shown in Fig. 2. In object detection, we exploit the features from final convolutional layer of S-CNN to activate the receptive fields (RFs) to obtain the regions where an object might appear. These proposals are fed into O-CNN architecture to extract features. Then, with these features we train class-specific binary SVM classifiers that gives us the probability of appearance of an object in a test set. In scene classification, we extract features of an image from S-CNN that represents global information. Thereafter, we exploit object detection scores which are fed into a network that consists of three hidden layers in order to model the interaction of scene and objects in an image as shown in Fig. 2. The object level information gives us the local features of the scene. By combining both global and local features, we model the output layer based on softmax regression. Finally, we fine tune both S-CNN and O-CNN (please see Sec. III for details) for better performance.

Our ***main contributions*** in this paper are as follows:

- In our joint scene classification and object recognition framework, both deep networks- S-CNN and O-CNN - take advantage of interdependence of scene and objects in order to improve performance.
- Receptive fields from last convolutional layer in S-CNN provide the region proposals for object detection. The number of proposed regions is significantly reduced (1-3 orders less compared to [7]) which leads to less computation cost with high recognition accuracy.
- As scene and objects co-occur, objects provide useful information about scene. So, we also take into account object detections along with features from S-CNN which are processed through three hidden layers. In this way, we compute robust features to achieve better performance on scene prediction.

II. Prior Work

Many of the scene classification methods use low dimensional features such as GIST [9], and SIFT descriptor [10]. In [11], spatial pooling regions are learned to construct mid-level representation to classify scenes. In [3], convolutional neural network is trained on places-205 dataset to learn deep features for scene recognition. In [12], the authors train a multi-scale convolutional network from raw pixels to extract dense feature vectors for scene labeling. In [13], context-specific objects and their layout are used to learn scene structure which is further used as low dimensional features to classify scenes. However, the authors do not use any interactive approach where recognizing one can benefit the other.

In object detection, some of the efficient techniques exploit sliding window [14] and boosting [15]. One of the promising approaches in recognition tasks has been to exploit the relationships between objects in a scene using a graphical model [5], [6]. Several different methods [16], [17] have been employed to represent context models to achieve higher accuracy in recognizing objects. In [18], objects are represented using mixtures of deformable part models (DPM). In [2], regions with CNN (R-CNN) features are used for object detection and their deep networks have shown very good results. However, finding regions by selective search [7] and then classifying approximately 2400 warped regions are indeed computationally expensive. In [19], authors present object detection using an additional spatial pyramid pooling layer that use selective search to find the candidate windows which are put on feature map of the CNN.

In [20], a framework is proposed for joint scene and object classification by exploiting contextual relationship between scene and objects. In [6], the authors present joint segmentation, object detection and scene classification that involves graphical models [6] to delineate the contextual information. However, these context-based approaches are highly dependent on initial prediction of scene or object labels in order to improve the accuracy. Unlike these approaches, we consider two CNNs- one for scene and other for objects, where interaction between them are exploited for efficient scene and object classification.

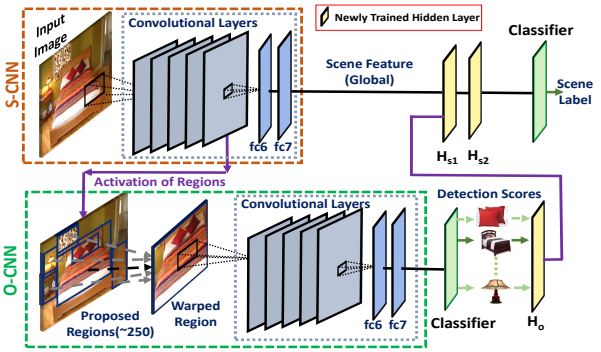


Fig. 2. Joint framework for scene and object classification. Orange and green dotted boxes represent the scene and object deep network respectively. Purple color implies the interlink between these two network. The flow from S-CNN to O-CNN gives us the probable object locations in an image. Then, the detection scores of the objects are taken as input to the hidden layer H_o which represents the local information of a scene. Finally, both global and local features are propagated through two hidden layers H_{s1} and H_{s2} to the output layer where scene labels are predicted. Best viewable in color.

III. Joint Scene and Object Model

Our goal in this paper is to jointly model the scene and objects where both can benefit each other in the recognition process. Fig. 2 shows overall framework.

