BusBot: A Stain Learning Approach

Anthony Chen and Detian Shi

***Abstract*— In this paper we decided to investigate the application of robotics to the domestic chore of using a sponge to wipe down a table after a meal. We divided the problem into two main components; stain identification and cleaning kinematics. For the stain identification sub-problem we mainly focused on developing algorithms that would enable a robot to correctly identify stains. We first developed and experimented with a series of methods to extract features from stains in images. To increase accuracy we then use machine learning techniques to classify stained sections on images.**

I. Motivation

While domestic applications of robots have been highlighted in popular culture practically applying robotics to accurately perform household tasks remains an open and interesting problem. In this paper we investigate the application of robots to a particular household task, specifically the task of using a sponge to wipe down a table after a meal.

II. Related Work

Although very little has been done in terms of general stain detection, there has been some work in specialized instances of stain detection. For example, Mertens et Al. [3], examined the use of computer vision algorithms in detecting stains on eggs with some success. In general, however, these works focus on a specific situation, with extensive prior knowledge about the scene and controlled conditions under which the algorithm is expected to work. In this project, we consider the problem more generally.

III. Approach

We divided the problem into two main components; stain identification and cleaning kinematics. For the stain identification sub-problem we mainly focused on developing algorithms that would enable a robot to correctly identify stains. First, we clearly established our definition of a "stain" as a discoloration produced by foreign matter having penetrated into or chemically reacted with a material. The most important property of these stains is the discoloration of the stained area from the background texture color.

A. Image Segmentation

Since a stain will usually stand out in comparison with the background, one natural approach we explored is to segment the image into background and stain parts. To accomplish this, we used a standard Watershed algorithm to segment the image. Specifically, watershed by flooding [1] was utilized. The segmentation process initially erodes the image to obtain seed points, which are defined as pixels whose values are representative of their neighbors. The watershed algorithm is then run on the seed points to obtain a segmentation. The initial seed points allow distinct segments to be treated contiguously and to partion the output in two, as a normal segmentation might produce many layers of segments. The watershed algorithm was chosen because of its speed and ability to clearly segment trivial test cases.

B. Average Window Approach

After initial testing, the initial Watershed segmentation turned out to be too sensative to background textures. As a result, we set about to create methods that can overcome this. In the window approach, we split up the image into fixed size square blocks. We then compare each window with the average window. Finally, we normalize the differences by scaling each difference with respect to the maximum and minimum difference in the photo. Windows with a difference greater than a threshold are then classified as stains. The motivation of this window approach is that although each window may differ from some background texture, a window with a stain in it will differ more relatively to thoses that do not. In this way, we account for the differences created by a background, especially any repeating one.

C. Learning Features

A natural extension to the above approach is to apply classical Machine Learning methods to improve robustness. Rather than hand specifying how thresholds should be specified, we can learn them through supervised training methods. For this, we choose SVM's for its extensibitiliy and effectiveness. For the features, we first split the image on a per channel basis, one for each of the three color channels. For each channel, we used the absolute difference between the average intensity in the window and the average, mean and mode intensity (per channel) of the image as features. In addition, the difference in variation of intensities within the window as well as the edginess of the window were also used as features. The motivation for the variance is that a stain might cover up an otherwise colorful background, so a decrease in variance might tell us something about whether the window is indeed a stain. Similarly, a window with a stain will have different degree of edginess as compared to those without.

IV. Results and Discussion

The complete dataset used to evaluate the algorithms contains about 70 images, which results in roughly 70,000 training examples (at a window size of 10x10). To ensue that distribution of positive and negative examples is roughly evenly split, we took a simple random sample of the negative

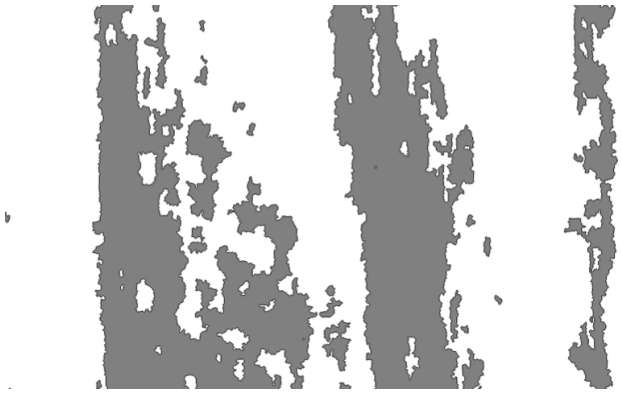

Fig. 1. Segmentation fail

Fig. 2. Segmentation fail

training examples. For every picture, we divide it into rectangular regions as governed by a window size parameter. These rectangular regions are used to classify windows as stained or non-stained. If a window overlaps with a labeled stain region, that window is considered to be correctly identified.

