SUPPORT VECTOR MACHINES: SUNSET DETECTOR

Aaron Mercier
Larry Gates

Computer Science at Rose-Hulman Institute of Technology
Email: mercieal@rose-hulman.edu
gateslm@rose-hulman.edu

ABSTRACT

Scene classification in image recognition is not easily done by using single features such as color. Using a support vector machine (SVM), scenes can be classified with reasonable accuracy by using training images to determine the thresholds more appropriately. Determining the mean and standard deviation for a given block defined in the image in the RGB color space converted to LST color space allowed for a strongly descriptive set of features. To improve on the baseline sunset detector, different weights were applied to the individual bands of the LST color space in an attempt to maximize accuracy. Without the improvement to the baseline sunset detector, running the SVM with the a set of test images resulted in an accuracy of about 87.6% with σ (*sigma*) $= 0.1$ and $C = 20$. Including the improvement to the baseline, the accuracy was not greater than the original 87.6% accuracy at a threshold of 0 found with no weights before normalization. The weights did in fact cause a significant variation in final accuracy, TPR, and FPR of the SVM results when applied after normalization. We found that while doubling all of the bands did not change the accuracy, the SVM did have lower FPR and higher TPR values.

Keywords— RGB, LST, ROC Curve, classification, SVM, Sunset

1. INTRODUCTION

Extracting information about the content of an image has been been a challenge that has seen major growth in need over the years, and as such the techniques have been improving rapidly. Detecting if an image contains a sunset or not is just one small part of this massive goal, but it does show that it is possible to extract this information and with reasonably good accuracy. Detecting a sunset could have several practical uses as well. One example could be an auto-detecting filter that automatically detects a sunset and filters accordingly to bring out the brighter colors and make the sunset stand out. Another example includes using the detected sunset to provide further information regarding the direction the picture was taken as well as the estimated time the picture was taken.

Detecting and classifying anything in an image is not trivial, and sunsets are no different. One of biggest challenges of detecting sunsets is that really the only feature that distinguishes between scenes with or without sunsets is color. The problems with color are that there can be many images of anything that could potentially be very similar in color to a sunset, and sunsets also can vary greatly in color. For example, consider figure 1 below. Figure 1a shows what appears to be a volcano crater at night, while figure 1b shows a sunset on a cloudy day. The colors of both images are strikingly similar, which makes it incredibly challenging to determine if the volcano is a sunset or not.

(a) (b)

Figure 1. An image of a volcano crater at night (a) and an image of a sunset on a cloudy day (b)

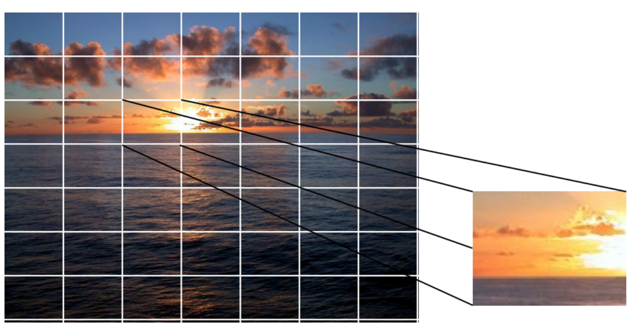
The proposed solution uses a series of advanced image recognition technologies similar to the baseline sunset detector implemented in the original sunset paper plus some additional minor changes in an attempt to solve the problem of detecting sunsets with reasonable accuracy [1]. Because there are images with no sunsets that are extremely similar to a sunset in terms of color as explained earlier, the solution needs to be extremely fine-tuned at classifying the colors of these images. Therefore the solution uses a machine that is able to be trained on pre-classified images and uses this trained model to classify the images as containing sunsets or non-sunsets.

2. PROCESS

The process used to extract the features and classify the images starts by breaking up the image into smaller, more consistently colored blocks, then extracting the features from each block, normalizing the features, and finally using the normalized features to train and test the SVM. This process follows the work done for the baseline detector done by M. Boutell, J. Luo, R. Gray [1].

