Explicit Image Filter

CS229 Final Project, Dec 2005

CYRILLE HABIS
habis@stanford.edu

FILIP KRSMANOVIC
filipk@cs.stanford.edu

Abstract.
We propose to create an automatic image filter to recognize offensive digital images that contain violent, sexual and other explicit material. Our initial system focuses on filtering sexually explicit and nude images. To build it, we first identify relevant features and train learners to recognize them. We use Computer Vision (CV) to extract both color and edge based features, and supervised Machine Learning (ML) algorithms such as Logistic Regression and SVM to train. Currently considered features are skin pixels, skin cluster layouts, simple body parts, specific edges and skin variations across edges. We then create single feature-based or related feature set-based weak image classifiers that we test and tune for lowest classification error. Finally, we combine these weak classifiers to test how different features do together. Our main combination method is to attach weights to each classifier and train them to get a final strong hypothesis. The overall result is a fairly accurate explicit image classification framework (87.11% accurate), where modular weak classifiers can be added and removed for further improvement. We test our features and algorithms on image sets returned by Google's image search. Our main concern is accuracy, and our secondary concern is speed, major limitations on the current state of the art. Our approach is novel since, although ML work on general object recognition exists, ML has not been used by most current solutions that classify sexually explicit images, and none have used it with CV.

1. Introduction

In today's age of the media Internet, all manner of images have become available online, in copious amounts. In fact, one of the more rapidly growing areas in search technology today is image search. With this availability comes the natural need to filter offensive content, to prevent explicit images from reaching the wrong eyes. Many solutions exist today, such as filters that are part of image search engines like those of Google or Yahoo!, and commercially available stand-alone filters like ContentProtect or NetNanny. As any user of these services is aware, while they are very good at filtering websites and text, they often fail to remove offensive images. The reasons are clear in that "current Internet image search technology is based upon words, rather than image content," [2], as images are obtained by using the image filename or text that surrounds the image on a webpage [3]. Even though attempts have been made to search using actual image semantics [1, 2], it remains a largely unsolved problem and thus, it is no surprise that image filters also use the same methods as search, in that they screen for the text and web pages to do with the image, but not the image itself.

Methods that process actual images exist, such as extracting nudes using skin filtering and geometric shape matching of limbs [1], or classification methods using wavelet transforms of images and matching them to those in a database [4]. Also, companies like VIMA Technologies have actual commercially available solutions [7]. While these methods directly address the problem, many suffer from inaccuracy (the first one only finds 43% of positive nudes [1]), or exhibit a lack of speed that hinders use in real world situations (the second method requires up to 5 seconds for an image to be processed [4]). Most of the solutions do not employ Machine Learning (ML) techniques, and those that do, like VIMA Technologies, do not use Computer Vision (CV).

Due to the nature of ML and the fact that it has been used in object recognition [2], as well as the fact that visual image features need to be extracted, applying ML and CV together to the abstract problem of classifying offensive images seems like a natural choice, and it is our aim to do so. We thus create weak classifiers for single relevant features or sets of related features, and we elaborate on those methods below. For all the classifiers and their combination, we also display the results obtained. Currently, our system is concerned with images containing nude and sexually explicit material. To test the algorithms and features of

our framework, we use Google's image search to provide representative sets of images, appropriate since these images represent those most popularly viewed or linked to on the Internet.

2. Method & Results

Overall Strategy

To construct our system we first identify relevant features by studying our images of interest, and by referring to previous work on the subject such as [1]. For each feature or related set of features, we then attempt to extract either color data, edge data or both as appropriate, using CV, and then train a hypothesis to recognize the features from this data. Thus, we build a single weak classifier based on this related feature set, which is trained on our actual images to recognize when one is explicit. The weak classifiers are created using modified ML algorithms such as Logistic Regression and SVM, and we test with different algorithms and tuned values to produce the least estimated generalization error. Thus, we can compare individual features and related feature sets to empirically evaluate which are most valuable, in a manner close to Forward Search of Wrapper Model Feature Selection. We finally go on to combine unrelated feature sets to find which combined sets are most valuable. Currently, we combine by weighing each separate classifier according to their performances on small test sets, resulting in a single weighted classifier. The entire system is a general explicit image classification framework, where new feature-based weak classifiers can be added and tuned to facilitate future work and improvements. For a clearer picture of the system architecture, please see Diagram 1 in the Appendix.

