

AUTOMATICALLY DISTINGUISHING BETWEEN SUNSET AND NON-SUNSET SCENES USING MACHINE LEARNING

Nicholas Kamper and Eric Henderson

Rose-Hulman Institute of Technology

Email: kampernj@rose-hulman.edu and henderea@rose-hulman.edu

ABSTRACT

Scene detection is a complicated problem with many commercial applications in consumer products. For this paper, we implemented detection of sunset versus non-sunset scenes using support vector machine

Keywords— sunset detector, scene classification, support vector machine, image classification

1. INTRODUCTION

For the sunset detector project, we set out to devise an accurate way to classify whether a given image is a sunset or not using supervised machine learning techniques.

Scene classification has several practical applications. One such use is in consumer imaging devices to determine the best post-processing steps to deliver a high quality image. Another would be automated image sorting based on the scene type (e.g., being able to classify portraits, nature shots, and so on, to categorize them in an album).

Scene classification is difficult, even in the binary case that we consider in this paper. Certain scenes would be very difficult to classify, as they would share the same characteristics, such as hue and brightness features, as other scenes.

2. PROBLEM DESCRIPTION

In this particular scenario, we wanted to develop a classifier that would give a binary classification as to whether a scene contained a sunset or not, only using a set of approximately 500 training images.

3. IMPLEMENTATION

To solve this problem, we started out by implementing a baseline detector. First, we converted the source image to the LST (luminance and chroma) bands. Once in LST, to obtain the feature vector, we split the image into a 7x7 equidistant grid and took the mean and standard deviation of each of the LST bands for each of the 49 regions. This resulted in a 294 dimension feature vector for each image.

The feature vector was then normalized using the zero-mean and one-standard deviation method. With the 294 dimension



Fig. 1. A Sunset Image Split into a 7x7 Grid

normalized training feature vector, we trained an SVM using several different kernels, both polynomial and radial basisfunction kernels, varying the kernel parameter to obtain the best accuracy.

After training the SVM, we then ran the SVM on a set of test data, obtaining a set of labels. We then normalized those labels using the same zero-mean and one-standard deviation method and used a threshold of 0 to assign integer labels to each image. We compared the results of various kernels and parameters to obtain the best results.

In an attempt to improve our accuracy, in addition to normalizing the Y1 label values from the SVM, we also tried varying the grid size. However, this technique did not result in any improvements to the accuracy over the baseline 7x7 grid.

4. EXPERIMENTAL SETUP AND RESULTS

4.1. Input

| | Training | Test |
|------------|----------|------|
| Sunset | 223 | 254 |
| Non-Sunset | 276 | 245 |

4.2. SVM Parameter Results

4.2.1. Polynomial Kernel

| p | TP | TN | FP | FN | TPR | FPR | Acc |
|---|-----|-----|-----|----|--------|--------|--------|
| 2 | 211 | 226 | 19 | 43 | 83.07% | 7.76% | 87.58% |
| 3 | 218 | 239 | 6 | 36 | 85.83% | 2.45% | 91.58% |
| 4 | 211 | 222 | 23 | 43 | 83.07% | 9.39% | 86.77% |
| 5 | 206 | 240 | 5 | 48 | 81.10% | 2.04% | 89.38% |
| 6 | 196 | 131 | 114 | 58 | 77.17% | 46.53% | 65.53% |

4.2.2. Radial Basis Function Kernel

| σ | TP | TN | FP | FN | TPR | FPR | Acc |
|----------|-----|-----|----|----|--------|-------|--------|
| 5 | 234 | 234 | 11 | 20 | 92.13% | 4.49% | 93.79% |
| 7 | 231 | 235 | 10 | 23 | 90.94% | 4.08% | 93.39% |
| 9 | 232 | 237 | 8 | 22 | 91.34% | 3.27% | 93.99% |
| 11 | 234 | 238 | 7 | 20 | 92.13% | 2.86% | 94.59% |
| 13 | 234 | 238 | 7 | 20 | 92.13% | 2.86% | 94.59% |
| 15 | 235 | 237 | 8 | 19 | 92.52% | 3.27% | 94.59% |

