Meta-Evaluation of Image Segmentation Using Machine Learning

Hui Zhang, Sharath Cholleti, Sally A. Goldman
Dept. of Computer Science and Engineering, Washington University,
One Brookings Drive, St. Louis, MO 63130
{huizhang, cholleti, sg}@wustl.edu

Jason E. Fritts
Dept. of Mathematics and Computer Science, St. Louis University
221 N. Grand Blvd, St. Louis, MO 63103
jefritts@slu.edu

Abstract

Image segmentation is a fundamental step in many computer vision applications. Generally, the choice of a segmentation algorithm, or parameterization of a given algorithm, is selected at the application level and fixed for all images within that application. Our goal is to create a stand-alone method to evaluate segmentation quality. Stand-alone methods have the advantage that they do not require a manually-segmented reference image for comparison, and can therefore be used for real-time evaluation. Current stand-alone evaluation methods often work well for some types of images, but poorly for others. We propose a meta-evaluation method in which any set of base evaluation methods are combined by a machine learning algorithm that coalesces their evaluations based on a learned weighting function, which depends upon the image to be segmented. The training data used by the machine learning algorithm can be labeled by a human, based on similarity to a human-generated reference segmentation, or based upon system-level performance. Experimental results demonstrate that our method performs better than the existing stand-alone segmentation evaluation methods.

1. Introduction

Image segmentation is a fundamental step in many image, video and computer vision applications. Many segmentation methods have been developed, but there is still no satisfactory performance measure, which makes it hard to compare different segmentation methods, or even different parameterizations of a single method. However, the ability to compare two segmentations (generally obtained via two different methods/parameterizations) in an application-independent way is important: (1) to autonomously select among two possible segmentations within a segmentation algorithm or a broader application; (2) to place a segmentation algorithm on a solid experimental and scientific ground [2]; and (3) to monitor the segmentation results on the fly, so that segmentation performance can be guaranteed and consistency can be maintained [9].

Designing a good measure for segmentation quality is a known hard problem – some researchers even feel it is impossible. Each person has his/her distinct standard for a good segmentation and different applications may function better using different segmentations. While the criteria of a good segmentation are often application-dependent and hard to explicitly define, for many applications the difference between a favorable segmentation and an inferior one is noticeable. It is possible (and necessary) to design performance measures to capture such differences.

Human-aided methods are widely used in segmentation evaluation, either by relative measures that compute the discrepancy between one segmentation with a human-generated reference segmentation [10, 14], or by subjective measures in which humans evaluate the segmentation visually or qualitatively [11]. Although usually deemed more satisfactory, human-aided methods are subjective, tedious and time-consuming.

In contrast, stand-alone evaluation methods [3] evaluate a segmentation based on how well it matches the desired characteristics of a good segmentation, as based on human judgment [1, 12, 13, 17, 20, 23, 24]. Since they do not require reference images, these methods can operate over a wide range of image types and varying conditions, and can be used in real-time systems where a large number of unknown images need to be processed.

Often a segmentation algorithm is used as a preprocessing step for a larger system. It is natural to use the overall performance of an end system to evaluate the segmentation quality. A system-level evaluation would typ-

ically segment all images with each of a set of segmentation techniques (or parameterizations) being considered, and then select the one giving the best overall performance. However, even for a given application, one segmentation technique (or parameterization) may be best for some of the images, and another technique may be best for other images. An advantage of a stand-alone method is that it could be applied to each image to adaptively select the best segmentation technique to use on that particular image, possibly improving the performance obtained when using any one segmentation method for all images.

Current stand-alone evaluation methods usually examine different fundamental criteria for segmentation quality, or examine the same criteria in different ways. Each of them typically work well for some types of images, but poorly for others. We propose a meta-evaluation method in which any existing evaluation methods (called *base evaluators*) are combined by a machine learning algorithm that coalesces their evaluations based on a learned weighting function that is dependent upon the characteristics of the image being segmented. An advantage of our approach is that any base evaluator can be used without any change in our learning algorithm. Also, the training data used by the machine learning algorithm can be labeled by a human, based on similarity to a human-generated reference segmentation, or based upon system-level performance. An advantage of such a machine learning approach is that the resulting segmentation evaluator is tuned for the types of images upon which it is trained. Also, our method creates a decision tree for each base evaluator that provides information about which features are important in determining when that evaluator is reliable. The decision tree tailors the influence given to each base evaluator according to the image being segmented.

