### Sunset Detector

Ethan Russell, Scott Cai

Rose-Hulman Institute of Technology

CSSE-463 Image Recognition

Email: [russelec@rose-hulman.edu](mailto:russelec@rose-hulman.edu), [cais1@rose-hulman.edu](mailto:cais1@rose-hulman.edu)

#### Abstract

Images of sunsets can usually be easily recognized by human eyes. However, machines struggle to differentiate them because sunsets can vary widely in terms of color and shape. In this paper, we will discuss the use of a trained support vector machine to identify sunset images. We feed a support vector machine data based on the mean and standard deviation of pixels from images broken up into many sub-images. We optimize the hyper-parameters of the SVM with a set of validation images, then determine the best threshold to use with the output of the SVM. Our classification algorithm is able to attain a true positive rate of 0.874, and a false positive rate of 0.098.

**1. Introduction**

Scene classification always plays a vital role when dealing with image recognition, including natural scenes and human-made buildings. Compared to buildings made up of straight lines and regular geometric shapes, natural scenes are very irregular and organic. In this paper, we attempt to classify sunset images. This could be useful, for example, in a smart automobile which adjusts the tint of the windows if it detects a sunset in the peripheral view of the driver. Sunset detection can be quite challenging because sunsets can encompass a very wide range of colors and shapes depending on clouds, air quality, humidity, season, and part of the world. Figure 0 demonstrates the variability of these sunset images.



Figure 0: This contains two sunset images that are vastly different.

One approach to this classification would be to hand-tune thresholds based on shape, color, position, size, etc. This would be time intensive and very difficult to attain accurate results. We decided to use a different method-- compute features and train the computer to find the best way possible to classify the images. We first extract many features from a large set of images, both sunset and non-sunset. We use these features to train a support vector machine. We tune the support vector machine with another set of images, and finally test the classifier. This method works fairly well, though is naturally a bit more computationally expensive than a hand-tuned algorithm.

**2. Feature Extraction**

**2.1 Image Segmentation**

To better facilitate image classification, we segmented our images into smaller sub-images, so we could have a finer resolution with our image features. We divided the original images into 49 images using a 7x7 grid. Note that our sub-images extend to the edges of the image, and as a result are not exactly the same size due to rounding. This does not affect any of the data metrics used. [1]

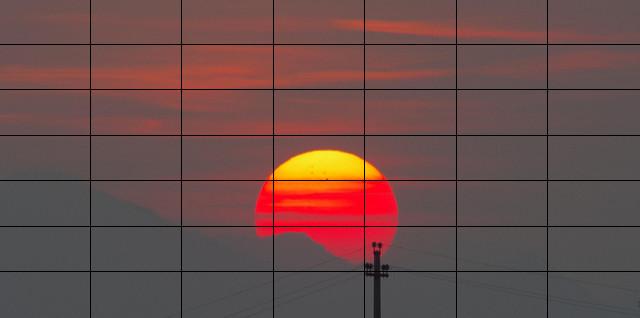


Figure 1: This shows a sample sunset image divided using the 7x7 grid.

**2.2 Conversion to LSV colorspace**

Next, we converted from RGB to LST colorspace:

L = R+G+B (1)

S = R-B

T = R-2\*G+B

The ranges for these values are:

0 ≤ L ≤ 765

-255 ≤ S ≤ 255

-510 ≤ T ≤ 510

We chose to use the LSV colorspace because it seemed to work better than RGB through former experimentation. [1]

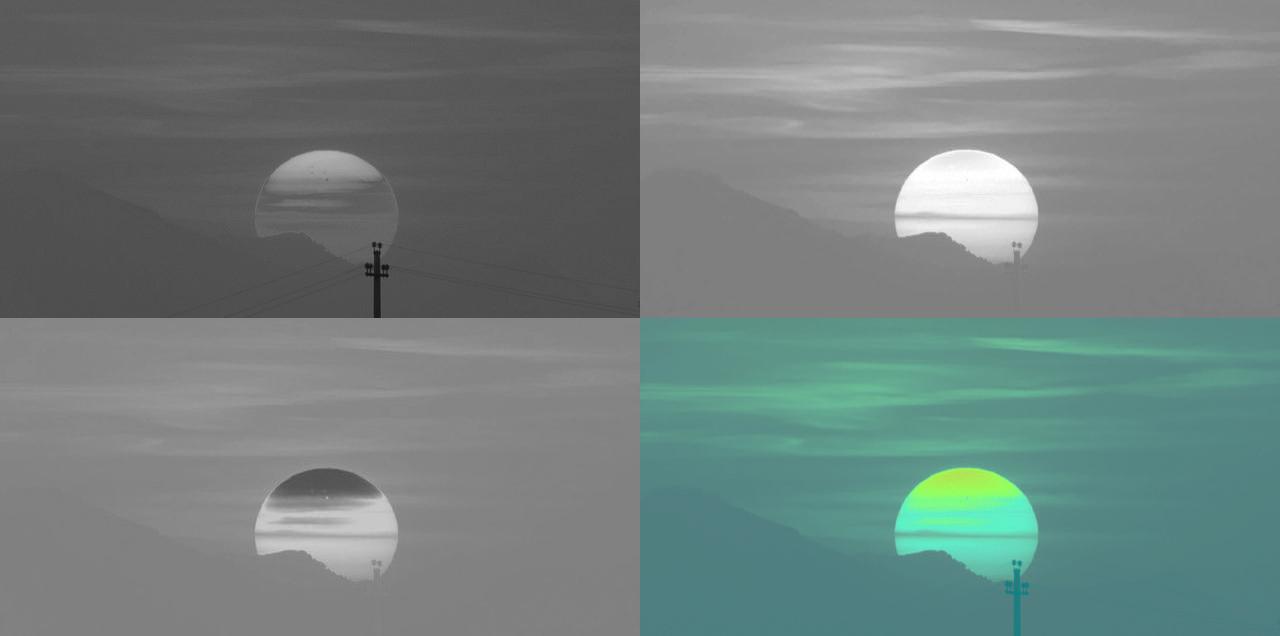


Figure 2: This displays L (top left), S (top right), T (bottom left) and LST displayed instead as an RGB (bottom right). All of these have been normalized to be between 0 and 255.