A. Object Detection

Our object detection model follows three steps- region proposal, feature extraction and training binary classifiers.

Region Proposal. We adopt the region proposal technique as presented in [8] where the proposals are fewer in number and are mostly related to the objects in an image. Given an image, the region proposal approach provides some regions to localize the objects. We use final convolutional layer features of S-CNN to activate the possible regions for object localization using

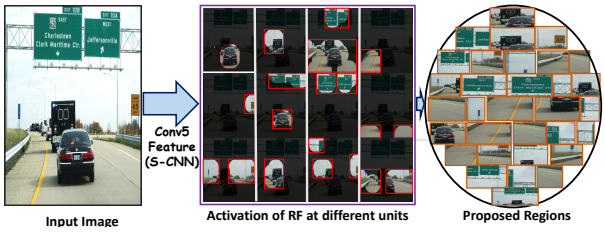


Fig. 3. Representation of region proposal approach for localizing objects. Given an image, receptive fields at final convolutional layer produce the activation regions. Bounding boxes have been placed around the segmentation using contour approximation method to obtain regions. Best viewable in color.

empirical receptive field for each unit. Along this process, we obtain approximately 450 regions on average from all the units of last convolutional layer in S-CNN. Contour approximation method has been used to place a bounding box around the activated region.

Fig. 3 shows some activated regions of an image using receptive field. We observe that all the objects of an image have been almost covered by the activation of RF of all the units from last convolutional layer of the S-CNN. We eliminate some of the proposals that overlap among themselves significantly to locate the same object. In order to do that, we first calculate Intersection over Union (IoU) among the bounding boxes of region proposals. If $IoU > \lambda$, we keep one from them and adjust the bounding boxes by multiplying a factor β. The elimination process of redundant windows helps us to prevent from high false positive rate. Then, we order the candidate windows according to the area of the bounding boxes. We approximately keep 250 regions with larger area. We do not consider the bounding boxes with very small area as those boxes imply part of the objects most of the time. For each region, we extract the feature with 4096-dimension from last fully connected layer (fc7).

Training Binary Classifiers. In order to detect objects, we train class specific binary classifiers. Let, $f_O \in \Re^{4096\times 1}$ be the feature vector from the $fc7$ layer. Each binary classifier will compute the probability, $P(O_i = 1|f_O)$ where i denotes the object class. For each binary classifier, we split the training set into positive and negative examples. With all regions proposals, we have the bounding boxes of different IoU overlap with ground-truth bounding box. We select the $IoU \leq 0.3$ as negative examples. We only consider the ground-truth boxes for each class as positive. With training features and labels, we get one linear SVM model per class. To fit the large training data we implement standard hard negative mining method as presented in [18]. To reduce the object localization error, we adopt the regression method as presented in [18].

B. Scene Classification

Scene classification employs two steps- feature extraction and classification.

Feature Extraction. To categorize scenes, we apply deep network to extract the features. Feature extraction involves three steps. We first extract global feature from fc7 layer of the S-CNN as shown in Fig. 2. Given an image, let $f_S \in \Re^{4096\times 1}$ be a vector obtained from the $fc7$ layer of S-CNN. The feature from S-CNN gives us global information of an image. Objects are important entities for a scene since they provide useful

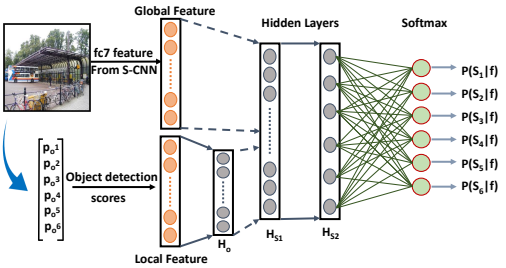


Fig. 4. This figure illustrates the feature extraction technique for scene classification. At first, object detection scores are propagated through a hidden layer H_o. Then, H_{S1} combines the feature from S-CNN for whole image and the feature of processed detection scores from H_o. Finally, one more fully connected layer connects these global and local features to the output layer in order to predict scene labels. Please see in color monitor for better view.

information about the scene. In this paper, we represent the semantic interaction between scene and objects by training three hidden layers. Given the object proposals obtained from S-CNN, object detectors detect objects in an image with some probability score. From the detection of objects, we can form a vector that implies whether a particular object is present or not. With the probability of appearing an object, we get object level (local) feature. For example, let consider a vector $p_o = [p_{o^1}, p_{o^2}, \dots p_{o^N}]$ that contains the probabilities of appearing an object in the scene. Here, N is the total number of object categories. The model structure shown to capture the object level features is shown in Fig. 4.

