Irrigation Decision based on SVM with RGB images of Green Spaces

João Doroana
joao.doroana@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2020

Abstract

Water is a fundamental resource in our world. We need it, we use it and we waste it. In every city we see a lot of green spaces. To maintain the green colour in those spaces we need to consume a lot of water. To control the amount of water to irrigate green spaces, classical solutions use soil sensors that are quite localised and expensive. Image based solutions have a wider range of analysis but most work has been done on satellite imagery that include multispectral image. To design a more accessible solution, we propose to interpret RGB images acquired from normal off-the-shelf cameras. We adopt a data driven approach: create a dataset of green spaces labeled by experts with 1 of the 4 categories: (i) *definitely needs irrigation*, (ii) *needs irrigation*, (iii) *doesn't need irrigation*, and (iv) *definitely doesn't need irrigation*. From this dataset we train SVM classification and regression models with different types of images and strategies, comparing results in order to find the best method. Our final model shows an accuracy of 80 % in a diversity of green spaces.
Keywords: Irrigation, RGB, SVR, svc, SVMs, Color

1. Introduction

Agriculture accounts for, on average, 70 percent of all water withdrawals globally, and an even higher share of "consumptive water use", due to the evapotranspiration requirements of crops. Competition for water resources is expected to increase in the future due to population growth, urbanization, industrialization, and climate change. Thus, we need to find more efficient ways of using our water.

There are already some approaches, mentioned in [3] and [15], to save water such as: use weather stations, to estimate the evapotranspiration of the plant and the precipitation on that area, to predict how much water will be missing in the following days; or placing humidity sensors in the soil to measure the water is present in a small area around the sensor. With this type of data an autonomous system can turn on the sprinklers and manage the water in the field. Other approaches are to use images from drones and satellites observing the crops. With these images, the farmer can decide if there is a need for more irrigation or not.

Agriculture is a complex science because, there are lots of types of plants and each plant has their unique characteristics: it reacts differently to different soils and days of year and other conditions, as explained in [2]. Besides the water needs for agriculture there are green spaces inside our cities that also require appropriate irrigation [9]. As the name implies, they are supposed to stay "green" in order to be healthy. In this thesis we focus on the image analysis of green spaces to optimize irrigation.

We propose to create a system that, given an RGB image, computes regions of that image with a numerical classification, corresponding to how much water is missing in that region. To achieve this numerical classification we have created a dataset with the help of an agronomy specialist, that has labeled several regions of images of grass fields as: *definitely needs irrigation*, *needs irrigation*, *doesn't need irrigation*, *definitely doesn't need irrigation*. With this information we will train a support vector regression in order to obtain a continuous classification of each region. Those values will be between 0 and 1, with the need for irrigation decreasing with the value.

With this work we want to contribute with a new way of creating vegetation indexes. This method would be able to work in parallel with the weather stations and humidity sensors in order to make better autonomous decisions in the process of irrigation.

2. State-of-the-art

In this section, we will cover some state-of-the-art methods that are most relevant to our work.

2.1. Support Vector Machines

SVMs construct a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. A good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin) since, in general, the larger the margin the lower the generalization error of the classifier [7].

2.1.1 Classification

When SVMs are used for classification, they are called SVC. Given training vectors $x_i \in \mathbb{R}^p$, $i = 1, ..., n$, in two classes and a vector $y \in \{1, -1\}^n$, our goal is to find $w \in \mathbb{R}^p$ and $b \in \mathbb{R}$. The primal problem is defined as

$$\begin{array}{ll} \underset{w,b,\zeta}{min} & \frac{1}{2}w^T w + C\sum_{i=1}^{n}\zeta_i \\ s.t. & y_i(w^T\phi(x_i) + b) \geq 1 - \zeta_i, \\ & \zeta_i \geq 0, i = 1, .., n. \end{array} \tag{1}$$

We are trying to maximize the margin, by minimizing $w^T w$, while applying a penalty when a sample is misclassified or within the margin boundary.

2.1.2 Regression

For regression in SVMs the SVR is used. The data to fit is composed of training vectors $x_i \in \mathbb{R}^p$, $i = 1, ..., n$, and a vector $y \in \mathbb{R}^n$ [4], The primal problem is now defined

$$\begin{array}{ll} \underset{w,b,\zeta,\zeta^*}{min} & \frac{1}{2}w^T w + C\sum_{i=1}^{n}(\zeta_i + \zeta_i^*) \\ s.t. & y_i - w^T\phi(x_i) + b \leq \varepsilon + \zeta_i, \\ & w^T\phi(x_i) + b - y_i \leq \varepsilon + \zeta_i^* \\ & \zeta_i\zeta_i^* \geq 0, i = 1, .., n. \end{array} \tag{2}$$

Here, we penalize samples whose prediction is at least ϵ away from their target. These samples penalize the objective ζ^* or ζ, depending on whether their predictions lie above or below the tube created by ϵ.