5. RESULTS OF VARIOUS SIZED GRID RUNS

5.1. Calculating for block size = 5

5.1.1. Polynomial Kernel

Kernel: poly; Param: 2.000 TP: 223/499; TN: 225/499; FP: 20/499; FN: 31/499 TPR: 87.80%; FPR: 8.16%; Accuracy: 89.78%

Kernel: poly; Param: 3.000 TP: 224/499; TN: 238/499; FP: 7/499; FN: 30/499 TPR: 88.19%; FPR: 2.86%; Accuracy: 92.59%

Kernel: poly; Param: 4.000 TP: 208/499; TN: 224/499; FP: 21/499; FN: 46/499 TPR: 81.89%; FPR: 8.57%; Accuracy: 86.57%

Kernel: poly; Param: 5.000 TP: 208/499; TN: 240/499; FP: 5/499; FN: 46/499 TPR: 81.89%; FPR: 2.04%; Accuracy: 89.78%

5.1.2. Radial Basis Function Kernel

Kernel: rbf; Param: 5.000 TP: 231/499; TN: 231/499; FP: 14/499; FN: 23/499 TPR: 90.94%; FPR: 5.71%; Accuracy: 92.59%

Kernel: rbf; Param: 7.000 TP: 235/499; TN: 233/499; FP: 12/499; FN: 19/499 TPR: 92.52%; FPR: 4.90%; Accuracy: 93.79%

Kernel: rbf; Param: 9.000 TP: 234/499; TN: 233/499; FP: 12/499; FN: 20/499 TPR: 92.13%; FPR: 4.90%; Accuracy: 93.59%

Kernel: rbf; Param: 11.000 TP: 236/499; TN: 233/499; FP: 12/499; FN: 18/499 TPR: 92.91%; FPR: 4.90%; Accuracy: 93.99%

Kernel: rbf; Param: 13.000 TP: 235/499; TN: 232/499; FP: 13/499; FN: 19/499 TPR: 92.52%; FPR: 5.31%; Accuracy: 93.59%

Kernel: rbf; Param: 15.000 TP: 235/499; TN: 231/499; FP: 14/499; FN: 19/499 TPR: 92.52%; FPR: 5.71%; Accuracy: 93.39%

Kernel: rbf; Param: 17.000 TP: 235/499; TN: 232/499; FP: 13/499; FN: 19/499 TPR: 92.52%; FPR: 5.31%; Accuracy: 93.59%

Kernel: rbf; Param: 19.000 TP: 235/499; TN: 233/499; FP: 12/499; FN: 19/499 TPR: 92.52%; FPR: 4.90%; Accuracy: 93.79%

5.2. Calculating for block size = 6

5.2.1. Polynomial Kernel

Kernel: poly; Param: 2.000 TP: 217/499; TN: 226/499; FP: 19/499; FN: 37/499 TPR: 85.43%; FPR: 7.76%; Accuracy: 88.78%

Kernel: poly; Param: 3.000 TP: 218/499; TN: 238/499; FP: 7/499; FN: 36/499 TPR: 85.83%; FPR: 2.86%; Accuracy: 91.38%

Kernel: poly; Param: 4.000 TP: 213/499; TN: 226/499; FP: 19/499; FN: 41/499 TPR: 83.86%; FPR: 7.76%; Accuracy: 87.98%

Kernel: poly; Param: 5.000 TP: 210/499; TN: 242/499; FP: 3/499; FN: 44/499 TPR: 82.68%; FPR: 1.22%; Accuracy: 90.58%

5.2.2. Radial Basis Function Kernel

Kernel: rbf; Param: 5.000 TP: 236/499; TN: 230/499; FP: 15/499; FN: 18/499 TPR: 92.91%; FPR: 6.12%; Accuracy: 93.39%

Kernel: rbf; Param: 7.000 TP: 235/499; TN: 233/499; FP: 12/499; FN: 19/499 TPR: 92.52%; FPR: 4.90%; Accuracy: 93.79%

Kernel: rbf; Param: 9.000 TP: 235/499; TN: 234/499; FP: 11/499; FN: 19/499 TPR: 92.52%; FPR: 4.49%; Accuracy: 93.99%

Kernel: rbf; Param: 11.000 TP: 235/499; TN: 234/499; FP: 11/499; FN: 19/499 TPR: 92.52%; FPR: 4.49%; Accuracy: 93.99%

