SUPERVISED IMAGE SEGMENTATION VIA GROUND TRUTH DECOMPOSITION

Ilya Levner, Russell Greiner, and Hong Zhang

Department of Computing Science
University of Alberta
{ilya|greiner|zhang}@cs.ualberta.ca

ABSTRACT

This paper proposes a data driven image segmentation algorithm, based on *decomposing the target output (ground truth)*. Classical pixel labeling methods utilize machine learning algorithms that induce a mapping from pixel features to individual pixel labels. In contrast we propose to first extract features from both images *and* labels. Subsequently we induce a mapping from pixel features to label features and synthesize the final output by combining the newly derived label components. We demonstrate the effectiveness of the proposed approach by applying log-Gabor filters to both input and ground truth images of mineral ore. Subsequently we train perceptrons and regression trees to produce individual output components that are combined in frequency space to create the final segmentation. Experimental results show significant improvements over contextual pixel labeling and over ensemble methods.

1. INTRODUCTION

Segmenting an image into foreground regions versus background is an important task in image processing and computer vision. However, this pixel labeling task has yet to be solved, in a fully automated fashion, for non-trivial domains. This research aims to take us one step closer to realizing this goal by presenting an extension to the classical supervised segmentation approach to the pixel labeling problem.

While classical pixel labeling methods aim to map pixels or their extracted features into labels, we propose to first decompose the ground truth and map input features into output features. Once output features have been produced, by employing an ensemble of function approximators, they are synthesized into the final image labeling. The output decomposition scheme allows each function approximator to focus on a different aspect of the pixel labeling problem. The goal of the approach is to operate in between low level pixel labels and high level objects. As an example consider the rock objects depicted in Figure 1. The contour of each rock is composed of differently oriented edges. For each oriented edge we can train a specific detector and then synthesize the object contour by combing the outputs of each detector. Hence by decomposing the ground truth into a set of object parts describing a patch of ground truth labels we can brake down a difficult pixel labeling problem into a set of easier structure labeling problems.

Section 2 reviews the basic methodology of pixel labeling and supervised segmentation as well as presenting a high level description of our Output Decomposition Mixtures of Experts (OD-MoE) algorithm. Section 3 outlines the specific details of OD-MoE. Section 4 presents experimental results on segmentation of mineral ore images. Section 5 concludes the paper with final thoughts and future research directions.

2. CONTEXTUAL PIXEL LABELING

Formally let (i, j) index a discrete set of sites on a spatially regular $N \times M$ lattice:

$$S = \{(i, j) | 1 \leq i \leq N, 1 \leq j \leq M\} \tag{1}$$

For a training image $\boldsymbol{I}$ and the corresponding ground truth $\boldsymbol{L}$, let $\boldsymbol{I}(i, j)$ and $\boldsymbol{L}(i, j) \in \{0, 1\}$ respectively denote the intensity values of image pixels and the corresponding (binary) labels at site (i, j). Classical pixel labeling attempts to find the mapping:

$$h_{pl} : \boldsymbol{I}(i, j) \mapsto \boldsymbol{L}(i, j) \tag{2}$$

The process in Equation 2 treats individual pixels as i.i.d. (independent identically distributed). Unfortunately, this assumption is rarely satisfied in practice, since most non-trivial domains exhibit complex pixel interactions. Therefore, simply using raw pixel values for classification results in very poor segmentation. To overcome this problem, contextual pixel labeling defines a feature extraction function $\boldsymbol{f}^{\boldsymbol{I}}(i, j)$ that computes local (and possibly global) contextual features for each image pixel $\boldsymbol{I}(i, j)$. Subsequently the newly formed feature vectors are used to learn the mapping:

$$h_{cpl} : \boldsymbol{f}^{\boldsymbol{I}}(i, j) \mapsto \boldsymbol{L}(i, j) \tag{3}$$

