### 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 three strategies to identify sunset images. We train 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. This classification algorithm is able to attain a true positive rate of 0.874, and a false positive rate of 0.098.

We perform the same process using features extracted from the images by a pre-trained convolutional neural network. This classifier is able to obtain a true positive rate of 0.952 and a false positive rate of 0.088. Finally, we use a pretrained convolutional network, but retrain the last few layers of a convolutional neural network to identify sunsets. This yields a true positive rate of **TPR** and a false positive rate of **FPR**.

**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 1 demonstrates the variability of these sunset images.



Figure 1: This contains two sunset images that are vastly different.

**1.1 Support Vector Machine Classification**

One approach to this classification would be to manually tune thresholds based on shape, color, position, size, etc. This would be time intensive and very difficult to attain accurate results. Instead, 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.

**1.2 Classification by Convolutional Neural Network Feature Extraction**

A slightly more computationally expensive alternative to calculating features from the color space of each image is to use a convolutional neural network to extract features. We can then use the same support vector machine technique to classify the images. This method is a bit more effective than the manual feature extraction because the features are more descriptive of the image.

**1.3 Classification by Convolutional Neural Network Transfer Learning**

The final classification option we explore is transfer learning using a pre-trained convolutional neural network. This entails replacing the final three layers of the network with un-trained layers, then retraining the network on our data sets. This is a particularly interesting option because the machine does the optimizations autonomously and we do not need to deal with features.

**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]

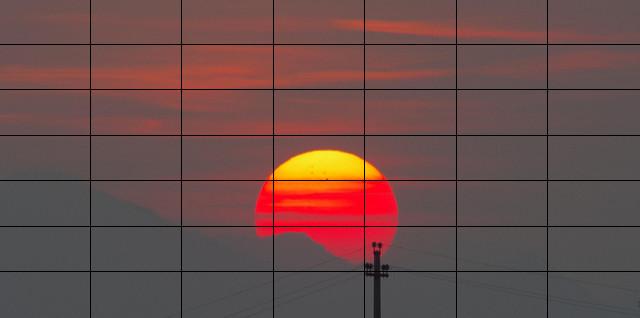


Figure 2: This shows a sample sunset image divided using the 7x7 grid.

**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]

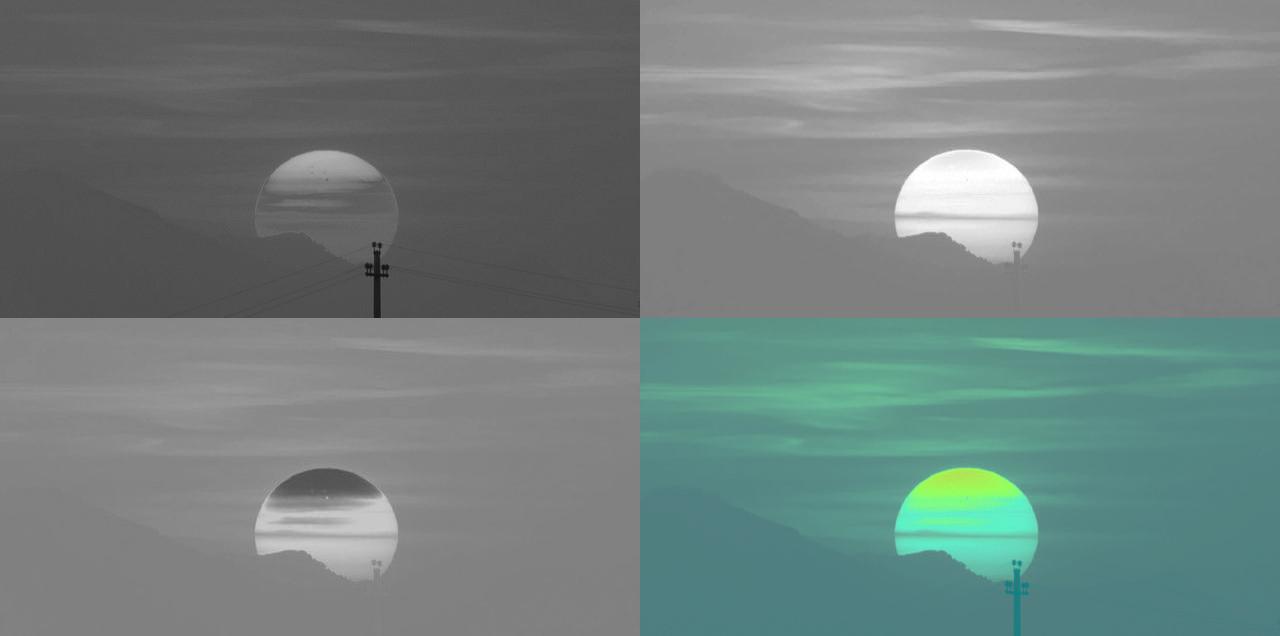


Figure 3: This displays L (top left), S (top right), T (bottom left) and LST displayed instead as an RGB (bottom right). All of these have been normalized to be between 0 and 255.

# 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.

# 3.2 Convolutional Neural Networks

Convolutional neural networks (CNN’s) are another powerful classification tool. Neural networks perform a series of weighted convolutions to determine an output classification. When training a CNN, the network is fed a training set of data, and the output classification is compared with the true classification and the weights of each convolutional layer are adjusted. After running through several iterations using the training set, it checks to see how the network performs on the validation set. The accuracy on the validation set increases as the classification fits the training set better until the classification boundary “overfits” the training set. The validation set is checked at the end of each epoch, and ideally, the training continues to repeat until convergence. After the training is complete, the performance can be verified using a test set of data.

There are multiple ways to use a CNN for classification. The first is way is to train a new network to recognize the desired classes. However, this is not applicable in all situations. Training a new neural network requires a large amount of computing power, as well as a wealth of training data. A second way is to use the classification output of a pre-existing trained neural network. However, this is not always possible – a pre-trained neural network may not contain the necessary classification. Another alternative is to generate features using the neural network. Features can be found by extracting the output of an intermediate layer prior to the output layer. That layer can be used as a feature set, which can be classified using an SVM. This method is called Feature Extraction. A final way to classify using a CNN is to use the majority of a pre-existing neural network, and re-train the last few layers. This is called transfer learning. In transfer learning, the CNN computes the features in the same way, but computes the classes from the features using different layers that are created by the re-training.

**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. These images are given to each of our classification algorithms, which output the classification. We generate an ROC curve using the output of each classifier, so we can compare performance.

**5a. SVM Results**

**5A.1 Training the SVM**

To train our SVM, we used a training set with 800 sunset and 800 non-sunset images. For each image, we computed the 294 features as detailed in section 2. We used the training set features as an input to generate 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.

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.

**5A.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.

**5A.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 4: 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 | 49 |
| Non-Sunset | 63 | 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 performance.

**5B. CNN Results**

**5B.1 Extracting Features using CNN**

Our second classifier uses a fine-tuned support vector machine just as used in section 5A. However, as features to input to the to train the support vector, we used features generated by a pre-trained neural network. The neural network we used was “alexnet”, and we grabbed the output of layer “fc8” to use as the feature set. This feature vector contains 1000 features for each image. We trained an SVM using the set of features generated from the training set of images. We optimized the hyperparameters of the SVM with the validation set of images using the process outlined in section 5A.2, and the table with the results of the parametric sweep can be found in Appendix C. We chose to proceed with a kernel width of 32, and a box constraint of 16 because it yielded a high TPR (0.95) and low FPR (0.0467) with relatively few support vectors (346). Finally, we used the tuned SVM to classify the test set. We varied the threshold and generated an ROC curve, as seen in Figure 5.

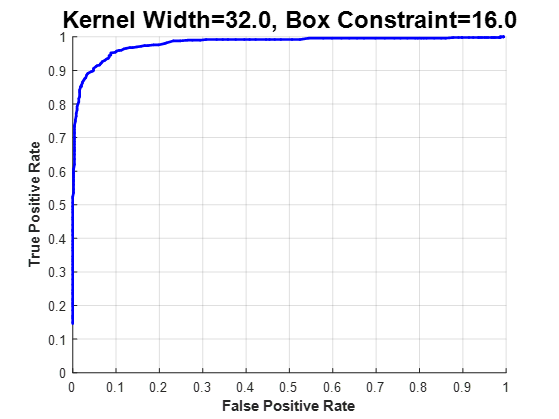


Figure 5: This is the ROC curve showing true positive vs false positive rate for the SVM trained with alexnet-generated features using the Kernel Width and Box Constraint as specified.

Table 3: 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 | 476 | 44 |
| Non-Sunset | 24 | 456 |

We found that a threshold of -0.2 yields a true positive rate of 0.952, and a false positive rate of 0.088. See Appendix D for a full chart of thresholds and performance.