Kernel: rbf; Param: 13.000 TP: 235/499; TN: 234/499; FP: 11/499; FN: 19/499 TPR: 92.52%; FPR: 4.49%; Accuracy: 93.99%

Kernel: rbf; Param: 15.000 TP: 235/499; TN: 235/499; FP: 10/499; FN: 19/499 TPR: 92.52%; FPR: 4.08%; Accuracy: 94.19%

Kernel: rbf; Param: 17.000 TP: 234/499; TN: 234/499; FP: 11/499; FN: 20/499 TPR: 92.13%; FPR: 4.49%; Accuracy: 93.79%

Kernel: rbf; Param: 19.000 TP: 235/499; TN: 234/499; FP: 11/499; FN: 19/499 TPR: 92.52%; FPR: 4.49%; Accuracy: 93.99%

5.3. Calculating for block size = 7

5.3.1. Polynomial Kernel

Kernel: poly; Param: 2.000 TP: 212/499; TN: 226/499; FP: 19/499; FN: 42/499 TPR: 83.46%; FPR: 7.76%; Accuracy: 87.78%

Kernel: poly; Param: 3.000 TP: 218/499; TN: 239/499; FP: 6/499; FN: 36/499 TPR: 85.83%; FPR: 2.45%; Accuracy: 91.58%

Kernel: poly; Param: 4.000 TP: 211/499; TN: 222/499; FP: 23/499; FN: 43/499 TPR: 83.07%; FPR: 9.39%; Accuracy: 86.77%

Kernel: poly; Param: 5.000 TP: 205/499; TN: 240/499; FP: 5/499; FN: 49/499 TPR: 80.71%; FPR: 2.04%; Accuracy: 89.18%

5.3.2. Radial Basis Function Kernel

Kernel: rbf; Param: 5.000 TP: 234/499; TN: 234/499; FP: 11/499; FN: 20/499 TPR: 92.13%; FPR: 4.49%; Accuracy: 93.79%

Kernel: rbf; Param: 7.000 TP: 232/499; TN: 234/499; FP: 11/499; FN: 22/499 TPR: 91.34%; FPR: 4.49%; Accuracy: 93.39%

Kernel: rbf; Param: 9.000 TP: 232/499; TN: 238/499; FP: 7/499; FN: 22/499 TPR: 91.34%; FPR: 2.86%; Accuracy: 94.19%

Kernel: rbf; Param: 11.000 TP: 233/499; TN: 237/499; FP: 8/499; FN: 21/499 TPR: 91.73%; FPR: 3.27%; Accuracy: 94.19%

Kernel: rbf; Param: 13.000 TP: 234/499; TN: 238/499; FP: 7/499; FN: 20/499 TPR: 92.13%; FPR: 2.86%; Accuracy: 94.59%

Kernel: rbf; Param: 15.000 TP: 235/499; TN: 237/499; FP: 8/499; FN: 19/499 TPR: 92.52%; FPR: 3.27%; Accuracy: 94.59%

Kernel: rbf; Param: 17.000 TP: 234/499; TN: 237/499; FP: 8/499; FN: 20/499 TPR: 92.13%; FPR: 3.27%; Accuracy: 94.39%

Kernel: rbf; Param: 19.000 TP: 234/499; TN: 237/499; FP: 8/499; FN: 20/499 TPR: 92.13%; FPR: 3.27%; Accuracy: 94.39%

5.4. Calculating for block size = 8

5.4.1. Polynomial Kernel

Kernel: poly; Param: 2.000 TP: 217/499; TN: 223/499; FP: 22/499; FN: 37/499 TPR: 85.43%; FPR: 8.98%; Accuracy: 88.98%

88.18%

Kernel: poly; Param: 3.000 TP: 223/499; TN: 239/499; FP: 6/499; FN: 31/499 TPR: 87.80%; FPR: 2.45%; Accuracy: 92.59%

Kernel: poly; Param: 4.000 TP: 221/499; TN: 209/499; FP: 36/499; FN: 33/499 TPR: 87.01%; FPR: 14.69%; Accuracy: 86.17%