To further improve pixel classification accuracy, recursive contextual pixel classification [1] and, more recently, random field methods (e.g., Markov / Conditional / Discriminative Random Fields [2, 3]) have been designed to account for label interactions as well as input pixel interactions. These systems first use the regular contextual pixel labeling, as in Equation 3, to produce an initial labeling $\boldsymbol{L}_0$. Subsequently a recursive procedure iteratively computes $\boldsymbol{L}_d$ as follows:

$$h_{rcpl} : [\boldsymbol{f}^{\boldsymbol{I}}(i, j), \boldsymbol{f}^{\boldsymbol{L}_{d-1}}(i, j)] \mapsto \boldsymbol{L}_d(i, j) \tag{4}$$

where $\boldsymbol{f}^{\boldsymbol{L}_{d-1}}(i, j)$ is a function extracting features from $\boldsymbol{L}_{d-1}$ at lattice site (i, j). Typically the features extracted from $\boldsymbol{L}$ are very simple, usually just a neighborhood centered about (i, j) (i.e., cliques). In contrast to the aforementioned approaches, our approach tackles the problem from a different point of view. We propose to explicitly extract contextual features from both input images and ground truth images, and subsequently learn a mapping from the former to the latter:

$$h_{\text{OD-MoE}} : \boldsymbol{f}^{\boldsymbol{I}}(i, j) \mapsto \boldsymbol{f}^{\boldsymbol{L}}(i, j) \tag{5}$$

as depicted in Figure 1. The essence of the algorithm lies in extracting output features, $\boldsymbol{f}^{\boldsymbol{L}}$, that allow the synthesis of output $\boldsymbol{L}$. Once the input/output decomposition scheme has extracted the input and output features, we utilize machine learning algorithms to train a set of models that instantiate the mapping in Equation 5. At runtime, the output of individual models is fused together in order to produce the final segmentation $\tilde{\boldsymbol{L}}$.

3. OUTPUT DECOMPOSITION BASED MIXTURE-OF-EXPERTS

For simplicity, let us consider a single feature extraction function, extracting k features for each site (i, j), and applicable to both the input image $\boldsymbol{I}$ and the output labels $\boldsymbol{L}$. Using this function we produce a set of input feature maps $\{\boldsymbol{\Phi}_t^{\boldsymbol{I}}\}_{t=1}^k$ and a set of output feature maps $\{\boldsymbol{\Phi}_t^{\boldsymbol{L}}\}_{t=1}^k$. For lattice site (i, j), the feature vectors are therefore given by:

$$\boldsymbol{f}^{\boldsymbol{I}}(i,j) = [\Phi_1^I(i,j), ..., \Phi_k^I(i,j)]$$

$$\boldsymbol{f}^{\boldsymbol{L}}(i,j) = [\Phi_1^L(i,j), ..., \Phi_k^L(i,j)]$$

Using these input/output feature maps we train a set of function approximators $H = \{h_1, ..., h_k\}$ as:

$$h_t : \boldsymbol{f}^{\boldsymbol{I}}(i,j) \mapsto \tilde{\Phi}_t^L(i,j) \tag{6}$$

that map input feature vectors, to individual output components. Pictorially, each induced mapping h_t corresponds to an expert depicted by solid arrows in Figure 1.

Fourier based Feature Extraction and Output Synthesis

The discrete 2-D Fourier transform [4] of a function $g(i, j)$ and its inverse, each defined on lattice S from Equation 1, are given by:

$$G(u,v) = \frac{1}{\sqrt{NM}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} g(i,j) e^{-(\sqrt{-1})2\pi(\frac{ui}{N} + \frac{vj}{M})}$$

$$g(i,j) = \frac{1}{\sqrt{NM}} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} G(u,v) e^{(\sqrt{-1})2\pi(\frac{ui}{N} + \frac{vj}{M})}$$