The remainder of the paper is organized as follows. In Section 3, we present our Meta-Segmentation Evaluation Technique ($MSET$). Experimental results are presented in Section 4. Section 5 describes a possible application of $MSET$ to the problem of dynamically selecting a segmentation technique within a system. Section 6 concludes the paper and discusses future work.

2. Related Work

This work addresses the limitations of our earlier Co-Evaluation framework [26] that combines a set of base evaluators using a machine learning algorithm (Naive Bayes, a Support Vector Machine, or the Weighted Majority algorithm) to combine the evaluation results from the base evaluators to obtain an overall evaluation. One limitation of Co-Evaluation is that the weight for each base evaluators is independent of the image being considered. However, each base evaluator typically excels for some types of images/segmentations, yet works poorly for others. Conversely, our new method, $MSET$, determines when each base evaluator will perform best, so the weight given to each base evaluator depends upon the original image and its segmentations being evaluated.

A large number of stand-alone evaluation methods have been proposed. Most of these methods consider factors such as region uniformity, inter-region heterogeneity, region contrast, line contrast, line connectivity, texture, and shape measures [3, 4, 12, 19, 20].

Liu and Yang [13] proposed the evaluation function $F(I) = \sqrt{N}\sum_{j=1}^{N}\frac{e_j^2}{\sqrt{S_j}}$ where N is the number of segments, S_j is the number of pixels in segment j, and e_j^2 is the squared color error of region j. Unless the image has very well-defined regions with very little variation in luminance and chrominance, the F evaluation function has a very strong bias towards under-segmentation (segmentations with very few regions). Borsotti *et al.* [1] improved upon Liu and Yang's method, proposing a modified quantitative evaluation (Q), where $Q(I) = \frac{\sqrt{N}}{1000 \cdot S_I}\sum_{j=1}^{N}\left[\frac{e_j^2}{1+\log S_j} + \left(\frac{N(S_j)}{S_j}\right)^2\right]$. The variance e_j^2 was given more influence in Q by dividing by the logarithm of the region size, and Q is penalized strongly by $\frac{N(S_j)}{S_j}$ when there are a large number of segments. So Q is less biased towards both under-segmentation and over-segmentation.

More recently, Zhang *et al*. [25] proposed the evaluation function E, an information theoretic approach, for segmentation evaluation. Instead of using squared color error, they use the Shannon entropy of the luminance of all pixels in a region to measure its uniformity, and define the *expected region entropy* of image I as the entropy across all regions where each regions has weight (or probability) proportional to its area. To prevent a bias towards over-segmentation, they define the *layout entropy* as the Shannon entropy of the object features of all pixels in image I where any two pixels in the same region have the same object feature. The evaluation function E is the sum of expected region entropy and layout entropy. Pal and Bhandari [17] also proposed an entropy-based segmentation evaluation measure for intra-region uniformity based on the second-order local entropy.

Several evaluation metrics designed for frames of a video can be easily modified for image segmentation evaluation [7, 9]. Correia and Pereira [7] proposed a set of metrics for both intra-object measure (such as shape regularity, spatial uniformity, etc.) and inter-object measure (such as contrast). Also the metrics are weighted based on a measure of how much a human reviewer's attention is attracted by each object. While these metrics are proposed for video segmentation quality measures, Zhang *et al*. [26] converted them into measures V_s and V_m for image segmentation quality measures by removing the motion and temporal related portions. For each region, the metrics used

are circularity and elongation ($circ\_elong$), and compactness ($compact$). The inter-region metric *contrast* is defined by $\sum_{i,j}(2DY_{i,j} + DU_{i,j} + DV_{i,j})/(4 \times 255 \times N_b)$ where N_b is the number of border pixels for the region, and for each pixel i, j in the image, $DY_{i,j}$, $DU_{i,j}$ and $DV_{i,j}$ are the maximum difference between the Y, U and V components between that pixel and its four neighbors. Also, contextual relevance metrics are used to capture the importance of a region i in terms of the human visual system (HVS). The difference between V_s and V_m is in the way these metrics are weighted, with V_s weighting the contrast more and V_m weighting the circularity and elongation more.

3. MSET: Meta-Segmentation Evaluation Technique

Machine learning algorithms aim to find hypotheses that explain provided training data. Meta-learning aims to learn the conditions under which each of a set of learning algorithms or applications perform best [22]. In this paper, we apply meta-learning to learn conditions when the base evaluators perform best. The goal of Meta-Segmentation Evaluation Technique ($MSET$) is to create a classifier that given an image I, and two segmentations of I, S_1 (created by one algorithm/parameterization) and S_2 (created by an alternate algorithm/parameterization), can accurately predict if S_1 or S_2 is a better segmentation for I.