A. Segmentation

Several disadvantages were discovered through experimentation with the Watershed segmentation. It was difficult to determine which segmented portion of the mask was a stain. The watershed segmentation approach was unable to handle patterned backgrounds or textures such as wood, as figures 1 and 2 demonstrate. In addition it is hard to modify and improve the the algorithm besides through pre and post processing.

B. Window

Through experimentation it was qualitatively determined that in general the window approach performed more accurately than the initial segmentation approach. Figure 3 shows the result of running the window algorithm on a stain. While the window approach performed better than the segmentation approach results still varied on an image by image basis. Certain parameters of the algorithm could be adjusted to improve performance but the adjustments vary between pictures and cannot be set automatically. Table I displays the statistical results of the window detection approach. As a result of the varied performance machine learning we decided to apply machine learning to the data by using a SVM to improve performance.

Window Size	Average Precision	Average Recall
5	0.640206848	0.161015766
10	0.625077057	0.203232955
20	0.646491238	0.262179909
50	0.645036666	0.405706819

TABLE I
WINDOW RESULTS

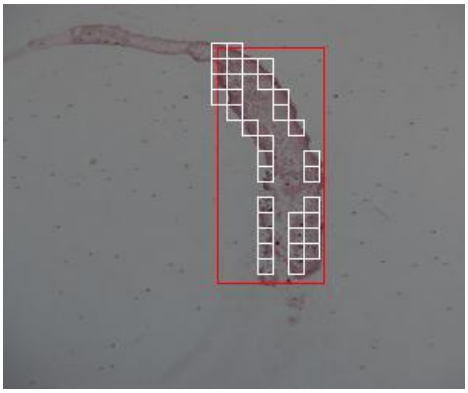

Fig. 3. Window-based

C. Learning

The SVM implementation used in this paper was SVM-light [4] and the window size is 10x10 for the examples used in this paper. Table II displays the result statistics of training on a sub-dataset labeled as “dataset 1” and testing on a sub-dataset labeled as “dataset 2”. Table III displays the result statistics of the reciprocal training/testing relationship. It is worth noting that the overall performance is higher in table III for multiple reasons. Dataset 1 is the first set of test images obtained and manually classified. The dataset contains images with slight contrast issues which results in the SVM misclassifying darker regions of pictures. The issue of contrast is further discussed later in this paper. Another reason for the poorer performance when training on dataset 1 results from misrepresentation of stains due to poor manual classification. Because most stains are not square in shape attempting to classify curved stains using square bounding boxes will inevitably result in areas misclassified as stains. As a result many pictures in the “dataset 1” contain misclassified examples and this source of error was noted before “dataset 2” was classified. The bounding boxes in “dataset 2” are drawn much closer to the shape of the stain in order to reduce error and the results show a definite reduction.

Qualitative comparisons between the SVM detected and window detected regions also highlight the improved performance brought on by machine learning. Figure 4 shows the results of window-based detection on the left and the results

k	Accuracy	Precision	Recall
0.00075	46.2	35.81	9.58
0.001	50.71	51.88	19.55
0.01	55.52	55.19	58.76
0.1	45.67	46.77	62.69
1	50.94	63.67	4.37
10	49.02	48.71	36.99
100	49.02	48.71	36.99
1000	49.02	48.71	36.99

TABLE II

SVM RESULTS FROM TRAINING ON DATASET 1 AND TESTING ON 2

k	Accuracy	Precision	Recall
0.00075	66.12	72.74	51.62
0.001	66.12	72.71	51.65
0.01	66.12	72.77	51.59
0.1	49.99	50.01	87.9
1	50.19	50.15	70.8
10	49.61	49.82	97.86
100	56.28	54.26	80.15
1000	56.28	54.26	80.15
10000	56.28	54.26	80.15

TABLE III

SVM RESULTS FROM TRAINING ON DATASET 2 AND TESTING ON 1

of learning on the right. It is worth noting that not only does the learning approach label much more of the stain, it also correctly identifies sections of the stain that were not classified initially. Figure 5 best highlights the advantages of the learning approach. In the top image the window-based detector incorrectly identified the background pattern as a series of stains. In the bottom image the SVM correctly labels the stained region while not misclassifying the background pattern.