To simplify notation we define:

$$\mathcal{F}[g] = G \; ; \; \mathcal{F}^{-1}[G] = g$$

to denote the Fourier Transform and its inverse. Next, we define a set of frequency filters $\{G_t(u, v)\}_{t=1}^k$ with:

$$\sum_{t=1}^{k} G_t(u,v) = 1, \forall (u,v) \in S$$

The frequency feature coefficients of an arbitrary function $q(i, j)$, defined over lattice S, are then simply the point-wise product of filter G_t with the frequency representation of the function, $Q \odot G_t = Q(u, v)G_t(u, v)$, $\forall (u, v) \in S$, with $Q = \mathcal{F}[q]$. The spatial feature maps of $q(i, j)$ are therefore defined as:

$$\Phi_t^q = \mathcal{F}^{-1}[Q \odot G_t], t = \{1, ..., k\}$$

Finally, for this specific decomposition scheme, the reconstruction (i.e., **the output synthesis**) function is defined in the Fourier domain as:

$$\tilde{Q} = \sum_{t=1}^{k} \mathcal{F}[\Phi_t^q] \odot G_t^{\boldsymbol{\alpha}} \tag{7}$$

where $\boldsymbol{\alpha} = 0$ is the default approach[1]. By setting $\boldsymbol{\alpha} \geq 1$, the algorithm is able to perform online filtering. Experimental results in Section 4, will demonstrate the ability of this filtering procedure to attenuate noise resulting from function approximation.

[1] In the case of $\boldsymbol{\alpha} = 0$ we can perform the summation in the spatial domain as $\tilde{q} = \sum_t \Phi_t^q$

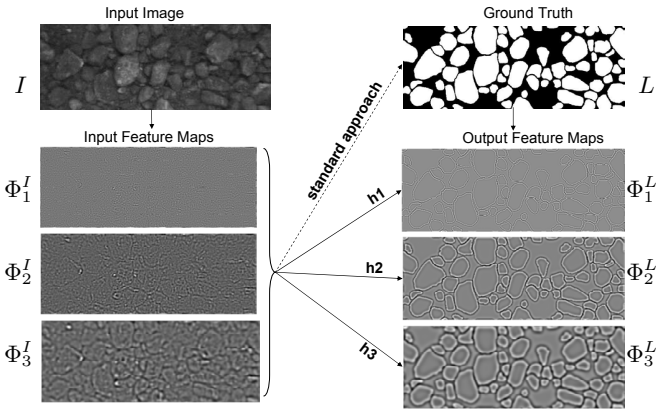

Fig. 1. Output Decomposition based Mixture of Experts approach. **Left:** Input Image, $\boldsymbol{I}$, and the extracted feature maps, $\Phi_1^I, \Phi_2^I, \Phi_3^I$. **Right:** Corresponding Ground Truth image, $\boldsymbol{L}$, and the extracted feature maps, $\Phi_1^L, \Phi_2^L, \Phi_3^L$. For demonstrational purposes we applied the Difference of Gaussians (DoG) decomposition scheme to both $\boldsymbol{I}$ and $\boldsymbol{L}$. The following mappings are represented: (dashed arrow) from input features to pixel labels as in the case of the **standard approach** defined by Equation 3; (solid arrows) from input features to output features as in the case of h_1, h_2, h_3 corresponding to the OD-MoE approach defined in Equation 6.

Runtime OD-MoE

At runtime the following steps are carried out :

1. Extract image features by convolving the input image with a filter bank $\{G_t\}_{t=1}^k$:

   $$\boldsymbol{\Phi}_t^{\boldsymbol{I}} = \mathcal{F}^{-1}\left[\mathcal{F}[\boldsymbol{I}] \odot G_t\right] \; \forall t = \{1, ..., k\}$$

   or equivalently:

   $$\boldsymbol{\Phi}_t^{\boldsymbol{I}} = \boldsymbol{I} * \mathcal{F}^{-1}[G_t] \; \forall t = \{1, ..., k\}$$

   where $*$ is the convolution operator.

2. Perform function approximation by applying the trained ensemble of function approximators defined by Equation 6

   $$\tilde{\boldsymbol{\Phi}}_t^{\boldsymbol{L}}(i,j) = h_t(\boldsymbol{f}^{\boldsymbol{I}}(i,j)) \; \; \forall (i,j) \in S$$

3. Produce the output labels by combining the output of function approximators:

   $$\tilde{\boldsymbol{L}} = \mathcal{F}^{-1}\left[\sum_{t=1}^{k} \mathcal{F}[\tilde{\Phi}_t^L] G_t^{\alpha}\right]$$

4. EMPIRICAL EVALUATION

Evaluation Criteria

To evaluate the performance of each algorithm several criteria were used. Respectively, TP, TN, FP, and FN, stand for the number of samples (i.e., pixels) being labeled as true positive, true negative, false positive, and false negative.