**5B.2 CNN Transfer Learning**

Our third type of classifier was to use a pretrained neural network to classify our images. However, the pretrained neural networks we wanted to use did not have a “sunset” class. Instead of starting from scratch and training a new network, we chose to use all but the last few layers of a pretrained network. We decided to try using two different neural networks to compare their performance: *alexnet* and *googlenet*. For each one, we removed the last three layers, and replaced them with three untrained layers. For each, we trained the last three layers with our training set, and validated using our validation set. We allowed them to each run to convergence. See the results of the training for *alexnet* and *googlenet* in Figure 6 and Figure 7

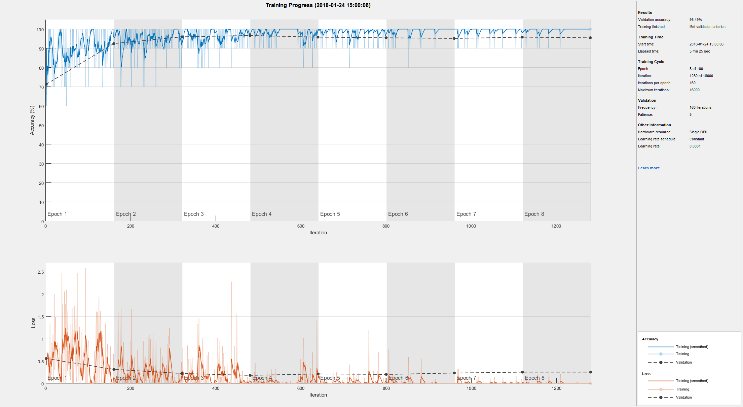


Figure 6: This shows the progress of the transfer learning training using Alexnet. Note that it converged in Epoch 8 after 1280 iterations. This took 5 minutes, 25 seconds using a GTX1060 graphics card.

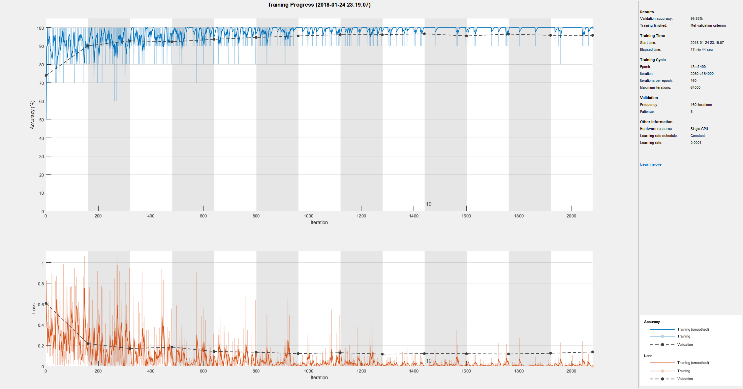


Figure 7: This shows the progress of the transfer learning training using Googlenet. Note that it converged in Epoch 13 after 2080 iterations. This took 17 minutes and 44 seconds using the same GTX1060 graphics card.

Table 4: This shows the results for transfer learning using both alexnet and googlenet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Network** | **Validation Accuracy** | **Total Iterations to Convergence** | **Epoch** | **Elapsed Time** |
| **alexnet** | 95.49% | 1280 | 8 | 5:25 |
| **googlenet** | 95.83% | 2080 | 13 | 17:44 |

After retraining the networks, we ran our test data on them. We wanted to optimize the threshold as we did for the SVM score parameter, but the networks output only classes. Instead of using the output of the networks, we used the penultimate layer output, *softmax*, to act as score. We generated ROC curves and found the best threshold to use. See the ROC curves for *alexnet* and *googlenet* in Figure 8 and Figure 9, respectively. See complete tables of FPR and TPR, and *softmax* threshold in Appendix E (*alexnet*) and Appendix F (*googlenet*).

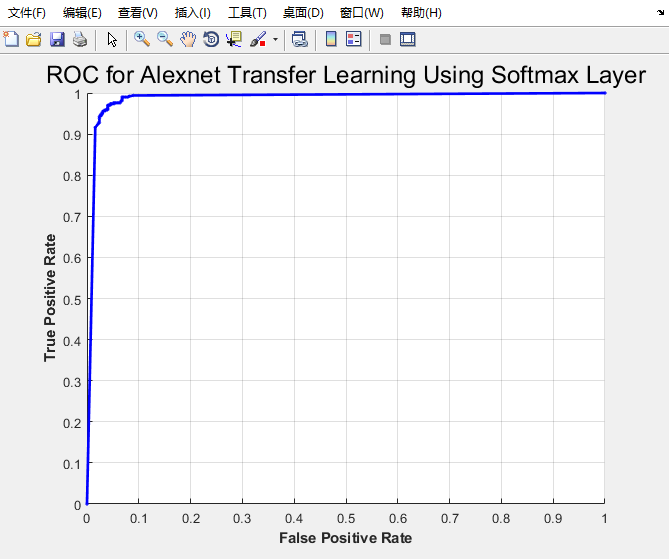


Figure 8: ROC curve for alexnet transfer learning using softmax layer. The best threshold found is anywhere from 0.51 to 0.56, which will yield a TPR of 0.97 and FPR of 0.04.

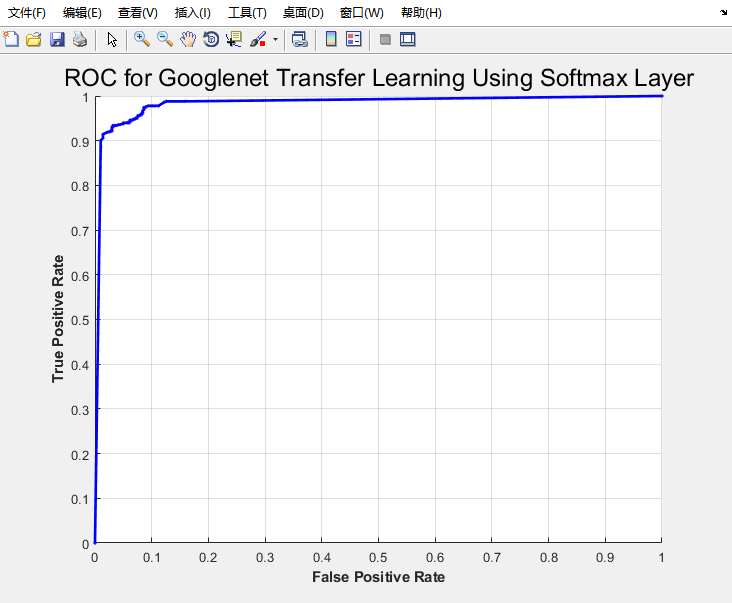


Figure 9: ROC curve for googlenet transfer learning using softmax layer. The best threshold 0.77, which will yield a TPR of 0.934 and FPR of 0.032.

The *alexnet* classifier with a *softmax* threshold of 0.51 yielded a TPR of 0.97, and a FPR of 0.04. The *googlenet* classifier with a *softmax* threshold of 0.77, yielded a TPR of 0.934 and a FPR of 0.032.

Table 5: This contains the detection results from alexnet on the test set using a softmax threshold of 0.51

|  |  |  |
| --- | --- | --- |
| Detected \ True | Sunset | Non-Sunset |
| Sunset | 485 | 20 |
| Non-Sunset | 15 | 480 |

Table 6: This contains the detection results from googlenet on the test set using a softmax threshold of 0.77

|  |  |  |
| --- | --- | --- |
| Detected \ True | Sunset | Non-Sunset |
| Sunset | 467 | 16 |
| Non-Sunset | 33 | 484 |

**6. discussion**

**6.1 Performance of SVM using LST Mean and Variance**

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 Figure 10 and Figure 11, there are two sunset images that our algorithm correctly identified. Figure 10 shows one with a relatively high score. It has much red, orange, and purple. Figure 11 has a sunset with only a very small amount of orange. It is possible that the SVM boundaries weigh the color orange heavily.



Figure 10: This shows the sunset image with the highest score in the test set. The score was 5.48



Figure 11: This shows a sunset image with a low score of 0.1738.

Figure 12 and Figure 13 show sunset images that were misidentified by the classifier. Figure 12 shows an image that was barely misidentified, and Figure 13 shows an image that was identified strongly as a non-sunset image. The image in Figure 12 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 13 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 12: This shows a sunset image that was misidentified by our algorithm with a score of 0.0680.



Figure 13: This shows the sunset image with the lowest score, -2.9085. Our algorithm misidentified it as a non-sunset.

Figure 14 and Figure 15 show non-sunset images that were correctly identified. Figure 14 has a very low score, and Figure 15 was almost identified as a sunset. Figure 14 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 15, on first inspection, is very close in color to the image in Figure 13. However, Figure 15 has a much higher score than the image in Figure 13. This could be explained by the presence of more orange hues in Figure 15 than in Figure 13.



Figure 14: This shows the non-sunset image with the lowest score: -6.2746.

****

Figure 15: This shows a non-sunset image that was correctly identified, but had a high score of 0.0929.

Figure 16 and Figure 17 contain non-sunset images which our algorithm identified as sunsets. Figure 16 was barely identified as a sunset, and Figure 17 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 16: This shows an image that was barely identified as a sunset. It had a score of 0.1815.

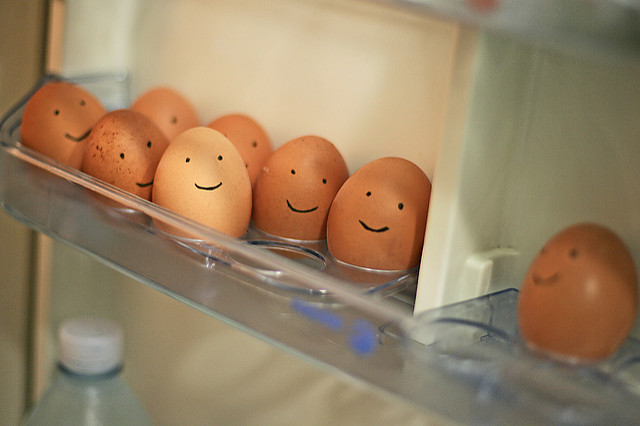


Figure 17: This shows the non-sunset image with the highest score, 1.7366.