We will now discuss our data sets, and then go through each classifier below. **Please note, all results shown are accuracies of weak or combined classifiers running on our actual test images to classify them as explicit or not.** Finally, note that we use the term "weak classifier" loosely, as some of our classifiers do fairly well on their own, especially when compared to the combined result.

Data and Testing

We have built an image crawler that, given a query, uses Google's image search and downloads a specified amount of images from those returned. We have used the query "sex" to get the set of positive data and the query "things", based on [2], to get negative data. All data was gone through and labeled by hand, and the positive data categorized, so that we could see what specific categories of sexually explicit images our feature sets or methods did well or poorly on. Categories included groups of people, more than one skin color, etc.

The data in the end was 882 images, specifically:

- 626 training examples (313 positive and 313 negative)
- 256 test examples (128 positive and 128 negative)

This is 71/29 % split, and we appropriately then use Hold-Out Cross Validation to measure the estimated generalization errors of different algorithms and feature sets. We argue that the number of training examples we have is enough for our current work according to Vapnik's theorem, since the largest amount of parameters in all our main classifier is 50. Thus, the number of training examples we need to minimize training error is linear in 50 at most. 10 * # of parameters is a good rule of thumb, as Prof. Ng mentioned in class, which we exceed.

Classifier 1: Skin Pixel Percentage Threshold

We use a skin filter to detect percentages of skin pixels in an image, based on the work of Forsyth and Fleck [1]. Our current use of skin differs already from that of Forsyth and Fleck as we consider skin a feature and not as necessary condition, nor do we assign a hard limit to how much skin is required, rather, we learn it. Please see figures 1 & 2 in the Appendix to see an example of the skin filter at work. Here, we learn

the skin colored pixel percentage threshold for the entire image that is most likely to indicate sexually explicit material. This is our most basic classifier, concerning just one value, so we use Logistic Regression.

Results:
Accuracy is not very high, but fair considering the simplicity of this classifier:
Training Set Error: 0.2013 – 79.87% accuracy
Test Set Error: 0.2422 – 75.78% accuracy

Classifier 2: Skin Cell Layout – Method 1

This classifier is also based on skin, but uses the information in a smarter way, via cell decomposition. The image is divided into a grid of cells, and each cell is labeled as a skin cell or not depending on whether 50% of it contains skin colored pixels. We obtain a feature vector with an entry for each cell that has the value 1 if the cell is skin and 0 if not, and pass this into an SVM.

Here we can vary the kernel, the constant c, and the number of rows & cols of the grid to find the best accuracy. We tried linear, Gaussian and spline kernels and found that the Gaussian and spline kernels overfit the data, as we get a very high training set accuracy but a much lower test set accuracy, so we suffer from high variance. Reducing the constant c improved performance somewhat, and taking outliers more into account seemed to help. Finally, increasing the grid granularity gave us poorer performance. We feel one of the reasons is that increasing granularity would increase the number of entries in the feature vector, and thus the amount of parameters in our hypothesis, which according to Vapnik means we may have too few training examples.

Results:
Our best accuracy here was with a linear kernel, c = 0.01 and a 10 X 10 grid:
Training Set Error: 0.147 – 85.3% accuracy
Test Set Error: 0.1914 – 80.86% accuracy

Classifier 3: Skin Cell Layout – Method 2

This is the same method as above, with one important difference. Intuitively, we thought that finer grid granularity to a certain extent, would help in our skin-based classification accuracy as we get more information, but we were met with poor results. Following our reasoning as to this cause from above, we no longer have an entry for each cell in the SVM input vector, but instead we aggregate the amount of skin cells in all rows and columns of the grid., and use this aggregation data. Thus, we hope to give the SVM the same relevant information, but with fewer entries in the feature vector, so that the amount of training examples we have is enough. Parameters and kernels were varied as above, with much the same observations. We thought it useful to graph how the grid size affects the training and test errors, please see graph 1 in the Appendix. The graph is very close to those we saw in class, in Learning Theory.