Next, we combine both global and local features. As shown in the Fig. 4, we use total three hidden layers, namely, H_o, H_{S1} and H_{S2} to propagate both the feature vectors f_S and p_o. The vector p_o is propagated to the H_{S1} through one hidden layer H_o. Here, every single hidden unit in H_o is connected to all the units in H_{S1}. H_{S1} is used as intermediate layer to combine both global (features from fc7 layer of S-CNN) and local (object level semantic information) features. The interaction between scene and objects is established through this hidden layer. Finally, one more hidden layer H_{S2} is incorporated to connect with softmax layer to predict the scene labels. Thus, appearance of the objects in a scene has an influence to predict the scene labels.

Classification. We use softmax to predict the label of the scene. Let, $\theta = [W_o, W_{S1}, W_{S2}]$ be the parameters of the 3 hidden layers where W_o, W_{S1}, W_{S2} denote the parameters of the layers H_o, H_{S1}, H_{S2} respectively. Given a feature f extracted from H_{S2}, the softmax function can be written as

$$P(S_i|f) = \frac{\exp^{(W_{S2,i}^T f)}}{\sum_{l=1}^{N_s} \exp^{(W_{S2,l}^T f)}} \quad (1)$$

Here, $i \in \{1, ..., N_s\}$ is the scene label. $W_{S2,i}$ implies the weight vector of the final layer corresponding to the class i. θ_i includes all the parameters of the hidden layers of H_o, H_{S1} and H_{S2} for class i. f is the test feature, which can be expressed as $f = [f_S\ ,\ p_o]$. Now, we get predicted label by maximizing $P(S_i|f)$ with respect to i. The predicted label can be expressed as, $S_{pred} = \arg\max_i \ P(S_i|f)$ The parameter θ can be found by solving an optimization problem. If we are given a set of training features along with labels, θ can be obtained by minimizing the cross entropy error.

$$L(\theta) = -\frac{1}{M}\sum_{p=1}^{M}\sum_{q=1}^{N_s} I(S^p = q)\log(S^p = q|x^p;\theta) \quad (2)$$

We find the optimal θ by using a standard gradient descent method. Here, $I(.)$ is the indicator function, it gives the value 1 if $p = q$, otherwise 0. S^p and x^p imply the scene label and the feature of the sample p. M is the number of samples.

Training the Hidden Layers. To train the hidden layers, we modify the cost function by adding a regularization term that penalizes large weights to improve generalization which can be expressed as

$$L'(\theta) = \frac{1}{M}\sum_{i}^{M} L_\theta(f^i) + \gamma R(\theta) \quad (3)$$

Here, $L_\theta(f^i)$ is the error averaged over instances and R is the regularization term. The data error $L_\theta(f^i)$ is computed from forward propagation. Loss L' is obtained from the output of the layers. We use stochastic gradient descent (SGD) to optmize the network in order to find the weights θ by minimizing the loss L' over the data. SGD coordinates between forward and backward to update the weights. We first need to compute the gradients of both error term and regularization term. With these gradients, we have $\bigtriangledown L'(\theta)$ which is the gradient of $L'(\theta)$ as in Eqn. 3. Now, we can update parameters as follows

$$V_{t+1} = \mu V_t - \alpha \bigtriangledown L'(\theta_t) \quad (4)$$

$$\theta_{t+1} = \theta_t + V_{t+1} \quad (5)$$

Where, μ is the momentum and α is the learning rate. V_{t+1} is the gradient of the model parameter θ. Finally, we update the parameters in each iterations using the Eqn. 5.