2.1. Dividing the image into a grid

The first part of the baseline sunset detector breaks the image into a 7x7 grid. This creates 49 smaller sub-images from the original image which makes finding the sunset much easier because the colors in a smaller sub-image should vary less. For example, consider figure 2a below. The full image contains a wide variety of colors other than the sunset, such as the blue from the water, the darker orange from the clouds, and other parts of the sky. If we break the image into the 7x7 regions as depicted by the grid lines (fig. 2a), we can extract the color features for each one and find that the image in figure 2b contains a very strong indication of a sunset.



(a) (b)

Figure 2. Splitting the image up into 7x7 grids (a) and producing a small subset of the image (b) used for extracting the mean and the standard deviation.

However if the image's width and/or height is not perfectly divisible by 7, some parts of the image will need to be cut off. We decided to clip off the excess (at most 6 pixels) from the right and from the bottom if this happens. While it would be better to clip an even amount (at most 3 pixels) from each side of the image, we decided that doing so would complicate the algorithm for little to no gain because the chances of a sunset being located 6 pixels away from the edges of the image are minimal.

2.2. Extracting color moments

The first step in extracting the features from these images in the grid is to convert the image from the RGB color space to the LST color space. This is because the LST color space is more uniform and more easily interpreted by a human, even though the features could have used the RGB space with little variation in results. The L band conversion equation simply sums the red, green, and blue values (R+G+B) at each pixel, the S band subtracts the red value by the blue value (R-B) at each pixel, and the T band subtracts the red value plus the blue value by twice the green value (R+B-2G) at each pixel. Once we have these unscaled LST values, we then find the first two color moments of each value and use those as the feature vectors that are used to train the SVM (support vector machine). The first color moment is the mean value of each band, which is found by adding up each value and dividing by the total number of pixels. The second color moment is the standard deviation which quantifies the amount of variation in the color band. Because there are 2 color moments per each of the 3 bands per each of the 49 sub-images of the full image, this makes 2*3*49 = 294 feature vectors per image input into the sunset detector.

2.3. Normalizing the feature vectors

The conversion from the RGB to the LST color space is left unscaled, therefore the theoretical range values of L is 0 to 765, S is -255 to 255, and T is -512 to 512. Because the ranges of values for L, S, and T are so different, the SVM could train with these feature vectors weighting one band more than the other. We want each band weighted evenly, so we have to normalize the data so that the range of each band is the same.

To normalize the feature vectors, we must first find the minimum value of that feature vector and subtract by it to shift all ranges to start at 0. We then find the maximum value of each band and divide by that so that each range has a maximum value of 1. Once this is complete for each feature vector, each feature vector has a normalized range from 0 to 1 and one does not outweigh the other.

2.4. SVM Classification

Once we have a set of training images, each broken up into 294 normalized feature vectors, we pass them into an SVM that will classify the images. A SVM first needs to take a kernel function that the SVM will use to map the nonseparable data into a higher dimension, where a decision boundary can be found to classify the images. We decided to use a RBF (Gaussian Radial-basis function) as the kernel function because it is known to provide more accurate decision boundaries. The RBF requires an extra parameter σ and the svm requires an extra parameter C, so we need to find a combination of these two parameters that give us the best accuracy when running a set of test images on the trained SVM.

To determine the optimum combination, we decided to loop through many possible combinations, retrain the SVM, and retest while storing each result into a matrix. We looped from $\sigma = 0.1$ to 17 in increments of .5 and C = 10 to 100 in increments of 5, thus creating 34 * 19 = 646 combinations of σ and C. Once we had the results for each combination, we analyzed the results and found which combination(s) returned the greatest accuracy with a reasonably low number of support vectors. We aimed for at most half as many support vectors as training images, which would mean that each training data is not represented by its own support vector and hurt performance. We found that the combination of $\sigma = 0.1$ and $C = 20$ gave us the highest accuracy with a reasonable number of support vectors.