Results:
Our reasoning seems correct and we see significant improvement here, with a linear kernel, c = 0.0005 and a 25 X 25 grid:
Training Set Error: 0.147 – 85.30% accuracy
Test Set Error: 0.1914 – 83.59% accuracy

Classifier 4: Simple Body Part Detection

We attempt to recognize certain body parts present in explicit images. This classifier is still in its infancy, and currently recognizes only nipples. We trained the detector using small image cutouts of nipples and control cutouts, grayscaling them, and running the pixels through an SVM. The detector is then run on

the actual images to classify positives if nipples are found. We recognize that alone this classifier is not very effective, but may be useful when combined with others.

Results:
Accuracy was low, but is as expected, since not all positive images have nipples and the classifier is still in preliminary stages. We have no training set error for our actual images, as the classifier was trained on small image cutouts. Its training accuracy on those there was 70%.
Test Set Error: 0.4535 – 54.65% accuracy

Classifier 5: Skin Variations Across Edges

After going through many explicit images by hand, we observed that a common occurrence was limbs and body parts of different people next to or over each other. Recognizing such a feature could help especially in images where skin detection may fail. Please see figures 5, 6 and 7 to see an example where skin detection does not help due to clothes, but this new detector would catch such an image due to one person's hand on the other's naked shoulder. To achieve this, we locate edges and find whether there are different skin hues on each side of the edge. If the difference is above a certain trained threshold, and there are enough such occurrences in an image, we classify it as explicit. To achieve this we trained a detector on image cutouts, much as we trained the body part detector above, and then we ran the resulting detector within skin regions only (which boosted performance and accuracy) directly on the test set images.

For edge detection, we use the Canny edge detector, and we run it only on areas labeled as skin to boost speed and accuracy. Please see figures 3 and 4 in the appendix to see edge detection at work being run on the entire image as well as just within skin pixels.

Results:
We obtain the best accuracy of any of our individual classifiers:
Test Set Error: 0.1367 – 86.33% accuracy

Classifier 6: Specific Edge Detection

This classifier is still a work in progress. We are trying to separate specific edges resulting from edge detection, and learn which edges may indicate explicit images. Once way to use this is building a dictionary of representative edges found in pornography, and then treating edges in a similar fashion to how words are treated in spam classification. A large challenge lies in separating edges and knowing where an edge ends, and this is a challenge we are still working on.

Combining Weak Classifiers

Finally, we also attempted to combine some of our weak classifiers to see how they would fare together. We combined classifiers 3, 4 and 5. The current combination method works by assigning weights to each individual classifier, and splitting the entire data into 3 segments. The first segment is used to train the individual classifiers, and the second segment is used to train our weights with higher weights going to classifiers that have smallest error, proportionally. The third set is for testing the final classifier, h. The combination equation for our 3 classifiers is:

$$h = \frac{(\varepsilon 2 + \varepsilon 3) * h1 + (\varepsilon 1 + \varepsilon 3) * h2 + (\varepsilon 1 + \varepsilon 2) * h3}{2 * (\varepsilon 1 + \varepsilon 2 + \varepsilon 3)}$$

where εi is hypothesis hi's test error on the second data segment above.

Results:
We achieved our best accuracy for the entire project here, but it is still a little less than we hoped for. We suspect this is due to the fact that our body part classifier is still very weak and not a big help, and that the other two features are too close, both being based on skin. However, we feel that combining features still has great potential and is worth trying, especially after we complete new different classifiers such as Classifier 6.
Test Set Error: 0.1299 – 87.11% accuracy

Please note that, throughout these tests, our system always classified twice as many false positives as false negatives, fairly consistently across the board.

3. Discussion and Conclusions

Please note, the discussion here is general, and conclusions and analysis to do with specific classifiers are mainly written above under the method and result descriptions of that classifier.

Overall, we found this a very challenging problem, as expected. However, we did manage to create a system that worked, and delivers a fairly decent accuracy of 87.11%. Of course, we feel that this accuracy still needs to improve. We found it was manageable to get to a reasonable accuracy of about 80%, but as we try to get to higher ones, the journey becomes what appears to be exponentially more difficult. The main culprits here are outliers, images such as figure 8, where skin filtering does not help much. Images where there was no skin present but were still pornography, such as people wearing clothes, or images where there was a good deal of skin but were innocuous, such as people at the beach, are what prevent high accuracy.

An additional observation worth noting is that our system classifies twice as many false positives as negatives, so it is a stricter filter. We feel this is better for our purposes, such as preventing children from seeing pornography. However, this view varies from person to person.