**6.2 Performance of SVM using Features Extracted by Alexnet CNN**

Our CNN feature SVM classifier performed even better than the SVM trained with manually generated features, yielding a true positive rate of true positive rate of 0.952, and a false positive rate of 0.088. Even so, we will examine some of the images identified by the classifier.

Figure 18 and Figure 19 contain correctly-identified sunset images. For both of them, a large portion of the image is sky, with the color ranging from deep purple, to yellow, to light blue.



Figure 18: This shows the sunset image with the highest score in the test set. The score was 2.9158.

****

Figure 19: This shows a sunset image with a low score of 0.2481.

Figure 20 and Figure 21 show sunset images that were misidentified by the classifier. Figure 20 is puzzling because it is an image with a large amount of sky and the sun, with very warm colors. Figure 21 is somewhat understandable, as the color of the sky is not distinctly sunset-like, and a large portion of the image contains things that are not sunset-like like cars and buildings.



Figure 20: This shows a sunset image that was misidentified by our algorithm, with a score of 0.1955.



Figure 21: This shows a sunset image that was misidentified by our algorithm with a score of -1.8134

Figure 22 and Figure 23 contain images of non-sunsets that were correctly identified by our algorithm. Figure 22 has a very low score, which is surprising because it contains many pixels that are similarly colored to a sunset. Figure 23 was nearly detected as a sunset, which is very surprising considering that it is a grayscale image. It may be scoring high because it is an image of the sky with clouds. Both of these images show that the algorithm weighs features other than color.



Figure 22: This shows a non-sunset image that was correctly identified with a very low score of -2.5181



Figure 23: This shows a non-sunset image that was correctly identified with a score near the threshold: 0.1959

Figure 24 and Figure 25 show non-sunset images that were incorrectly identified as sunsets. Figure 24 was barely misidentified, but this could be due to the presence of large amounts of orange, red, and purple. Figure 25 almost looks like a sunset, as the sky is a little yellow. However, we can see where the sky is a little brighter near the top right corner of the image, which indicates that the sun is not setting.



Figure 24: This shows a non-sunset image incorrectly identified as a sunset with a score near the threshold of 0.2427.



Figure 25: This shows a non-sunset image incorrectly identified as a sunset with a high score of 1.1873

**6.3 Performance of CNN Transfer Learning**

From Table 4, Table 5 and Table 6 we can see that the two CNN’s performed very differently. *Alexnet* achieved a higher TPR of 0.97, but was not able to achieve quite as low of a FPR (0.032) as *googlenet.* *Alexnet* only consumed 5 minutes, 25 seconds, while *googlenet* consumed 17 minutes, 44 seconds. *Alexnet* consumed 30.5% of the time that *googlenet* consumed. If we were to proceed with any of the classification algorithms, we would choose the trained *alexnet* due to its relatively low time (and therefore computation) resources required and relatively high TPR and low FPR. It is able to achieve both higher TPR and lower FPR than either of the SVM’s we trained.

**7. conclusions**

From the analysis above, we can see that CNNs have greater potential to classify correctly. Optimizing an SVM can take quite a long time, looping through hyperparameters to determine those that achieve maximum performance can take just as long as training a CNN. However, after a SVM has been generated, it is much more efficient to use than a trained CNN. The CNN was helpful in that it eliminated the need to determine what features are useful for distinguishing sunsets, but it uses a large amount of computation to classify the data. Using a CNN for feature extraction is not reasonable for small devices, considering it requires a large amount of memory and computation due to the large number of layers. The best classifier depends on the type of device needed to classify on. If resources are no issue, better performance can be achieved with a CNN. If resources are limited, computing features manually and using an SVM will still achieve reasonably good results.

If we had more time to improve our algorithm, we would try to train our own CNN instead of only using transfer learning on a pre-existing network. That way, it would only extract features necessary for sunset detection. It would be an interesting challenge to make our algorithm distinguish between sunsets and sunrises.

**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: Hyperparameter

**Optimization for LST feature SVM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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