2.2. Color models and Color attributes

A color model is an abstract mathematical model describing the way colors can be represented. We will approach the RGB color model and the color attributes with the respective formulas derived from the RGB values.

2.2.1 RGB

It is an additive color space where colors are obtained by a linear combination of Red, Green, and Blue values [20]. The value of each of the three channels is correlated to the amount of light hitting the surface of the respective color. This means that different pictures of the same spot taken at different times of the day can have completely different RGB values.

2.2.2 Chroma and Hue

Hue is the "attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors: red, yellow, green, and blue, or to a combination of two of them"[8]. Chroma is the "colorfulness relative to the brightness of a similarly illuminated white"[8]. In order to better understand this definitions, there is a graphical representation of these concepts in Figure 1.

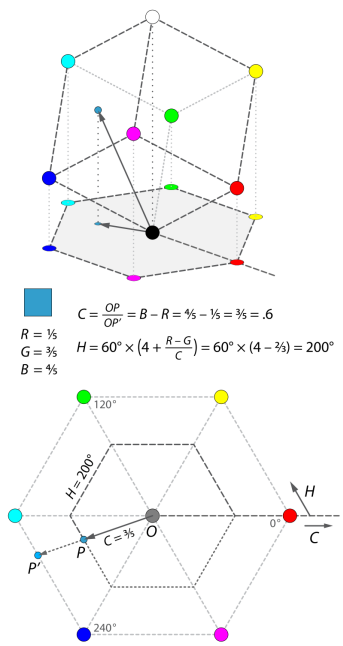

Figure 1: Both hue and chroma are defined based on the projection of the RGB cube onto a hexagon in the "chromaticity plane". Chroma is the relative size of the hexagon passing through a point, and hue is how far around that hexagon's edge the point lies. Assuming that the input is the RGB color [1/5,3/5, 4/5] where R,G,B $\in [0, 1]$, we would have hue $= 200°$ and chroma $= \frac{3}{5}$. Jacob Rus / CC BY-SA (https://creativecommons.org/licenses/by-sa/3.0)

The chroma is the proportion of the distance from the origin to the edge of the hexagon. This is the ratio of lengths OP/OP'. This ratio is the difference between the largest and smallest values among R, G, or B in a color [1]. Chroma can be computed as:

$$M = Max(R, G, B) \tag{3}$$
$$m = Min(R, G, B) \tag{4}$$
$$C = range(R, G, B) = M - m \tag{5}$$

The hue is the proportion of the distance around the edge of the hexagon which passes through the projected point[18], originally measured on the range [0, 1]. However, in some programs, this range is adjusted to [0, 360]. For points which project onto the origin in the chromatic plane, $R = G = B$ (grey colors) the hue is undefined. Hue can be computed as in equation 7.

$$H' = \begin{cases} \text{undefined}, & \text{if C = 0} \\ \frac{G-B}{C} \bmod 6, & \text{if M = R} \\ \frac{B-R}{C} + 2, & \text{if M = G} \\ \frac{R-G}{C} + 4, & \text{if M = B} \end{cases} \tag{6}$$
$$H = 60^{\text{o}} * H'x \tag{7}$$

2.2.3 Brightness and Lightness

Brightness is the "attribute of a visual sensation according to which an area appears to emit more or less light"[8]. Where Lightness is the "brightness relative to the brightness of a similarly illuminated white"[8]. In other words brightness is an absolute measure changing with the illumination where lightness is relative to a white point, being able to be constant with different illumination. When an image is darker, we expect to see a low value of lightness and when it is lighter we expect a higher value of lightness [14].

There are multiple ways to compute lightness, being the following the most commonly used:

- Intensity: Is the average of the three components assuming that the 3 channels contribute equally to our perceptions of lightness.

$$I = \frac{1}{3}(R + G + B) \tag{8}$$

- Value: Is the maximum of the three components assuming that only the highest value contributes to our perceptions of lightness.

$$V = M \tag{9}$$

- "Lightness": Is the average between the highest and lowest color components.

$$L = \frac{1}{2}(M + m) \tag{10}$$

In the case of a gray color (R=G=B) the lightness will be equal to R,G or B using any of the above methods. Figure 2 illustrates those methods.