The framework we have built is far from a perfect classifier, but achieves the purpose of being a great starting point for further improvements, as well as being a formidable classifier in its own right. We have built it in such a modular way that new classifiers based on other features can easily be added to the framework or taken away, which begs for future work as we elaborate below.

Future Directions:

Our main hopes for the future are to build more classifiers based on other features or related sets of features, and to test their different combinations. We feel we have almost exhausted skin related features. These are very appropriate and useful, but we feel that now we need to look at other types, such as edges and more sophisticated body part detection. These features would help in areas where skin detection fails such as the outliers above. We would also like to find efficient ways of separating specific edges to build our edge dictionary. In terms of body parts, we would like to learn more and better ways of detecting those body parts that are found in explicit images. Yet other features to explore are hair, specifically hair surrounded by skin.

New ways of combining our weaker classifiers are also on the agenda. We would like to try Boosting, for example, as it historically has done very well in such tasks. The traditional approach of Boosting with Decision Trees is also viable, as almost all our classifiers can be converted to Decision Trees.

We have a few more avenues to explore. Our images have been hand-labeled into categories, and we could try specific filters for each category, such as for groups of people, people of very different skin colors, etc. We would also like to get more data for our training, a current limitation, as we have to go through all the images by hand. Eventually, we aim to consider speed of our algorithms as well. We could also combine our system with text and web page filtering currently used, which we are fairly certain would boost our performance. Finally, when we are done with sexually explicit images, we would like to consider other offensive images such as violent ones to complete a truly general explicit image filter, one fit for the real world.