**Optimization for LST feature SVM**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 |

### Appendix C: Hyperparameter

**Optimization for CNN feature SVM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kernel Width | Box Constraint | True Positive Rate | False Positive Rate | Number of Support Vectors |
| 0.03125 | 0.03125 | 1 | 1 | 394 |
| 0.03125 | 0.0625 | 1 | 1 | 72 |
| 0.03125 | 0.125 | 1 | 1 | 72 |
| 0.03125 | 0.25 | 1 | 1 | 62 |
| 0.03125 | 0.5 | 1 | 1 | 46 |
| 0.03125 | 1 | 0 | 0 | 18 |
| 0.03125 | 2 | 0 | 0 | 18 |
| 0.03125 | 4 | 0 | 0 | 18 |
| 0.03125 | 8 | 0 | 0 | 18 |
| 0.03125 | 16 | 0 | 0 | 18 |
| 0.03125 | 32 | 0 | 0 | 18 |
| 0.03125 | 64 | 0 | 0 | 18 |
| 0.03125 | 128 | 0 | 0 | 18 |
| 0.03125 | 256 | 0 | 0 | 18 |
| 0.03125 | 512 | 0 | 0 | 18 |
| 0.03125 | 1024 | 0 | 0 | 18 |
| 0.0625 | 0.03125 | 1 | 1 | 1600 |
| 0.0625 | 0.0625 | 1 | 1 | 1600 |
| 0.0625 | 0.125 | 1 | 1 | 1600 |
| 0.0625 | 0.25 | 1 | 1 | 1600 |
| 0.0625 | 0.5 | 1 | 1 | 976 |
| 0.0625 | 1 | 1 | 1 | 72 |
| 0.0625 | 2 | 1 | 1 | 72 |
| 0.0625 | 4 | 1 | 1 | 72 |
| 0.0625 | 8 | 1 | 1 | 72 |
| 0.0625 | 16 | 1 | 1 | 72 |
| 0.0625 | 32 | 1 | 1 | 72 |
| 0.0625 | 64 | 1 | 1 | 72 |
| 0.0625 | 128 | 1 | 1 | 72 |
| 0.0625 | 256 | 1 | 1 | 72 |
| 0.0625 | 512 | 1 | 1 | 72 |
| 0.0625 | 1024 | 1 | 1 | 72 |
| 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 | 1 | 1 | 1597 |
| 0.125 | 2 | 0 | 0 | 1594 |
| 0.125 | 4 | 0 | 0 | 1594 |
| 0.125 | 8 | 0 | 0 | 1594 |
| 0.125 | 16 | 0 | 0 | 1594 |
| 0.125 | 32 | 0 | 0 | 1594 |
| 0.125 | 64 | 0 | 0 | 1594 |
| 0.125 | 128 | 0 | 0 | 1594 |
| 0.125 | 256 | 0 | 0 | 1594 |
| 0.125 | 512 | 0 | 0 | 1594 |
| 0.125 | 1024 | 0 | 0 | 1594 |
| 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 | 1598 |
| 0.25 | 2 | 0 | 0 | 1594 |
| 0.25 | 4 | 0 | 0 | 1594 |
| 0.25 | 8 | 0 | 0 | 1594 |
| 0.25 | 16 | 0 | 0 | 1594 |
| 0.25 | 32 | 0 | 0 | 1594 |
| 0.25 | 64 | 0 | 0 | 1594 |
| 0.25 | 128 | 0 | 0 | 1594 |
| 0.25 | 256 | 0 | 0 | 1594 |
| 0.25 | 512 | 0 | 0 | 1594 |
| 0.25 | 1024 | 0 | 0 | 1594 |
| 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 | 1598 |
| 0.5 | 2 | 0 | 0 | 1594 |
| 0.5 | 4 | 0 | 0 | 1594 |
| 0.5 | 8 | 0 | 0 | 1594 |
| 0.5 | 16 | 0 | 0 | 1594 |
| 0.5 | 32 | 0 | 0 | 1594 |
| 0.5 | 64 | 0 | 0 | 1594 |
| 0.5 | 128 | 0 | 0 | 1594 |
| 0.5 | 256 | 0 | 0 | 1594 |
| 0.5 | 512 | 0 | 0 | 1594 |
| 0.5 | 1024 | 0 | 0 | 1594 |
| 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 | 1598 |
| 1 | 2 | 0 | 0 | 1594 |
| 1 | 4 | 0 | 0 | 1594 |
| 1 | 8 | 0 | 0 | 1594 |
| 1 | 16 | 0 | 0 | 1594 |
| 1 | 32 | 0 | 0 | 1594 |
| 1 | 64 | 0 | 0 | 1594 |
| 1 | 128 | 0 | 0 | 1594 |
| 1 | 256 | 0 | 0 | 1594 |
| 1 | 512 | 0 | 0 | 1594 |
| 1 | 1024 | 0 | 0 | 1594 |
| 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 | 1598 |
| 2 | 2 | 0 | 0 | 1594 |
| 2 | 4 | 0 | 0 | 1594 |
| 2 | 8 | 0 | 0 | 1594 |
| 2 | 16 | 0 | 0 | 1594 |
| 2 | 32 | 0 | 0 | 1594 |
| 2 | 64 | 0 | 0 | 1594 |
| 2 | 128 | 0 | 0 | 1594 |
| 2 | 256 | 0 | 0 | 1594 |
| 2 | 512 | 0 | 0 | 1594 |
| 2 | 1024 | 0 | 0 | 1594 |
| 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 | 1600 |
| 4 | 1 | 0 | 0 | 1597 |
| 4 | 2 | 0 | 0 | 1594 |
| 4 | 4 | 0 | 0 | 1594 |
| 4 | 8 | 0 | 0 | 1594 |
| 4 | 16 | 0 | 0 | 1594 |
| 4 | 32 | 0 | 0 | 1594 |
| 4 | 64 | 0 | 0 | 1594 |
| 4 | 128 | 0 | 0 | 1594 |
| 4 | 256 | 0 | 0 | 1594 |
| 4 | 512 | 0 | 0 | 1594 |
| 4 | 1024 | 0 | 0 | 1594 |
| 8 | 0.03125 | 0 | 0 | 1600 |
| 8 | 0.0625 | 0 | 0 | 1600 |
| 8 | 0.125 | 0 | 0 | 1600 |
| 8 | 0.25 | 0 | 0 | 1600 |
| 8 | 0.5 | 0 | 0 | 1600 |
| 8 | 1 | 0.223333 | 0 | 1594 |
| 8 | 2 | 0.283333 | 0.003333 | 1592 |
| 8 | 4 | 0.283333 | 0.003333 | 1592 |
| 8 | 8 | 0.283333 | 0.003333 | 1592 |
| 8 | 16 | 0.283333 | 0.003333 | 1592 |
| 8 | 32 | 0.283333 | 0.003333 | 1592 |
| 8 | 64 | 0.283333 | 0.003333 | 1592 |
| 8 | 128 | 0.283333 | 0.003333 | 1592 |
| 8 | 256 | 0.283333 | 0.003333 | 1592 |
| 8 | 512 | 0.283333 | 0.003333 | 1592 |
| 8 | 1024 | 0.283333 | 0.003333 | 1592 |
| 16 | 0.03125 | 0.256667 | 0.003333 | 1596 |
| 16 | 0.0625 | 0.443333 | 0.013333 | 1521 |
| 16 | 0.125 | 0.573333 | 0.02 | 1419 |
| 16 | 0.25 | 0.7 | 0.02 | 1325 |
| 16 | 0.5 | 0.776667 | 0.016667 | 1257 |
| 16 | 1 | 0.83 | 0.016667 | 1232 |
| 16 | 2 | 0.836667 | 0.016667 | 1254 |
| 16 | 4 | 0.84 | 0.016667 | 1262 |
| 16 | 8 | 0.84 | 0.016667 | 1265 |
| 16 | 16 | 0.84 | 0.016667 | 1265 |
| 16 | 32 | 0.84 | 0.016667 | 1265 |
| 16 | 64 | 0.84 | 0.016667 | 1265 |
| 16 | 128 | 0.84 | 0.016667 | 1265 |
| 16 | 256 | 0.84 | 0.016667 | 1265 |
| 16 | 512 | 0.84 | 0.016667 | 1265 |
| 16 | 1024 | 0.84 | 0.016667 | 1265 |
| 32 | 0.03125 | 0.883333 | 0.186667 | 1076 |
| 32 | 0.0625 | 0.903333 | 0.143333 | 894 |
| 32 | 0.125 | 0.913333 | 0.07 | 742 |
| 32 | 0.25 | 0.93 | 0.05 | 627 |
| 32 | 0.5 | 0.936667 | 0.043333 | 552 |
| 32 | 1 | 0.943333 | 0.04 | 511 |
| 32 | 2 | 0.946667 | 0.043333 | 340 |
| 32 | 4 | 0.953333 | 0.05 | 537 |
| 32 | 8 | 0.946667 | 0.043333 | 330 |
| 32 | 16 | 0.95 | 0.046667 | 346 |
| 32 | 32 | 0.94 | 0.046667 | 572 |
| 32 | 64 | 0.94 | 0.046667 | 572 |
| 32 | 128 | 0.94 | 0.046667 | 572 |
| 32 | 256 | 0.94 | 0.046667 | 572 |
| 32 | 512 | 0.94 | 0.046667 | 572 |
| 32 | 1024 | 0.94 | 0.046667 | 572 |
| 64 | 0.03125 | 0.936667 | 0.276667 | 1203 |
| 64 | 0.0625 | 0.936667 | 0.233333 | 976 |
| 64 | 0.125 | 0.936667 | 0.176667 | 802 |
| 64 | 0.25 | 0.943333 | 0.093333 | 666 |
| 64 | 0.5 | 0.946667 | 0.07 | 555 |
| 64 | 1 | 0.96 | 0.066667 | 472 |
| 64 | 2 | 0.953333 | 0.056667 | 406 |
| 64 | 4 | 0.96 | 0.053333 | 383 |
| 64 | 8 | 0.923333 | 0.06 | 171 |
| 64 | 16 | 0.933333 | 0.073333 | 233 |
| 64 | 32 | 0.943333 | 0.076667 | 178 |
| 64 | 64 | 0.943333 | 0.05 | 330 |
| 64 | 128 | 0.943333 | 0.063333 | 348 |
| 64 | 256 | 0.95 | 0.046667 | 375 |
| 64 | 512 | 0.95 | 0.046667 | 375 |
| 64 | 1024 | 0.95 | 0.046667 | 375 |
| 128 | 0.03125 | 0.946667 | 0.336667 | 1582 |
| 128 | 0.0625 | 0.953333 | 0.316667 | 1365 |
| 128 | 0.125 | 0.94 | 0.276667 | 1122 |
| 128 | 0.25 | 0.953333 | 0.233333 | 920 |
| 128 | 0.5 | 0.956667 | 0.16 | 762 |
| 128 | 1 | 0.96 | 0.1 | 633 |
| 128 | 2 | 0.966667 | 0.08 | 532 |
| 128 | 4 | 0.966667 | 0.053333 | 455 |
| 128 | 8 | 0.936667 | 0.39 | 60 |
| 128 | 16 | 0.856667 | 0.273333 | 67 |
| 128 | 32 | 0.936667 | 0.166667 | 114 |
| 128 | 64 | 0.953333 | 0.086667 | 109 |
| 128 | 128 | 0.853333 | 0.386667 | 154 |
| 128 | 256 | 0.883333 | 0.12 | 246 |
| 128 | 512 | 0.93 | 0.053333 | 353 |
| 128 | 1024 | 0.93 | 0.05 | 350 |
| 256 | 0.03125 | 0.95 | 0.35 | 1600 |
| 256 | 0.0625 | 0.95 | 0.35 | 1600 |
| 256 | 0.125 | 0.95 | 0.353333 | 1570 |
| 256 | 0.25 | 0.956667 | 0.32 | 1336 |
| 256 | 0.5 | 0.946667 | 0.276667 | 1106 |
| 256 | 1 | 0.946667 | 0.22 | 909 |
| 256 | 2 | 0.96 | 0.16 | 754 |
| 256 | 4 | 0.963333 | 0.113333 | 634 |
| 256 | 8 | 0.99 | 0.543333 | 27 |
| 256 | 16 | 1 | 0.533333 | 30 |
| 256 | 32 | 0.943333 | 0.57 | 36 |
| 256 | 64 | 0.913333 | 0.306667 | 30 |
| 256 | 128 | 0.65 | 0.536667 | 42 |
| 256 | 256 | 0.866667 | 0.08 | 81 |
| 256 | 512 | 0.906667 | 0.163333 | 157 |
| 256 | 1024 | 0.843333 | 0.316667 | 238 |
| 512 | 0.03125 | 0.953333 | 0.353333 | 1600 |
| 512 | 0.0625 | 0.953333 | 0.353333 | 1600 |
| 512 | 0.125 | 0.953333 | 0.353333 | 1600 |
| 512 | 0.25 | 0.953333 | 0.353333 | 1600 |
| 512 | 0.5 | 0.953333 | 0.353333 | 1566 |
| 512 | 1 | 0.96 | 0.32 | 1331 |
| 512 | 2 | 0.946667 | 0.273333 | 1102 |
| 512 | 4 | 0.95 | 0.22 | 906 |
| 512 | 8 | 1 | 0.996667 | 23 |
| 512 | 16 | 1 | 0.683333 | 34 |
| 512 | 32 | 1 | 0.57 | 34 |
| 512 | 64 | 0.946667 | 0.4 | 27 |
| 512 | 128 | 0.956667 | 0.293333 | 45 |
| 512 | 256 | 1 | 1 | 3 |
| 512 | 512 | 0.663333 | 0.253333 | 27 |
| 512 | 1024 | 0.81 | 0.126667 | 108 |
| 1024 | 0.03125 | 0.953333 | 0.353333 | 1600 |
| 1024 | 0.0625 | 0.953333 | 0.353333 | 1600 |
| 1024 | 0.125 | 0.953333 | 0.353333 | 1600 |
| 1024 | 0.25 | 0.953333 | 0.353333 | 1600 |
| 1024 | 0.5 | 0.953333 | 0.353333 | 1600 |
| 1024 | 1 | 0.953333 | 0.353333 | 1600 |
| 1024 | 2 | 0.953333 | 0.35 | 1564 |
| 1024 | 4 | 0.963333 | 0.32 | 1330 |
| 1024 | 8 | 1 | 1 | 23 |
| 1024 | 16 | 1 | 1 | 23 |
| 1024 | 32 | 1 | 1 | 23 |
| 1024 | 64 | 1 | 0.666667 | 36 |
| 1024 | 128 | 0.993333 | 0.523333 | 31 |
| 1024 | 256 | 0.946667 | 0.406667 | 27 |
| 1024 | 512 | 0.916667 | 0.416667 | 39 |
| 1024 | 1024 | 1 | 1 | 3 |