# 2.3 Feature Calculation

For each sub-image, we extract the mean and covariance for each of the three bands. The theoretical range for the mean is the same as the range for each band, as shown in section 2.2. To find the maximum value for standard deviation, we can assume half the values are on the upper bound of the range, and half on the lower bound of the range. The range of standard deviation values are:

0 ≤ σL ≤ sqrt(n/(n-1))\*375.5

0 ≤ σS ≤ sqrt(n/(n-1))\*255

0 ≤ σT ≤ sqrt(n/(n-1))\*510

\*n = number of pixels of image.

This gave us a total of 294 features to use for classification (7\*7\*6).

**3. Classification**

# 3.1 Support Vector Machines

Support vector machines are one tool we used to classify our images. Support vector machines are fed data points and the class of the data points, and create boundaries to separate the two classes. Support vectors are specific data points along the boundary separating the two classes that define the boundary. To train an SVM, it is best to feed a large set of data points, so that they may fully represent the two classes. There are a few hyper-parameters that can be changed to optimize the performance of the SVM. The kernel function can map the data to a higher dimension to better fit the classes. The kernel can be linear, polynomial, or gaussian, among others. The Gaussian kernel usually has the highest performance using real data, and has two adjustable hyperparameters. Box constraints change the cost of misclassification of a data point. Kernel width changes the radius of curvature of the boundary. Ideally, these hyperparameters should be trained to match the general shape of the training set without matching it too closely, as this would make the SVM specific to the training set and it would perform badly in the future.

**4. Experimental Setup**

We have 3200 images total, divided between 3 sets.

Table 1: This contains the number of images in each of our image sets.

|  |  |  |
| --- | --- | --- |
|  | Sunset | Non-sunset |
| Training | 800 | 800 |
| Validation | 300 | 300 |
| Test | 500 | 500 |

These images vary significantly. They range from 138x771 to 2048x6144, and account for a wide variety of sunsets and non-sunsets. For each image, we computed the 294 features as detailed in section 2. We used these features as an input to create the SVM. We used only the training set of images to create the SVM. Then, we tuned the SVM’s hyperparameters using the validation set of images. After finalizing the classifier, we calculated the accuracy of the classifier using the test set of images.

**5. Results**

**5.1 Training the SVM**

To train our SVM, we used a training set with 800 sunset and 800 non-sunset images. We gave the SVM all of the extracted features from each of these images (a 1600x294 array), as well as the class of each image. We let MATLAB’s fitcsvm function standardize the data, so each feature would be weighted equally. We chose to use the gaussian kernel to best represent our data.

**5.2 Optimizing Hyper-Parameters**

After choosing the kernel, we needed to optimize the hyper-parameters. We varied the values for the kernel width and the box constraints over a wide range [2], and tested that SVM with a validation set composed of 300 sunset and 300 non-sunset images. For each set of hyper-parameters, we computed the score using MATLAB’s predict function. We thresholded the score with a default threshold of zero to classify each image. Using the known classes, we computed the true positive rate, false positive rate, and number of support vectors used to produce that boundary. See Appendix A for this parametric sweep. We decided to use a kernel width value of 32, and a box constraint value of 128, which would yield a true positive rate of 0.9200, false positive rate of 0.0767, and 508 support vectors.

**5.3 Choosing a Threshold**

After choosing hyper-parameters, we are done changing the support vector machine. Now we can determine which threshold value would give the highest true positive rate and lowest false positive rate using the test set. To do this, we ran the test set through our SVM. To maximize TPR while minimizing FPR, we found the threshold with the minimum distance from (FPR, TPR) to (0,1). This threshold is 0.10.



Figure 3: This is the ROC curve showing true positive rate vs false positive rate using the Kernel Width and Box Constraints as specified.

Using a threshold of 0.10, we compute the following:

Table 2: This contains the performance of our completed classifier on the test set of images using a threshold of 0.10.

|  |  |  |
| --- | --- | --- |
| Detected \ True | Sunset | Non-Sunset |
| Sunset | 437 | 63 |
| Non-Sunset | 49 | 451 |

The threshold of 0.10 yields a true positive rate of 0.874, and a false positive rate of 0.098. See Appendix B for a chart of thresholds and associated classification performance.

**6. discussion**

**6.1 Performance**

Our classifier performed relatively well, yielding a true positive rate of 0.874 and a false positive rate of 0.098. However, we will examine some of the images identified by the classifier.

In Figures 4 and 5, there are two sunset images that our algorithm correctly identified. Figure 4 shows one with a relatively high score. It has lots of red, orange, and purple. Figure 5 has a sunset with only a very small amount of orange. It is possible that the SVM boundaries weigh the color orange heavily.



Figure 4: This shows the sunset image with the highest score in the test set. The score was 5.48



Figure 5: This shows a sunset image with a low score of 0.1738.

Figures 6 and 7 show sunset images that were misidentified by the classifier. Figure 6 shows an image that was barely misidentified, and Figure 7 shows an image that was identified strongly as a non-sunset image. The image in Figure 6 has faint glimmers of orange in the center of the image, but otherwise has little to no red, orange or purple. It is possible that the small amount of orange raised the score. Figure 7 is somewhat perplexing – it contains mostly yellow pixels, which is close to orange. However, very few of them look actually orange. It is possible that the algorithm discriminates against the color yellow, or possibly it has a very specific distinction between yellow and orange.



Figure 6: This shows a sunset image that was misidentified by our algorithm with a score of 0.0680.



Figure 7: This shows the sunset image with the lowest score, -2.9085. Our algorithm misidentified it as a non-sunset.

Figures 8 and 9 show non-sunset images that were correctly identified. Figure 8 has a very low score, and Figure 9 was almost identified as a sunset. Figure 8 is mostly composed of black and white, so it has none of the colors of a sunset, yielding a very low score. The image in Figure 9, on first inspection, is very close to the image in Figure 7. However, Figure 9 has a much higher score than the image in Figure 7. This could be explained by the presence of more orange hues in Figure 9 than in Figure 7.



Figure 8: This shows the non-sunset image with the lowest score: -6.2746.

****

Figure 9: This shows a non-sunset image that was correctly identified, but had a high score of 0.0929.

Figures 10 and 11 contain non-sunset images which our algorithm identified as sunsets. Figure 10 was barely identified as a sunset, and Figure 11 was strongly identified as a sunset. Both of these images contain a large amount of orange pixels, which suggest that the algorithm strongly weighs orange pixels as sunsets.