APPENDIX

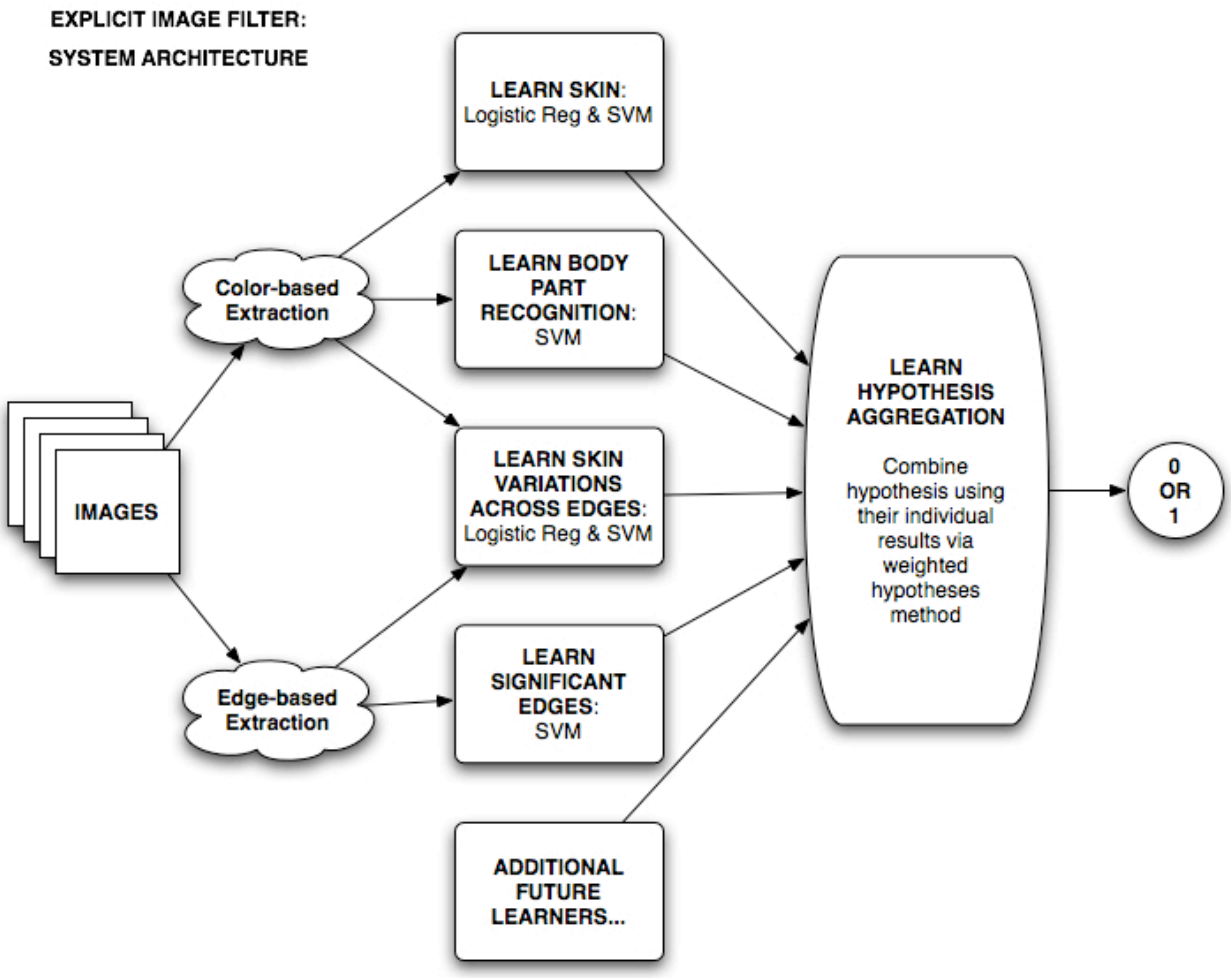


Diagram 1 – Entire system layout and overview.

Figure 1.

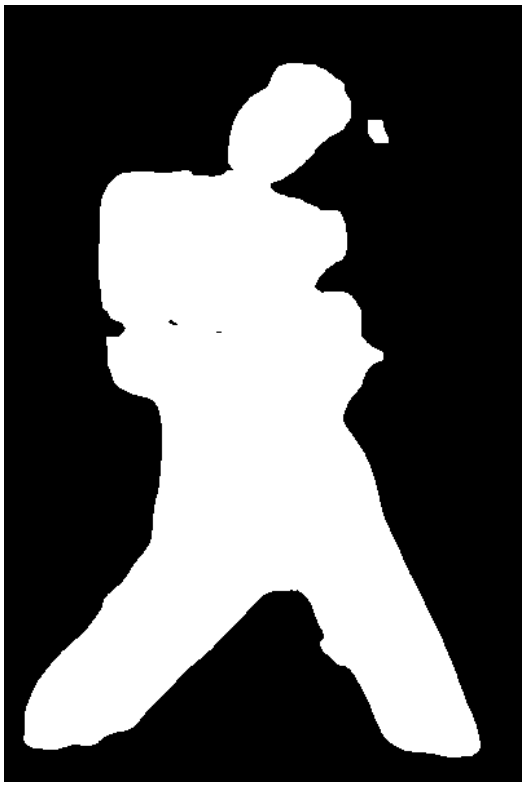


Figure 2 – Skin filtering of Figure 1.

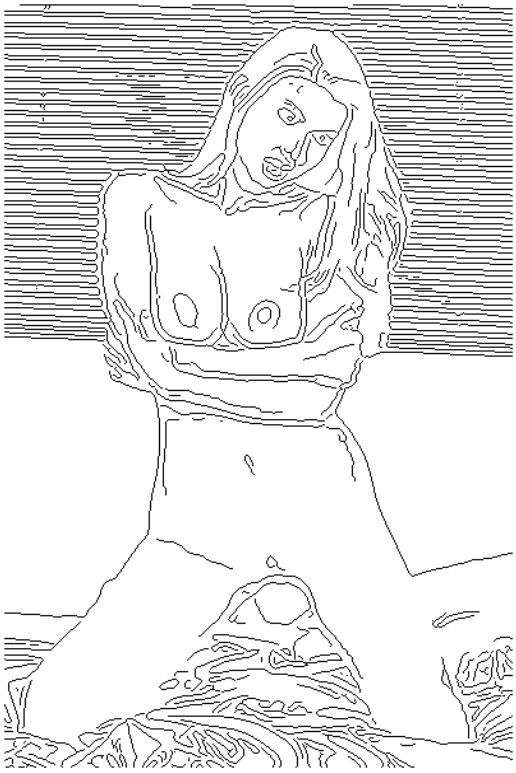


Figure 3 – Canny edge detection on Figure 1.