Fine-Tuning of S-CNN and O-CNN. We use two separate pre-trained models- one for S-CNN and other for O-CNN. The pre-trained models, for scene or objects, are trained on large number of categories. Here, our objective is to fine-tune the CNN parameters with new datasets both for S-CNN and O-CNN. We consider object proposals obtained from the new technique to tune the parameters. We use $N+1$ classes where N is the number of object class and additional 1 is for background. In order to fine-tune the pre-trained model of O-CNN, we split the dataset to form training and validation set. We consider Intersection over Union (IoU) (≥ 0.5) between the bounding boxes of a proposed region and ground-truth as positive and rest of the regions as negatives. We fine-tune the CNN parameters using stochastic gradient descent solver [21]. Similarly, for scene, we feed the data with two sources. One in input layer of S-CNN and other one in hidden layer H_o with the ground-truth detections of the objects. With the input data samples and corresponding labels, we fine tune S-CNN network as well.

IV. Experimental Results

In this section, we evaluate our object recognition and scene classification results on five challenging datasets and compare our method to other approaches.

Datasets. In SUN [23] dataset, we choose 150 scene classes and 120 object categories to evaluate scene classification and object detection performance. MIT-67 indoor [24] scene dataset consists of 67 indoor scene categories. It also provides large varieties of object categories. In MSRC [25] dataset, we evaluate our results with the ground truth which is available in [6] to classify 15 object categories and 21 scene classes. For Scene-15 [26] dataset, we cannot show our object detection results on scene-15 as there is no annotation for objects. We evaluate object detection on VOC2010 [22] dataset that contains 20 object categories. Since scene ground-truths are not provided, we can not perform scene classification on VOC2010.

Experimental Setup. We use pre-trained model *'VGG net'* [3] which is trained on 'places-205' dataset to extract the scene features from S-CNN. For O-CNN, we use pre-trained model *ILSVRC2012* as used in [2] to extract the features. We use caffe [21] to implement the CNN architecture. For the convenience of notation, we call the object detection framework *(Region proposals + O-CNN)* as R'-CNN in the rest of the paper.

As discussed in Sec. III-A, we choose approximately 250 regions from proposed candidate bounding boxes using the parameters λ and β. We choose the value of λ and β as 0.9 and 1.1. In MSRC dataset, 1-3 objects are present in an image and the object occupies significant portion of the whole image. So, we choose the top 10 activated regions with larger area from proposed candidate bounding boxes for MSRC dataset. Similarly, We choose 50 regions for VOC2010.

Training Data. In this paper, we split the dataset to form training and validation set in order to fine-tune (FT) the O-CNN parameters. Fine-tuning is required to adjust the CNN parameters to new datasets. We fine-tune our object-CNN with respect to VOC2010 [22], SUN [23] and MIT-67 datasets [24]. We consider our proposed object proposals with $IoU \geq 0.5$ of ground-truth as positive and rest of the regions as negatives. With $50k$ iterations we fine-tune the CNN parameters using SGD solver [21]. We do not finetune the parameters on MSRC since this dataset does not contain enough samples.

Evaluation Criteria. We calculate the average precision (AP) of each category comparing with the ground truth. Precision depends on both correct labeling and localization (overlap between object detection box and ground truth box). Let the computed bounding box of an object be O_b and the ground truth box be G_b, then the overlap ratio, $OR = \frac{O_b \cap G_b}{O_b \cup G_b}$. $OR \geq 0.5$ is considered as correct recognition of an object if the label of the object is also correct. From the average precision (AP) of each category, we calculate the mean AP (mAP) over all the categories.

Baseline Methods. Before presenting our results, we define all the abbreviations that will be used as baseline methods.

◇ **R′-SPP:** Our region proposal technique (flow from S-CNN to O-CNN) with spatial pyramid pooling [4] network.

◇ **R′-CNN:** Our region proposal with O-CNN (proposed).

◇ **R′-CNN-FT$_1$:** R'-CNN with fine-tuned (FT) O-CNN.

◇ **R′-CNN-FT$_2$:** R'-CNN with FT-O-CNN and FT-S-CNN.

◇ **CNN$_1$:** 'fc7' feature from S-CNN with 'Alexnet' model.

◇ **CNN$_2$:** 'fc7' feature from S-CNN with 'VGGnet' model.

◇ **S-CNN+H$_o$:** S-CNN feature concatenated with feature extracted from H_o layer.

◇ **S-CNN+H$_{S1}$:** S-CNN feature concatenated with feature extracted from H_{S1} layer.

◇ **S-CNN+H$_{S2}$(S′-CNN):** S-CNN feature + feature extracted from H_{S2} layer.