### Appendix D: threshold

**Optimization for CNN feature SVM**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Threshold | True Positive Rate | False Positive Rate | True Positives | True Negatives | False Positives | False Negatives |
| -2 | 1 | 0.994 | 500 | 3 | 497 | 0 |
| -1.98 | 1 | 0.994 | 500 | 3 | 497 | 0 |
| -1.96 | 1 | 0.994 | 500 | 3 | 497 | 0 |
| -1.94 | 1 | 0.994 | 500 | 3 | 497 | 0 |
| -1.92 | 1 | 0.994 | 500 | 3 | 497 | 0 |
| -1.9 | 1 | 0.994 | 500 | 3 | 497 | 0 |
| -1.88 | 1 | 0.99 | 500 | 5 | 495 | 0 |
| -1.86 | 1 | 0.988 | 500 | 6 | 494 | 0 |
| -1.84 | 1 | 0.986 | 500 | 7 | 493 | 0 |
| -1.82 | 1 | 0.986 | 500 | 7 | 493 | 0 |
| -1.8 | 0.998 | 0.986 | 499 | 7 | 493 | 1 |
| -1.78 | 0.998 | 0.986 | 499 | 7 | 493 | 1 |
| -1.76 | 0.998 | 0.986 | 499 | 7 | 493 | 1 |
| -1.74 | 0.998 | 0.986 | 499 | 7 | 493 | 1 |
| -1.72 | 0.998 | 0.982 | 499 | 9 | 491 | 1 |
| -1.7 | 0.998 | 0.976 | 499 | 12 | 488 | 1 |
| -1.68 | 0.998 | 0.972 | 499 | 14 | 486 | 1 |
| -1.66 | 0.998 | 0.972 | 499 | 14 | 486 | 1 |
| -1.64 | 0.998 | 0.968 | 499 | 16 | 484 | 1 |
| -1.62 | 0.998 | 0.966 | 499 | 17 | 483 | 1 |
| -1.6 | 0.998 | 0.958 | 499 | 21 | 479 | 1 |
| -1.58 | 0.998 | 0.954 | 499 | 23 | 477 | 1 |
| -1.56 | 0.998 | 0.95 | 499 | 25 | 475 | 1 |
| -1.54 | 0.998 | 0.944 | 499 | 28 | 472 | 1 |
| -1.52 | 0.998 | 0.94 | 499 | 30 | 470 | 1 |
| -1.5 | 0.998 | 0.936 | 499 | 32 | 468 | 1 |
| -1.48 | 0.998 | 0.936 | 499 | 32 | 468 | 1 |
| -1.46 | 0.998 | 0.924 | 499 | 38 | 462 | 1 |
| -1.44 | 0.998 | 0.914 | 499 | 43 | 457 | 1 |
| -1.42 | 0.998 | 0.898 | 499 | 51 | 449 | 1 |
| -1.4 | 0.998 | 0.898 | 499 | 51 | 449 | 1 |
| -1.38 | 0.998 | 0.89 | 499 | 55 | 445 | 1 |
| -1.36 | 0.998 | 0.878 | 499 | 61 | 439 | 1 |
| -1.34 | 0.996 | 0.86 | 498 | 70 | 430 | 2 |
| -1.32 | 0.996 | 0.85 | 498 | 75 | 425 | 2 |
| -1.3 | 0.996 | 0.834 | 498 | 83 | 417 | 2 |
| -1.28 | 0.996 | 0.816 | 498 | 92 | 408 | 2 |
| -1.26 | 0.996 | 0.81 | 498 | 95 | 405 | 2 |
| -1.24 | 0.996 | 0.804 | 498 | 98 | 402 | 2 |
| -1.22 | 0.996 | 0.796 | 498 | 102 | 398 | 2 |
| -1.2 | 0.996 | 0.79 | 498 | 105 | 395 | 2 |
| -1.18 | 0.996 | 0.772 | 498 | 114 | 386 | 2 |
| -1.16 | 0.996 | 0.756 | 498 | 122 | 378 | 2 |
| -1.14 | 0.996 | 0.742 | 498 | 129 | 371 | 2 |
| -1.12 | 0.996 | 0.72 | 498 | 140 | 360 | 2 |
| -1.1 | 0.996 | 0.698 | 498 | 151 | 349 | 2 |
| -1.08 | 0.996 | 0.672 | 498 | 164 | 336 | 2 |
| -1.06 | 0.996 | 0.656 | 498 | 172 | 328 | 2 |
| -1.04 | 0.996 | 0.638 | 498 | 181 | 319 | 2 |
| -1.02 | 0.996 | 0.62 | 498 | 190 | 310 | 2 |
| -1 | 0.996 | 0.598 | 498 | 201 | 299 | 2 |
| -0.98 | 0.996 | 0.578 | 498 | 211 | 289 | 2 |
| -0.96 | 0.996 | 0.544 | 498 | 228 | 272 | 2 |
| -0.94 | 0.992 | 0.524 | 496 | 238 | 262 | 4 |
| -0.92 | 0.992 | 0.506 | 496 | 247 | 253 | 4 |
| -0.9 | 0.992 | 0.48 | 496 | 260 | 240 | 4 |
| -0.88 | 0.992 | 0.466 | 496 | 267 | 233 | 4 |
| -0.86 | 0.992 | 0.454 | 496 | 273 | 227 | 4 |
| -0.84 | 0.992 | 0.446 | 496 | 277 | 223 | 4 |
| -0.82 | 0.992 | 0.422 | 496 | 289 | 211 | 4 |
| -0.8 | 0.992 | 0.408 | 496 | 296 | 204 | 4 |
| -0.78 | 0.992 | 0.396 | 496 | 302 | 198 | 4 |
| -0.76 | 0.992 | 0.376 | 496 | 312 | 188 | 4 |
| -0.74 | 0.992 | 0.358 | 496 | 321 | 179 | 4 |
| -0.72 | 0.992 | 0.346 | 496 | 327 | 173 | 4 |
| -0.7 | 0.992 | 0.332 | 496 | 334 | 166 | 4 |
| -0.68 | 0.992 | 0.308 | 496 | 346 | 154 | 4 |
| -0.66 | 0.99 | 0.294 | 495 | 353 | 147 | 5 |
| -0.64 | 0.99 | 0.276 | 495 | 362 | 138 | 5 |
| -0.62 | 0.99 | 0.264 | 495 | 368 | 132 | 5 |
| -0.6 | 0.988 | 0.248 | 494 | 376 | 124 | 6 |
| -0.58 | 0.988 | 0.232 | 494 | 384 | 116 | 6 |
| -0.56 | 0.982 | 0.216 | 491 | 392 | 108 | 9 |
| -0.54 | 0.98 | 0.212 | 490 | 394 | 106 | 10 |
| -0.52 | 0.976 | 0.198 | 488 | 401 | 99 | 12 |
| -0.5 | 0.976 | 0.186 | 488 | 407 | 93 | 12 |
| -0.48 | 0.974 | 0.174 | 487 | 413 | 87 | 13 |
| -0.46 | 0.974 | 0.166 | 487 | 417 | 83 | 13 |
| -0.44 | 0.972 | 0.158 | 486 | 421 | 79 | 14 |
| -0.42 | 0.97 | 0.148 | 485 | 426 | 74 | 15 |
| -0.4 | 0.968 | 0.138 | 484 | 431 | 69 | 16 |
| -0.38 | 0.968 | 0.132 | 484 | 434 | 66 | 16 |
| -0.36 | 0.966 | 0.124 | 483 | 438 | 62 | 17 |
| -0.34 | 0.964 | 0.118 | 482 | 441 | 59 | 18 |
| -0.32 | 0.96 | 0.112 | 480 | 444 | 56 | 20 |
| -0.3 | 0.96 | 0.108 | 480 | 446 | 54 | 20 |
| -0.28 | 0.96 | 0.108 | 480 | 446 | 54 | 20 |
| -0.26 | 0.958 | 0.102 | 479 | 449 | 51 | 21 |
| -0.24 | 0.956 | 0.1 | 478 | 450 | 50 | 22 |
| -0.22 | 0.954 | 0.098 | 477 | 451 | 49 | 23 |
| -0.2 | 0.952 | 0.088 | 476 | 456 | 44 | 24 |
| -0.18 | 0.946 | 0.088 | 473 | 456 | 44 | 27 |
| -0.16 | 0.944 | 0.084 | 472 | 458 | 42 | 28 |
| -0.14 | 0.94 | 0.084 | 470 | 458 | 42 | 30 |
| -0.12 | 0.936 | 0.082 | 468 | 459 | 41 | 32 |
| -0.1 | 0.934 | 0.078 | 467 | 461 | 39 | 33 |
