Dense Image Labeling Using Deep Convolutional Neural Networks

Md Amirul Islam, Neil Bruce, Yang Wang
Department of Computer Science
University of Manitoba
Winnipeg, MB
{*amirul, bruce, ywang*}*@cs.umanitoba.ca*

***Abstract*—We consider the problem of dense image labeling. In recent work, many state-of-the-art techniques make use of Deep Convolutional Neural Networks (DCNN) for dense image labeling tasks (e.g. multi-class semantic segmentation) given their capacity to learn rich features. In this paper, we propose a dense image labeling approach based on DCNNs coupled with a support vector classifier. We employ the classifier based on DCNNs outputs while leveraging features corresponding to a variety of different labels drawn from a number of different datasets with distinct objectives for prediction. The principal motivation for using a support vector classifier is to explore the strength of leveraging different types of representations for predicting class labels, that are not directly related to the target task (e.g. predicted scene geometry may help assigning object labels). This is the first approach where DCNNs with predictions tied to different objectives are combined to produce better segmentation results. We evaluate our model on the Stanford background (semantic, geometric) and PASCAL VOC 2012 datasets. Compared to other state-of-the-art techniques, our approach produces state-of-the-art results for the Stanford background dataset, and also demonstrates the utility of making use of intelligence tied to different sources of labeling in improving upon baseline PASCAL VOC 2012 results.**

***Keywords*-Deep Convolutional Neural Network, Support Vector Classifier, Semantic Segmentation, Geometric Labeling**

I. Introduction

Many computer vision problems involve producing a pixel-wise dense labeling of a given image as the output. One example of dense image labeling is the task of semantic segmentation. The goal of semantic segmentation is to label each pixel in an image according to the object classes that this pixel belongs to. Another example is geometric labeling, where the goal is to label each pixel according to its geometric class (e.g. sky, vertical, horizontal). Traditionally, these dense image labeling tasks are often solved by learning a classifier that classifies each pixel based on some manually defined visual features (e.g. [19]). Probabilistic models (e.g. conditional random fields (CRFs)) are then used to refine the final results.

In recent years, deep convolution neural networks (DCNNs) have shown tremendous success in high-level visual recognition tasks, such as image classification, object detection and a variety of other problems. This has included work on extending DCNNs for dense image labeling problems, such as semantic segmentation. Most DCNN-based semantic segmentation methods (e.g. [12]) use convolutional neural networks to produce a coarse label map, then use upsampling (sometimes referred to as deconvolution) to produce dense outputs.

In order to properly train deep convolutional neural networks, one typically needs a large amount of labeled training data. Compared with image classification, training data for dense image labeling tasks is much more onerous to produce. For example, the current semantic segmentation datasets are orders of magnitude smaller than datasets that address the problem of image classification. Most DCNN-based semantic segmentation methods use pre-trained image classification models and fine-tune those models for semantic segmentation.

The computer vision community has produced several benchmark datasets for various dense image labeling tasks over the years. For example, the Standard background dataset [6] contains pixel-level annotations of 8 semantic classes and 3 geometric classes. The PASCAL VOC dataset [4] contains pixel-level annotations for 21 semantic classes (20 object classes and the background class). The possibility of combining these datasets (and others as they become available) presents the opportunity to make use of larger datasets, and more variety in labeling to learn DCNNs for dense image labeling. Given that the class labels corresponding to different datasets may not be identical, there is a challenge in directly combining these datasets to address a specific problem. For example, in considering the semantic segmentation problem posed by the PASCAL VOC dataset, it is unclear whether value may be derived from considering images and labels resident in the Stanford background dataset.

Our work is motivated by the aforementioned observation. Even though the annotations on different datasets are not compatible, the visual representations that are of value in solving these different dense labeling tasks may overlap. In making use of these diverse datasets to learn a good visual feature representation, this may present the opportunity for better performance for any specific dense labeling task corresponding to any of the problems associated with a specific subset of data corresponding to this larger dataset comprised of heterogeneous and incompatible labels.

In this paper, we therefore propose a new approach

for dense image labeling by taking advantage of multiple datasets, wherein class labels corresponding to different datasets might not be compatible. We learn a DCNN for each of the datasets separately. The outputs of the DCNNs learned from different datasets are concatenated to provide feature vectors that are directly driven by the target problem, but also corresponding to other types of labels. An SVM classifier is learned based on this feature vector to perform dense image labeling on a particular dataset. This presents a natural avenue for determining the extent to which added value may be derived from making use of labeled data that are not directly related to the target classification problem.

II. Related Work

Semantic segmentation is fundamental to image understanding as it assigns class labels to individual pixels in an image. The problem of semantic segmentation has been studied over decades but remains a challenging task. Some of the difficulty is due to variation in the appearance of objects, background clutter, pose variations, scale changes, occlusions, and other factors.