Methods	VOC2010 [22]		SUN [23] dataset		MIT-67 [24] dataset		MSRC [25] dataset	
	accuracy	N_R	accuracy	N_R	accuracy	N_R	accuracy	N_R
DPM [18]	24.78%	-	18.79%	-	19.61%	-	48.20%	-
SPP [4]	50.78%	~ 2400	-	-	-	-	-	-
R'-SPP	52.32%	250	-	-	-	-	-	-
R-CNN [2]	50.46%	~ 2400	36.28%	~ 2400	32.07%	~ 2400	76.22%	~ 2400
R'-CNN	52.80%	50	37.63%	250	32.88%	250	**76.79**%	10
R'-CNN-FT_1	54.20%	50	39.76%	250	32.95%	250	-	-
R'-CNN-FT_2	**56.12%**	50	**42.86**%	250	**34.72**%	250	-	-

TABLE I
MEAN AVERAGE PRECISION (MAP) OF STATE-OF-THE-ART METHODS AND OUR METHOD ON FOUR DATASETS. WITH NUMBER OF REGION PROPOSALS (N_R) WHICH ARE 1-3 ORDERS LESS THAN R-CNN [2], R'-CNN ACHIEVES BETTER PERFORMANCE.

$\diamond$ **S′-CNN +FT$_1$**: S'-CNN with fine-tuned (FT) S-CNN.
$\diamond$ **S′-CNN +FT$_2$**: S'-CNN with FT-S-CNN and FT-O-CNN.

a) Object Recognition Results:

Comparison against Other Detectors. We perform experiments of our method, R'-CNN on four datasets- VOC2010 [22], SUN [23], MIT-67 Indoor [24] and MSRC [25] datasets. In Table I, we compare our R'-CNN detector with DPM [18] and R-CNN [2]. We implement both DPM and R-CNN on aforementioned datasets. From Table I, we can see that R'-CNN outperforms both DPM [18] and R-CNN [2]. After fine-tuning both S-CNN and O-CNN, the performance of our method is further improved. From Table I, the best comparable result is found with R-CNN [2]. *Our method achieves better performance with compared to R-CNN with very less number of region proposals.*

We also analyze preliminary results on VOC dataset for SPP [4] detector as presented in Table I. In SPP, proposals are generated using [7] which are then projected into the feature map of CNN to extract feature. Then, these features are used to train and classify binary SVM. We use our object proposal strategy instead of [7] in SPP net. We can see from the Table I that our region selection strategy performs better in SPP on VOC2010 dataset.

Reduction of Computational Cost. The main advantage of using our region proposal technique in object recognition is that it reduces the computational time as less number of proposals are proposed. Our R'-CNN detector is approximately ***9*** times faster than R-CNN [2] as we only classify around 250 regions instead of 2400 regions presented in [2] on SUN [23] and MIT-67 [24] datasets. For VOC2010 [22] and MSRC [25] dataset, our method is ***48*** and ***240*** times faster than R-CNN respectively. Results are shown in Table I. In Fig. 6 (b), we observe that how object detection performance varies with varying number of region proposals. It is not always useful to have large number of object proposals as it might also increase the rate of false positives which will be discussed next.

Is the proposed region proposal semantically meaningful? In order to measure the region proposal quality, we also calculate the ratio between the number of false positives (FP) and the number of proposed regions, $FPR = \frac{FP}{N_R}$ (N_R is the number of proposed regions). From our analysis, R-CNN [2] has higher FPR than ours by approximately 1.43%, 2.32%, 1.06% and 1.21% on VOC2010 [22], SUN [23], MIT-67 Indoor scene [24] and MSRC [25] datasets. Due to low FPR, our approach achieves higher performance than R-CNN. Fig. 5 shows the object recognition results on some example images. From Fig. 5, we observe that objects with larger pixel size are more inclined to have correct label and localization.