Since $MSET$ is constructed from a set of components that can be independently modified, it provides a lot of flexibility. The way in which these components are combined is illustrated in Figure 1. Observe that if the base evaluators are stand-alone segmentation evaluation methods, we obtain a new stand-alone evaluation method. However, unlike typical stand-alone methods, both the selection of the features to use in the decision tree and the training data allows our evaluation method to be tailored to a particular application, yet still be applied without the need for human intervention.

We now describe the components of $MSET$.

Base Evaluator: Any segmentation evaluation method can be used as a base evaluator. In fact, system-level evaluation methods, or even human-aided evaluation methods, can be used as base evaluators if desired, enabling fundamentally different evaluation methods to be coalesced. However, in this paper, we focus on using existing stand-alone methods.

Features: Both the base evaluators and the learning component use a set of features to capture the important qualities of the image and the segmentation methods. Some of these features depend only on the image (*e.g.* overall color and texture information for the image itself), and other features depend upon the particular segmentation (*e.g.* number of segments, average color, texture and shape features

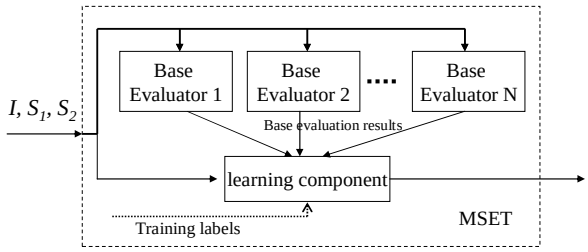

Figure 1. An Overview of $MSET$.

across all segments). Application-specific features can also be added.

Training Data: The learning component tailors its performance to a particular set of images through the training data, which is composed of a set of examples, each of which includes a raw image I, two segmentations (S_1 and S_2) for I, and a label indicating which of S_1 and S_2 is a better segmentation. The label could be provided by a human evaluator, by measuring the similarity to a human-generated reference segmentation, or based on which of S_1 and S_2 produces the better system-level performance.

Learning Algorithm: $MSET$ first constructs a decision tree [18] for each base evaluator. However, unlike the standard ID3 decision tree algorithm in which each leaf node gives a predicted label, each leaf node in the decision tree constructed by $MSET$ gives the predicted accuracy for that base evaluator for the input image. These decision trees are then used as a basis for defining a weighted vote of the base evaluators.

The decision tree for each evaluator is computed in the following manner. Each example in the training data is labeled as positive (if the evaluator agrees with the label), or otherwise negative. Each internal node in the decision tree partitions all of the examples into two or three sets according to the value of the selected attribute. In particular, the following options that depend upon the image itself are considered as possible internal nodes:

- Based on the LUV color space, we compute the average value across the image for color_L, color_U, and color_V. Then for each $Y \in \{L, U, V\}$ and each possible threshold value $t \in \{90, 128, 150\}$, we define an internal node where one partition contains images where color_Y $< t$, and the other where color_Y $\geq t$.
- Similarly, we compute the average value for wavelet coefficients in horizontal (HL), vertical (LH) and diagonal (HH) directions. For each of these texture features, we define an internal node with possible threshold values of 1.0, 1.5 and 2.0.

We also consider the following features that depend upon the segmentation: the number of segments (NoS), shape and texture features (perimeter, compactness, circularity,

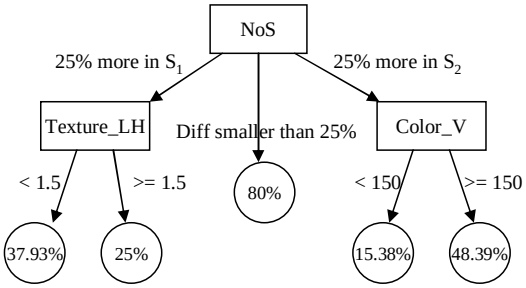

Figure 2. An example decision tree output by $MSET$ for base evaluator F.

elongation, Sobel and contrast as defined in [6]), averaged over all segments. For each of these features, and $x \in \{10, 25, 50\}$, we define a potential internal node with three branches based on whether S_1's value for the attribute is $x\%$ greater than S_2's value, the difference between S_1 and S_2's value is less than $x\%$, or S_2's value for the attribute is $x\%$ greater than S_1's value.