Jaccard measure is defined as $\frac{TP}{TP+FP+FN}$, and for binary label images A and B is also known as intersection-over-union measure, denoted by $\frac{|A \cap B|}{|A \cup B|}$.

Pixel Accuracy is defined as $\frac{TP+TN}{TP+TN+FP+FN}$.

Precision is defined as $\frac{TP}{TP+FP}$.

Recall is defined as $\frac{TP}{TP+FN}$.

Label score is defined as $L = min(S(A,B), S(B,A))$,

$$S(A,B) = \sum_{j}^{m} \left[\sum_{i}^{n} \left(\frac{|A_j \cap B_i|}{|A_j \cup B_i|} \frac{B_i}{\bigcup\limits_{|A_j \cap B_i| \neq 0} B_i} \right) \frac{A_j}{\bigcup\limits_{j} A_j} \right]$$

where A_j is a connected component in image A and B_i is a connected component in image B. This labeling score is a form of local intersection-over-union (I/U) whereby both errors at the pixel level and object level are penalized.

Experimental Design

To test the proposed approach, we had a granulometry expert manually label nine, 236×637 pixel images containing mineral ore (see Figures 1 and 2 for examples). All algorithms, described below, were trained on image 1 and performance tested on images 2-9[2]. Feature extraction, for both the input images and output labels, was performed using a bank of log-Gabor filters as in [5], with 7 scales and 6 orientations for a total of 42 filters. Since the inverse Fourier transform of filter responses contains both real and imaginary parts, we further separated the feature maps into two components corresponding to the even (real) and odd (imaginary) filter responses. As a result, both inputs and outputs were decomposed into 84 feature maps. Each of the 84 experts was focused on a single output feature map and was trained using either: (a) linear regression, or (b) regression trees [6]. The gating function was designed as in the previous section. To test the efficacy of frequency domain filtering we run our system with $\boldsymbol{\alpha} \in \{0, 1\}$, corresponding to, **Raw** (i.e., non-filtered) and **Filtered** outputs in Equation 7. For further comparison we also implemented the **standard** contextual pixel labeling approach, defined by Equation 3, that used the same input features to directly output labels (depicted by a dashed arrow in Figure 1). In addition, we also tested several typical ensemble methods based on bagging. The first version was based on the original bootstrap version of bagging from [7] whereby 60% of the training samples were randomly selected for training each ensemble member. The second version was loosely based on [8], whereby we randomly permuted 10% of the labels within the training set for each member of the ensemble. Each version was tested with 40 and 80 ensemble members.

Results

Experimental results are presented in Table 1 with examples of test output presented in Figure 2. In terms of pixel accuracy and Jaccard measures, all algorithms performed comparably. However, significant differences exist in terms of precision, recall and label score. Regardless of the base classifier, the OD-MoE system produces results with high precision and low recall when compared to the rest of the tested algorithms (standard regression and bagging). In turn, that results in significantly better label score which recall was designed specifically to evaluate object level information, namely the number of objects, their location and boundaries. From this perspective, our algorithm is far more suitable to the object delineation task(s) as indicated by a label score, of 0.43 for OD-MoE(Filtered) using regression trees, which is almost three times better than the competition. As mentioned in the introduction, the identification of object parts lies at the heart of the output decomposition function. Each log-Gabor filter identifies various frequency components comprising the target objects. In turn these components are easier to learn than the unstructured pixel labels. It is thus no surprise that the algorithm improves the labeling at the object rather than the pixel level. The difference in segmentation quality can be readily observed in Figure 2. With respect to bagging, although there was an increase in performance at the pixel level when compared to using a single regression tree, the label score clearly remains unaffected. Visually the output looked very similar to the output of a single regression tree[3] which provides a stark contrast to the output of OD-MoE.