Figure 10: This shows an image that was barely identified as a sunset. It had a score of 0.1815.

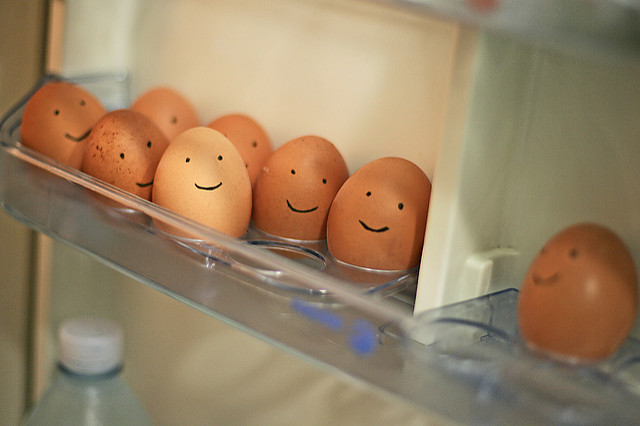


Figure 11: This shows the non-sunset image with the highest score, 1.7366.

**7. conclusions**

From the analysis above, we clearly see that our SVM weighs orange colors during its process of detecting sunsets. This is somewhat expected, as the SVM only uses mean and standard deviation of the colors. It might not be possible to separate all sunsets and non-sunsets using just these metrics. To improve its performance, edges and shapes could be additional features to train the SVM with. We would also like to try splitting the images into a larger number of sub-images. This could theoretically give a little more weight to shapes or positions. A larger training set could also provide better boundaries. We could also use a convolutional neural network instead of a support vector machine.

**8. References**

|  |  |
| --- | --- |
| [1] | M. Boutell, J. Luo and R. T. Gray, "Sunset Scene Classification using Simulated Image Recomposition," in *IEEE International Conference on Multimedia and Expo*, Baltimore, MD, 2003. |
| [2] | C.-. W. Hsu, C.-C. Chang and C.-J. Lin, "A Practical Guide to Support Vector Classification," National Taiwan University, Taipei, Taiwan, 2003. |