Kernel: poly; Param: 5.000 TP: 203/499; TN: 241/499; FP: 4/499; FN: 51/499 TPR: 79.92%; FPR: 1.63%; Accuracy: 88.98%

5.4.2. Radial Basis Function Kernel

Kernel: rbf; Param: 5.000 TP: 231/499; TN: 232/499; FP: 13/499; FN: 23/499 TPR: 90.94%; FPR: 5.31%; Accuracy: 92.79%

Kernel: rbf; Param: 7.000 TP: 232/499; TN: 230/499; FP: 15/499; FN: 22/499 TPR: 91.34%; FPR: 6.12%; Accuracy: 92.59%

Kernel: rbf; Param: 9.000 TP: 233/499; TN: 231/499; FP: 14/499; FN: 21/499 TPR: 91.73%; FPR: 5.71%; Accuracy: 92.99%

Kernel: rbf; Param: 11.000 TP: 232/499; TN: 232/499; FP: 13/499; FN: 22/499 TPR: 91.34%; FPR: 5.31%; Accuracy: 92.99%

Kernel: rbf; Param: 13.000 TP: 233/499; TN: 232/499; FP: 13/499; FN: 21/499 TPR: 91.73%; FPR: 5.31%; Accuracy: 93.19%

Kernel: rbf; Param: 15.000 TP: 237/499; TN: 232/499; FP: 13/499; FN: 17/499 TPR: 93.31%; FPR: 5.31%; Accuracy: 93.99%

Kernel: rbf; Param: 17.000 TP: 237/499; TN: 231/499; FP: 14/499; FN: 17/499 TPR: 93.31%; FPR: 5.71%; Accuracy: 93.79%

Kernel: rbf; Param: 19.000 TP: 235/499; TN: 230/499; FP: 15/499; FN: 19/499 TPR: 92.52%; FPR: 6.12%; Accuracy: 93.19%

5.5. Calculating for block size = 9

5.5.1. Polynomial Kernel

Kernel: poly; Param: 2.000 TP: 217/499; TN: 221/499; FP: 24/499; FN: 37/499 TPR: 85.43%; FPR: 9.80%; Accuracy: 87.78%

Kernel: poly; Param: 3.000 TP: 223/499; TN: 242/499; FP: 3/499; FN: 31/499 TPR: 87.80%; FPR: 1.22%; Accuracy: 93.19%

Kernel: poly; Param: 4.000 TP: 219/499; TN: 209/499; FP: 36/499; FN: 35/499 TPR: 86.22%; FPR: 14.69%; Accuracy: 85.77%

Kernel: poly; Param: 5.000 TP: 203/499; TN: 241/499; FP: 4/499; FN: 51/499 TPR: 79.92%; FPR: 1.63%; Accuracy: 88.98%

5.5.2. Radial Basis Function Kernel

Kernel: rbf; Param: 5.000 TP: 236/499; TN: 230/499; FP: 15/499; FN: 18/499 TPR: 92.91%; FPR: 6.12%; Accuracy: 93.39%

Kernel: rbf; Param: 7.000 TP: 234/499; TN: 230/499; FP: 15/499; FN: 20/499 TPR: 92.13%; FPR: 6.12%; Accuracy: 92.99%

Kernel: rbf; Param: 9.000 TP: 234/499; TN: 230/499; FP: 15/499; FN: 20/499 TPR: 92.13%; FPR: 6.12%; Accuracy: 92.99%

Kernel: rbf; Param: 11.000 TP: 234/499; TN: 231/499; FP: 14/499; FN: 20/499 TPR: 92.13%; FPR: 5.71%; Accuracy: 93.19%

Kernel: rbf; Param: 13.000 TP: 234/499; TN: 232/499; FP: 13/499; FN: 20/499 TPR: 92.13%; FPR: 5.31%; Accuracy: 93.39%

Kernel: rbf; Param: 15.000 TP: 235/499; TN: 232/499; FP: 13/499; FN: 19/499 TPR: 92.52%; FPR: 5.31%; Accuracy: 93.59%

Kernel: rbf; Param: 17.000 TP: 235/499; TN: 233/499; FP: 12/499; FN: 19/499 TPR: 92.52%; FPR: 4.90%; Accuracy: 93.79%