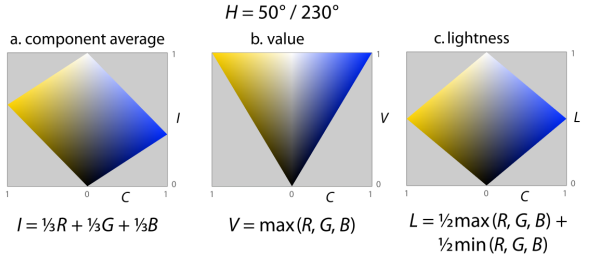

Figure 2: Three different possible "lightness" dimensions, plotted against chroma, for a pair of complementary hues. Jacob Rus / CC BY-SA (https://creativecommons.org/licenses/by-sa/3.0)

2.2.4 Saturation

As represented in Figure 2, there are some values that are greyed, corresponding to intervals that fall outside the RGB range. Saturation defines those intervals of values that cannot be used [21].

- Saturation in Intensity

$$S_I = \begin{cases} 0, & \text{if I = 0} \\ 1 - \frac{m}{I}, & \text{otherwise} \end{cases} \tag{11}$$

- Saturation in Value

$$S_V = \begin{cases} 0, & \text{if V = 0} \\ \frac{C}{V}, & \text{otherwise} \end{cases} \tag{12}$$

- Saturation in Lightness

$$S_L = \begin{cases} 0, & \text{if L = 0} \\ \frac{C}{1-|2L-1|}, & \text{otherwise} \end{cases} \tag{13}$$

With this information we can compute the HSI, HSV and HSL color models.

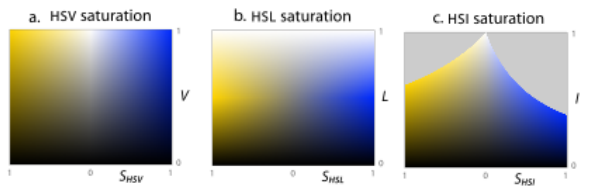

Figure 3: For H = 50º/230º, we can see how does saturation influence intensity, value or lightness. Jacob Rus / CC BY-SA (https://creativecommons.org/licenses/by-sa/3.0)

2.3. KNN

The KNN is a data classification algorithm that attempts to determine what group a data point is in by looking at the data points around it. An algorithm, looking at one point on a grid trying to determine if a point is in group A or B, looks at the

states of the points that are near it [10]. If the majority of the points are in group A, then it is likely that the data point in question will be A rather than B, and vice versa. This kind of classification can be used for other purposes such as data reduction where the KNN algorithm is used to separate the data into two sets: (i) the prototypes that are used for the classification decisions and (ii) the absorbed points that can be correctly classified by KNN using prototypes. The absorbed points can then be removed from the training set [11].

2.4. Vegetation Index

Vegetation indices are designed to maximize sensitivity to the vegetation characteristics while minimizing confounding factors such as soil background reflectance, directional, or atmospheric effects [13].

2.4.1 NDVI

The NDVI is calculated from these individual measurements as follows:

$$NDVI = \frac{NIR - RED}{NIR + RED} \tag{14}$$

Where Red and NIR stand for the spectral reflectance measurements acquired in the red (visible) and near-infrared regions, respectively. These spectral reflectances are themselves ratios of the reflected radiation over the incoming radiation in each spectral band individually, hence they take on values between 0.0 and 1.0 [17]. This range is usually associated to a range of colors like in figure 4.

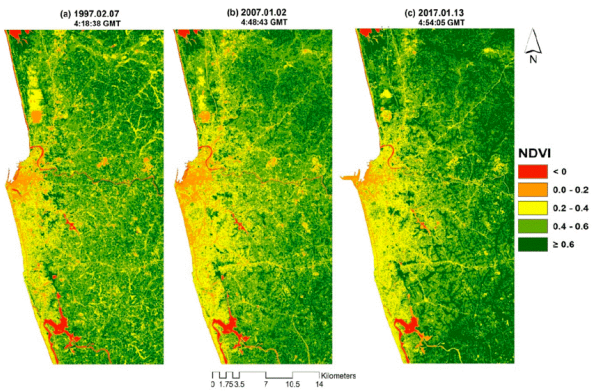

Figure 4: NDVI applied to satellite imagery [19]

Although the NDVI was created with the purpose to simply detect the presence of vegetation, it is commonly used in agriculture as an health index, where healthier plants are associated with greener pixels.

2.4.2 VARI

The VARI was developed to be used in later stage crops to estimate the fraction of vegetation [6], having the best results in crops of wheat and corn. It is obtained as:

$$VARI = \frac{GREEN - RED}{GREEN + RED - BLUE} \tag{15}$$

As in NDVI, a color code between red and green is associated to values.