### Appendix A

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kernel Width | Box Constraint | True Positive Rate | False Positive Rate | Number of Support Vectors |
| 0.03125 | 0.03125 | 0 | 0 | 1600 |
| 0.03125 | 0.0625 | 0 | 0 | 1600 |
| 0.03125 | 0.125 | 0 | 0 | 1600 |
| 0.03125 | 0.25 | 0 | 0 | 1600 |
| 0.03125 | 0.5 | 0 | 0 | 1600 |
| 0.03125 | 1 | 0 | 0 | 1598 |
| 0.03125 | 2 | 0 | 0 | 1594 |
| 0.03125 | 4 | 0 | 0 | 1594 |
| 0.03125 | 8 | 0 | 0 | 1594 |
| 0.03125 | 16 | 0 | 0 | 1594 |
| 0.03125 | 32 | 0 | 0 | 1594 |
| 0.03125 | 64 | 0 | 0 | 1594 |
| 0.03125 | 128 | 0 | 0 | 1594 |
| 0.03125 | 256 | 0 | 0 | 1594 |
| 0.03125 | 512 | 0 | 0 | 1594 |
| 0.03125 | 1024 | 0 | 0 | 1594 |
| 0.0625 | 0.03125 | 0 | 0 | 1600 |
| 0.0625 | 0.0625 | 0 | 0 | 1600 |
| 0.0625 | 0.125 | 0 | 0 | 1600 |
| 0.0625 | 0.25 | 0 | 0 | 1600 |
| 0.0625 | 0.5 | 0 | 0 | 1600 |
| 0.0625 | 1 | 0 | 0 | 1597 |
| 0.0625 | 2 | 0 | 0 | 1593 |
| 0.0625 | 4 | 0 | 0 | 1593 |
| 0.0625 | 8 | 0 | 0 | 1593 |
| 0.0625 | 16 | 0 | 0 | 1593 |
| 0.0625 | 32 | 0 | 0 | 1593 |
| 0.0625 | 64 | 0 | 0 | 1593 |
| 0.0625 | 128 | 0 | 0 | 1593 |
| 0.0625 | 256 | 0 | 0 | 1593 |
| 0.0625 | 512 | 0 | 0 | 1593 |
| 0.0625 | 1024 | 0 | 0 | 1593 |
| 0.125 | 0.03125 | 0 | 0 | 1600 |
| 0.125 | 0.0625 | 0 | 0 | 1600 |
| 0.125 | 0.125 | 0 | 0 | 1600 |
| 0.125 | 0.25 | 0 | 0 | 1600 |
| 0.125 | 0.5 | 0 | 0 | 1600 |
| 0.125 | 1 | 0 | 0 | 1597 |
| 0.125 | 2 | 0 | 0 | 1593 |
| 0.125 | 4 | 0 | 0 | 1593 |
| 0.125 | 8 | 0 | 0 | 1593 |
| 0.125 | 16 | 0 | 0 | 1593 |
| 0.125 | 32 | 0 | 0 | 1593 |
| 0.125 | 64 | 0 | 0 | 1593 |
| 0.125 | 128 | 0 | 0 | 1593 |
| 0.125 | 256 | 0 | 0 | 1593 |
| 0.125 | 512 | 0 | 0 | 1593 |
| 0.125 | 1024 | 0 | 0 | 1593 |
| 0.25 | 0.03125 | 0 | 0 | 1600 |
| 0.25 | 0.0625 | 0 | 0 | 1600 |
| 0.25 | 0.125 | 0 | 0 | 1600 |
| 0.25 | 0.25 | 0 | 0 | 1600 |
| 0.25 | 0.5 | 0 | 0 | 1600 |
| 0.25 | 1 | 0 | 0 | 1597 |
| 0.25 | 2 | 0 | 0 | 1593 |
| 0.25 | 4 | 0 | 0 | 1593 |
| 0.25 | 8 | 0 | 0 | 1593 |
| 0.25 | 16 | 0 | 0 | 1593 |
| 0.25 | 32 | 0 | 0 | 1593 |
| 0.25 | 64 | 0 | 0 | 1593 |
| 0.25 | 128 | 0 | 0 | 1593 |
| 0.25 | 256 | 0 | 0 | 1593 |
| 0.25 | 512 | 0 | 0 | 1593 |
| 0.25 | 1024 | 0 | 0 | 1593 |
| 0.5 | 0.03125 | 0 | 0 | 1600 |
| 0.5 | 0.0625 | 0 | 0 | 1600 |
| 0.5 | 0.125 | 0 | 0 | 1600 |
| 0.5 | 0.25 | 0 | 0 | 1600 |
| 0.5 | 0.5 | 0 | 0 | 1600 |
| 0.5 | 1 | 0 | 0 | 1597 |
| 0.5 | 2 | 0 | 0 | 1593 |
| 0.5 | 4 | 0 | 0 | 1593 |
| 0.5 | 8 | 0 | 0 | 1593 |
| 0.5 | 16 | 0 | 0 | 1593 |
| 0.5 | 32 | 0 | 0 | 1593 |
| 0.5 | 64 | 0 | 0 | 1593 |
| 0.5 | 128 | 0 | 0 | 1593 |
| 0.5 | 256 | 0 | 0 | 1593 |
| 0.5 | 512 | 0 | 0 | 1593 |
| 0.5 | 1024 | 0 | 0 | 1593 |
| 1 | 0.03125 | 0 | 0 | 1600 |
| 1 | 0.0625 | 0 | 0 | 1600 |
| 1 | 0.125 | 0 | 0 | 1600 |
| 1 | 0.25 | 0 | 0 | 1600 |
| 1 | 0.5 | 0 | 0 | 1600 |
| 1 | 1 | 0 | 0 | 1597 |
| 1 | 2 | 0 | 0 | 1593 |
| 1 | 4 | 0 | 0 | 1593 |
| 1 | 8 | 0 | 0 | 1593 |
| 1 | 16 | 0 | 0 | 1593 |
| 1 | 32 | 0 | 0 | 1593 |
| 1 | 64 | 0 | 0 | 1593 |
| 1 | 128 | 0 | 0 | 1593 |
| 1 | 256 | 0 | 0 | 1593 |
| 1 | 512 | 0 | 0 | 1593 |
| 1 | 1024 | 0 | 0 | 1593 |
| 2 | 0.03125 | 0 | 0 | 1600 |
| 2 | 0.0625 | 0 | 0 | 1600 |
| 2 | 0.125 | 0 | 0 | 1600 |
| 2 | 0.25 | 0 | 0 | 1600 |
| 2 | 0.5 | 0 | 0 | 1600 |
| 2 | 1 | 0 | 0 | 1596 |
| 2 | 2 | 0 | 0 | 1593 |
| 2 | 4 | 0 | 0 | 1593 |
| 2 | 8 | 0 | 0 | 1593 |
| 2 | 16 | 0 | 0 | 1593 |
| 2 | 32 | 0 | 0 | 1593 |
| 2 | 64 | 0 | 0 | 1593 |
| 2 | 128 | 0 | 0 | 1593 |
| 2 | 256 | 0 | 0 | 1593 |
| 2 | 512 | 0 | 0 | 1593 |
| 2 | 1024 | 0 | 0 | 1593 |
| 4 | 0.03125 | 0 | 0 | 1600 |
| 4 | 0.0625 | 0 | 0 | 1600 |
| 4 | 0.125 | 0 | 0 | 1600 |
| 4 | 0.25 | 0 | 0 | 1600 |
| 4 | 0.5 | 0 | 0 | 1599 |
| 4 | 1 | 0.116667 | 0.003333 | 1592 |
| 4 | 2 | 0.14 | 0.01 | 1590 |
| 4 | 4 | 0.14 | 0.01 | 1590 |
| 4 | 8 | 0.14 | 0.01 | 1590 |
| 4 | 16 | 0.14 | 0.01 | 1590 |
| 4 | 32 | 0.14 | 0.01 | 1590 |
| 4 | 64 | 0.14 | 0.01 | 1590 |
| 4 | 128 | 0.14 | 0.01 | 1590 |
| 4 | 256 | 0.14 | 0.01 | 1590 |
| 4 | 512 | 0.14 | 0.01 | 1590 |
| 4 | 1024 | 0.14 | 0.01 | 1590 |
| 8 | 0.03125 | 0.173333 | 0.003333 | 1600 |
| 8 | 0.0625 | 0.226667 | 0.013333 | 1594 |
| 8 | 0.125 | 0.3 | 0.016667 | 1531 |
| 8 | 0.25 | 0.4 | 0.043333 | 1452 |
| 8 | 0.5 | 0.526667 | 0.04 | 1380 |
| 8 | 1 | 0.703333 | 0.036667 | 1365 |
| 8 | 2 | 0.723333 | 0.03 | 1383 |
| 8 | 4 | 0.72 | 0.03 | 1388 |
| 8 | 8 | 0.72 | 0.03 | 1388 |
| 8 | 16 | 0.72 | 0.03 | 1388 |
| 8 | 32 | 0.72 | 0.03 | 1388 |
| 8 | 64 | 0.72 | 0.03 | 1388 |
| 8 | 128 | 0.72 | 0.03 | 1388 |
| 8 | 256 | 0.72 | 0.03 | 1388 |