For all possible internal nodes (e.g. attributes and threshold choices to define the partitions), the one that maximizes the information gain is selected as the root where the information gain is the difference of entropy of the node and the sum of the entropies of the children based upon the proportion of the examples that are positive and negative in each branch [18]. The process is recursively repeated until either: the information gain is less than 0.01, or the number of examples for any child is less than 10, or the number of examples is less than 15% of the total examples in the training set. The final decision tree partitions the set of all images into equivalence classes with one equivalence class per leaf. Each leaf holds the percentage of the training examples that reach that leaf are correctly classified (as whether S_1 or S_2 is best) by the given base evaluator.

An example decision tree (for F) is shown in Figure 2. The number of segments (NoS) with $x = 25$ had the highest information gain and thus was selected as the root. The branch for image pairs in which the number of segments are within 25% of each other becomes a leaf node where the accuracy on the training data is 80%. The other two equivalence classes defined by the root are split again. The leftmost one is split using Texture_LH with a threshold of 1.5, and the rightmost one is split using Color_V with a threshold of 150. Their children are not split again and become leaf nodes. The accuracy of the evaluator on the entire data set is 46.27%, however, the decision tree has discovered attribute values for which this classifier performs very well (namely, when there is less than a 25% difference between the number of segments), achieving 80% accuracy on this class of images.

For each base classifier and its complement (where the selection of S_1 and S_2 is best is reversed), a decision tree is built as described above. We now describe how these decision trees are combined to predict the label for a new example X consisting of image I, and segmentations S_1 and S_2 created by the same segmentation algorithm/parameterization as for the training data. For decision tree T_i of the ith base evaluator, let $L_i(X)$ be the leaf reached for example X, and let $\alpha_{i,j}$ be the accuracy on the training examples for leaf j of tree T_i.

Let BC_1 be the set of base classifiers (or their complements) that indicate that S_1 is better than S_2, and let BC_2 be the set of base classifiers (or their complements) that indicate that S_2 is better than S_1. $MSET$ predicts that S_1 is better than S_2 if and only if

$$\sum_{i \in BC_1} 2^{|\alpha_{i,L_i(X)} - 0.5| \times 10} \geq \sum_{i \in BC_2} 2^{|\alpha_{i,L_i(X)} - 0.5| \times 10}.$$

That is, the final prediction is made according to a weighted vote where the weights are defined by the decision tree leaves reached by X. The multiplicative factor of 10 in the exponent is included so that a 10% increase in accuracy causes the weight given to that evaluator to be doubled. For instance, examples ending in a leaf with an accuracy of 0.6 receive a weight of 2, whereas examples ending in a leaf with an accuracy of 0.7 receive a weight of 4, and so on.

4. Experimental Results

In this section we describe experimental results to evaluate $MSET$. One obstacle in the research of segmentation evaluation is the lack of benchmark image sets. Due to availability, as well as to compare the performance of $MSET$ with the Co-Evaluation method, we acquired the same image sets as used in Co-Evaluation [26], which are in turn based on the Berkeley Segmentation Dataset [14] and the Military Graphics Collection [15]. We performed three sets of experiments that differ in both the type of images and the methods in which the segmentations are obtained, which result in radical differences in the performance of the base evaluators.

Since $MSET$ coalesces the results of base evaluators, the selection of base evaluators is crucial to its performance. A good selection of base evaluators must include those methods that evaluate as many criteria of a good segmentation as possible, thereby enabling $MSET$ to combine the results from the most comprehensive perspectives. However, in this paper, to compare with Co-Evaluation, we use the same five evaluators used by the Co-Evaluation method: F [13], Q [1], E [25] and V_s and V_m [26].

For each of the three experiments, we select a set of images $\mathcal{I}$ and for each $I \in \mathcal{I}$, we generate segmentations S_1 and S_2. We define a label of which segmentation is best using a subjective measure based on human visual evaluation. We then split these examples into two approximately equal-sized training and test sets. For the training set, the label is

Evaluation Methods	Accuracy
F	$17.99\% \pm 1.06\%$
Q	$14.07\% \pm 0.94\%$
E	$83.12\% \pm 1.00\%$
V_s	$17.41\% \pm 1.00\%$
V_m	$19.91\% \pm 0.96\%$
$CoE\text{-}WM$	$90.44\% \pm 1.07\%$
$MSET$	$93.89\% \pm 0.85\%$

Table 1. The results for Experiment 1 (mean $\pm$ 95% confidence interval).