2.4.3 TGI

The TGI is used to estimate leaf chlorophyl using visible-spectrum imagers [12]. TGI is defined as the area of the triangle. Its graphical computation can be observed in figure 5.

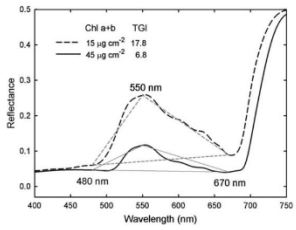

Figure 5: Graphical TGI computation [16]

This results in the equation 16

$$TGI = GREEN - 0.39 * RED - 0.69 * BLUE \tag{16}$$

As in the above method, a color range is defined between red and green in order to be comparable with other methods.

3. Approach

Our approach can be divided in 4 steps, as we show in Figure 6.

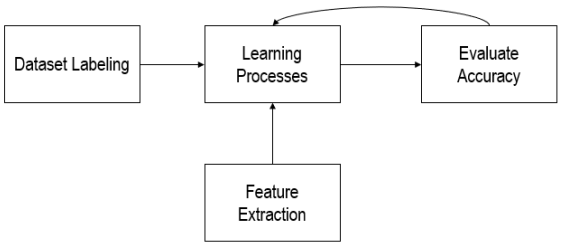

Figure 6: Overview of the proposed approach

The data labeling (Section 3.1) is where we will describe how each image is labelled and show the type of images that we will be focused on obtaining. The feature extraction (Section 3.2) is composed of several methods to obtain the relevant information from each image. The learning process (Section

3.3) defines the strategies that can be adopted to reach our goal. The last process will evaluate the results and compare it with other approaches.

3.1. Data Labeling

Visually, a green space is well irrigated when it is "green". However, this value of green can change due to light exposure since the RGB colorspace is influenced by the light exposure, as mentioned in 2.2. In order to make sure that we capture different values of "green" we will take pictures at different hours of the day and with different weather conditions, obtaining this different types of photos, like the ones present in Figure 7.

Figure 7: Example of captured images, having different levels of brightness and different quantity of brightness

After obtaining all the images in these different conditions we will proceed with the labeling. The user will have to choose 1 of 4 options: *definitely needs irrigation*, *needs irrigation*, *doesn't need irrigation*, *definitely doesn't need irrigation*. In each image there are several regions where we would like to give different classifications. So, as in [22], we will segment each green space picture in smaller rectangles, Figure 8, and label each of them.

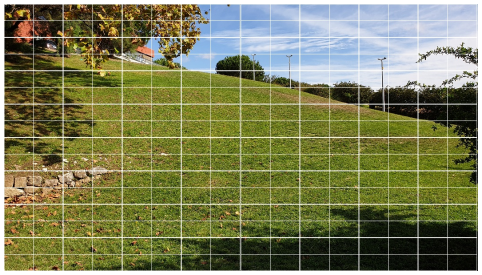

Figure 8: Image segmented in several rectangles

A picture will be divided in equally sized rectangles. If a rectangle contains background information, the user can skip the classification of that rectangle, which will be classified as background.

3.2. Feature Extraction

Given the labeled images we need to define how and what to extract from those images. In order to have a better idea of what we are dealing with, we have an example of the labeled rectangles in Figure 9

Figure 9: Example of segmented rectangles

In order to train our SVMs, an histogram of each rectangle and the corresponding label could be used. However, farmers and agronomy specialists are used to use vegetation indexes, as mentioned in Section 2.4, which associate a value to each pixel regardless of the surrounding pixels. With this in mind, we want to train our SVMs with pixels.

A user will first label images patches based on their dominant color. For instance he labels as *needs irrigation* because it has more brown than green. So, from that picture, we would like to extract the brown pixels. Or it is almost all green, being important to retain those green pixels and not confuse our system with the little white dots or black dots. For this, we need to know which pixels are safe to extract from each image.

3.2.1 Data Selection

In order to select the group of pixels that correspond to the defined patch label we will use KNN to pick groups of different colors. For that, we treat all the pixels in the image like points on a 3-dimensional space.

For patches labeled as *definitely doesn't need irrigation* and *definitely needs irrigation*, we will use KNN to compute two clusters. The biggest cluster will be assumed to have the biggest group of correctly labeled pixels. The euclidean distance between the color of two pixels in the selected color space will be used in order to agglomerate the clusters. In order to remove some outliers, only a percentage of that group will be selected, being removed the ones that have higher distances from their neighbours. For the labels *need irrigation* and *doesn't need irrigation*, we will use three clusters and choose a smaller percentage of pixels. This decision was made based because of higher uncertainty due to a mix of *irrigate* and *don't irrigate*.