| 8 | 512 | 0.72 | 0.03 | 1388 |
| 8 | 1024 | 0.72 | 0.03 | 1388 |
| 16 | 0.03125 | 0.8 | 0.096667 | 1328 |
| 16 | 0.0625 | 0.84 | 0.093333 | 1127 |
| 16 | 0.125 | 0.86 | 0.086667 | 977 |
| 16 | 0.25 | 0.876667 | 0.083333 | 859 |
| 16 | 0.5 | 0.9 | 0.066667 | 788 |
| 16 | 1 | 0.903333 | 0.056667 | 758 |
| 16 | 2 | 0.913333 | 0.056667 | 763 |
| 16 | 4 | 0.91 | 0.063333 | 766 |
| 16 | 8 | 0.906667 | 0.053333 | 775 |
| 16 | 16 | 0.903333 | 0.043333 | 760 |
| 16 | 32 | 0.903333 | 0.043333 | 753 |
| 16 | 64 | 0.903333 | 0.043333 | 753 |
| 16 | 128 | 0.903333 | 0.043333 | 753 |
| 16 | 256 | 0.903333 | 0.043333 | 753 |
| 16 | 512 | 0.903333 | 0.043333 | 753 |
| 16 | 1024 | 0.903333 | 0.043333 | 753 |
| 32 | 0.03125 | 0.91 | 0.186667 | 1320 |
| 32 | 0.0625 | 0.91 | 0.16 | 1123 |
| 32 | 0.125 | 0.916667 | 0.133333 | 957 |
| 32 | 0.25 | 0.923333 | 0.13 | 824 |
| 32 | 0.5 | 0.916667 | 0.11 | 730 |
| 32 | 1 | 0.91 | 0.1 | 660 |
| 32 | 2 | 0.903333 | 0.093333 | 611 |
| 32 | 4 | 0.926667 | 0.076667 | 582 |
| 32 | 8 | 0.92 | 0.076667 | 573 |
| 32 | 16 | 0.916667 | 0.08 | 549 |
| 32 | 32 | 0.916667 | 0.086667 | 543 |
| 32 | 64 | 0.91 | 0.083333 | 533 |
| 32 | 128 | 0.92 | 0.076667 | 508 |
| 32 | 256 | 0.916667 | 0.083333 | 505 |
| 32 | 512 | 0.916667 | 0.083333 | 505 |
| 32 | 1024 | 0.916667 | 0.083333 | 505 |
| 64 | 0.03125 | 0.953333 | 0.37 | 1568 |
| 64 | 0.0625 | 0.936667 | 0.246667 | 1410 |
| 64 | 0.125 | 0.93 | 0.193333 | 1215 |
| 64 | 0.25 | 0.93 | 0.183333 | 1036 |
| 64 | 0.5 | 0.92 | 0.17 | 894 |
| 64 | 1 | 0.913333 | 0.136667 | 781 |
| 64 | 2 | 0.91 | 0.116667 | 706 |
| 64 | 4 | 0.916667 | 0.13 | 635 |
| 64 | 8 | 0.906667 | 0.133333 | 594 |
| 64 | 16 | 0.896667 | 0.113333 | 551 |
| 64 | 32 | 0.913333 | 0.096667 | 529 |
| 64 | 64 | 0.916667 | 0.1 | 521 |
| 64 | 128 | 0.903333 | 0.103333 | 509 |
| 64 | 256 | 0.906667 | 0.096667 | 497 |
| 64 | 512 | 0.903333 | 0.103333 | 470 |
| 64 | 1024 | 0.903333 | 0.096667 | 447 |
| 128 | 0.03125 | 0.996667 | 0.796667 | 1600 |
| 128 | 0.0625 | 0.993333 | 0.72 | 1598 |
| 128 | 0.125 | 0.95 | 0.356667 | 1548 |
| 128 | 0.25 | 0.936667 | 0.236667 | 1380 |
| 128 | 0.5 | 0.926667 | 0.196667 | 1189 |
| 128 | 1 | 0.926667 | 0.19 | 1021 |
| 128 | 2 | 0.923333 | 0.176667 | 882 |
| 128 | 4 | 0.91 | 0.146667 | 778 |
| 128 | 8 | 0.906667 | 0.14 | 695 |
| 128 | 16 | 0.91 | 0.15 | 641 |
| 128 | 32 | 0.906667 | 0.156667 | 597 |
| 128 | 64 | 0.9 | 0.143333 | 558 |
| 128 | 128 | 0.893333 | 0.136667 | 535 |
| 128 | 256 | 0.9 | 0.13 | 506 |
| 128 | 512 | 0.89 | 0.116667 | 496 |
| 128 | 1024 | 0.883333 | 0.136667 | 475 |
| 256 | 0.03125 | 0.996667 | 0.886667 | 1600 |
| 256 | 0.0625 | 0.996667 | 0.886667 | 1600 |
| 256 | 0.125 | 0.996667 | 0.886667 | 1600 |
| 256 | 0.25 | 0.99 | 0.696667 | 1598 |
| 256 | 0.5 | 0.95 | 0.353333 | 1542 |
| 256 | 1 | 0.936667 | 0.236667 | 1373 |
| 256 | 2 | 0.926667 | 0.2 | 1187 |
| 256 | 4 | 0.923333 | 0.19 | 1016 |
| 256 | 8 | 0.923333 | 0.17 | 882 |
| 256 | 16 | 0.913333 | 0.15 | 775 |
| 256 | 32 | 0.913333 | 0.15 | 691 |
| 256 | 64 | 0.906667 | 0.153333 | 635 |
| 256 | 128 | 0.906667 | 0.163333 | 593 |
| 256 | 256 | 0.893333 | 0.16 | 564 |
| 256 | 512 | 0.893333 | 0.146667 | 543 |
| 256 | 1024 | 0.876667 | 0.16 | 523 |
| 512 | 0.03125 | 0.996667 | 0.896667 | 1600 |
| 512 | 0.0625 | 0.996667 | 0.896667 | 1600 |
| 512 | 0.125 | 0.996667 | 0.896667 | 1600 |
| 512 | 0.25 | 0.996667 | 0.896667 | 1600 |
| 512 | 0.5 | 0.996667 | 0.896667 | 1600 |
| 512 | 1 | 0.99 | 0.696667 | 1598 |
| 512 | 2 | 0.95 | 0.356667 | 1542 |
| 512 | 4 | 0.936667 | 0.24 | 1370 |
| 512 | 8 | 0.926667 | 0.2 | 1185 |
| 512 | 16 | 0.923333 | 0.193333 | 1015 |
| 512 | 32 | 0.923333 | 0.166667 | 882 |
| 512 | 64 | 0.913333 | 0.146667 | 779 |
| 512 | 128 | 0.913333 | 0.15 | 693 |
| 512 | 256 | 0.906667 | 0.156667 | 633 |
| 512 | 512 | 0.903333 | 0.17 | 592 |
| 512 | 1024 | 0.89 | 0.16 | 561 |
| 1024 | 0.03125 | 1 | 0.896667 | 1600 |
| 1024 | 0.0625 | 1 | 0.896667 | 1600 |
| 1024 | 0.125 | 1 | 0.896667 | 1600 |
| 1024 | 0.25 | 1 | 0.896667 | 1600 |
| 1024 | 0.5 | 1 | 0.896667 | 1600 |
| 1024 | 1 | 1 | 0.896667 | 1600 |
| 1024 | 2 | 1 | 0.896667 | 1600 |
| 1024 | 4 | 0.99 | 0.7 | 1598 |
| 1024 | 8 | 0.95 | 0.356667 | 1542 |
| 1024 | 16 | 0.936667 | 0.24 | 1370 |
| 1024 | 32 | 0.926667 | 0.2 | 1185 |
| 1024 | 64 | 0.923333 | 0.193333 | 1014 |
| 1024 | 128 | 0.923333 | 0.166667 | 882 |
| 1024 | 256 | 0.913333 | 0.146667 | 777 |
| 1024 | 512 | 0.913333 | 0.153333 | 692 |
| 1024 | 1024 | 0.906667 | 0.156667 | 633 |