| -0.08 | 0.93 | 0.074 | 465 | 463 | 37 | 35 |
| -0.06 | 0.926 | 0.068 | 463 | 466 | 34 | 37 |
| -0.04 | 0.922 | 0.066 | 461 | 467 | 33 | 39 |
| -0.02 | 0.92 | 0.064 | 460 | 468 | 32 | 40 |
| 0 | 0.916 | 0.062 | 458 | 469 | 31 | 42 |
| 0.02 | 0.914 | 0.056 | 457 | 472 | 28 | 43 |
| 0.04 | 0.912 | 0.054 | 456 | 473 | 27 | 44 |
| 0.06 | 0.906 | 0.048 | 453 | 476 | 24 | 47 |
| 0.08 | 0.9 | 0.048 | 450 | 476 | 24 | 50 |
| 0.1 | 0.898 | 0.046 | 449 | 477 | 23 | 51 |
| 0.12 | 0.896 | 0.042 | 448 | 479 | 21 | 52 |
| 0.14 | 0.892 | 0.036 | 446 | 482 | 18 | 54 |
| 0.16 | 0.89 | 0.034 | 445 | 483 | 17 | 55 |
| 0.18 | 0.886 | 0.032 | 443 | 484 | 16 | 57 |
| 0.2 | 0.878 | 0.03 | 439 | 485 | 15 | 61 |
| 0.22 | 0.876 | 0.028 | 438 | 486 | 14 | 62 |
| 0.24 | 0.872 | 0.026 | 436 | 487 | 13 | 64 |
| 0.26 | 0.868 | 0.024 | 434 | 488 | 12 | 66 |
| 0.28 | 0.864 | 0.022 | 432 | 489 | 11 | 68 |
| 0.3 | 0.862 | 0.022 | 431 | 489 | 11 | 69 |
| 0.32 | 0.856 | 0.02 | 428 | 490 | 10 | 72 |
| 0.34 | 0.852 | 0.02 | 426 | 490 | 10 | 74 |
| 0.36 | 0.852 | 0.018 | 426 | 491 | 9 | 74 |
| 0.38 | 0.85 | 0.018 | 425 | 491 | 9 | 75 |
| 0.4 | 0.842 | 0.016 | 421 | 492 | 8 | 79 |
| 0.42 | 0.836 | 0.016 | 418 | 492 | 8 | 82 |
| 0.44 | 0.822 | 0.016 | 411 | 492 | 8 | 89 |
| 0.46 | 0.816 | 0.014 | 408 | 493 | 7 | 92 |
| 0.48 | 0.806 | 0.014 | 403 | 493 | 7 | 97 |
| 0.5 | 0.8 | 0.012 | 400 | 494 | 6 | 100 |
| 0.52 | 0.792 | 0.01 | 396 | 495 | 5 | 104 |
| 0.54 | 0.786 | 0.01 | 393 | 495 | 5 | 107 |
| 0.56 | 0.782 | 0.01 | 391 | 495 | 5 | 109 |
| 0.58 | 0.772 | 0.008 | 386 | 496 | 4 | 114 |
| 0.6 | 0.764 | 0.008 | 382 | 496 | 4 | 118 |
| 0.62 | 0.752 | 0.006 | 376 | 497 | 3 | 124 |
| 0.64 | 0.748 | 0.006 | 374 | 497 | 3 | 126 |
| 0.66 | 0.736 | 0.004 | 368 | 498 | 2 | 132 |
| 0.68 | 0.734 | 0.004 | 367 | 498 | 2 | 133 |
| 0.7 | 0.732 | 0.004 | 366 | 498 | 2 | 134 |
| 0.72 | 0.724 | 0.004 | 362 | 498 | 2 | 138 |
| 0.74 | 0.716 | 0.004 | 358 | 498 | 2 | 142 |
| 0.76 | 0.712 | 0.004 | 356 | 498 | 2 | 144 |
| 0.78 | 0.702 | 0.004 | 351 | 498 | 2 | 149 |
| 0.8 | 0.696 | 0.004 | 348 | 498 | 2 | 152 |
| 0.82 | 0.688 | 0.004 | 344 | 498 | 2 | 156 |
| 0.84 | 0.68 | 0.004 | 340 | 498 | 2 | 160 |
| 0.86 | 0.67 | 0.004 | 335 | 498 | 2 | 165 |
| 0.88 | 0.664 | 0.004 | 332 | 498 | 2 | 168 |
| 0.9 | 0.66 | 0.004 | 330 | 498 | 2 | 170 |
| 0.92 | 0.65 | 0.004 | 325 | 498 | 2 | 175 |
| 0.94 | 0.636 | 0.004 | 318 | 498 | 2 | 182 |
| 0.96 | 0.62 | 0.004 | 310 | 498 | 2 | 190 |
| 0.98 | 0.606 | 0.002 | 303 | 499 | 1 | 197 |
| 1 | 0.606 | 0.002 | 303 | 499 | 1 | 197 |
| 1.02 | 0.596 | 0.002 | 298 | 499 | 1 | 202 |
| 1.04 | 0.594 | 0.002 | 297 | 499 | 1 | 203 |
| 1.06 | 0.588 | 0.002 | 294 | 499 | 1 | 206 |
| 1.08 | 0.582 | 0.002 | 291 | 499 | 1 | 209 |
| 1.1 | 0.574 | 0.002 | 287 | 499 | 1 | 213 |
| 1.12 | 0.57 | 0.002 | 285 | 499 | 1 | 215 |
| 1.14 | 0.56 | 0.002 | 280 | 499 | 1 | 220 |
| 1.16 | 0.546 | 0.002 | 273 | 499 | 1 | 227 |
| 1.18 | 0.536 | 0.002 | 268 | 499 | 1 | 232 |
| 1.2 | 0.524 | 0 | 262 | 500 | 0 | 238 |
| 1.22 | 0.52 | 0 | 260 | 500 | 0 | 240 |
| 1.24 | 0.506 | 0 | 253 | 500 | 0 | 247 |
| 1.26 | 0.496 | 0 | 248 | 500 | 0 | 252 |
| 1.28 | 0.494 | 0 | 247 | 500 | 0 | 253 |
| 1.3 | 0.486 | 0 | 243 | 500 | 0 | 257 |
| 1.32 | 0.478 | 0 | 239 | 500 | 0 | 261 |
| 1.34 | 0.462 | 0 | 231 | 500 | 0 | 269 |
| 1.36 | 0.456 | 0 | 228 | 500 | 0 | 272 |
| 1.38 | 0.44 | 0 | 220 | 500 | 0 | 280 |
| 1.4 | 0.426 | 0 | 213 | 500 | 0 | 287 |
| 1.42 | 0.424 | 0 | 212 | 500 | 0 | 288 |
| 1.44 | 0.42 | 0 | 210 | 500 | 0 | 290 |
| 1.46 | 0.41 | 0 | 205 | 500 | 0 | 295 |
| 1.48 | 0.392 | 0 | 196 | 500 | 0 | 304 |
| 1.5 | 0.382 | 0 | 191 | 500 | 0 | 309 |
| 1.52 | 0.372 | 0 | 186 | 500 | 0 | 314 |
| 1.54 | 0.364 | 0 | 182 | 500 | 0 | 318 |
| 1.56 | 0.346 | 0 | 173 | 500 | 0 | 327 |
| 1.58 | 0.338 | 0 | 169 | 500 | 0 | 331 |
| 1.6 | 0.332 | 0 | 166 | 500 | 0 | 334 |
| 1.62 | 0.322 | 0 | 161 | 500 | 0 | 339 |
| 1.64 | 0.312 | 0 | 156 | 500 | 0 | 344 |
| 1.66 | 0.306 | 0 | 153 | 500 | 0 | 347 |
| 1.68 | 0.292 | 0 | 146 | 500 | 0 | 354 |
| 1.7 | 0.278 | 0 | 139 | 500 | 0 | 361 |
| 1.72 | 0.262 | 0 | 131 | 500 | 0 | 369 |
| 1.74 | 0.246 | 0 | 123 | 500 | 0 | 377 |
| 1.76 | 0.238 | 0 | 119 | 500 | 0 | 381 |
| 1.78 | 0.234 | 0 | 117 | 500 | 0 | 383 |
| 1.8 | 0.22 | 0 | 110 | 500 | 0 | 390 |
| 1.82 | 0.216 | 0 | 108 | 500 | 0 | 392 |
| 1.84 | 0.208 | 0 | 104 | 500 | 0 | 396 |
| 1.86 | 0.204 | 0 | 102 | 500 | 0 | 398 |
| 1.88 | 0.198 | 0 | 99 | 500 | 0 | 401 |
| 1.9 | 0.186 | 0 | 93 | 500 | 0 | 407 |
| 1.92 | 0.17 | 0 | 85 | 500 | 0 | 415 |
| 1.94 | 0.166 | 0 | 83 | 500 | 0 | 417 |
| 1.96 | 0.162 | 0 | 81 | 500 | 0 | 419 |
| 1.98 | 0.15 | 0 | 75 | 500 | 0 | 425 |
| 2 | 0.146 | 0 | 73 | 500 | 0 | 427 |