Approaches based on DCNNs have shown tremendous success in computer vision. Krizhevsky et al. [9] introduced a DCNN architecture named AlexNet for the image classification task. This model consists of millions of parameters and neurons. Simonyan and Zisserman [18] proposed a deep network named VGG-16 to examine the impact of network depth on classification accuracy. Recent, there has been work on extending DCNNs for dense image labeling tasks, such as semantic segmentation. Mostajabi et al. [14] introduced a feed-forward architecture for semantic segmentation based on super-pixels. Starting from a super-pixel, they consider small regions containing the super-pixel along with the regions around it to extract rich features using a CNN. Instead of predicting labels for each pixel, their approach classifies super-pixels using a feedforward multilayer network. Classifying super-pixels using a CNN leads to significant improvement in accuracy.

Recent techniques apply DCNNs to the whole image in a sliding window fashion. Long et al. [12] proposed the first work which trains fully convolutional networks end-to-end. They derived these results based on fine-tuning of the VGG-16 [18] and GoogLeNet [20], models towards performing segmentation by defining a novel skip architecture which combines semantic information with deep, coarse, and appearance based information. Chen et al. [3] proposed a novel architecture for semantic segmentation using a DCNN and fully connected CRF. This is the first model where DCNNs and CRFs are combined to produce accurate segmentation results.

The fixed sized receptive fields in CNNs imply certain limitations during training and label prediction. If the image size is larger or smaller than the defined receptive field, it is more likely to be mislabeled as label prediction only depends on local information for large objects. Noh et al. [15] proposed a deconvolution network consisting of deconvolution and un-pooling layers for semantic segmentation which improves the limitations of fully convolutional networks. The proposed deconvolutional network [15] identifies more detailed structure in upsampling the image, and leverages cross-scale relationships to predict instance-wise segmentation labels which are combined to produce the resulting dense segmentation. Zheng et al. [23] proposed a new segmentation model named CRFasRNN by expressing the mean field inference of dense CRFs with Gaussian pairwise potentials in the form of a Recurrent Neural Network (RNN). CRFasRNN [23] can be trained end-to-end by conventional back-propagation algorithms which yields a new state-of-the art in terms of accuracy. For the pixel-level prediction produced by a pre-trained CNN, CRFasRRN [23] use the CRF inference as a post-processing method. Liu et al. [11] introduced a CRF based segmentation model. Instead of using conventional feature extractors they used 4096 dimensional CNN features to learn the CRF in order to predict multi-class label.

Geometric labeling is another dense image labeling task we consider in this paper. The goal of geometric labeling is to label each pixel (or superpixel) in the image according to its geometric class. Hoiem et al. [7] proposed a method for geometric labeling by classifying each superpixel. Gould et al. [6] used a CRF model to decompose an image scene into geometric and semantically consistent regions.

III. Our Approach

Given an input image, our goal is to produce a dense labeling of the pixels in the image. In this paper, we consider two dense labeling tasks, namely semantic segmentation [12] and geometric labeling [6]. The goal of semantic segmentation aims to label each pixel according to the labeled object category (e.g. people, car, building, etc.). The goal of geometric labeling is to label each pixel according to its geometric class. Over the years, the computer vision community has created several benchmark datasets for these problems, but these datasets often consider different sets of classes. For example, the Stanford background dataset [6] contains 8 categorical classes (sky, tree, road, grass, water, building, mountain, foreground) and 3 geometric classes (sky, horizontal, vertical) while the PASCAL VOC data [4] contains 21 classes (20 object classes + background).

In this paper, we propose an approach for dense image labeling based on deep convolutional neural networks (DCNNs). The novelty of our approach is that we take advantage of multiple datasets even though they are defined by different sets of class labels. In particular, we use the Standard background dataset (with both semantic labels and geometric labels) and the PASCAL VOC dataset (with object class labels) in this paper. An overview of our approach is illustrated in Fig. 1. First, we train three separate DCNNs. The