### Appendix B

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Threshold | True Positive Rate | False Positive Rate | True Positives | True Negatives | False Positives | False Negatives |
| -2 | 0.992 | 0.592 | 496 | 204 | 296 | 4 |
| -1.98 | 0.992 | 0.588 | 496 | 206 | 294 | 4 |
| -1.96 | 0.992 | 0.582 | 496 | 209 | 291 | 4 |
| -1.94 | 0.992 | 0.582 | 496 | 209 | 291 | 4 |
| -1.92 | 0.992 | 0.576 | 496 | 212 | 288 | 4 |
| -1.9 | 0.992 | 0.568 | 496 | 216 | 284 | 4 |
| -1.88 | 0.99 | 0.562 | 495 | 219 | 281 | 5 |
| -1.86 | 0.99 | 0.558 | 495 | 221 | 279 | 5 |
| -1.84 | 0.988 | 0.554 | 494 | 223 | 277 | 6 |
| -1.82 | 0.988 | 0.552 | 494 | 224 | 276 | 6 |
| -1.8 | 0.988 | 0.55 | 494 | 225 | 275 | 6 |
| -1.78 | 0.988 | 0.542 | 494 | 229 | 271 | 6 |
| -1.76 | 0.988 | 0.542 | 494 | 229 | 271 | 6 |
| -1.74 | 0.986 | 0.538 | 493 | 231 | 269 | 7 |
| -1.72 | 0.982 | 0.536 | 491 | 232 | 268 | 9 |
| -1.7 | 0.98 | 0.53 | 490 | 235 | 265 | 10 |
| -1.68 | 0.98 | 0.53 | 490 | 235 | 265 | 10 |
| -1.66 | 0.98 | 0.526 | 490 | 237 | 263 | 10 |
| -1.64 | 0.978 | 0.524 | 489 | 238 | 262 | 11 |
| -1.62 | 0.978 | 0.516 | 489 | 242 | 258 | 11 |
| -1.6 | 0.976 | 0.514 | 488 | 243 | 257 | 12 |
| -1.58 | 0.974 | 0.506 | 487 | 247 | 253 | 13 |
| -1.56 | 0.974 | 0.504 | 487 | 248 | 252 | 13 |
| -1.54 | 0.974 | 0.494 | 487 | 253 | 247 | 13 |
| -1.52 | 0.974 | 0.492 | 487 | 254 | 246 | 13 |
| -1.5 | 0.974 | 0.478 | 487 | 261 | 239 | 13 |
| -1.48 | 0.974 | 0.474 | 487 | 263 | 237 | 13 |
| -1.46 | 0.974 | 0.462 | 487 | 269 | 231 | 13 |
| -1.44 | 0.974 | 0.458 | 487 | 271 | 229 | 13 |
| -1.42 | 0.974 | 0.456 | 487 | 272 | 228 | 13 |
| -1.4 | 0.974 | 0.452 | 487 | 274 | 226 | 13 |
| -1.38 | 0.974 | 0.448 | 487 | 276 | 224 | 13 |
| -1.36 | 0.974 | 0.446 | 487 | 277 | 223 | 13 |
| -1.34 | 0.974 | 0.442 | 487 | 279 | 221 | 13 |
| -1.32 | 0.974 | 0.438 | 487 | 281 | 219 | 13 |
| -1.3 | 0.974 | 0.436 | 487 | 282 | 218 | 13 |
| -1.28 | 0.972 | 0.426 | 486 | 287 | 213 | 14 |
| -1.26 | 0.97 | 0.42 | 485 | 290 | 210 | 15 |
| -1.24 | 0.968 | 0.41 | 484 | 295 | 205 | 16 |
| -1.22 | 0.966 | 0.398 | 483 | 301 | 199 | 17 |
| -1.2 | 0.964 | 0.392 | 482 | 304 | 196 | 18 |
| -1.18 | 0.964 | 0.386 | 482 | 307 | 193 | 18 |
| -1.16 | 0.962 | 0.38 | 481 | 310 | 190 | 19 |
| -1.14 | 0.96 | 0.376 | 480 | 312 | 188 | 20 |
| -1.12 | 0.958 | 0.362 | 479 | 319 | 181 | 21 |
| -1.1 | 0.958 | 0.356 | 479 | 322 | 178 | 21 |
| -1.08 | 0.956 | 0.35 | 478 | 325 | 175 | 22 |
| -1.06 | 0.956 | 0.346 | 478 | 327 | 173 | 22 |
| -1.04 | 0.954 | 0.342 | 477 | 329 | 171 | 23 |
| -1.02 | 0.952 | 0.338 | 476 | 331 | 169 | 24 |
| -1 | 0.952 | 0.332 | 476 | 334 | 166 | 24 |
| -0.98 | 0.952 | 0.324 | 476 | 338 | 162 | 24 |
| -0.96 | 0.952 | 0.318 | 476 | 341 | 159 | 24 |
| -0.94 | 0.952 | 0.316 | 476 | 342 | 158 | 24 |
| -0.92 | 0.952 | 0.312 | 476 | 344 | 156 | 24 |
| -0.9 | 0.948 | 0.31 | 474 | 345 | 155 | 26 |
| -0.88 | 0.948 | 0.306 | 474 | 347 | 153 | 26 |
| -0.86 | 0.948 | 0.302 | 474 | 349 | 151 | 26 |
| -0.84 | 0.948 | 0.296 | 474 | 352 | 148 | 26 |
| -0.82 | 0.944 | 0.294 | 472 | 353 | 147 | 28 |
| -0.8 | 0.942 | 0.29 | 471 | 355 | 145 | 29 |
| -0.78 | 0.942 | 0.278 | 471 | 361 | 139 | 29 |
| -0.76 | 0.938 | 0.278 | 469 | 361 | 139 | 31 |
