PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA

A PATTERN RECOGNITION STRATEGY FOR VISUAL GRAPE BUNCH DETECTION IN VINEYARDS

RODRIGO ANDRÉS PÉREZ ZAVALA

Thesis submitted to the Office of Research and Graduate Studies in partial fulfillment of the requirements for the Degree of Master of Science in Engineering

Advisor:
MIGUEL A. TORRES TORRITI

Santiago de Chile, August 2017

PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA

A PATTERN RECOGNITION STRATEGY FOR VISUAL GRAPE BUNCH DETECTION IN VINEYARDS

RODRIGO ANDRÉS PÉREZ ZAVALA

Members of the Committee

MIGUEL ATTILIO TORRES TORRITI

MARCELO ALEJANDRO ARENAS SAAVEDRA

GIANCARLO TRONI PERALTA

FERNANDO ALFREDO AUAT CHEEIN

Thesis submitted to the Office of Research and Graduate Studies in partial fulfillment of the requirements for the Degree of Master of Science in Engineering

Santiago de Chile, August 2017

Gratefully to my family and friends

ACKNOWLEDGEMENTS

This work was supported by the National Commission for Science and Technology Research of Chile (CONICYT) under grants Fondecyt 1140575 and Basal Project FB0008. I express gratitude to the Center for Research and Innovation of Concha y Toro Winery for making available their data.

The author would like to thank Manuel Reis and Carlos Pereira from UTAD, Portugal, for sharing their dataset. Thanks to the Robotic and Automation Laboratory (RAL) of the Pontificia Universidad Catolica de Chile, specially to my advisor Miguel Torres for his time and guidance, and to Jorge Reyes for his support.

Special thanks to the Advanced Center for Electrical and Electronic Engineering (AC3E) of Universidad Tecnica Federico Santa Maria for their valuable remarks.

Finally, but most importantly, to my family and friends for their unconditional love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In addition to reducing the well-known problems in labor shortage, the use of robots for precision viticulture in the wine and table grape industry could help increase the efficiency in crop monitoring, fertilization, management, phenotyping and harvesting. However, one of the main challenges is the automated detection of grape clusters, problem that has slowed down the adoption of robotic units. This work presents a method that employs visible spectrum cameras for robust grape berries recognition and grape bunch detection on vineyards in different illumination and occlusion scenarios. A comparative study of different feature vector and support vector classifiers is presented for grape recognition. Three different gradient information features (Histogram of Oriented Gradients, Dense Scale Invariant Feature Transform and Daisy) and one texture descriptor (Local Binary Pattern) were tested, along with the study of Support Vector Machine (SVM) and one-class Support Vector Data Descriptor (SVDD) classifiers. A HOG+LBP feature fusion and SVM-RBF kernel classifier, show better results with an average accuracy of 96%, average precision of 99% and average recall of 93% in grape/non-grape image recognition. The proposed method for grape bunch detection in field images, uses a Fast Radial Symmetry Transform as salient point detector. Then, feature extraction and classification are computed in each salient point with a multiscale approach. Afterwards, a DBSCAN method defines cluster number and allows to create the non-convex envelop using Alpha Shape algorithm. Each cluster's spatial distribution and shape is analyzed for an improved cluster segmentation. Grape bunch detection and a comparison of berry and non-berry pixels was studied, using a hand labeled ground-truth. Four different datasets, with diverse illumination and acquisition protocol were tested, with an average precision of 83% and average recall of 82% in bunch detection and an average precision of 81% and average recall of 71% in area classification. Results show good performance with no need of special illumination, no color feature used which allows recognition for red and green grapes and a bunch detection scheme working in varying scenarios.

Keywords: grape cluster, grape detection, precision viticulture, histogram of oriented gradients, local binary pattern, support vector machine.

RESUMEN

Además de reducir los conocidos problemas de escasez de mano de obra, el uso de robots para la viticultura de precisión en la industria vitivinícola podría ayudar a aumentar la eficiencia en el control de los cultivos, la fertilización, la gestión, el fenotipado y la cosecha. Sin embargo, uno de los principales retos es la detección automatizada de racimos de uva, lo que ha frenado la adopción de unidades robóticas. Por lo tanto, este trabajo presenta un método que emplea cámaras de espectro visible para el reconocimiento de uvas y la detección de racimos de uva en terreno en diferentes escenarios de iluminación. Se presenta un estudio comparativo de diferentes descriptores y clasificadores para el reconocimiento de bayas, junto con un método para la detección del racimo de uva utilizando imágenes obtenidas en terreno. Se probaron tres descriptores de información de gradiente (*Histogram of Oriented Gradients*, *Dense Scale Invariant Feature Transform* y *Daisy*) y un descriptor de textura (*Local Binary Pattern*), junto con la comparación de clasificación entre *Support Vecto Machine* (SVM) y *Support Vector Data Descriptor* (SVDD) para el reconocimiento de bayas. La mezcla de descriptores HOG + LBP, junto al clasificador SVM, supera a los otros descriptores con una *accurac*y de 96%, *precision* de 99% y *recall* de 93% en clasificación de imágenes en categoría uva o no-uva. Se propone un método para la detección de racimo de uva usando imágenes, aplicando el método *Fast Radial Symmetry Transform* como detector de puntos claves. Luego, la extracción y clasificación de características tiene lugar en cada punto de interés en múltiples escalas. Posteriormente, un método DBSCAN define el número de clúster y permite crear la envolvente no convexo para cada racimo, utilizando la técnica de *Alpha Shapes*, para consecutivamente analizar la distribución espacial y separa racimos vecinos. Los resultados se compararon con imágenes clasificadas manualmente, obteniéndose 83% de *precisión* y 82% de *recall* en la detección de racimos de uvas. A su vez se analizó el área clasificada obteniéndose *accurac*y del 96%, *precision* 81% y un *recall* de 71%. Los resultados muestran un buen desempeño sin la necesidad de iluminación externa, sin utilizar información de color y detectando racimos cercanos en ambientes variables.