3. NOVEL WORK

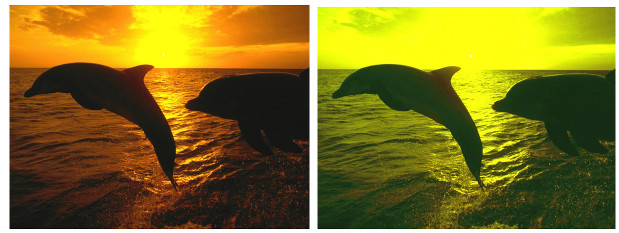
To extend the original work done in the baseline detector, a novel approach was taken by applying weights to the different bands of the LST color scheme. The approach was to attempt to see if applying different weights to the bands of the LST color scheme would contribute more to the distance calculations of an SVM. Based off of the intuition that one band may contribute more to the detection of a sunset, we weight a band more than another in an attempt to improve the overall accuracy obtained when running the training and testing data through the classifier. This technique was a novel concept that was thought of the previous year and had not been attempted. The changes from the initial structure of the code were minimal.

3.1 Weights Before Normalization

Initially, we tested the novel approach with scaling the weights before the normalization. The normalization scales all of the values between the range of 0 and 1. The weights would essentially be unnoticeable.

3.1.1 Modifying the Weight

To modify the weights of the color bands, additional parameters were added to the RGB-to-LST conversion function to multiply the matrix representing each band of the LST by the appropriate weight. Determining the maximized value of each band required the method of guessing and checking. With the RGB scheme, multiplying a certain band by a factor will change the appearance of the image. As pictured in figure 3, doubling the green band notably made the image brighter, and caused the sun and sunrays to blend into a single object.



(a) (b)

Figure 3. Comparison of the original image (a) to the same image with the green band of RGB doubled (b).

Following the idea of doubling a band to affect the image, the weighting one or more of the LST bands should modify the image's appearance, as well as the mean and the standard deviation of the color bands. This is under the assumption that the result would not be the same as weighting one of the bands for RGB.

3.1.2. Testing the Novel Work

A wide variety of weights were tested due to the fact that the weights needed to maximize the accuracy were unknown. The initial idea was to start off by applying the same weight to all the bands, then applying a different weight to a single band, followed by alternating two of the bands to be the same value, as well as having each band being different value.

The use of weights that were the same value across all bands were expected to keep the accuracy very similar because each band would be contributing the same amount. When using a weight that was 0 or close to 0 for any of the bands, the predicted response would be to distort the bands to a value of 0 or close to 0. This would skew the accuracy significantly depending on the influence the band has on the color model. The same result was expected for bands that had weights greater than 1.

3.2 Weights After Normalization

To show that weights do impact the accuracy and TPR over the FPR rate, we also applied weights to the mean and standard deviation of each band. This would make the range of the weighted bands no longer the range from 0 to 1.

3.2.1 Modifying the Weights

The matrix that holds the 294 feature vector for each image contains the mean and standard deviation of each color band for each image block. The mean and standard deviation are normalized by the normalization function. To scale the weights of each band, the mean and standard deviation need to be scaled to impact the feature matrix.

3.2.2 Testing the Novel Work

The weights used are close to 1 or zero, using an outlier of 100. The use of weights that were the same value across all bands were expected to keep the accuracy very similar because each band would be contributing the same amount. When using a weight that was 0 the predicted response would be to distort the TPR over FPR rate extremely because that band would essentially contribute nothing to the classifier. This would skew the accuracy significantly depending on the influence the band has on the color model.

4. RESULTS

The SVM was trained using 499 total images: 276 images containing no sunset and 223 images containing sunsets of varying degree and sizes. Once the SVM was trained, we tested the SVM using 605 images: 50 images containing sunsets difficult to detect, 222 images containing sunsets that are easier to detect, 92 images without sunsets that could easily be misclassified, and 241 images without sunsets that are easier to classify.