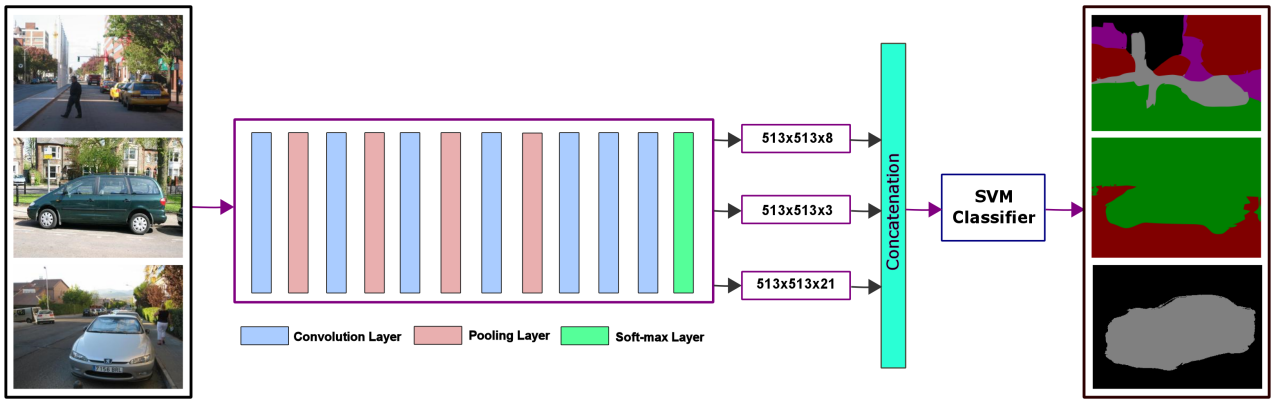

Figure 1. An overview of our proposed approach. Leftmost images are samples from different datasets. From each dataset, we learn a deep convolutional neural network (DCNN). The architecture of the DCNN is shown in the middle of the figure and is described in detail in Sec. III-A. Convolution, pooling and soft-max layers in the DCNNs are shown in different colors. ReLu layers are omitted from the box. The DCNN learned from each dataset will produce a dense labeling for a given image. We concatenate the outputs from these DCNNs to form a feature vector for each pixel in the image. We then train an SVM classifier based on these feature vectors to obtain the final labeling of each pixel in the image.

first DCNN is trained on the Stanford background dataset to produce one of the 8 semantic classes for each pixel. The second DCNN is trained on the Stanford background dataset to produce one of the three geometric classes. The third DCNN is trained on the PASCAL dataset to produce one of the 20 object classes for each pixel (Sec. III-A). For a given image, we apply these three DCNNs and concatenate their outputs to form a feature vector. We then learn a SVM classifier based on this feature vector to predict the label of each pixel in the image (Sec. III-B).

A. Deep Convolutional Neural Network

The architecture of our deep network is based on DeepLab [3], which in turn is based on the VGG-16 network [18] trained on the ImageNet classification task. In total, the network has 15 convolutional layers and 5 max-pooling layers. Table I summarizes the different layers in the network and their parameters.

An input image is passed through a stack of convolutional layers with very small kernel sizes. Spatial pooling is carried out by five max-pooling layers, which follow the convolution layers. Two fully-connected layers of VGG-16 [18] network are transformed to convolutional layers in order to get pixel-wise prediction. The last 1x1 convolution (fc8) layer is used to make sure that the number of output matches the number of labels. For example, if we train this network on the Standard background dataset to predict geometric classes for each pixel, the number of labels will be 3. If we train this network to predict object classes for each pixel, the number of labels will be 21. We use Caffe [8] for training the network.

Suppose that we want to train the network to predict semantic classes on the Stanford background dataset. There are 8 semantic categories on this dataset. Each image is rescaled to 513x513 during training. Through convolution and pooling, the deep network extracts multi-class visual deep features and generates 8 coarse score maps. Each feature map indicates the probabilistic label map of each semantic category. The resulting feature maps are up-sampled to 513x513 using bilinear interpolation to equate the size of the input image. Therefore, the network predicts 513x513x8 labels in the end. Similarly, we will get 513x513x3 labels by training a DCNN for the geometric labeling task on the Stanford background dataset, and 513x513x21 labels by training a DCNN for the semantic labeling task on the PASCAL VOC dataset. We concatenate these three sets of features maps together in the end to get a feature map of 513x513x(8+3+21). Each pixel corresponds to a (8+3+21) dimensional feature vector in the feature map.

B. SVM Learning

In this section, we consider how the feature maps obtained from the DCNNs in Sec. III-A are processed and used to train a SVM classifier for producing the final the dense labeling on a particular dataset.

For ease of presentation, let us consider the semantic segmentation problem on the PASCAL VOC dataset. This problem requires labeling each pixel as one of the 21 semantic classes defined in the PASCAL VOC. We first run the three DCNNs from Sec. III-A on both training and test images in the PASCAL VOC datasets (these DCNNs are trained from both the PASCAL VOC and the Standard background datasets with heterogeneous labels). Each pixel in an image is then represented by a (8+3+21=32) dimensional feature vector. We then learn a linear SVM classifier to predict the 21 semantic classes on the PASCAL VOC dataset using this 32 dimensional feature vector.

	1	2	3	4	5	6	7	8	9	10	11	12	13
layer	2× **conv**	**max**	2 × **conv**	**max**	2× **conv**	**max**	3× **conv**	**max**	3× **conv**	**max**	**fc6**	**fc7**	**fc8**
filter-stride	3-12	3-2	3-12	3-2	3-12	3-2	3-12	3-1	3-12	3-1	3-12	1-12	1-12
#channel	64	64	128	128	256	256	512	512	512	512	1024	1024	**#label**
activation	relu	idn	relu	idn	relu	idn	relu	idn	relu	idn	relu	relu	soft
size	321	161	161	81	81	41	41	41	41	41	41	41	41

Table I
DETAILS OF THE ARCHITECTURE OF THE CONVOLUTIONAL NEURAL NETWORK.