Since we want to train our models using pixels, we don't need to save repeated pixels, discarding all the repeated pixels during this step.

To simplify the process of finding the best kernel to apply in our SVMs, we will test different color spaces, namely the HSV, HSL and HSI models. As explained in section 2.2, these models represent the color and the brightness better than using

the RGB model. Depending on the model used, a restriction in the brightness of the labeled rectangle is made. Note that the labeling of too bright or too dark rectangles will be discarded. For that, we will ignore rectangles that have average brightness higher than 220/255 or lower than 20/255.

3.3. Learning Processes

To accomplish our goal, two methods are used. In the first method, that we denote svr_simple, we will train an SVR in order to have a regression between *don't irrigate* and *irrigate*. We choose to apply a regression instead of a binary classification, because the vegetation indexes also return continuous values. In the second, two models are created. One for rectangles that are in the shadow(svr_shadow) and another for rectangles that are in the light(svr_light). With this we are trying to cope with images with significantly different values of brightness. In order to distinguish what is light and what is shadow, a SVC model is trained to classify each pixel as light or shadow (svc_brightness). Depending on this classification, the svr_shadow or the svr_light is used.

To train the models we will extract different color attributes, as in [5]. Those tests will use different testing datasets that are referred in Section 4. In the SVR models we have 4 labels. We will apply the values of 1 and 0 to *definitely doesn't need irrigate* and *definitely needs irrigation*, respectively. To the *doesn't need irrigation* and *need irrigation* we will apply 0.4 and 0.2. This interval was created based on the yellow intermediary color used in NDVI, as shown in Figure 4.

3.4. Evaluate Accuracy

In order to choose the best color attributes, our models need to be ranked. For that, evaluation metrics have to be specified. For the svc_brightness model we will use a confusion matrix, where the number of true light(TL), true shadow(TS), false light(FL) and false shadow(FL) is obtained, being the sum of these 4 numbers the total population (TP). With these values we can compute our accuracy as in equation 17.

$$ACC = \frac{TL + TS}{TP} \tag{17}$$

For the svr models we will use the coefficient of determination R^2 of the prediction. The coefficient R^2 is defined

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2}, \tag{18}$$

where SS_{RES} is the residual sum of squares and SS_{TOT} is the total sum of squares.

4. Experiments and Results

In this chapter the datasets that were used are explained, namely, pictures captured from a fixed camera(fixed images), pictures captured in different places(mixed images) and satellite images. Those datasets enabled several experiments, represented in Table 1

Table 1: Experiments performed.

	Training Set	Test Set
Exp. A	Mixed Images	Mixed Images
Exp. B	Mixed Images	Fixed Images
Exp. C	Fixed Images	Fixed Images
Exp. D	Mixed Images	Satellites Images

4.1. Datasets

In this section we will explain how the datasets where obtained in order to be used in the realized experiments

4.1.1 Mixed images

To form this dataset, we captured pictures from different green places at different times and days. To minimize the number of pictures we choose places that had, at the same time, light and shadow, as seen in Figure 10.

Figure 10: Example of an image with different light and shadow.

With this, we expect to have, in a single picture, multiple examples of groups of pixels when the image subdivision mentioned in Section 3.1 is applied.

4.1.2 Fixed images

This dataset contains photos captured over two months from the same green space in different days and times of the day. In Figure 11 we have some photos captured in the same day. Depending on the hour of the day, we have very different images of the same place. Although the status of the green space didn't change over the day, different colors during the day will let us extract much more information about what the user defines as *need irrigation* than just classifying different green spaces.

Figure 11: Examples of the pictures captured with a fixed camera on the same day but at different hours.

Figure 12 shows some photos captured at different days but at the same hour. With this, we hope to have a similar light exposure in each photo while getting different classifications from the user since it is normal to have different zones needing irrigation.

Figure 12: Examples of pictures captured with a fixed camera on different days but at the same hour.

Assuming that the camera is fixed and that the irrigation needs of the plants don't change over 24 hours, a user only needs to label one picture from one day and it will have labeled all the pictures of that day. Using this assumption, we can reduce our labeling time. However, we still need to remove from the dataset images that are not focused or are dazzled by the sun.

4.1.3 Satellite Images

This dataset is composed of RGB images present in the paper [16], where it's made a comparison between NDVI, TGI, VARI and RGB, as in Figure 13.

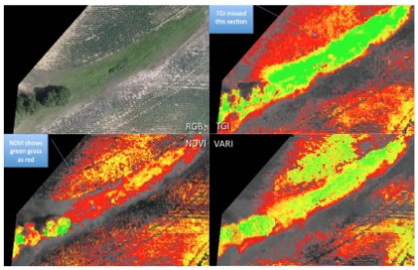

Figure 13: 1 RGB image represented by 3 diferent vegetation indexes: NDVI, TGI, VARI.