4.1. Original Sunset Paper Implementation Results

The values of $\sigma = 0.1$ and $C = 20$ as parameters in the classifier resulted in an accuracy of about 87.6% when running the classifier on the 605 testing images and 145 support vectors which is about 30% of the total 499 training images input into the SVM. Table 1 below shows a small portion of the accuracies obtained when testing different combinations of σ and C.

Table 1. Subset of the testing to find the maximum accuracy based on the varying sigma and c-parameter. The subset contains the accuracy of the sigma and c-parameter used. This is 30 samples out of the 646 values, as discussed in section 2.4.

Sigma	C	Accuracy	Sigma	C	Accuracy
0.1	10	0.871074	1.6	10	0.861157025
0.1	15	0.871074	1.6	15	0.854545455
0.1	20	0.876145	1.6	20	0.852892562
0.1	25	0.867769	1.6	25	0.854545455
0.1	30	0.867769	1.6	30	0.856198347
0.6	10	0.856198	2.1	10	0.85785124
0.6	15	0.859504	2.1	15	0.861157025
0.6	20	0.866116	2.1	20	0.854545455
0.6	25	0.867769	2.1	25	0.852892562
0.6	30	0.871074	2.1	30	0.854545455
1.1	10	0.856198	2.6	10	0.856198347
1.1	15	0.852893	2.6	15	0.856198347
1.1	20	0.856198	2.6	20	0.85785124
1.1	25	0.857851	2.6	25	0.854545455
1.1	30	0.861157	2.6	30	0.852892562

Once we had the optimal values for σ and C, we then ran the SVM with the test dataset and did some statistical analysis to determine the number of true positives, true negatives, false positives, and false negatives and displayed them in a confusion matrix. These values were then used to determine the accuracy shown in Table 1, as well as the true positive rate (TPR) and false positive rate (FPR) needed to generate an receiver operating characteristic (ROC) curve. We generated the ROC curve (figure 3) by varying the threshold that is compared with the signed distance returned by the SVM to classify the image. By finding the minimum and maximum values of the distances returned by the SVM which were about -4.0 and 4.0 respectively, we determined that these would be good values to vary the threshold from.

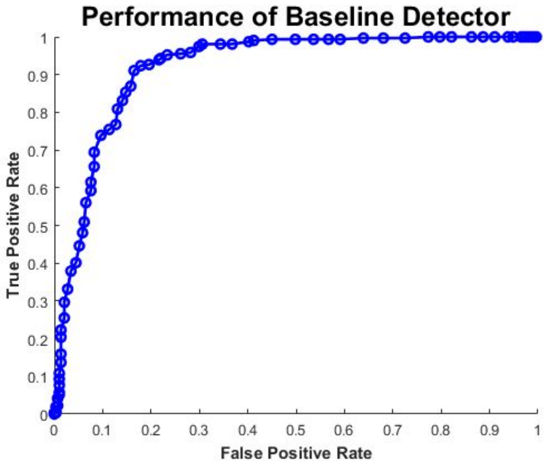


Figure 3. Resulting ROC curve by varying the threshold in a range of -4.0 to 4.0 with incrementing values of .1. The values of $\sigma = 0.1$ and $C = 20$ were used.

Figure 3 shows the results of TPR vs FPR when varying threshold (each dot represents a run with a different threshold value). As the threshold nears the minimum value of the distance (strongly detected as not a sunset), we can see that both TPR and FPR approach 100%. This is because everything is being classified as containing a sunset. Conversely, when the threshold nears the maximum value of the distance, the TPR and FPR approach 0% because nothing is classified as containing a sunset. Having a threshold value of 0 (which is the default for SVM's) results in a TPR of approximately 92.357% and an FPR of approximately 18.213% which is reasonably close the goal of 100% TPR and 0% FPR. The high FPR value is believed to be a result of the 92 difficult test images that appear to have sunsets but actually do not. The ROC shows that if we use a higher threshold we could get lower FPR values. For example, we could achieve a FPR lower than 10% which would lower the TPR to less than 70% as well. We could also get a higher TPR if we decrease threshold. For example, we could achieve a near 100% TPR which would increase the FPR to at least 30%.