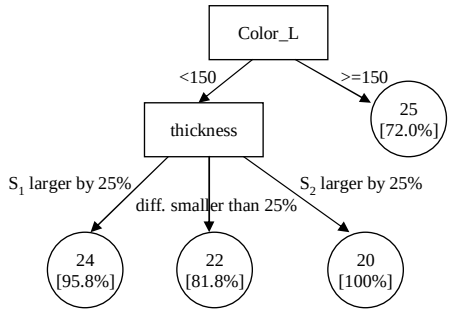

Figure 3. An example decision tree for base evaluator E.

included along with I, S_1, and S_2. For the test set, the final evaluator created by $MSET$ is given I, S_1 and S_2, but not the label. We measure performance by the *accuracy*, which is defined as the number of percentage of the test examples when the the predicted label (of whether S_1 or S_2 is a better segmentation of I) matches the unseen label.

For all experiments, we create 30 random splits of the examples into training and test sets, and report the mean accuracy and 95% confidence interval over the 30 runs. We compare the performance of $MSET$ with each of the five base evaluators, as well as that of the Co-Evaluation method using the weighted majority combiner ($CoE\text{-}WM$), which was reported to outperform other Co-Evaluation methods [26].

Experiment 1: Human segmentation results vs. machine segmentation results. In our first experiment, 189 images from the Berkeley Segmentation Database are used where S_1 is the provided segmentation created by a human [14] and S_2 is segmented by EDISON [8] to have the same number of segments as S_1. The label is always that S_1 is best (but none of the evaluators make use of this knowledge).

Table 1 shows the mean accuracy and the 95% confidence interval for each evaluation method. Recall that $MSET$ considers the base evaluators and their complements. Of these 10 evaluators, the best 5 accuracies are 82.01 (complement of F), 85.93 (complement of Q), 83.12 (E), 82.59 (complement of V_s), and 80.09 (complement of V_m). Observe that the improvement of $MSET$ over $CoE\text{-}WM$ is statistically significant, and both statistically outperform the base evaluators (and their complements).

The decision tree for E (for one of the 30 runs) is shown in Figure 3. (The overall accuracy of E on the training data used to create this tree was 86.8%.) The root of this decision tree partitions the examples based on whether or not the luminance of I is above 150. When the luminance is at least 150, a leaf node is created. For the images used in this experiment, high average luminance usually means there are large areas of uniform light background, such as sky, snow or sea. Those images are relatively simple and EDISON can segment them well. Consequently, the difference between human segmentation and EDISON segmentation is small, and it is more difficult for E to differentiate them. As a result E does not perform as well for the 25 examples whose luminance is over 150. For the remaining 66 examples (with luminance less than 150), E's accuracy is 92.4%.

For those examples where the luminance is low, a further split is based on the thickness [7] where the average thickness of a segmentation describes both the average area of each segment and its shape. A larger thickness indicates both a larger area and more circularity. When the percentage difference between S_1's and S_2's thickness is greater than 25%, E achieves a very high accuracy (95.8% and 100%). Since by definition E prefers those segmentations whose regions are less equally-sized, if the difference between thickness is high, one of the segmentations is more favorable to E. So, the decision tree achieves its goal of finding the type of examples in which E performs very well. For this data set, E is given more weight if the test image has low luminance, and the difference in thickness between the two segmentations is large.

Experiment 2: Results from different parameterizations of a segmentation method. In our second experiment, 249 aircraft images from the Military Graphics Collection are segmented by the Improved Hierarchical Segmentation (IHS) algorithm [27] with different parameterizations (namely, the number of segments in the final segmentation). The images used for this experiments were the ones where the human evaluators all agreed which segmentation was best. The label is given to the segmentation that was agreed to be better. In general, the variation in the number of segments was fairly high so that the human evaluators were all in agreement.

The results for this experiment are shown in Table 2. Of these 5 evaluators and their complements, the best 5 accuracies are 53.2 (complement of F), 73.67 (Q), 66.27 (complement E), 55.72 (V_s), and 62.88 (V_m). Thus, as compared with the first set of experiments, the base evaluators do not perform as well. One reason for this is that in about two-thirds of the training data, the segmentation with more segments is labeled as the better one. Both F and E have a

Evaluation Methods	Accuracy
F	$46.80\% \pm 1.00\%$
Q	$73.67\% \pm 1.13\%$
E	$33.73\% \pm 0.86\%$
V_s	$55.72\% \pm 1.07\%$
V_m	$62.88\% \pm 0.99\%$
$CoE\text{-}WM$	$77.14\% \pm 2.17\%$
$MSET$	$83.13\% \pm 1.12\%$

Table 2. The average evaluation accuracy (mean $\pm$ 95% confidence interval) for each method in experiment two.