Since the previous paper concludes which vegetation index is the best for each case, and even mentions where the best of them fails, we used the author information to label each image. Using *definitely needs irrigation* for red and orange, *needs irrigation* for yellow, *doesn't need irrigation* for light green, and *definitely doesn't need irrigation* for green, in order to compare our models predictions with the vegetation indexes present in the paper.

4.2. Results

In this section we will provide the score that was obtained during the training of the SVMs models. For the svr_simple, svr_shadow and svr_light models we will compute the score using (18) and for the svc_brightness we will compute the score using (17). We will also compare one RGB image from each dataset to the numerical score obtained with our algorithm. In order be more graphical we will represent them using the following color code:

Values	Color Code
[0.0, 0.1[	#d73027
[0.1, 0.2[	#f46d43
[0.2, 0.3[	#fdae61
[0.3, 0.4[	#fee08b
[0.4, 0.5[	#ffffbf
[0.5, 0.6[	#d9ef8b
[0.6, 0.7[	#a6d96a
[0.7, 0.8[	#66bd63
[0.8, 0.9[	#1a9850
[0.9, 1.0]	#006837

Figure 14: Numerical intervals represented by each hexadecimal color code.

4.2.1 Tests on Mixed

Table 2 shows the results obtained for experiment A using 80% of the mixed dataset as training set and 20% as testing set or the light/shadow classification task. It is important to note that the images were chosen randomly to belong to each dataset. However, when we refer to this dataset we will always use the same images as used for this experiment.

	svr_simple(R^2)	svc_brightness(Acc)	svr_light(R^2)	svr_shadow(R^2)
HSL	0.56	0.94	0.83	0.76
HSV	0.60	0.92	0.89	0.73
HSI	0.61	0.93	0.87	0.70
RGB	0.49	0.47	0.62	0.54

Table 2: Results obtained during the experiment A

Using the best results from each model applied to the image in Figure 15, we obtained the shadow classification represented in Figure 16. Then, the SVR models were applied to the respective shadow and light, obtaining Figure 17.

Figure 15: Original image.

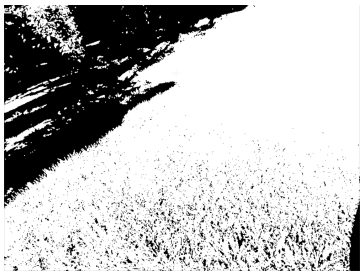

Figure 16: Image after the svr_brightness.

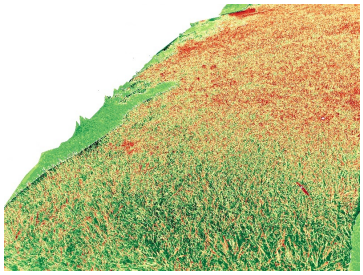

Figure 17: Image after the combination of svr_light and svr_shadow.

Comparing the real image(Figure 15) with the image classified as light and shadow (Figure 17), we can see that the two main shadows are represented correctly. Yet, it seems like there are some random dots in the middle down region of the image. In reality, if zoom in on the picture, we observe that those shadows exist and are made by the vegetation. In the image containing our irrigation decision we can see that due to the shadows it is not reacting as we wanted too. The pixels that had lower intensity ($I <= 20$) had a wrong classification, this was already expected since these values are too low to get a valid color.

4.2.2 Tests on Fixed

In Table 3 we have the results of experiment B, obtained using the training set previously defined in subsection 4.2.1(Mixed Dataset) and 20% of the Fixed dataset as testing set.

	svr_simple(R^2)	svc_brightness(Acc)	svr_light(R^2)	svr_shadow(R^2)
HSL	0.46	0.87	0.82	0.89
HSV	0.54	0.79	0.91	0.86
HSI	0.57	0.84	0.80	0.81
RGB	0.39	0.37	0.55	0.6

Table 3: Results obtained using pictures from different places as train dataset and pictures from fixed camera as test dataset

For experiment C we maintained the test set, but used the remaining 80% of the fixed dataset as training set, obtaining Table 4

	svr_simple(R^2)	svc_brightness(Acc)	svr_light(R^2)	svr_shadow(R^2)
HSL	0.67	0.96	0.85	0.95
HSV	0.69	0.95	0.92	0.93
HSI	0.69	0.96	0.91	0.92
RGB	0.61	0.72	0.77	0.74

Table 4: Results obtained using pictures from the fixed camera as train and test set