### APPendix E: Threshold

**Optimization for Alexnet**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Threshold | True Positive Rate | False Positive Rate | True Positives | True Negatives | False Positives | False Negatives |
| 0 | 1 | 1 | 500 | 0 | 500 | 0 |
| 0.01 | 0.994 | 0.09 | 497 | 455 | 45 | 3 |
| 0.02 | 0.992 | 0.082 | 496 | 459 | 41 | 4 |
| 0.03 | 0.99 | 0.078 | 495 | 461 | 39 | 5 |
| 0.04 | 0.99 | 0.074 | 495 | 463 | 37 | 5 |
| 0.05 | 0.99 | 0.07 | 495 | 465 | 35 | 5 |
| 0.06 | 0.99 | 0.068 | 495 | 466 | 34 | 5 |
| 0.07 | 0.988 | 0.068 | 494 | 466 | 34 | 6 |
| 0.08 | 0.988 | 0.068 | 494 | 466 | 34 | 6 |
| 0.09 | 0.986 | 0.068 | 493 | 466 | 34 | 7 |
| 0.1 | 0.982 | 0.068 | 491 | 466 | 34 | 9 |
| 0.11 | 0.982 | 0.068 | 491 | 466 | 34 | 9 |
| 0.12 | 0.98 | 0.066 | 490 | 467 | 33 | 10 |
| 0.13 | 0.98 | 0.066 | 490 | 467 | 33 | 10 |
| 0.14 | 0.98 | 0.066 | 490 | 467 | 33 | 10 |
| 0.15 | 0.976 | 0.064 | 488 | 468 | 32 | 12 |
| 0.16 | 0.976 | 0.062 | 488 | 469 | 31 | 12 |
| 0.17 | 0.976 | 0.062 | 488 | 469 | 31 | 12 |
| 0.18 | 0.976 | 0.058 | 488 | 471 | 29 | 12 |
| 0.19 | 0.976 | 0.056 | 488 | 472 | 28 | 12 |
| 0.2 | 0.976 | 0.056 | 488 | 472 | 28 | 12 |
| 0.21 | 0.976 | 0.054 | 488 | 473 | 27 | 12 |
| 0.22 | 0.976 | 0.054 | 488 | 473 | 27 | 12 |
| 0.23 | 0.976 | 0.054 | 488 | 473 | 27 | 12 |
| 0.24 | 0.976 | 0.054 | 488 | 473 | 27 | 12 |
| 0.25 | 0.976 | 0.052 | 488 | 474 | 26 | 12 |
| 0.26 | 0.976 | 0.052 | 488 | 474 | 26 | 12 |
| 0.27 | 0.974 | 0.052 | 487 | 474 | 26 | 13 |
| 0.28 | 0.974 | 0.052 | 487 | 474 | 26 | 13 |
| 0.29 | 0.974 | 0.052 | 487 | 474 | 26 | 13 |
| 0.3 | 0.974 | 0.052 | 487 | 474 | 26 | 13 |
| 0.31 | 0.974 | 0.052 | 487 | 474 | 26 | 13 |
| 0.32 | 0.974 | 0.052 | 487 | 474 | 26 | 13 |
| 0.33 | 0.974 | 0.05 | 487 | 475 | 25 | 13 |
| 0.34 | 0.974 | 0.048 | 487 | 476 | 24 | 13 |
| 0.35 | 0.974 | 0.048 | 487 | 476 | 24 | 13 |
| 0.36 | 0.974 | 0.046 | 487 | 477 | 23 | 13 |
| 0.37 | 0.974 | 0.046 | 487 | 477 | 23 | 13 |
| 0.38 | 0.974 | 0.046 | 487 | 477 | 23 | 13 |
| 0.39 | 0.972 | 0.046 | 486 | 477 | 23 | 14 |
| 0.4 | 0.972 | 0.046 | 486 | 477 | 23 | 14 |
| 0.41 | 0.972 | 0.046 | 486 | 477 | 23 | 14 |
| 0.42 | 0.972 | 0.044 | 486 | 478 | 22 | 14 |
| 0.43 | 0.97 | 0.044 | 485 | 478 | 22 | 15 |
| 0.44 | 0.97 | 0.044 | 485 | 478 | 22 | 15 |
| 0.45 | 0.97 | 0.044 | 485 | 478 | 22 | 15 |
| 0.46 | 0.97 | 0.044 | 485 | 478 | 22 | 15 |
| 0.47 | 0.97 | 0.042 | 485 | 479 | 21 | 15 |
| 0.48 | 0.97 | 0.042 | 485 | 479 | 21 | 15 |
| 0.49 | 0.97 | 0.042 | 485 | 479 | 21 | 15 |
| 0.5 | 0.97 | 0.042 | 485 | 479 | 21 | 15 |
| 0.51 | 0.97 | 0.04 | 485 | 480 | 20 | 15 |
| 0.52 | 0.97 | 0.04 | 485 | 480 | 20 | 15 |
| 0.53 | 0.97 | 0.04 | 485 | 480 | 20 | 15 |
| 0.54 | 0.97 | 0.04 | 485 | 480 | 20 | 15 |
| 0.55 | 0.97 | 0.04 | 485 | 480 | 20 | 15 |
| 0.56 | 0.97 | 0.04 | 485 | 480 | 20 | 15 |
| 0.57 | 0.968 | 0.04 | 484 | 480 | 20 | 16 |
| 0.58 | 0.968 | 0.04 | 484 | 480 | 20 | 16 |
| 0.59 | 0.968 | 0.04 | 484 | 480 | 20 | 16 |
| 0.6 | 0.968 | 0.04 | 484 | 480 | 20 | 16 |
| 0.61 | 0.968 | 0.04 | 484 | 480 | 20 | 16 |
| 0.62 | 0.966 | 0.04 | 483 | 480 | 20 | 17 |
| 0.63 | 0.962 | 0.04 | 481 | 480 | 20 | 19 |
| 0.64 | 0.96 | 0.04 | 480 | 480 | 20 | 20 |
| 0.65 | 0.96 | 0.04 | 480 | 480 | 20 | 20 |
| 0.66 | 0.96 | 0.04 | 480 | 480 | 20 | 20 |
| 0.67 | 0.96 | 0.038 | 480 | 481 | 19 | 20 |
| 0.68 | 0.958 | 0.038 | 479 | 481 | 19 | 21 |
| 0.69 | 0.958 | 0.038 | 479 | 481 | 19 | 21 |
| 0.7 | 0.958 | 0.038 | 479 | 481 | 19 | 21 |
| 0.71 | 0.958 | 0.038 | 479 | 481 | 19 | 21 |
| 0.72 | 0.958 | 0.038 | 479 | 481 | 19 | 21 |
| 0.73 | 0.958 | 0.036 | 479 | 482 | 18 | 21 |
| 0.74 | 0.958 | 0.034 | 479 | 483 | 17 | 21 |
| 0.75 | 0.958 | 0.034 | 479 | 483 | 17 | 21 |
| 0.76 | 0.958 | 0.034 | 479 | 483 | 17 | 21 |
| 0.77 | 0.956 | 0.034 | 478 | 483 | 17 | 22 |
| 0.78 | 0.956 | 0.032 | 478 | 484 | 16 | 22 |
| 0.79 | 0.954 | 0.03 | 477 | 485 | 15 | 23 |
| 0.8 | 0.954 | 0.03 | 477 | 485 | 15 | 23 |
| 0.81 | 0.954 | 0.03 | 477 | 485 | 15 | 23 |
| 0.82 | 0.954 | 0.03 | 477 | 485 | 15 | 23 |
| 0.83 | 0.95 | 0.03 | 475 | 485 | 15 | 25 |
| 0.84 | 0.95 | 0.03 | 475 | 485 | 15 | 25 |
| 0.85 | 0.95 | 0.03 | 475 | 485 | 15 | 25 |
| 0.86 | 0.95 | 0.028 | 475 | 486 | 14 | 25 |
| 0.87 | 0.948 | 0.028 | 474 | 486 | 14 | 26 |
| 0.88 | 0.948 | 0.028 | 474 | 486 | 14 | 26 |
| 0.89 | 0.946 | 0.028 | 473 | 486 | 14 | 27 |
| 0.9 | 0.946 | 0.026 | 473 | 487 | 13 | 27 |
| 0.91 | 0.946 | 0.026 | 473 | 487 | 13 | 27 |
| 0.92 | 0.944 | 0.026 | 472 | 487 | 13 | 28 |
| 0.93 | 0.944 | 0.026 | 472 | 487 | 13 | 28 |
| 0.94 | 0.942 | 0.024 | 471 | 488 | 12 | 29 |
| 0.95 | 0.938 | 0.024 | 469 | 488 | 12 | 31 |
| 0.96 | 0.93 | 0.024 | 465 | 488 | 12 | 35 |
| 0.97 | 0.928 | 0.024 | 464 | 488 | 12 | 36 |
| 0.98 | 0.926 | 0.022 | 463 | 489 | 11 | 37 |
| 0.99 | 0.916 | 0.016 | 458 | 492 | 8 | 42 |
| 1 | 0 | 0 | 0 | 500 | 0 | 130 |