Feature extraction	SUN	MIT-67	MSRC	Scene15
GIST [9]	57.47%	29.08%	57.69%	68.02%
dSIFT [10]	61.13%	34.44%	71.36%	75.21%
CNN_1 feature [3]	75.95%	70.00%	90.15%	91.06%
CNN_2 feature [3]	76.62%	71.48%	91.67%	**91.88**%
S-CNN + H_o	76.36%	71.62%	90.88%	-
S-CNN + H_{S1}	76.92%	71.24%	92.08%	-
S-CNN + H_{S2}	77.85%	72.76%	**93.14**%	91.56%
S'-CNN+FT_1	78.21%	73.68%	-	-
S'-CNN+FT_2	**79.49**%	**74.12**%	-	-

TABLE II
SCENE CLASSIFICATION ACCURACY.

b) Scene Classification Results:

Comparison with State-of-the-art Methods. Table II shows the scene categorization accuracy of other state-of-the-art methods and S'-CNN feature extraction technique. We first compare proposed feature extraction method to other methods such as GIST [9], dSIFT [10] and CNN [3]. S'-CNN technique outperforms GIST, dSIFT by large margin. From Table II, the best comparable result is found with CNN_2 feature [3]. With the tuned parameters of S-CNN and O-CNN, S'-CNN feature based classification is further improved. For Scene-15 dataset, as we do not have any annotation for objects, we use R-CNN [2] to detect the objects. As detectors are not trained on this dataset, we do not obtain the best performance with S'-CNN. From Table II, we can conclude that *information flow from O-CNN to S-CNN improves the performance on scene categorization*.

Is Selection of Regions Important in Scene Categorization? Our proposed feature extraction technique exploits the local features from O-CNN. However, the scene classification performance depends on the quality of region proposals. We carry out an experiment on SUN dataset with 7240 images that have full annotation for objects. We select three different region selection strategies namely, selective search [7] used in R-CNN [2], proposed object proposal technique and ground truth regions to detect objects. Region proposals with tight boundary over objects give good detection results. Since object detection scores are used to represent feature for scenes, object proposals have direct impact in classifying scene. Table III demonstrates the scene classification accuracy. From Table III, we observe that our region selection strategy, S'-CNN outperforms the CNN_2 feature [3] method where no region selection is used. We also compare with ground-truth as proposals, and our proposal performance is close to the result of scene classification when ground truth is used as region proposals.

Comparison with Other Joint Scene and Object Model. We also compare our joint model with the holistic [6] approach

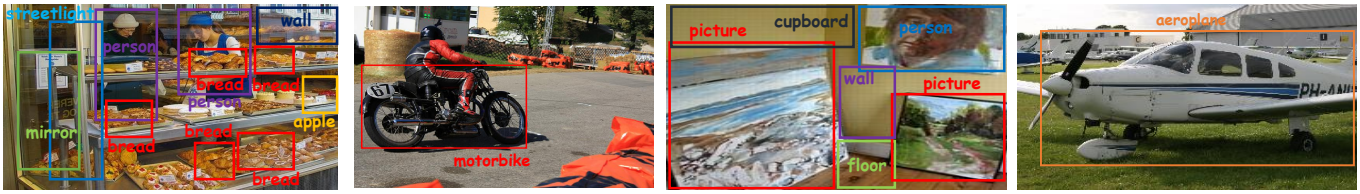
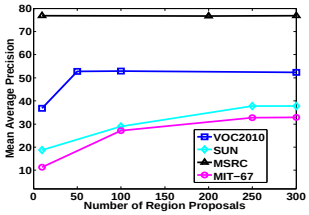


Fig. 5. Some examples showing the object localizations with label of different object categories in an image.

Region Selection Strategy			
CNN_2 feature [3]	S-CNN + R-CNN [2]	S'-CNN	Ground-truth
83.92%	83.64%	85.28%	**86.14%**

TABLE III
SCENE CLASSIFICATION PERFORMANCE WITH DIFFERENT REGION SELECTION STRATEGIES.

Method	Scene Prediction	Object Detection
MIT-67 Indoor Dataset		
holistic [6]	72.04%	32.48%
Ours	**74.12%**	**34.72%**
MSRC Dataset		
holistic[6]	92.28%	**76.83%**
Ours	**93.14%**	76.79%



(a) (b)

Fig. 6. (a) Comparison of other combined scene and object model, and (b) Plot of mAP vs number of region proposals. Here, we can see that how the detection performance changes with the number of object proposals.

for joint scene and object classification. Fig. 6(a) demonstrates the comparison results on MIT-67 Indoor and MSRC datasets. To make a fair comparison, we implement their holistic model [6] on top of CNN_2 based scene classification method and R-CNN [2] based object detectors. Holistic model [6] does not perform well when an image contains multiple objects. Thus, our method outperforms [6] on MIT-67 Indoor dataset. On MSRC dataset, both our model and holistic [6] approach are comparable in classifying objects.