Comparing both tables we can see that the experiment C had in general better results for the different color attributes, however for the best color attribute of each model, the difference was much slighter. With this in mind, we choose the models obtained from the mixed test fixed to create Figures 15, 16 and 17

Figure 18: Original image

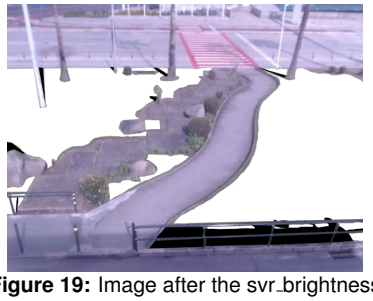

Figure 19: Image after the svr_brightness

Figure 20: Image after the combination of svr_light and svr_shadow

Our shadow classifier successfully detected the shadows in the region of interest. And this time, having higher intensity shadows, our irrigation decision algorithm could ignore the shadows and classify correctly the darker areas.

4.2.3 Tests on Satellite

In this dataset the great majority of the images didn't have any pixel that could be considered shadow and the images that had shadows, were formed by a combination of pixels with a very low intensity, being unclassifiable by our algorithm. So we tested directly the svr_light to the entire picture obtaining Table 5.

	svr_simple(R^2)	svr_light(R^2)
HSL	0.77	0.92
HSV	0.80	0.97
HSI	0.86	0.93
RGB	0.65	0.78

Table 5: Results obtained using pictures from different places as train and test set

Applying the svr_light to the figure 21 we obtained figure 22

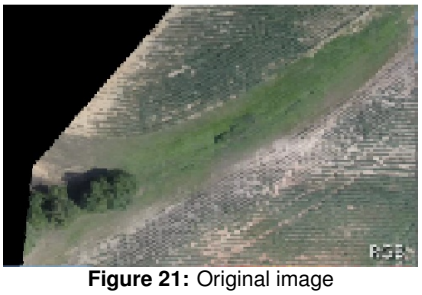

Figure 21: Original image

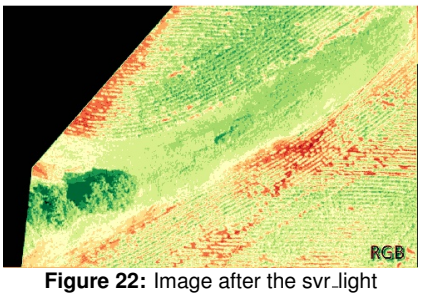

Figure 22: Image after the svr_light

As mentioned in Section 4.1.3 the main objective of testing with this dataset was to be able to observe the differences between our algorithm and the vegetation indexes present in the paper [16]. For that we changed the color codes of the previous figure in order to be easier to compare, obtaining the figure 23.

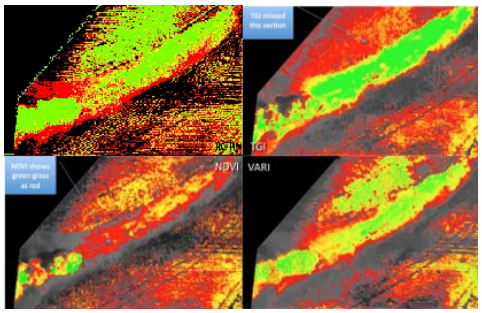

Figure 23: Image that has on the up left corner the result of our algorithm applied to Figure 21. The other images show the 3 vegetation indexes, mentioned in paper [16], that were applied to the same figure.

Comparing our algorithm with the NDVI and TGI vegetation indexes we can see that we overcome the failing areas, represent in Figure 23. However, the percentage of green and red is clearly different between our algorithm and the VARI vegetation index.

5. Conclusions

In this thesis we developed a method to automatize an irrigation decision based on a image with a segmented green space. This is done by using SVMs in order to generate models capable of computing a numerical score to each pixel of the green space. To do this, we had to overcome two main challenges: (i) labeling the images and(ii) understanding how the different color models could be exploited in order to extract the best information for the task.

To surpass the referred challenges, we segment each image in smaller images, and label each in four classes. Then, using clustering methods, we use only a percentage of the biggest cluster of pixels, assuming that the other are outliers.

With our approach, we were able to create a quick dataset with less than 100 photos, which had enough diversity to train our models to different images. Our best model had approximately 83 % of accuracy on the mixed set, taking in account the average of the classifications at light and shadow. When interpreting the images, we can clearly see that our algorithm improves with the increase of the camera altitude, being in its optimal state when working on satellite images. On lower altitudes (less than 3 meters) the probability of the present vegetation creating tiny shadows on the image will increase. These tiny shadows usually have a very

low intensity, creating classification errors in our algorithm, so it is not advised to use our algorithm on these altitudes.