| -0.74 | 0.938 | 0.274 | 469 | 363 | 137 | 31 |
| -0.72 | 0.936 | 0.268 | 468 | 366 | 134 | 32 |
| -0.7 | 0.934 | 0.262 | 467 | 369 | 131 | 33 |
| -0.68 | 0.934 | 0.258 | 467 | 371 | 129 | 33 |
| -0.66 | 0.932 | 0.254 | 466 | 373 | 127 | 34 |
| -0.64 | 0.928 | 0.254 | 464 | 373 | 127 | 36 |
| -0.62 | 0.928 | 0.248 | 464 | 376 | 124 | 36 |
| -0.6 | 0.928 | 0.24 | 464 | 380 | 120 | 36 |
| -0.58 | 0.926 | 0.232 | 463 | 384 | 116 | 37 |
| -0.56 | 0.926 | 0.226 | 463 | 387 | 113 | 37 |
| -0.54 | 0.926 | 0.224 | 463 | 388 | 112 | 37 |
| -0.52 | 0.926 | 0.21 | 463 | 395 | 105 | 37 |
| -0.5 | 0.924 | 0.208 | 462 | 396 | 104 | 38 |
| -0.48 | 0.924 | 0.202 | 462 | 399 | 101 | 38 |
| -0.46 | 0.922 | 0.202 | 461 | 399 | 101 | 39 |
| -0.44 | 0.92 | 0.194 | 460 | 403 | 97 | 40 |
| -0.42 | 0.92 | 0.194 | 460 | 403 | 97 | 40 |
| -0.4 | 0.918 | 0.194 | 459 | 403 | 97 | 41 |
| -0.38 | 0.918 | 0.192 | 459 | 404 | 96 | 41 |
| -0.36 | 0.916 | 0.19 | 458 | 405 | 95 | 42 |
| -0.34 | 0.916 | 0.182 | 458 | 409 | 91 | 42 |
| -0.32 | 0.914 | 0.176 | 457 | 412 | 88 | 43 |
| -0.3 | 0.906 | 0.176 | 453 | 412 | 88 | 47 |
| -0.28 | 0.906 | 0.174 | 453 | 413 | 87 | 47 |
| -0.26 | 0.906 | 0.17 | 453 | 415 | 85 | 47 |
| -0.24 | 0.904 | 0.168 | 452 | 416 | 84 | 48 |
| -0.22 | 0.904 | 0.166 | 452 | 417 | 83 | 48 |
| -0.2 | 0.904 | 0.16 | 452 | 420 | 80 | 48 |
| -0.18 | 0.904 | 0.156 | 452 | 422 | 78 | 48 |
| -0.16 | 0.902 | 0.148 | 451 | 426 | 74 | 49 |
| -0.14 | 0.9 | 0.146 | 450 | 427 | 73 | 50 |
| -0.12 | 0.898 | 0.144 | 449 | 428 | 72 | 51 |
| -0.1 | 0.896 | 0.14 | 448 | 430 | 70 | 52 |
| -0.08 | 0.892 | 0.132 | 446 | 434 | 66 | 54 |
| -0.06 | 0.884 | 0.13 | 442 | 435 | 65 | 58 |
| -0.04 | 0.884 | 0.128 | 442 | 436 | 64 | 58 |
| -0.02 | 0.882 | 0.124 | 441 | 438 | 62 | 59 |
| 0 | 0.882 | 0.12 | 441 | 440 | 60 | 59 |
| 0.02 | 0.882 | 0.112 | 441 | 444 | 56 | 59 |
| 0.04 | 0.88 | 0.11 | 440 | 445 | 55 | 60 |
| 0.06 | 0.876 | 0.108 | 438 | 446 | 54 | 62 |
| 0.08 | 0.874 | 0.104 | 437 | 448 | 52 | 63 |
| 0.1 | 0.874 | 0.098 | 437 | 451 | 49 | 63 |
| 0.12 | 0.87 | 0.096 | 435 | 452 | 48 | 65 |
| 0.14 | 0.866 | 0.096 | 433 | 452 | 48 | 67 |
| 0.16 | 0.864 | 0.096 | 432 | 452 | 48 | 68 |
| 0.18 | 0.86 | 0.094 | 430 | 453 | 47 | 70 |
| 0.2 | 0.86 | 0.092 | 430 | 454 | 46 | 70 |
| 0.22 | 0.858 | 0.092 | 429 | 454 | 46 | 71 |
| 0.24 | 0.854 | 0.092 | 427 | 454 | 46 | 73 |
| 0.26 | 0.852 | 0.088 | 426 | 456 | 44 | 74 |
| 0.28 | 0.852 | 0.086 | 426 | 457 | 43 | 74 |
| 0.3 | 0.848 | 0.084 | 424 | 458 | 42 | 76 |
| 0.32 | 0.844 | 0.084 | 422 | 458 | 42 | 78 |
| 0.34 | 0.842 | 0.082 | 421 | 459 | 41 | 79 |
| 0.36 | 0.836 | 0.082 | 418 | 459 | 41 | 82 |
| 0.38 | 0.828 | 0.08 | 414 | 460 | 40 | 86 |
| 0.4 | 0.826 | 0.08 | 413 | 460 | 40 | 87 |
| 0.42 | 0.824 | 0.078 | 412 | 461 | 39 | 88 |
| 0.44 | 0.82 | 0.07 | 410 | 465 | 35 | 90 |
| 0.46 | 0.818 | 0.07 | 409 | 465 | 35 | 91 |
| 0.48 | 0.818 | 0.07 | 409 | 465 | 35 | 91 |
| 0.5 | 0.812 | 0.068 | 406 | 466 | 34 | 94 |
| 0.52 | 0.806 | 0.068 | 403 | 466 | 34 | 97 |
| 0.54 | 0.802 | 0.064 | 401 | 468 | 32 | 99 |
| 0.56 | 0.8 | 0.06 | 400 | 470 | 30 | 100 |
| 0.58 | 0.8 | 0.056 | 400 | 472 | 28 | 100 |
| 0.6 | 0.8 | 0.054 | 400 | 473 | 27 | 100 |