We have experimented with several approaches for constructing the training data for learning the SVM classifier.

ConvNet-SVM: This approach randomly selects a set of pixels from all the training images on PASCAL VOC. Each pixel will be a training instance with 21 dimensional feature vector from DCNN. Since we know the ground-truth semantic labels of these pixels, we can learn a SVM classifier using the ground-truth labels.

ConvNet-CSVM: This approach randomly selects a set of pixels from all the training images on PASCAL VOC. Each pixel will be a training instance with 32 dimensional feature vector, corresponding to the concatenated feature set.

Both ConvNet-SVM and ConvNet-CSVM learn the SVM classifier from the pixels sampled from the PASCAL training images. We have also experimented with sampling the pixels from the PASCAL test images.

ConvNet-CSVM2: This approach randomly selects a set of pixels from all the test images on PASCAL VOC. Each pixel will be a training instance with 32 dimensional feature vector. An image-specific SVM technique is then applied to each test image separately assuming the predicted labels (i.e. based on ConvNet-SVM) represent the ground-truth for that specific image. That is, given a general classifier, and associated label predictions, these may be refined by training an SVM specific to the image under consideration by making the assumption that the assigned class labels are mostly correct and treating this as the ground truth. This stage of image-specific SVM classification takes full advantage of the support vector classification stage to refine the initial generic predictions to produce those that may better characterize a specific test image. An overview of the process is shown in Fig. 2

ConvNet-CSVM3: This approach is similar to Convnet-CSVM2 apart from how pixels are sampled from the test images on PASCAL VOC. We select class-wise positive pixels (those for which ConvNet-SVM produces a higher score among all other categories) with a 32 dimensional feature vector.

ConvNet-WSVM: This approach randomly selects class-wise positive pixels with a 32 dimensional feature vector from all the test images on PASCAL VOC. Then instead of linear SVM, we train a weighted SVM by assigning a different weight to each pixel to vary the contribution of each pixel in the second SVM training stage. We use the Kernel-based Possible C-means (KPCM) algorithm [21] to generate a weight for each pixel. Finally the refined pixel-wise segmentation is predicted by the refined SVM.

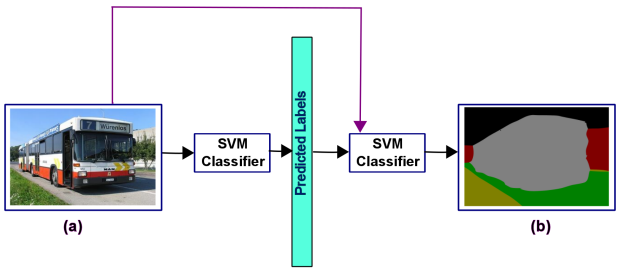

Figure 2. (a) Test image. Predicted labels produced by the classifier are used to train image specific classifier again along with the corresponding test image. Note that the known label values themselves are not used in training, but rather the labels produced by the *generic* SVM are assumed to be mostly correct and define the image specific ground truth for subsequent SVM training based on 1 image. (b) Output image.

Table II summarizes these different approaches.

Method	Procedure
ConvNet-SVM	Trained linear SVM by choosing random samples from training images
ConvNet-CSVM	Trained linear SVM by choosing samples from training images with concatenated feature maps
ConvNet-CSVM2	Trained linear SVM by choosing samples from test images with concatenated feature maps (note that for this case and those that follow, the ground truth labels are assumed (from the preceding classifiers), and do not come from the known test image labels)
ConvNet-CSVM3	Trained linear SVM by choosing class-wise positive samples (where the DCNN produces a higher score among all other categories) from test images with concatenated feature maps
ConvNet-WSVM	Trained weighted SVM by choosing class-wise positive samples from test images with concatenated feature maps.

Table II
DESCRIPTION OF DIFFERENT CONFIGURATIONS USED IN OUR EXPERIMENTS. IN EACH CASE, WE PERFORM IMAGE-SPECIFIC SVM TO GET FINAL PREDICTIONS.

IV. EXPERIMENTAL EVALUATION

Method	sky	tree	road	grass	water	building	mountain	foreground	average (%)	overall (%)
ConvNet	95.5	85.4	93.7	94.4	92.1	88.6	86.2	77.4	89.1	90.4
ConvNet-SVM	93.2	90.5	93.1	92.7	91.3	89.4	85.8	78.6	89.3	90.8
ConvNet-CSVM	94.4	85.9	95.1	91.0	92.7	90.9	85.5	86.5	89.6	91.2
ConvNet-CSVM2	93.2	90.3	96.5	92.7	92.6	96.2	65.3	75.3	87.8	**91.6**
ConvNet-CSVM3	93.9	90.1	94.4	94.2	91.7	94.2	90.7	75.4	**90.4**	90.9
ConvNet-WSVM	94.0	90.1	94.2	94.2	91.4	94.2	90.8	74.5	**90.4**	91.0