Kernel: rbf; Param: 19.000 TP: 235/499; TN: 233/499; FP: 12/499; FN: 19/499 TPR: 92.52%; FPR: 4.90%; Accuracy: 93.79%

5.6. Calculating for block size = 10

5.6.1. Polynomial Kernel

Kernel: poly; Param: 2.000 TP: 224/499; TN: 220/499; FP: 25/499; FN: 30/499 TPR: 88.19%; FPR: 10.20%; Accuracy: 88.98%

Kernel: poly; Param: 3.000 TP: 225/499; TN: 241/499; FP: 4/499; FN: 29/499 TPR: 88.58%; FPR: 1.63%; Accuracy: 93.39%

Kernel: poly; Param: 4.000 TP: 219/499; TN: 205/499; FP: 40/499; FN: 35/499 TPR: 86.22%; FPR: 16.33%; Accuracy: 84.97%

Kernel: poly; Param: 5.000 TP: 186/499; TN: 243/499; FP: 2/499; FN: 68/499 TPR: 73.23%; FPR: 0.82%; Accuracy: 85.97%

5.6.2. Radial Basis Function Kernel

Kernel: rbf; Param: 5.000 TP: 234/499; TN: 232/499; FP: 13/499; FN: 20/499 TPR: 92.13%; FPR: 5.31%; Accuracy: 93.39%

Kernel: rbf; Param: 7.000 TP: 233/499; TN: 230/499; FP: 15/499; FN: 21/499 TPR: 91.73%; FPR: 6.12%; Accuracy: 92.79%

Kernel: rbf; Param: 9.000 TP: 234/499; TN: 230/499; FP: 15/499; FN: 20/499 TPR: 92.13%; FPR: 6.12%; Accuracy: 92.99%

Kernel: rbf; Param: 11.000 TP: 234/499; TN: 228/499; FP: 17/499; FN: 20/499 TPR: 92.13%; FPR: 6.94%; Accuracy: 92.59%

Kernel: rbf; Param: 13.000 TP: 234/499; TN: 229/499; FP: 16/499; FN: 20/499 TPR: 92.13%; FPR: 6.53%; Accuracy: 92.79%

Kernel: rbf; Param: 15.000 TP: 234/499; TN: 229/499; FP: 16/499; FN: 20/499 TPR: 92.13%; FPR: 6.53%; Accuracy: 92.79%

Kernel: rbf; Param: 17.000 TP: 234/499; TN: 230/499; FP: 15/499; FN: 20/499 TPR: 92.13%; FPR: 6.12%; Accuracy: 92.99%

Kernel: rbf; Param: 19.000 TP: 234/499; TN: 230/499; FP: 15/499; FN: 20/499 TPR: 92.13%; FPR: 6.12%; Accuracy: 92.99%

5.6.3. Summary

The best accuracy came from the Radial Basis Function kernel with σ of 11, 13, and 15. All three of these had identical accuracies. The σ s of 11 and 13 were completely identical in their results, but the σ of 15 had one more true positive and one fewer true negative when compared to the other two.

None of the other grid sizes were able to give improved accuracy over the 7x7 grid, even losing accuracy in most cases.

6. CONCLUSION

We found that a basic technique using a 7x7 grid was able to classify the test set with greater than 94% accuracy. Our normalization technique for normalizing the Y1 values greatly improved the accuracy of our code without requiring us to do manual threshold adjustments.

6.1. Misclassifications

Our classifier ended up misclassifying several images. The images that were false-positives typically contained hue and intensity values very similar to that of a sunset, such as evident by Figure 2, while false-negatives typically displayed low intensity values compared to other images in the data set.

7. FUTURE WORK

One possible area of future research would be to use other features aside from color to classify against, such as shape, edges, and color gradients.

One area that we would like to investigate in particular is dynamically determining the grid region sizes on an image-by-image basis. In addition, we would also like to investigate the use of irregularly sized grid regions. These regions would be optimized based on the training data and what would provide the smallest standard deviation of values within the region.

Another area we would also like to investigate would be the possibility of using multiple SVM models, each trained on a sub-



Fig. 2. A non-sunset misclassified as a sunset

set of sunset images, and combining the results to increase the accuracy and confidence of our system.