4.2. Novel Work Results

When we applied weights to the bands before normalization, all of the values became scaled to a range from 0 to 1 so the changes in any weights applied would be unnoticeable. After applying the weight after the normalization, a band's mean and standard deviation would be in a range that is not guaranteed to be directly from 0 to 1. Both tests use the values of $\sigma = 0.1$ and $C = 20$ from the initial testing were used as the kernel function and parameter input into the SVM. The threshold values used ranged from -4.0 to 4.0, in incrementing values of 0.1 as in the initial testing.

4.2.1 Weights Before Normalization

A total of 16 sets of three weight values. Refer to table 2 for the corresponding accuracy for a set of weights.

Table 2. Weights used for each band on a given test with accuracies. The threshold was set to 0. The ordering of the table was not the order of the individual testing.

L Weights	S Weights	T Weights	Max. Accuracy
1	1	1	0.8760
1	1	100	0.8744
1	1	.0000000001	0.8727
1	.0000000001	1	0.8727
1	.5	.5	0.8744
1	.25	.25	0.8727
2	2	2	0.8727
100	.01	.01	0.8760
1000	.001	1000	0.8727
.0000000001	1	1	0.8760
.01	100	.01	0.8727
.001	1000	1000	0.8760
.01	.01	100	0.8744
.001	.001	1000	0.8744
.5	.5	.5	0.8727
.25	.25	.25	0.8727

The ordering of the test was not a factor when producing the results. Figure 4 shows the overlay of each one of the weight sets. As noted by the overlaying ROC curves and the maximized accuracies from each weight set run, the accuracy was never greater than the maximum value that was found.

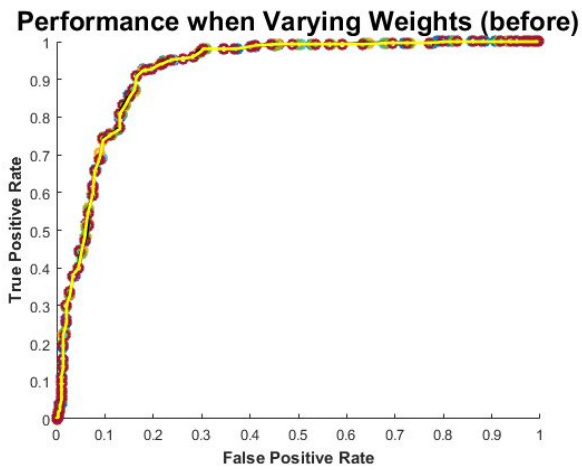


Figure 4. Result of tests using different weights applied before normalization on each band as described in section 4.2.1.

The ROC curve in figure 4 shows the different false positive rates versus the true positive rate as the threshold changes with 16 weight sets. The overlaying of the 16 weight sets shows that TPR over FPR rate did not vary from the original accuracy with no weights.

While initially determining values to use for weights, the weight zero could not be used. The normalization of the band would divide by the maximum value, which would be zero. This arithmetic error would prevent the calculations from continuing. Instead of removing the normalization, using a value that would take any number close to zero was used.

4.2.2 Weights After Normalization

A total of 13 sets of three weight values. Refer to table 2 for the corresponding accuracy for a set of weights.

Table 3. Weights used for each band on a given test with a given accuracy. The threshold was set to 0. The ordering of the table was not the order of the individual testing. The values were changed from the values used in section 4.2.1.