### APPendix E: Threshold

**Optimization for googlenet**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Threshold | True Positive Rate | False Positive Rate | True Positives | True Negatives | False Positives | False Negatives |
| 0 | 1 | 1 | 500 | 0 | 500 | 0 |
| 0.01 | 0.988 | 0.16 | 494 | 420 | 80 | 6 |
| 0.02 | 0.988 | 0.126 | 494 | 437 | 63 | 6 |
| 0.03 | 0.986 | 0.122 | 493 | 439 | 61 | 7 |
| 0.04 | 0.978 | 0.112 | 489 | 444 | 56 | 11 |
| 0.05 | 0.978 | 0.106 | 489 | 447 | 53 | 11 |
| 0.06 | 0.978 | 0.102 | 489 | 449 | 51 | 11 |
| 0.07 | 0.978 | 0.098 | 489 | 451 | 49 | 11 |
| 0.08 | 0.978 | 0.094 | 489 | 453 | 47 | 11 |
| 0.09 | 0.976 | 0.09 | 488 | 455 | 45 | 12 |
| 0.1 | 0.974 | 0.088 | 487 | 456 | 44 | 13 |
| 0.11 | 0.974 | 0.086 | 487 | 457 | 43 | 13 |
| 0.12 | 0.972 | 0.086 | 486 | 457 | 43 | 14 |
| 0.13 | 0.97 | 0.086 | 485 | 457 | 43 | 15 |
| 0.14 | 0.966 | 0.084 | 483 | 458 | 42 | 17 |
| 0.15 | 0.966 | 0.084 | 483 | 458 | 42 | 17 |
| 0.16 | 0.964 | 0.084 | 482 | 458 | 42 | 18 |
| 0.17 | 0.964 | 0.084 | 482 | 458 | 42 | 18 |
| 0.18 | 0.964 | 0.084 | 482 | 458 | 42 | 18 |
| 0.19 | 0.962 | 0.084 | 481 | 458 | 42 | 19 |
| 0.2 | 0.96 | 0.082 | 480 | 459 | 41 | 20 |
| 0.21 | 0.958 | 0.082 | 479 | 459 | 41 | 21 |
| 0.22 | 0.958 | 0.082 | 479 | 459 | 41 | 21 |
| 0.23 | 0.958 | 0.082 | 479 | 459 | 41 | 21 |
| 0.24 | 0.958 | 0.08 | 479 | 460 | 40 | 21 |
| 0.25 | 0.958 | 0.08 | 479 | 460 | 40 | 21 |
| 0.26 | 0.958 | 0.08 | 479 | 460 | 40 | 21 |
| 0.27 | 0.958 | 0.08 | 479 | 460 | 40 | 21 |
| 0.28 | 0.956 | 0.078 | 478 | 461 | 39 | 22 |
| 0.29 | 0.956 | 0.078 | 478 | 461 | 39 | 22 |
| 0.3 | 0.956 | 0.076 | 478 | 462 | 38 | 22 |
| 0.31 | 0.952 | 0.074 | 476 | 463 | 37 | 24 |
| 0.32 | 0.95 | 0.074 | 475 | 463 | 37 | 25 |
| 0.33 | 0.95 | 0.072 | 475 | 464 | 36 | 25 |
| 0.34 | 0.95 | 0.072 | 475 | 464 | 36 | 25 |
| 0.35 | 0.95 | 0.072 | 475 | 464 | 36 | 25 |
| 0.36 | 0.95 | 0.072 | 475 | 464 | 36 | 25 |
| 0.37 | 0.95 | 0.072 | 475 | 464 | 36 | 25 |
| 0.38 | 0.95 | 0.072 | 475 | 464 | 36 | 25 |
| 0.39 | 0.95 | 0.072 | 475 | 464 | 36 | 25 |
| 0.4 | 0.95 | 0.072 | 475 | 464 | 36 | 25 |
| 0.41 | 0.948 | 0.07 | 474 | 465 | 35 | 26 |
| 0.42 | 0.948 | 0.068 | 474 | 466 | 34 | 26 |
| 0.43 | 0.946 | 0.066 | 473 | 467 | 33 | 27 |
| 0.44 | 0.946 | 0.066 | 473 | 467 | 33 | 27 |
| 0.45 | 0.946 | 0.066 | 473 | 467 | 33 | 27 |
| 0.46 | 0.946 | 0.064 | 473 | 468 | 32 | 27 |
| 0.47 | 0.946 | 0.064 | 473 | 468 | 32 | 27 |
| 0.48 | 0.946 | 0.064 | 473 | 468 | 32 | 27 |
| 0.49 | 0.946 | 0.062 | 473 | 469 | 31 | 27 |
| 0.5 | 0.946 | 0.062 | 473 | 469 | 31 | 27 |
| 0.51 | 0.942 | 0.062 | 471 | 469 | 31 | 29 |
| 0.52 | 0.942 | 0.06 | 471 | 470 | 30 | 29 |
| 0.53 | 0.94 | 0.06 | 470 | 470 | 30 | 30 |
| 0.54 | 0.94 | 0.06 | 470 | 470 | 30 | 30 |
| 0.55 | 0.94 | 0.058 | 470 | 471 | 29 | 30 |
| 0.56 | 0.94 | 0.058 | 470 | 471 | 29 | 30 |
| 0.57 | 0.94 | 0.058 | 470 | 471 | 29 | 30 |
| 0.58 | 0.94 | 0.056 | 470 | 472 | 28 | 30 |
| 0.59 | 0.94 | 0.056 | 470 | 472 | 28 | 30 |
| 0.6 | 0.94 | 0.05 | 470 | 475 | 25 | 30 |
| 0.61 | 0.94 | 0.05 | 470 | 475 | 25 | 30 |
| 0.62 | 0.94 | 0.05 | 470 | 475 | 25 | 30 |
| 0.63 | 0.938 | 0.05 | 469 | 475 | 25 | 31 |
| 0.64 | 0.936 | 0.044 | 468 | 478 | 22 | 32 |
| 0.65 | 0.936 | 0.044 | 468 | 478 | 22 | 32 |
| 0.66 | 0.936 | 0.044 | 468 | 478 | 22 | 32 |
| 0.67 | 0.936 | 0.044 | 468 | 478 | 22 | 32 |
| 0.68 | 0.936 | 0.044 | 468 | 478 | 22 | 32 |
| 0.69 | 0.936 | 0.044 | 468 | 478 | 22 | 32 |
| 0.7 | 0.936 | 0.042 | 468 | 479 | 21 | 32 |
| 0.71 | 0.934 | 0.038 | 467 | 481 | 19 | 33 |
| 0.72 | 0.934 | 0.038 | 467 | 481 | 19 | 33 |
| 0.73 | 0.934 | 0.034 | 467 | 483 | 17 | 33 |
| 0.74 | 0.934 | 0.034 | 467 | 483 | 17 | 33 |
| 0.75 | 0.934 | 0.034 | 467 | 483 | 17 | 33 |
| 0.76 | 0.934 | 0.034 | 467 | 483 | 17 | 33 |
| 0.77 | 0.934 | 0.032 | 467 | 484 | 16 | 33 |
| 0.78 | 0.932 | 0.032 | 466 | 484 | 16 | 34 |
| 0.79 | 0.932 | 0.032 | 466 | 484 | 16 | 34 |
| 0.8 | 0.932 | 0.032 | 466 | 484 | 16 | 34 |
| 0.81 | 0.932 | 0.032 | 466 | 484 | 16 | 34 |
| 0.82 | 0.932 | 0.032 | 466 | 484 | 16 | 34 |
| 0.83 | 0.932 | 0.032 | 466 | 484 | 16 | 34 |
| 0.84 | 0.93 | 0.03 | 465 | 485 | 15 | 35 |
| 0.85 | 0.928 | 0.03 | 464 | 485 | 15 | 36 |
| 0.86 | 0.928 | 0.03 | 464 | 485 | 15 | 36 |
| 0.87 | 0.928 | 0.03 | 464 | 485 | 15 | 36 |
| 0.88 | 0.922 | 0.03 | 461 | 485 | 15 | 39 |
| 0.89 | 0.922 | 0.03 | 461 | 485 | 15 | 39 |
| 0.9 | 0.922 | 0.028 | 461 | 486 | 14 | 39 |
| 0.91 | 0.922 | 0.028 | 461 | 486 | 14 | 39 |
| 0.92 | 0.92 | 0.026 | 460 | 487 | 13 | 40 |
| 0.93 | 0.92 | 0.024 | 460 | 488 | 12 | 40 |
| 0.94 | 0.918 | 0.02 | 459 | 490 | 10 | 41 |
| 0.95 | 0.914 | 0.014 | 457 | 493 | 7 | 43 |
| 0.96 | 0.914 | 0.014 | 457 | 493 | 7 | 43 |
| 0.97 | 0.906 | 0.014 | 453 | 493 | 7 | 47 |
| 0.98 | 0.9 | 0.01 | 450 | 495 | 5 | 50 |
| 0.99 | 0.886 | 0.01 | 443 | 495 | 5 | 57 |
| 1 | 0 | 0 | 0 | 500 | 0 | 196 |