Table III
QUANTITATIVE RESULTS OF DIFFERENT APPROACHES FOR THE SEMANTIC SEGMENTATION TASK ON THE STANFORD BACKGROUND DATASET [6]. WE SHOW THE ACCURACY FOR EACH SEMANTIC CLASS, THE AVERAGE ACCURACY OF THESE EIGHT CLASSES (MCA), AND THE OVERALL PIXEL ACCURACY (OVERALL).

	sky	horizontal	vertical	average (%)	overall (%)
ConvNet	90.1	92.8	93.2	92.0	93.5
ConvNet-SVM	90.5	94.2	93.8	92.8	94.0
ConvNet-CSVM	90.6	93.9	94.6	93.1	93.8
ConvNet-CSVM2	90.5	95.3	96.1	**94**	95.1
ConvNet-WSVM	90.6	94.7	96.6	**94**	**95.2**

Table IV
QUANTITATIVE RESULTS OF DIFFERENT APPROACHES FOR THE GEOMETRY LABELING TASK ON THE STANFORD BACKGROUND DATASET [6].

We present experimental results on two different datasets: the Stanford background dataset (SBD) [6] and the PASCAL VOC 2012 [4] segmentation benchmark dataset. On the Stanford backgound dataset, we report several metrics for measuring the pixel accuracy. Let n_{ij} be the number of pixels of class i predicted to be class j, and $t_i = \sum_j n_{ij}$ be the total number of pixels of class i. Let K be the total number of classes. We compute:

- per-class accuracy for the i-th class: n_{ii}/t_i
- average per-class accuracy: $(1/K)\sum_i n_{ii}/t_i$
- overall accuracy: $\sum_i n_{ii}/\sum_i t_i$

On the PASCAL VOC 2012 dataset, we report results using intersection-over-union (IoU) for each class and the mean IoU overall all classes as follows:

- IoU for the i-th class: $n_{ii}/(t_i + \sum_j n_{ji} - n_{ii})$
- mean IoU: $(1/K)\sum_i n_{ii}/(t_i + \sum_j n_{ji} - n_{ii})$

A. Implementation

ConvNet-CSVM model is trained and tested with Caffe [8] on a machine with 10 cores (2.3GHZ Intel Xeon E5-2630V3 CPU), 64GB RAM, 4TB hard drive, and two NVIDIA Titan X GPUs. We use some of the parameters from DeepLab [3] to initialize the network. Following [3], the batch size and initial learning rate are initialized to 30 and 0.001 respectively. We fix the momentum to 0.9, weight decay of 0.0005 and maximum iteration to 6000. The total number of parameters in the model is approximately 20.5M and training requires approximately 6 hours.

Detailed description of different configurations used in our experiments are presented in Table II. Each configuration differs in the approach of creating training proposals for linear and weighted SVM. For the image specific SVM, each test image is trained separately assuming the predicted labels produced by linear or weighted SVM as ground-truth.

B. Dataset

We evaluate the proposed method on the Stanford Background Dataset [6] and the PASCAL VOC 2012 [4] segmentation challenge dataset. The Stanford background dataset contains total 715 images of urban and rural scenes. Each pixel is labeled with one of the 8 semantic classes (sky, tree, road, grass, water, building, mountain, and foreground) and one of the 3 geometric classes (sky, horizontal, and vertical). Each image is approximately 240×320 and contains at least one foreground object. The PASCAL VOC 2012 dataset [4] consists of 1464 training and 1456 test images. Each pixel in this dataset is labeled with one of the 21 categories (20 object categories and the background class).

C. Evaluation on Stanford background dataset

In this section, we report our evaluation results on the Stanford background dataset. Following [6], [11], [14], [16], we use 5-fold cross-validation which splits the dataset into 572 training images and 143 test images. A challenging class within this dataset is the foreground class, since it includes a wide range of objects like person, cow, bicycle, sheep, car as a singular class. The appearance of the foreground class can vary drastically across different object types. Another challenging class is the mountain class, since it appears in very few images. In order to explore the strength of leveraging different types of representations for predicting labels, we report results for different configurations. The quantitative results of semantic and geometric classes are shown in Table III and Table IV, respectively. We can see that our method performs quite well on this dataset. Some qualitative semantic segmentation examples are shown in Fig. 3. Some qualitative examples for geometric labeling are shown in Fig. 4

We compare our approach with several baseline methods. The comparisons are summarized in Table V and Table VI.