L Weights	S Weights	T Weights	Accuracy	Line Color
1	1	1	0.87438	Yellow
2	2	2	0.87438	Magenta
100	100	100	0.480992	Cyan
0.25	0.25	0.25	0.852893	Red
1	1	100	0.480992	Green
1	100	1	0.480992	Blue
100	1	1	0.480992	Black
1	1	0	0.849587	Yellow
1	0	1	0.831405	Magenta
0	1	1	0.803306	Cyan
1	0	0	0.809917	Red
0	1	0	0.743802	Green
0	0	1	0.717355	Blue

The ordering of the test was not a factor when producing results. Figure 5 shows the overlay of each one of the weight sets. As noted by the overlaying ROC curves and the maximized accuracies from each weight set run, the accuracies now vary far more when applying the weights after the normalization than being applied before normalization.

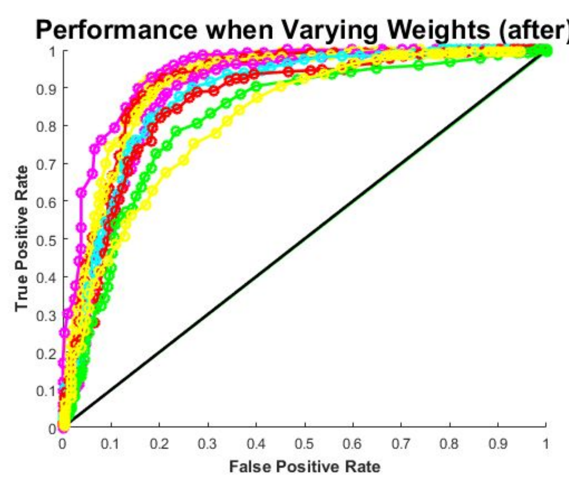


Figure 5. Result of tests using different weights applied after normalization on each band as described in section 4.2.2.

The ROC curve in figure 5 shows the different false positive rates versus the true positive rate as the threshold changes with 13 weight sets. The TPR over the FPR rate begins to vary with different weights as the threshold is around 0. When the thresholds are close to 4 or -4, the TPR over the FPR rate stays close together. One thing to notice is that the magenta line where all bands are doubled provided a higher TPR for low values of FPR compared to the other lines. Another includes the straight lines going from (0,0) to (1,1) This is because with the extremely high weights around 100, images were either being classified as either completely not having a sunset or completely having a sunset. Therefore as threshold varies, the images go from nothing classified as sunsets to everything classified sunsets and thus results in points only at (0,0) and (1,1).

5. DISCUSSION AND CONCLUSION

The baseline results for the sunset detector appeared to match the results found in the original sunset paper [1].This algorithm does not have as much of a limitation as a classifier that uses a single feature to identify and classify objects. Improving the classifier involved using a superior feature vector, having 6 features for every block defined in a picture. Using SVMs showed the importance of having multiple dimensions of features by allowing for a nonlinear boundary line. Even though the accuracy was similar through the varying of the weights on the color bands, the images in each iteration could have not been the same as the previous iteration. While running the tests, we came across the problem where we would receive different accuracies but the kernel function, SVM parameters and weights were kept the same. We were unable to determine the cause of this.

5.1. Examining Images from Confusion Matrix [2]

The images in figure 6 were classified as false positives, meaning the images processed by the program were detected as containing a sunset. As humans, it is apparent that a sunset is not visible in either image. The hat on the child's head (Fig. 6a) could be confused as a sun. The sand in the background also starts to have a different shade at the start of the hat, appearing almost as the change from landscape to sky. The image 6b is a mystery as to the misclassification. The distance returned from the SVM was .8695, in a range of about -4.0 to 4.0. The qualities that would help influence the false positive would be the sky and the smoke stack having a color that stands out against the blue sky.