There is a tradeoff between recall and precision by selecting k values.

Contrast.

D. Future Work

While the learning approach towards stain detection has proven to be more effective than segmentation and window comparison-based solutions the process could benefit from additional image preprocessing to reduce error.

Also we need to program the ROBOT.

APPENDIX

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word acknowledgment in America is without an e after the g. Avoid the stilted expression, One of us (R. B. G.) thanks . . . Instead, try R. B. G. thanks. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

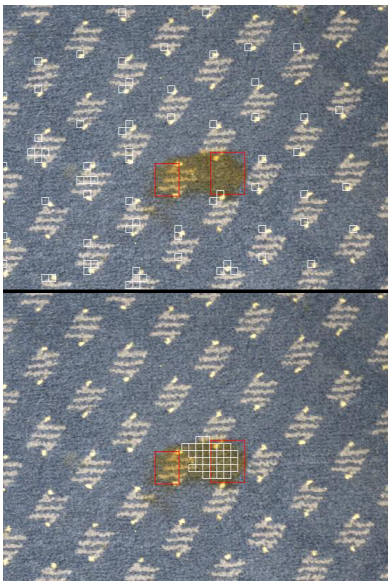

Fig. 5. Comparison of window feature detection (left) and SVM detection (right)

Fig. 6. Window-based

REFERENCES

[1] Serge Beucher and Christian Lantujoul. Use of Watersheds in Contour Detection. In International workshop on image processing, real-time edge and motion detection (1979).

[2] Abid, K. (2012, July 11). stackoverflow Retrieved from http://stackoverflow.com/questions/11294859/how-to-define-the-markers-for-watershed-in-opencv/11438165

[3] Mertens, K. (2005). Dirt detection on brown eggs by means of color computer vision. Manuscript submitted for publication, Egg Quality and Incubation Research Group, , Available from US National Library of Medicine. (16335136)Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/16335136

[4] T. Joachims, Making large-Scale SVM Learning Practical. Advances

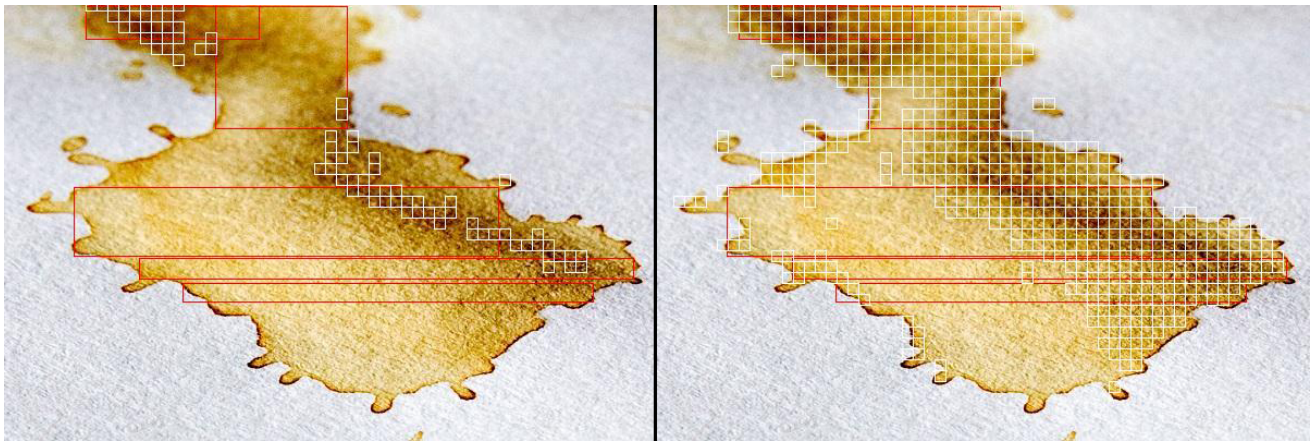

Fig. 4. Comparison of window feature detection (left) and SVM detection (right)

in Kernel Methods - Support Vector Learning, B. Schlkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.