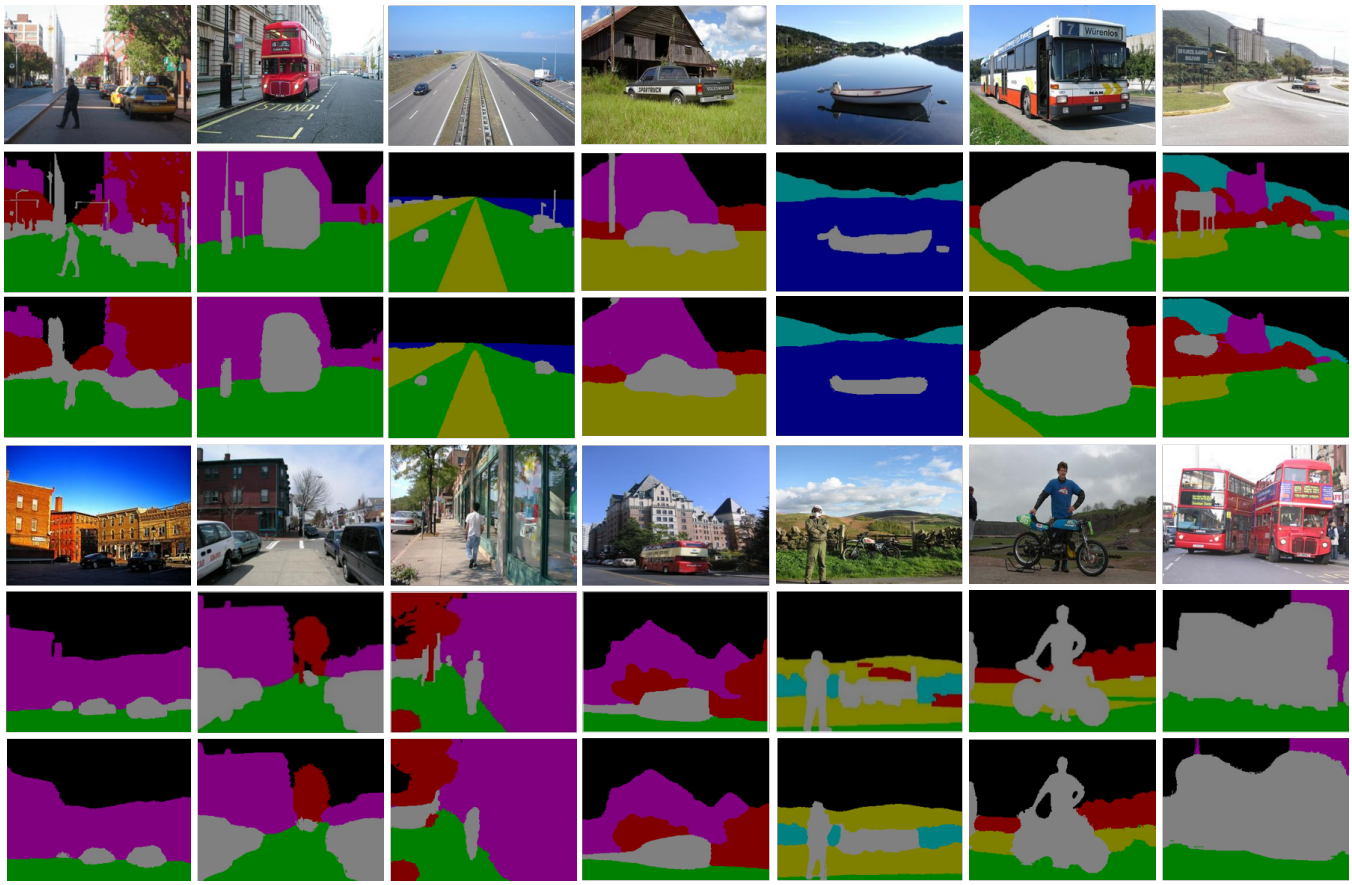

Figure 3. Sample results of semantic segmentation on the Stanford background dataset [6]. 1st row: test images; 2nd row: ground-truth semantic segmentations; 3rd-row: segmentation results produced by ConvNet-CSVM2. Different semantic classes are represented by different colors.