(a) (b)

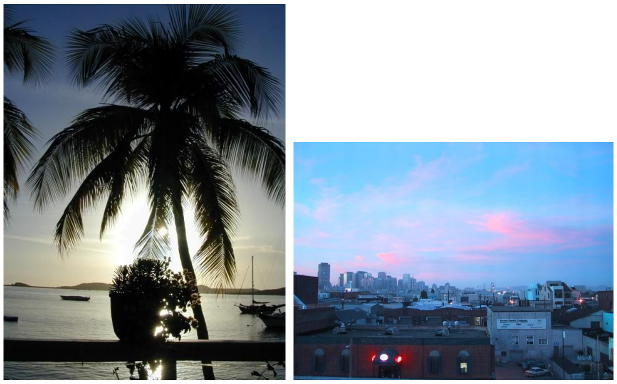
Figure 6. Images incorrectly classified as a sunset.

The images in figure 7 were classified correctly as true positives. The coloring of the sky matches what humans would note in a sunset. While the landscape does provide a challenge to blocking the color of the complete sunset, the program was still able to detect the sunset because the image was broken up into 49 smaller blocks.

(a) (b)

Figure 7. Images correctly classified as a sunset.

Both images were classified as false negatives in figure 8, meaning that images are classified as the true results but detected as a negative result. The palm tree (Fig. 8a) blocks the view of the sunset and obscures the sun. The sun appears to be hidden in the leaves on the branch. The image also appears to have been filtered to remove the reddish orange tint that people assimilate with a sunset. The sun is not visible in figure 8b, only providing the color of the sky, as in figure 7b. Compared to figure 7b, figure 8b has brighter colors that are closer to the landscape, the buildings, which potentially blend in with the sky at a far enough distance.



(a) (b)

Figure 8. Images incorrectly classified as not a sunset.

The images in figure 9 were correctly classified as not containing a sunset, known as a true negative. Both images do not have an orange glow, which is typically present in a sunset.

(a) (b)

Figure 9. Images correctly classified as not a sunset.

5.2. Given 2 More Weeks

Given a short extension, more extensive results for the color band weights could be produced. Using a larger range of weights and varying the combination of weights to bands could be tested more comprehensively, both before and after normalization. The use of smaller increment values in a range for band weights could be incorporated. The time to calculate the results of varying the weights depended heavily on the speed of the program. To improve the training set, additional sunset and not sunset examples could be found and used as training. As discussed at the beginning of this section, additional testing could be done to check if any images are different when changing the color band weights. We could have also done further analysis on the ROC curve to determine which threshold value provided the optimal combination of high TPR and low FPR.

5.3. Given 1 More Year

Given a longer extension, more weighting can be done in the calculation of LST. By attempting to use weights on the RGB to LST conversion, different results might be more noticeable than just weights on the LST bands themselves. Parallelizing the program would speed up testing, as testing various weights can be done in parallel because the results of using one weight over the other does not require the next weight to be tested. The use of deep neural nets could improve the classifiers. Reading the documentation for the neural net packages would take time. Once understanding the full documentation of a neural net package, optimizing the neural net to different parameters would not be challenge. Using weighting the bands, neural nets, or not modifying the original paper while experimenting with kernel functions could produce different results. Since each of the novel approaches could have different results based on the use of a different kernel function, a large amount of the time would be spent varying values and examining the results. To add on top of the year extension, the short extension, and the testing done relative to the original paper, repeating all of those with different training and testing schemes [1]. Training and test schemes such as no training or training on both the test and training sets could produce many different results, which would vary extremely with the weighting of the color bands and the use of neural nets.

6. REFERENCES

[1] Matthew Boutell, Jiebo Luo, and Robert T. Gray. *IEEE International Conference on Multimedia and Expo*, Baltimore, MD, July 2003.

[2] M. Boutell, *CSSE463 - Image Recognition: In-class Quiz 11*. [Online] Available: http://www.rose-hulman.edu/class/csse/csse463/201720/Quizzes/Day11%20classifiers.docx. Accessed: Jan. 18, 2017.