Examining the merit of frequency domain filtering, we can see mixed results. When linear regression is used to construct the individual experts, the filtering step is detrimental to performance. However, when regression trees are employed at the base level, overall performance improves. Our results consistently demonstrate that regardless of the base regressor used, precision score improve as a result of the filtering step. Visually, in Figure 2, we can see a significant reduction in noise when frequency domain filtering is used in conjunction with regression trees.

Table 1. Average performance on test images. **Standard** denotes regression performed directly on the ground truth as is commonly done for pixel labeling. **Raw** denotes OD-MoE without filtering. **Filtered** denotes OD-MoE with filtering prior to output reconstruction. **Bag** denotes a standard bagging procedure with each member of the ensemble using randomly selected 60% of the training samples. **RPL_Bag** denotes bagging where 10% of the labels were randomly permuted for each ensemble member. Ensemble sizes are shown in brackets.

Linear Regression as base learner

Algorithm	jacq	acc	prec	recall	label score
Standard	**0.67**	0.75	0.78	0.84	0.14
OD-MoE(Raw)	0.64	**0.76**	0.86	0.72	0.40
OD-MoE(Filtered)	0.59	0.73	**0.87**	0.64	0.38

Regression Tree as base learner

Algorithm	jacq	acc	prec	recall	label score
Standard	0.52	0.62	0.68	0.69	0.04
OD-MoE(Raw)	0.61	0.74	0.85	0.69	0.41
OD-MoE(Filtered)	0.62	0.75	0.86	0.68	**0.43**

Bagged Regression Trees

	jacq	acc	prec	recall	label score
Bag(40)	0.63	0.70	0.71	**0.86**	0.04
Bag(80)	0.63	0.71	0.71	0.85	0.04
RLP_Bag(40)	0.59	0.69	0.74	0.74	0.14
RPL_Bag(80)	0.59	0.69	0.74	0.75	0.14

5. CONCLUSION

This paper presented the Output Decomposition Mixture-of-Experts (OD-MoE) algorithm designed for pixel labeling and object delineation tasks. By using an output decomposition function to identify coherent parts of the objects, each function approximator, comprising the mixture of experts, can be focused on specific object aspect. Experimental results indicate that these primitive object structures are easier to identify than the individual pixel labels and enable OD-

[2]Similar results to those presented in this paper were obtained using different train test splits.

[3]The output from Bag(40) and Bag(80) looked very similar to the output of standard regression tree, while the output from RLP_Bag(40) and RLP_Bag(80) looked very similar to the output of standard linear regression.