Evaluation Methods	Accuracy
F	$37.05\% \pm 1.06\%$
Q	$47.11\% \pm 1.12\%$
E	$41.81\% \pm 1.19\%$
V_s	$57.30\% \pm 0.89\%$
V_m	$60.11\% \pm 1.06\%$
$CoE\text{-}WM$	$62.43\% \pm 0.94\%$
$MSET$	$65.37\% \pm 1.21\%$

Table 3. The evaluation accuracy (mean $\pm$ 95% confidence interval) for each method in experiment three.

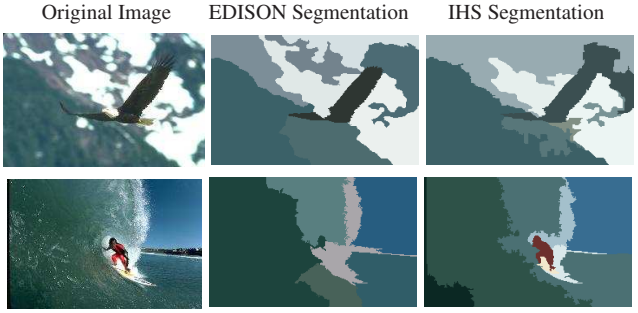

Figure 4. Image examples segmented by EDISON and IHS. In the top example, EDISON is labeled best, and in the bottom example IHS is labeled as best.

bias towards segmentations with fewer segments, and thus they didn't perform as well. However, the improvement of $MSET$ over $CoE\text{-}WM$ is statistically significant, and both statistically outperform the base evaluators (and their complements). Although the best base evaluator achieves only 73.57% accuracy, $MSET$ achieves an accuracy over 83%.

Experiment 3: Results from different segmentation methods. In our third experiment 268 images from the Berkeley Segmentation Database are segmented with both IHS and EDISON using approximately the same number of segments. Two examples are shown in Figure 4. A group of six human evaluators independently compare the segmentations from both algorithms. Only those images where at least four evaluators agreed which segmentation is best were used.

The results for this experiment are shown in Table 3. Of these 5 evaluators and their complements, the best 5 accuracies are 62.95 (complement of F), 52.89 (complement of Q), 58.19 (complement E), 57.30 (V_s), and 60.11 (V_m). Again, results show that the $MSET$ improvement over $CoE\text{-}WM$ is statistically significant, and they both outperform the base evaluators.

4.1. Discussion

In all three experiments, $MSET$ outperforms the other evaluation methods and this improvement is statistically significant at the 95% confidence level. Clearly, the performance of the base evaluators (and their complements) affects the performance of $MSET$. In experiment one, the accuracies of the evaluator are either very high (83.12%), or very low (14.07%$\sim$19.91%), so both $MSET$ and $CoE\text{-}WM$ have high accuracy (90.44%, 93.89%). In experiment three, when five evaluators have accuracy between 37% to 60%, i.e. most evaluators perform little better than random guess, both $MSET$ and $CoE\text{-}WM$ have low accuracy (62.43%, 65.37%). Experiment 2 falls between these two extremes.

For $MSET$ to obtain good results, two conditions are required: (1) For each image, some evaluator must perform well, and (2) the attributes used in constructing the decision tree must enable it to discriminate when each evaluator performs well. Thus, there are two ways in which the results for Experiment 3 could be improved. One possibility, is that by adding the right attributes (and thresholds) to use as possible internal nodes for the decision tree, conditions may be found in which each of the evaluators work well. As long as there is at least one high accuracy leaf reached for each example in the test set, then much better performance for $MSET$ could be achieved. We believe that larger improvements of $MSET$ over $CoE\text{-}WM$ can be obtained by finding more discriminative features to use in creating the decision trees.

Another way in which the results can be improved is to include better base evaluators. In fact, one nice feature of this work is that $MSET$ can improve as better stand-alone evaluation methods are developed. We are currently exploring the use of other stand-alone evaluation measures.

5. Possible Application for MSET

In this section, we briefly explore one possible application of a good stand-alone image segmentation method that dynamically selects the best segmentation technique to use for each individual image within an application (or even within the segmentation algorithm itself).