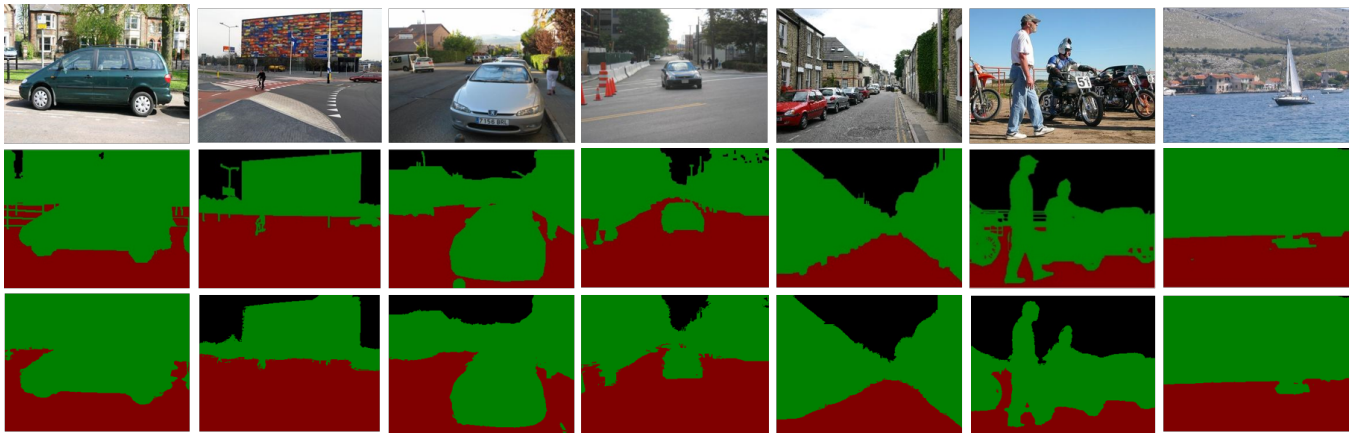

Figure 4. Sample results of geometric labeling on the Stanford background dataset [6]. 1st row: test images; 2nd row: ground-truth geometric labels; 3rd-row: geometric labeling results produced by ConvNet-CSVM2. Different geometric classes are represented by different colors.

	background	aeroplane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	**IoU**
FCN-8s [12]	91.2	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
Zoom-out [14]	89.8	81.9	35.1	78.2	57.4	56.5	80.5	74.0	79.8	22.4	69.6	53.7	74.0	76.0	76.6	68.8	44.3	70.2	40.2	68.9	55.3	64.4
DeepLab-CRF [3]	93.1	84.4	54.5	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	59.7	82.2	50.4	73.1	63.7	71.6
DeConvNet+CRF [15]	92.9	87.8	41.9	80.6	63.9	67.3	88.1	78.4	81.3	25.9	73.7	61.2	72.0	77.0	79.9	78.7	59.5	78.3	55.0	75.2	61.5	70.5
CRFasRNN [23]	-	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
ConvNet	89.5	72.1	29.9	73.5	56.7	64.3	81.1	73.9	77.4	27.2	62.0	49.6	70.8	61.3	66.8	75.8	42.3	66.3	41.5	73.3	49.7	62.1
ConvNet-CSVM	83.0	79.2	30.1	77.5	54.3	67.4	80.8	75.4	76.0	29.6	62.3	53.2	68.5	63.1	68.1	75.4	46.2	69.7	40.8	73.8	52.6	63.2
ConvNet-CSVM2	86.2	77.5	29.4	78.1	54	66.9	83.7	77.1	76.7	32.8	63.2	52.9	73.2	63.4	70.4	77.5	44.6	70.1	40.8	53.1	74.3	64.1
ConvNet-WSVM	87.0	77.7	29.5	78.0	57	67.1	83.7	77.0	78.3	32.9	62.9	53.0	73.5	63.2	70.9	77.4	45.8	70.2	40.2	54.7	74.4	64.5

Table VII
QUANTITATIVE RESULTS OF DIFFERENT APPROACHES FOR THE SEMANTIC SEGMENTATION TASK ON THE PASCAL VOC 2012 DATASET [4].

Figure 5. Sample results of semantic segmentation on the PASCAL VOC 2012 dataset [4]. 1st column: test images; 2nd column: ground-truth semantic segmentations; 3rd-column: segmentation results produced by ConvNet-CSVM2. Different semantic classes are represented by different colors.

Method	overall (PPA)	average (MCA)
Gould et al. [6]	76.4	-
Pylon [10]	81.9	72.4
RCN [16]	80.2	69.9
Multiscale Net [5]	81.4	76.0
TM-RCPN [17]	82.3	79.1
DeconvNet-16 [13]	84.2	78.4
LSTM-RNN [1]	78.56	68.79
CNN-CRF [11]	83.5	76.9
Zoom-Out [14]	86.1	80.9
ConvNet	90.4	89.1
ConvNet-SVM	90.8	89.3
ConvNet-CSVM	91.2	89.6
ConvNet-CSVM2	**91.6**	87.8
ConvNet-CSVM3	90.9	**90.4**
ConvNet-WSVM	91.0	**90.4**

Table V
COMPARISON WITH STATE-OF-THE-ART SEMANTIC SEGMENTATION APPROACHES ON STANFORD BACKGROUND (SEMANTIC) DATASET [6].

Method	overall (PPA)	average (MCA)
Gould et al. [6]	89.1	-
ConvNet	93.5	92.0
ConvNet-SVM	94.0	92.8
ConvNet-CSVM	93.8	93.1
ConvNet-CSVM2	95.1	**94.0**
ConvNet-WSVM	**95.2**	**94.0**

Table VI
COMPARISON WITH GOULD ET AL. [6] ON THE GEOMETRIC LABELING TASK ON THE STANFORD BACKGROUND DATASET.