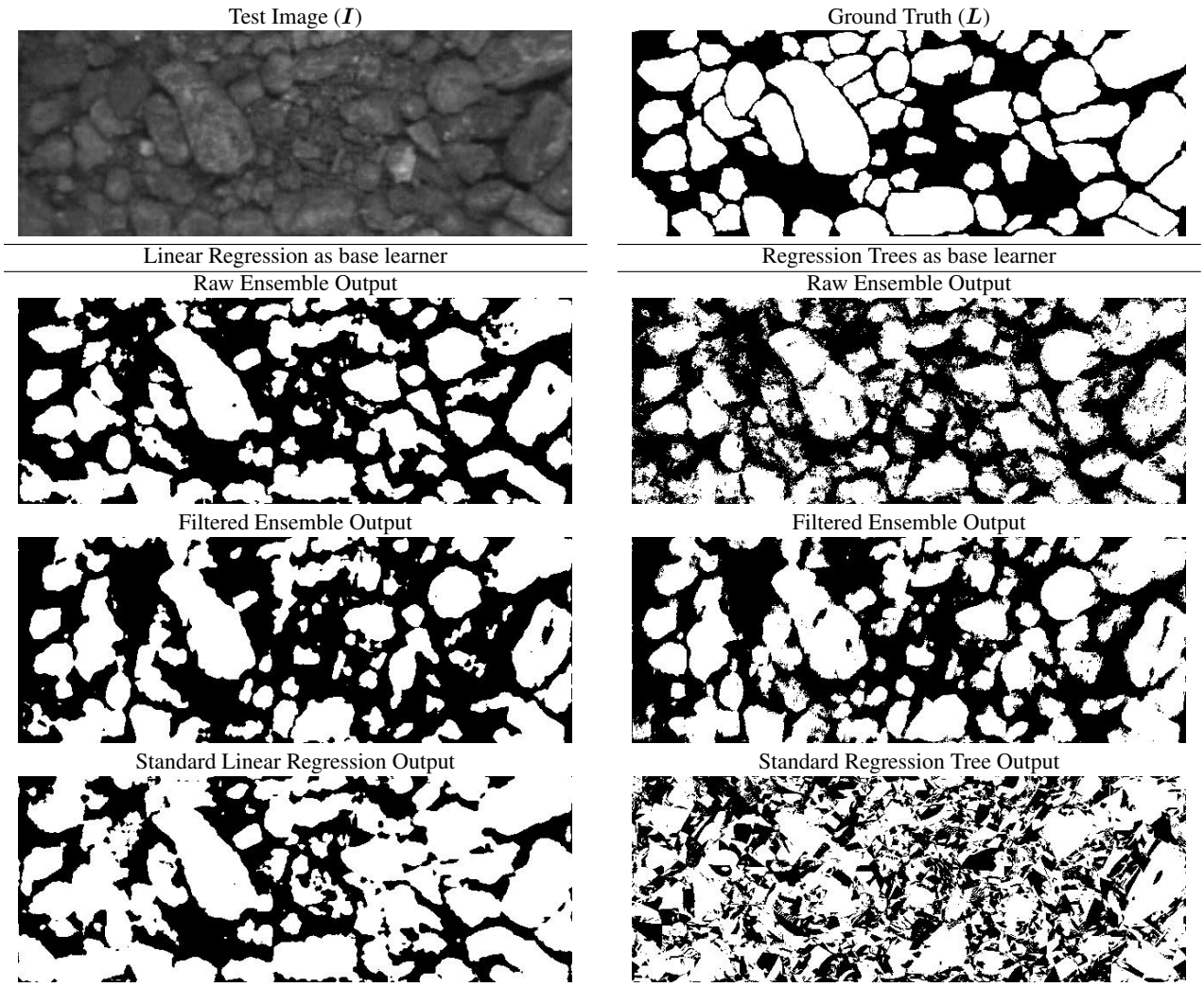

Fig. 2. **Top Row:** Test input image and corresponding ground truth. Test output using Linear Regression as base learner **Left**, and Regression Tree as base learner **Right**.

MoE to produce results superior to those of standard pixel labeling algorithms. More specifically, OD-MoE is able to separate out individual objects, while the basic pixel labeling algorithms, devoid of higher level knowledge of objects, consistently fused several objects together. In addition, the proposed log-Gabor filters, used for output decomposition, have additional uses as noise filters, which enable further mitigation of function approximation errors. Overall the experimental results indicate that the proposed approach is highly suitable for image interpretation tasks.

6. REFERENCES

[1] T.S. Yu and K.S. Fu, "Recursive contextual classification using a spatial stochastic model," *Pattern Recognition*, vol. 16, no. 1, pp. 89–108, 1983.

[2] John Lafferty, Andrew McCallum, and Fernando Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th International Conf. on Machine Learning*. 2001, pp. 282–289, Morgan Kaufmann, San Francisco, CA.

[3] Sanjiv Kumar and Martial Hebert, "Discriminative fields for modeling spatial dependencies in natural images," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, December 2003.

[4] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Prentice Hall, 2 edition, 2002.

[5] Peter Kovesi, "Image features from phase congruency," *Videre: A Journal of Computer Vision Research*, vol. 1, no. 2, 1999.

[6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics. Springer Verlag, New York, 2001.

[7] Leo Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[8] Leo Breiman, "Randomizing outputs to increase prediction accuracy," *Machine Learning*, vol. 40, no. 3, pp. 229–242, 2000.