Since image segmentation usually depends on the content of the images to be segmented, most segmentation al-

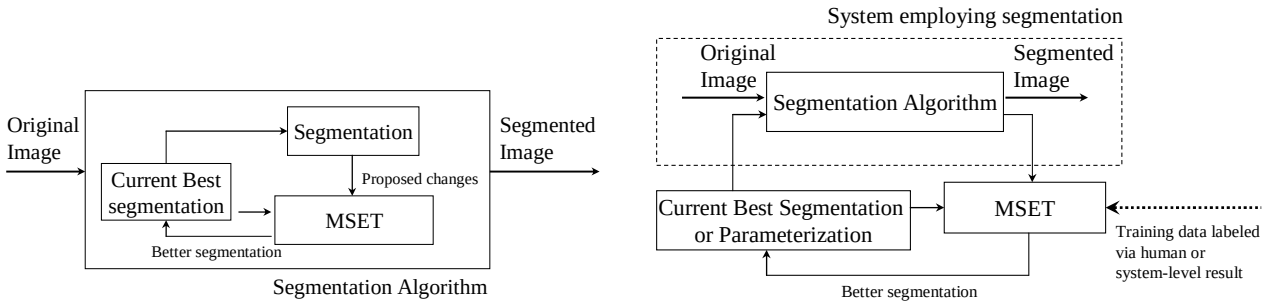

Figure 5. Self-tuning of image segmentation using $MSET$.

gorithms have tunable parameter(s) enabling them to be adjusted to the characteristics of different images. Some example parameters include feature distance thresholds for clustering or splitting, the minimum region and spatial bandwidth in mean-shift based segmentation [5], and the dataNcut in normalized cut segmentation [21].

Most segmentation methods are manually tuned using sample images. However, these parameters might not be appropriate for the segmentation of later images. A self-tunable segmentation method should be able to examine and evaluate its intermediate results, and automatically determine parameter options that generate better segmentation. $MSET$, when using stand-alone evaluations as base evaluation methods, can be embedded in a segmentation method or a system employing segmentation to do self-tuning.

There are two self-tuning paradigms, as illustrated in Figure 5. In *intra-algorithm tuning*, $MSET$ is embedded in a segmentation algorithm and evaluates each intermediate result of the segmentation, such that the resulting final output is improved over a segmentation using only fixed parameters values. For instance, in a region-growing segmentation method, $MSET$ can be used to evaluate the outcome of each merging of two regions and only complete the merge if the resulting segmentation is improved according to the $MSET$ evaluation. Such intra-algorithm tuning can be performed incrementally. Observe that any features that depend upon the image itself can be computed once, and when there are incremental changes in the segmentation, most of the features that depend upon the segmentation do not change. Thus incrementally maintaining the features can greatly reduce the computation needed to run the base evaluators, which is the dominant cost in using $MSET$ to select among a set of possible segmentations.

The second self-tuning paradigm is useful in situations where a segmentation algorithm is used as a black box. In such *algorithm-independent tuning*, $MSET$ evaluates every segmentation result by varying the tunable parameter(s) of a segmentation algorithm over a range of values, and choosing the parameters that lead to the most appropriate segmentation. This tuning paradigm can work with any segmentation algorithms, as long as $MSET$ has access to the segmentation results. For systems employing segmentation, the overall performance of the systems (*e.g.* the retrieval accuracy in content-based image retrieval system, or the recognition rate in target recognition system) can be used as the label of segmentation evaluation in the training process, so the segmentation can be self-tuned to attain the best system-level performance.

6. Conclusion

Current objective evaluation methods usually examine different fundamental criteria of good segmentation, and rely heavily on the image characteristics they are measuring. So they work well in some cases, or for some groups of images, and poorly for the others. To improve the evaluation accuracy, we propose a meta-segmentation evaluation technique, in which different evaluators judge the performance of the segmentation in different ways, and their measures are combined by a learning algorithm that determines how to coalesce the results from the constituent measures. Based on features extracted from the original image and each segmented image, and features defined for each segmentation, the learning module has the possibility of learning what base evaluator is most likely to generate reliable evaluations for each type of image, and can use this to weight the influence of each base evaluator in a way that is appropriate for the individual image.

The advantages of $MSET$ are its improved accuracy as compared to the current evaluation methods, the ability to use any base evaluators with it, the potential to combine fundamentally different types of evaluation methods, the possibility to evaluate the segmented images from different imaging technologies (*e.g.* segmentations of optical, radar and infra-red images of a target), and its parallel structure, which facilitates fast processing time. Also, the flexible nature of $MSET$ means that it is not necessary to find a single approach in order to obtain good stand-alone objective segmentation evaluation across all types of images. Rather, it is just necessary to find a set of base evaluation methods

such that at least one of them works for each type of image. Combined with a careful selection of attributes to use in constructing the decision tree, $MSET$ can combine such base evaluators to achieve good performance across all image types.