We can see that our proposed approach outperforms all the baseline methods.

D. Evaluation result on PASCAL VOC 2012

The segmentation results on PASCAL VOC 2012 test set for different configurations are reported in Table VII. Following [12], [3], [23], we have used augmented training data with extra annotation for training the deep network. However, for training the SVM model, we didn't use any images other than the PASCAL VOC 2012 training set. Initially we achieve performance of 63.2 mean IoU for ConvNet-SVM and 64.5 IoU for Convnet-WSVM. Sample

segmentation outputs are illustrated in Fig. 5. It is important to note that there is evidently an advantage in making use of the data and labels from the SBD that are not directly related to the PASCAL VOC 2012 problem, and this suggests value in the proposed approach in a more general sense given future availability of datasets that include dense pixel-wise labeling.

V. Conclusion

We present the problem of supervised semantic segmentation based on pixel-wise class label assignments at a coarse level of abstraction. Our proposed approach can produce semantically accurate predictions. The novelty of our approach is the integration of deep convolutional neural networks with image-specific weighted support vector classification, and demonstration of the value in leveraging distinct and heterogeneous datasets. Experimental results demonstrate the effectiveness of our approach.

There are many potential directions to extend this work. Continuation of this work will involve design of an end-to-end segmentation network motivated by the principles of the VGG-16 network [18] as well as aiming to study and visualize how the depth of neural nets interacts with semantic segmentation results. Further work will involve developing a framework to solve problems related to semantic segmentation improving on existing capabilities, and integrating additional sources of imagery and labeling.

Acknowledgment

This work is supported by NSERC and the University of Manitoba Research Grants Program (URGP). We gratefully acknowledge the support of NVIDIA Corporation with the GPU donation used in this research.

References

[1] W. Byeon, T. M. Breuel, F. Raue, M. Liwicki. Scene labelling with LSTM Recurrent Neural Networks. In *CVPR*, 2015.

[2] G. Cardinal, X. Boix, J. van de Weijer, A. D. Bagdanov, J.Serrat, and J. Gonzalez. Harmony Potentials for Joint Classification and Segmentation. In *CVPR*, 2010.

[3] L-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic Image Segmentation with Deep Convolutional nets and Fully Connected CRFs. In *ICLR*, 2015.

[4] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge a Retrospective. *International Journal of Computer Vision (ICCV)*, 111(1), pp. 98–136, 2015.

[5] C. Farabet, C. Couprie, L. Najman, Y. LeCun. Learning Hierarchical Features for Scene Labeling. *PAMI*, 35(8), pp. 1915–1929, 2013.

[6] S. Gould, R. Fulton, D. Koller. Decomposing a Scene into Geometric and Semantically Consistent Regions. *Proceedings of International Conference on Computer Vision (ICCV)*, 2009.

[7] D. Hoiem, A. Efros, M. Hebert. Geometric Context from a Single Image. *IEEE ICCV*, 2005.

[8] J.Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *CoRR, vol. abs/1408.5093, 2014*, 2014.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet Classication with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, pp. 1106–1114, 2012.

[10] V. Lempitsky, A. Vedaldi, A. Zisserman. Pylon Model for Semantic Segmentation. *Advances in Neural Information Processing Systems*, pp. 109–117, 2011.

[11] F. Liu, G. Lin, C. Shen. CRF Learning with CNN Features for Image Segmentation. In *Pattern Recognition.48*, 2015 pp. 2893–2992.

[12] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[13] R. Mohan. Deep Deconvolutional Network for Scene Parsing. In *CVPR*, 2014.

[14] M. Mojtabi, P. Yadollahpour, G. Shakhnarovich. Feed-forward Semantic Segmentation with Zoom-out Features. In *CVPR*, 2015.

[15] H. Noh, S. Hong, and B. Han. Learning Deconvolution Network for Semantic Segmentation. *IEEE International Conference on Computer Vision*, 2015.

[16] P. O. Pinheiro, R. Collobert. Recurrent Convolutional Neural Network for Scene Parsing. In *ICML*,China, 2015.

[17] A. Sharma, Oncel Tuzel, David W. Jacobs. Deep Hierarchical Parsing for Semantic Segmentation. In *CVPR*, 2015.

[18] K. Simonyan, A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

[19] J. Shotton, J. Winn, C. Rother, A. Criminisi. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation. *European Conference on Computer Vision*, 2006.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich. Going Deeper with Convolutions. In *CVPR*, 2014.

[21] X. Yang, Q. Song, A. Cao. Weighted Support Vector Machine for Data Classification. In *International Joint Conference on Neural Networks*, 2005.

[22] J. Yao, S. Fidler, R. Urtasun. Describing the Scene as a Whole: Joint Object Detection, Scene Classification and Semantic Segmentation. In *CVPR*, 2012.

[23] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional Random Fields as Recurrent Neural Networks. In *International Conference on Computer Vision (ICCV)*, 2015.