| 0.62 | 0.8 | 0.048 | 400 | 476 | 24 | 100 |
| 0.64 | 0.792 | 0.048 | 396 | 476 | 24 | 104 |
| 0.66 | 0.788 | 0.048 | 394 | 476 | 24 | 106 |
| 0.68 | 0.782 | 0.048 | 391 | 476 | 24 | 109 |
| 0.7 | 0.778 | 0.046 | 389 | 477 | 23 | 111 |
| 0.72 | 0.776 | 0.046 | 388 | 477 | 23 | 112 |
| 0.74 | 0.77 | 0.046 | 385 | 477 | 23 | 115 |
| 0.76 | 0.768 | 0.042 | 384 | 479 | 21 | 116 |
| 0.78 | 0.768 | 0.04 | 384 | 480 | 20 | 116 |
| 0.8 | 0.76 | 0.04 | 380 | 480 | 20 | 120 |
| 0.82 | 0.748 | 0.038 | 374 | 481 | 19 | 126 |
| 0.84 | 0.738 | 0.038 | 369 | 481 | 19 | 131 |
| 0.86 | 0.734 | 0.036 | 367 | 482 | 18 | 133 |
| 0.88 | 0.732 | 0.036 | 366 | 482 | 18 | 134 |
| 0.9 | 0.726 | 0.036 | 363 | 482 | 18 | 137 |
| 0.92 | 0.722 | 0.036 | 361 | 482 | 18 | 139 |
| 0.94 | 0.72 | 0.034 | 360 | 483 | 17 | 140 |
| 0.96 | 0.716 | 0.034 | 358 | 483 | 17 | 142 |
| 0.98 | 0.712 | 0.032 | 356 | 484 | 16 | 144 |
| 1 | 0.704 | 0.028 | 352 | 486 | 14 | 148 |
| 1.02 | 0.702 | 0.028 | 351 | 486 | 14 | 149 |
| 1.04 | 0.698 | 0.028 | 349 | 486 | 14 | 151 |
| 1.06 | 0.694 | 0.024 | 347 | 488 | 12 | 153 |
| 1.08 | 0.694 | 0.022 | 347 | 489 | 11 | 153 |
| 1.1 | 0.688 | 0.018 | 344 | 491 | 9 | 156 |
| 1.12 | 0.682 | 0.018 | 341 | 491 | 9 | 159 |
| 1.14 | 0.672 | 0.018 | 336 | 491 | 9 | 164 |
| 1.16 | 0.664 | 0.018 | 332 | 491 | 9 | 168 |
| 1.18 | 0.66 | 0.018 | 330 | 491 | 9 | 170 |
| 1.2 | 0.656 | 0.018 | 328 | 491 | 9 | 172 |
| 1.22 | 0.652 | 0.018 | 326 | 491 | 9 | 174 |
| 1.24 | 0.646 | 0.016 | 323 | 492 | 8 | 177 |
| 1.26 | 0.644 | 0.016 | 322 | 492 | 8 | 178 |
| 1.28 | 0.636 | 0.016 | 318 | 492 | 8 | 182 |
| 1.3 | 0.632 | 0.016 | 316 | 492 | 8 | 184 |
| 1.32 | 0.624 | 0.014 | 312 | 493 | 7 | 188 |
| 1.34 | 0.62 | 0.014 | 310 | 493 | 7 | 190 |
| 1.36 | 0.616 | 0.014 | 308 | 493 | 7 | 192 |
| 1.38 | 0.606 | 0.014 | 303 | 493 | 7 | 197 |
| 1.4 | 0.602 | 0.012 | 301 | 494 | 6 | 199 |
| 1.42 | 0.6 | 0.01 | 300 | 495 | 5 | 200 |
| 1.44 | 0.594 | 0.01 | 297 | 495 | 5 | 203 |
| 1.46 | 0.582 | 0.008 | 291 | 496 | 4 | 209 |
| 1.48 | 0.578 | 0.008 | 289 | 496 | 4 | 211 |
| 1.5 | 0.568 | 0.008 | 284 | 496 | 4 | 216 |
| 1.52 | 0.562 | 0.006 | 281 | 497 | 3 | 219 |
| 1.54 | 0.556 | 0.006 | 278 | 497 | 3 | 222 |
| 1.56 | 0.554 | 0.004 | 277 | 498 | 2 | 223 |
| 1.58 | 0.542 | 0.004 | 271 | 498 | 2 | 229 |
| 1.6 | 0.524 | 0.004 | 262 | 498 | 2 | 238 |
| 1.62 | 0.516 | 0.002 | 258 | 499 | 1 | 242 |
| 1.64 | 0.512 | 0.002 | 256 | 499 | 1 | 244 |
| 1.66 | 0.51 | 0.002 | 255 | 499 | 1 | 245 |
| 1.68 | 0.504 | 0.002 | 252 | 499 | 1 | 248 |
| 1.7 | 0.496 | 0.002 | 248 | 499 | 1 | 252 |
| 1.72 | 0.492 | 0.002 | 246 | 499 | 1 | 254 |
| 1.74 | 0.484 | 0 | 242 | 500 | 0 | 258 |
| 1.76 | 0.476 | 0 | 238 | 500 | 0 | 262 |
| 1.78 | 0.468 | 0 | 234 | 500 | 0 | 266 |
| 1.8 | 0.464 | 0 | 232 | 500 | 0 | 268 |
| 1.82 | 0.46 | 0 | 230 | 500 | 0 | 270 |
| 1.84 | 0.456 | 0 | 228 | 500 | 0 | 272 |
| 1.86 | 0.454 | 0 | 227 | 500 | 0 | 273 |
| 1.88 | 0.448 | 0 | 224 | 500 | 0 | 276 |
| 1.9 | 0.444 | 0 | 222 | 500 | 0 | 278 |
| 1.92 | 0.442 | 0 | 221 | 500 | 0 | 279 |
| 1.94 | 0.438 | 0 | 219 | 500 | 0 | 281 |
| 1.96 | 0.434 | 0 | 217 | 500 | 0 | 283 |
| 1.98 | 0.434 | 0 | 217 | 500 | 0 | 283 |
| 2 | 0.428 | 0 | 214 | 500 | 0 | 286 |