Overall we, give a new possible approach to the creation of vegetation indexes and in the application of irrigation metrics in the computer vision area.

References

[1] H. Agata, A. Yamashita, and T. Kaneko. Chroma key using a checker pattern background. *IEICE Transactions*, 90-D:242–249, 01 2007.

[2] R. Allen, L. Pereira, D. Raes, and M. Smith. Fao irrigation and drainage paper no. 56. *Rome: Food and Agriculture Organization of the United Nations*, 56:26–40, 01 1998.

[3] S. Andriyas. Analysis of irrigation decision behavior and forecasting future irrigation decisions. *All Graduate Theses and Dissertations*, 1359, 2012.

[4] D. Basak, S. Pal, and D. Patranabis. Support vector regression. *Neural Information Processing – Letters and Reviews*, 11, 11 2007.

[5] M. Benazzouz, I. Baghli, and M. Chikh. Microscopic image segmentation based on pixel classification and dimensionality reduction. *International Journal of Imaging Systems and Technology*, 23, 03 2013.

[6] L. Eng, R. Ismail, W. Hashim, and A. Baharum. The use of vari, gli, and vigreen formulas in detecting vegetation in aerial images. *International Journal of Technology*, 10:1385, 11 2019.

[7] T. Evgeniou and M. Pontil. *Support Vector Machines: Theory and Applications*, volume 2049. Springer, Berlin, Heidelberg, 01 2001.

[8] M. D. Fairchild. *Color Appearance Models*. Addison-Wesley, 2005.

[9] D. Fam, E. Mosley, A. Mellick Lopes, L. Mathieson, J. Morison, and G. Connellan. Irrigation of urban green spaces: a review of the environmental, social and economic benefits. *CRC for Irrigation Futures Technical Report No. 04/08*, 04 2008.

[10] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer. Knn model-based approach in classification. In Z. Tari and D. C. Schmidt, editors, *Meersman R*. CoopIS, DOA, and ODBASE. OTM 2003. Lecture Notes in Computer Science, vol 2888. Springer, Berlin, Heidelberg, On The Move to Meaningful Internet Systems 2003, 2003.

[11] G. Guo, H. Wang, D. Bell, and Z. Liao. Similarity-based data reduction and classification. In *Monitoring, Security, and Rescue Techniques in Multiagent Systems*, pages 227–238, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[12] E. Hunt, C. Daughtry, J. Eitel, and D. Long. Remote sensing leaf chlorophyll content using a visible band index. *Agronomy Journal*, 103:1090, 07 2011.

[13] Z. Jiang, A. Huete, J. Chen, Y. Chen, J. Li, G. Yan, and Y. Zou. Analysis of ndvi and scaled difference vegetation index retrievals of vegetation fraction. *Remote Sensing of Environment*, 101:366–378, 04 2006.

[14] V. A. Kotkar and S. S. Gharde. *Image contrast enhancement by preserving brightness using global and local features*. Third International Conference on Computational Intelligence and Information Technology (CIIT 2013). IET, Institution of Engineering and Technology, 2013.

[15] H. Krupakar, A. Jayakumar, and D. Ganesh. A review of intelligent practices for irrigation prediction. *All Graduate Theses and Dissertations*, 1359, 2012.

[16] T. Mckinnon and P. Hoff. Comparing rgb-based vegetation indices with ndvi fordrone based agricultural sensing. *Agribotix.*, pages 1–8, 2017.

[17] G. Meeragandhi, S. Parthiban, and A. Nagaraj. Ndvi: Vegetation change detection using remote sensing and gis – a case study of vellore district. *3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)*, 03 2015.

[18] S. Naik and C. Murthy. Hue-preserving color image enhancement without gamut problem. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 12:1591–8, 02 2003.

[19] M. Ranagalage, R. C. Estoque, and Y. Murayama. An urban heat island study of the colombo metropolitan area, sri lanka, based on landsat data (1997-2017). *ISPRS Int. J. Geo Inf.*, 6:189, 2017.

[20] H. Saini, R. Sayal, R. Buyya, and G. Aliseri. *Innovations in Computer Science and Engineering: Proceedings of 7th ICICSE*, volume 103 of *Lecture Notes in Networks and Systems*. Springer Singapore, 2020.

[21] S. Wang, W. Cho, J. Jang, M. Abidi, and J. Paik. Contrast-dependent saturation adjustment for outdoor image enhancement. *Journal of the Optical Society of America A*, 34:7, 01 2017.

[22] J. Yu, S. Sharpe, A. Schumann, and N. Boyd. Deep learning for image-based weed detection in turfgrass. *European Journal of Agronomy*, 104:78–84, 03 2019.