There are many interesting directions for future work. We will experiment with more extensive image sets and include a wider variety of base evaluators. We plan to further explore the selection of the features used by the decision tree algorithm. We also plan to consider learning techniques other than the variation of the ID3 decision tree algorithm that $MSET$ currently uses. Finally, we plan to explore the application of $MSET$ to create a self-tunable segmentation method as discussed in Section 5 to dynamically choose between different segmentation algorithms/parameterizations for each image.

7. Acknowledgment

The authors thank the generous support of NSF under grant 0329241 and we thank Rouhollah Rahmani, Ibrahim Noorzaie and Long Chen for helping create human evaluation results.

References

[1] M. Borsotti, P. Campadelli, and R. Schettini. Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters*, 19(8):741–747, 1998. 1, 2, 4

[2] K. W. Bowyer and P. J. Phillips. *Empirical Evaluation Techniques in Computer Vision*. Wiley-IEEE Computer Society Press, 1998. 1

[3] S. Chabrier, B. Emile, H. Laurent, C. Rosenberger, and P. Marche. Unsupervised evaluation of image segmentation application to multi-spectral images. *Proc. of the 17th Int. Conf. on Pattern Recognition*, 576–579, 2004. 1, 2

[4] H.-C. Chen and S.-J. Wang. The use of visible color difference in the quantitative evaluation of color image segmentation. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004. 2

[5] D. Comanicu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 24:603–619, 2002. 7

[6] P. Correia and F. Pereira. Estimation of video object's relevance. *Proceedings of European Signal Processing Conference*, 2000. 4

[7] P. Correia and F. Pereira. Objective evaluation of video segmentation quality. *IEEE Trans. on Image Processing*, 12(2):186–200, 2003. 2, 5

[8] Edge Detection and Image SegmentatioN System. http://www.caip.rutgers.edu/riul/research/code/EDISON/. 5

[9] C. E. Erdem, B. Sanker, and A. M. Tekalp. Performance measures for video object segmentation and tracking. *IEEE Trans. on Image Processing*, 13:937–951, 2004. 1, 2

[10] F. J. Estrada and A. D. Jepson. Quantitative evaluation of a novel image segmentation algorithm. *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2005. 1

[11] E. D. Gelasca, T. Ebrahimi, M. Farias, M. Carli, and S. Mitra. Towards perceptually driven segmentation evaluation metrics. *Proc. of Conf. on Computer Vision and Pattern Recognition Workshop*, 2004. 1

[12] M. D. Levine and A. M. Nazif. Dynamic measurement of computer generated image segmentations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(2):155–164, 1985. 1, 2

[13] J. Liu and Y.-H. Yang. Multi-resolution color image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(7):689–700, 1994. 1, 2, 4

[14] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Proc. 8th Int. Conf. Computer Vision*, 2:416–423, 2001. 1, 4, 5

[15] Military Graphics Collection. http://www.locked.de/en/ index.html. 4

[16] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[17] N. Pal and D. Bhandari. Image thresholding: some new techniques. *Signal Processing*, 33(2):139–158, 1993. 1, 2

[18] R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986. 3, 4

[19] C. Rosenberger and K. Chehdi. Genetic fusion: Application to multi-components image segmentation. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000. 2

[20] P. Sahoo, S. Soltani, A. Wong, and Y. Chen. A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing*, 41(2):233–260, 1988. 1, 2

[21] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 7

[22] R. Vilalta and Y. Drissi. A perspective view and survey of metalearning. *Artificial Intelligence Review*, 18(2):77–95, 2002. 3

[23] J. Weszka and A. Rosenfeld. Threshold evaluation techniques. *IEEE Trans. on Systems, Man and Cybernetics*, 8(8):622–629, 1978. 1

[24] Y. Yitzhaky and E. Peli. A method for objective edge detection evaluation and detector parameter selection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(8):1027–1033, 2003. 1

[25] H. Zhang, J. Fritts, and S. Goldman. An entropy-based objective evaluation method for image segmentation. *Proc. SPIE- Storage and Retrieval Methods and Applications for Multimedia*, 2004. 2, 4

[26] H. Zhang, J. Fritts, and S. Goldman. A co-evaluation framework for improving segmentation evaluation. *Proc. SPIE-Signal Processing, Sensor Fusion and Target Recognition*, 5809, 2005. 2, 4, 5

[27] H. Zhang, J. Fritts, and S. Goldman. An improved fine-grain hierarchical method of image segmentation. *Technical Report - Washington University in St. Louis*, 2005. 5