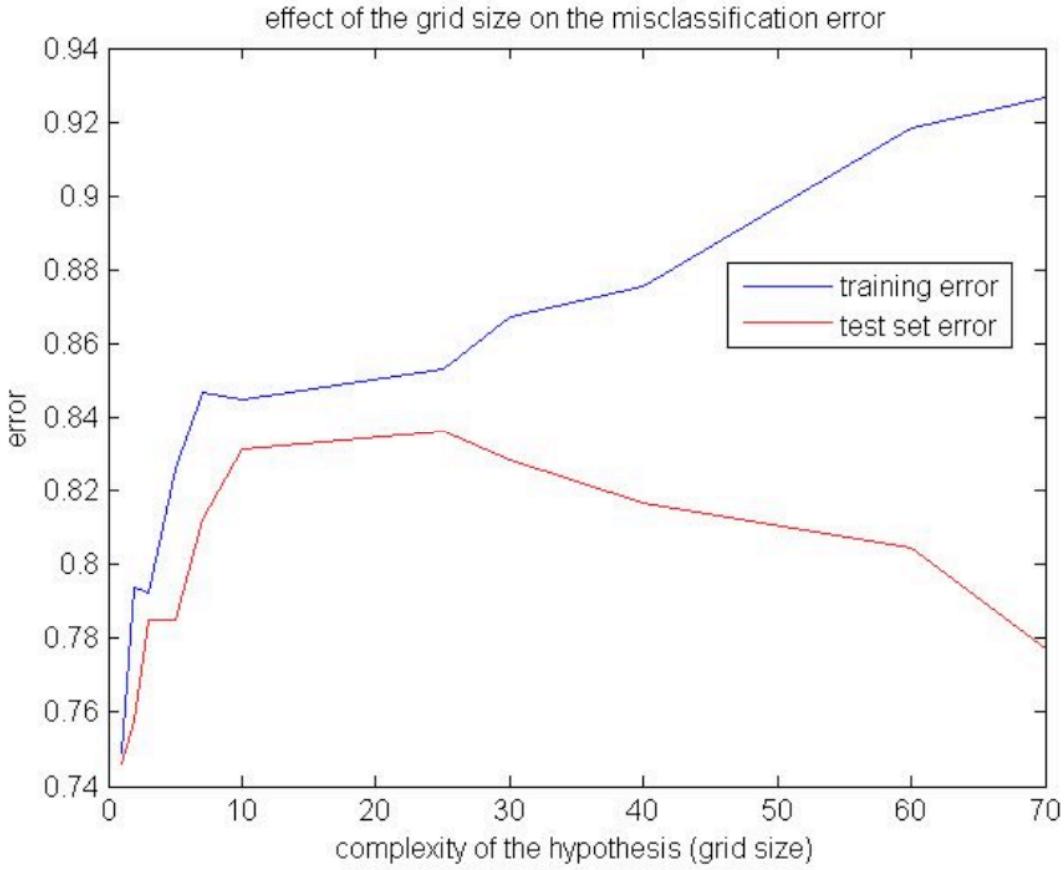
Figure 4 – Canny edge detection on Figure 1. Detection only in skin regions.

Figure 5.

Figure 6 – Skin filtering of Figure 5. Amount of skin too little to be of use.

Figure 7 – Canny edge detection on Figure 5. Edges where skin hues will vary are apparent.

Figure 8 – Skin filtering does not help.



Graph 1 – showing effects of bias and variance

Bibliography

1. D.A. Forsyth and M. M. Fleck. Automatic detection of human nudes, pages 3-4, 8-10, 1999.

2. R. Fergus, P. Perona and A Zisserman. A Visual Category Filter for Google Images, pages 1, 5-9, 2004.

3. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In 7th Int. WWW Conference, 1998.

4. G. Wiederhold and J. Z. Wang. WIPE (TM): Wavelet Image Pornography Elimination; A System for Screening Objectionable Images. http://stanfordtech.stanford.edu/4DCGI/docket?docket=97-096

5. J. P. Kapur. Face Detection in Color Images, *EE499 Capstone Design Project, University of Washington Department of Electrical Engineering*, 1997. http://www.geocities.com/jaykapur/face.html

6. J. Hoshen and R. Kopelman. Percolation and cluster distribution. i. *cluster multiple labeling technique and critical concentration algorithm*. Phys. Rev. B, 14(8):3438--3445, 1976.

7. VIMA Technologies, 2005 http://www.vimatech.com/

8. J. Shotton, A. Blake and R. Cipolla. Contour-Based Learning for Object Detection, *ICCV,* 2005.

We would also like to acknowledge and thank Prof. Andrew Ng and Ashutosh Saxena for their inputs throughout this project.