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

Our initial approach towards stain detection used the Watershed image segmentation algorithm. 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 close to their neighbors. The watershed algorithm is then run on the seed points to obtain a segmentation. The watershed algorithm was initially chosen because of its speed and ability to clearly segment trivial test cases.

B. Average Window Approach

One of the most important reasons why Watershed segmentation was abandoned was due to the algorithm's inability to ignore background textures. As a result an alternative method was created. The new method segments the image into square windows and compares each window with the average window value. Windows with a value difference greater than a threshold are classified as a stain.

C. Learning Features

SVM features were determined on a per channel basis for all three color channels. For each channel the difference between the window and average, mean and mode value were used as features. In addition the variation of values within the window as well as the edginess of the window were also used as features. The variation of values was used because stains might be monocolored as compared to the background texture pattern, due to the properties of the stain substance.

IV. Results and Discussion

The complete dataset used to evaluate the algorithms contains about 70 images, which together contain roughly 70,000 training examples. The distribution of positive and negative examples is roughly evenly split as well. For every picture rectangular regions are identified that contain stains. 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 a stain.

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

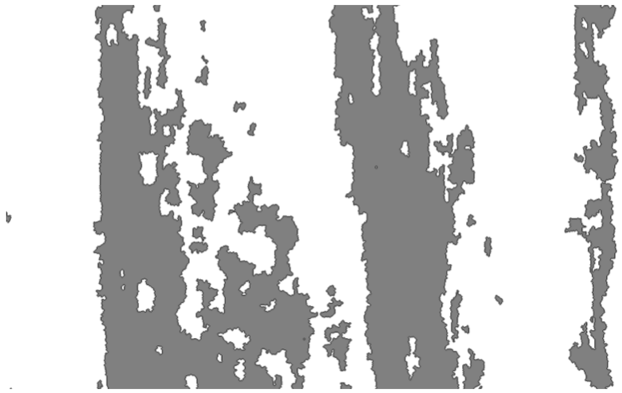

Fig. 1. Segmentation fail

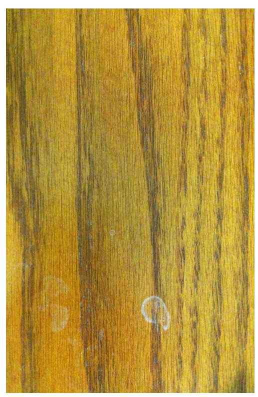

Fig. 2. Segmentation fail

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

C. Learning

The SVM implementation used in this paper was SVM-light and the window size is 10x10 for the examples used in this paper. Table II displays the result statistics of training on.

First test set had bad examples (overexposure, misrepresentation).

Figure

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

D. Future Work

Through

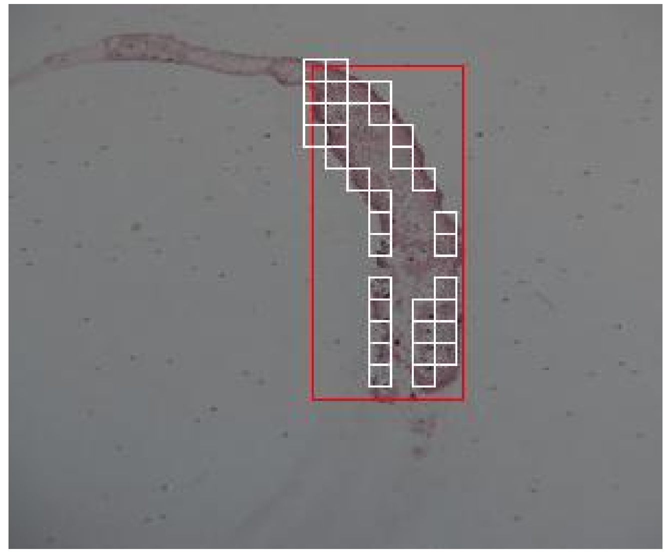

Fig. 3. Window-based

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

V. CONCLUSIONS

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

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

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

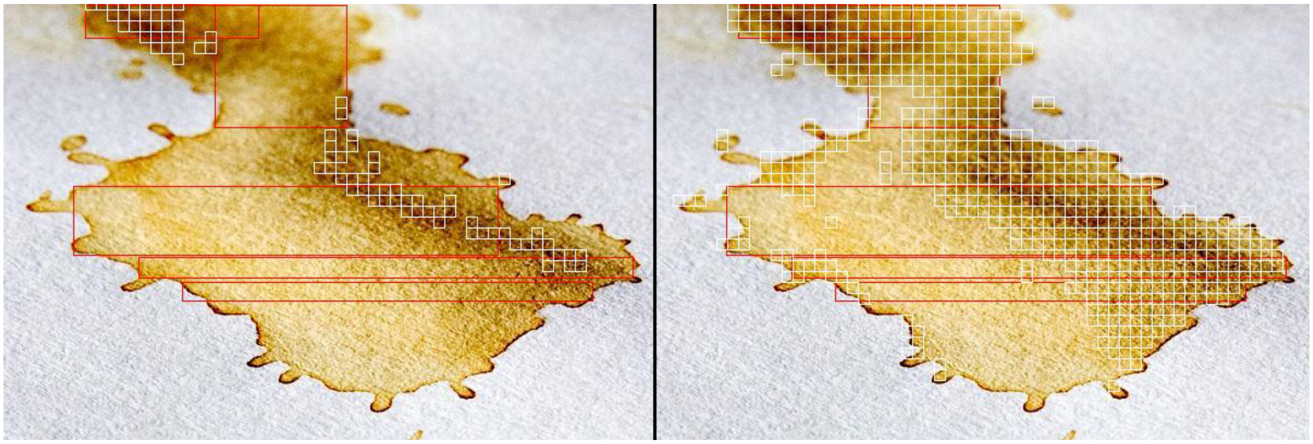

Fig. 4. Comparison of window feature detection (left) and SVM detection (right)

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.

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

Fig. 5. Window-based

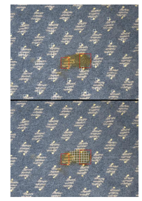

Fig. 6. Comparison of window feature detection (left) and SVM detection (right)