Discussion. Faster-RCNN [27] is one of the promising techniques that provides object proposals using region proposal network (RPN). However, faster-RCNN slides over the final convolutional feature map and proposes 9 anchor boxes (region proposals) for each spatial location which provides 20,000 anchors in total. After series of operations (ignoring cross-boundary anchors and non-maximal suppression), RPN ends up with 2000 proposals for training. Since sliding window is computationally expensive, our region proposal technique can provide significantly less number of initial anchor boxes (e.g. 250*9 2250, one magnitude less) which can reduce the computational burden. This is because our region proposals are derived from the scene classification network.

V. CONCLUSION

In this paper, we propose a novel framework for joint scene and object classification by exploiting the inter-dependence between scene and object CNN architectures. In our framework, S-CNN provides object proposals that improves the performance of object detection. Similarly, bottom-up flow from O-CNN to S-CNN aids to form robust features that discriminate the different scene categories.

Acknowledgment. This work is partially supported by NSF grant 1544969 and US Office of Naval Research contract N00014-15-C-5113 through Mayachitra, Inc.

REFERENCES

[1] K. A. Ehinger, A. Torralba, and A. Oliva, "A taxonomy of visual scenes: Typicality ratings and hierarchical classification," *Journal of Vision*, vol. 10, no. 7, pp. 1237–1237, 2010.
[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*. IEEE, 2014, pp. 580–587.
[3] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *NIPS*, 2014, pp. 487–495.
[4] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *PAMI*, 2015.
[5] A. Torralba, K. P. Murphy, and W. T. Freeman, "Using the forest to see the trees: exploiting context for visual object detection and localization," *Communications of the ACM*, vol. 53, no. 3, pp. 107–114, 2010.
[6] J. Yao, S. Fidler, and R. Urtasun, "Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation," in *CVPR*, 2012, pp. 702–709.
[7] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, "Segmentation as selective search for object recognition," in *ICCV*, 2011.
[8] J. H. Bappy and A. K. Roy-Chowdhury, "CNN based region proposals for efficient object detection," in *ICIP*, 2016.
[9] Z. Li and L. Itti, "Saliency and gist features for target detection in satellite images," *TIP*, vol. 20, no. 7, pp. 2017–2029, 2011.
[10] C. Liu, J. Yuen, and A. Torralba, "Dense scene alignment using sift flow for object recognition," in *CVPR*, 2009.
[11] D. Lin, C. Lu, R. Liao, and J. Jia, "Learning important spatial pooling regions for scene classification," in *CVPR*, 2014, pp. 3726–3733.
[12] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *PAMI*, vol. 35, no. 8, pp. 1915–1929, 2013.
[13] H. Izadinia, F. Sadeghi, and A. Farhadi, "Incorporating scene context and object layout into appearance modeling," in *CVPR*, 2014, pp. 232–239.
[14] A. Vedaldi, V. ulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *ICCV*, 2009, pp. 606–613.
[15] Z. Qi, Y. Xu, L. Wang, and Y. Song, "Online multiple instance boosting for object detection," *Neurocomputing*, vol. 74, pp. 1769–1775, 2011.
[16] Y. J. Lee and K. Grauman, "Object-graphs for context-aware visual category discovery," *PAMI*, vol. 34, no. 2, pp. 346–358, 2012.
[17] C. Galleguillos and S. Belongie, "Context based object categorization: A critical survey," *CVIU*, vol. 114, pp. 712–722, 2010.
[18] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *PAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.
[19] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *ECCV*, 2014.
[20] J. H. Bappy, S. Paul, and A. K. Roy-chowdhury, "Online adaptation for joint scene and object classification," in *ECCV*, 2016.
[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," *arXiv preprint arXiv:1408.5093*, 2014.
[22] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, 2010.
[23] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in *CVPR*. IEEE, 2010, pp. 129–136.
[24] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *CVPR*, 2009.
[25] T. Malisiewicz and A. A. Efros, "Improving spatial support for objects via multiple segmentations," *BMVC*, 2007.
[26] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.
[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.