Palabras Claves: detección de uva, racimos de uva, vitivinicultura de precisión, histograma de gradientes orientados, máquina de soporte de vectores.

INTENTIONALLY BLANK PAGE

1 INTRODUCTION

1.1 Motivation

The industry of fresh table and wine grapes represent a major economic activity worldwide, being the fruit crop with highest value, with a market size of approximately 70 billion dollars; see Figure 1 (FAO-OIV, 2016). The International Organization of Vine and Wine report that during 2015, more than 75 million tonnes of grape were harvested in 7.1 million hectares in surface, shown in Figure 2 (OIV, 2015b). Almost 50% of production goes to alcohol (270 million hectoliters of wine), while 33% of the production is used in fresh grape while the rest is divided between juice and dried grapes. Chile is one of the major wine producers, with 5% of the global production (OIV, 2015a), and world's leading fresh table grape exporter (USDA, 2014), increasing both surface and production each year. Nevertheless, the world viticulture industry is facing several difficulties concerning qualified fieldworkers due to the increase in employment costs and labor shortage (Canadian Agricultural Human Resource Council, 2016; Quackenbush, 2017; Subercaseaux & Contreras, 2013; Timmins, 2009), affecting productivity, quality, harvesting on time and crop monitoring. Additionally, current agricultural tasks and practices are manually done, destructive, expensive and highly time consuming, as well as inaccurate and subjectively influenced or bias by workers (Grossetête et al., 2012; Nuske et al., 2014). Because of these problems, the agricultural sector needs to include new technology aimed at helping the producers in different aspects such as productivity and quality.

The use of technology in the grape and wine industry has been an important research topic due to the relevance of precision viticulture. This area of study looks for the monitoring of relevant variables and data in order to obtain repeatable process, real time information for rapid response and improving the grape quality, while also reducing the environmental impact and operational costs (Fernández, Montes, Salinas, Sarria, &

Armada, 2013). The data obtained by sensors may help growers make more informed decision and manage in a better way their field, handling the high variability in their crops.

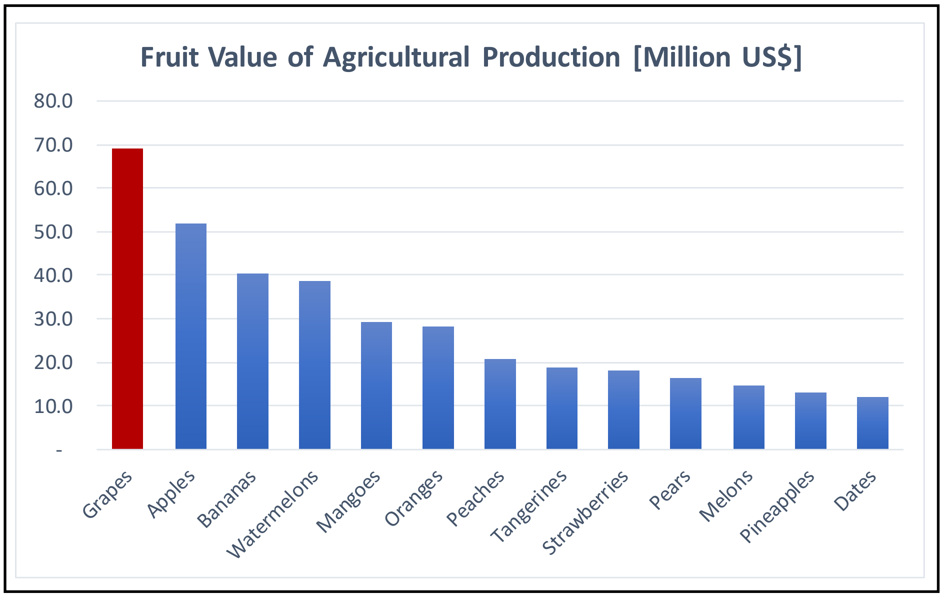


Figure 1. Top fruit value of agricultural production. Source: (FAO, 2017)

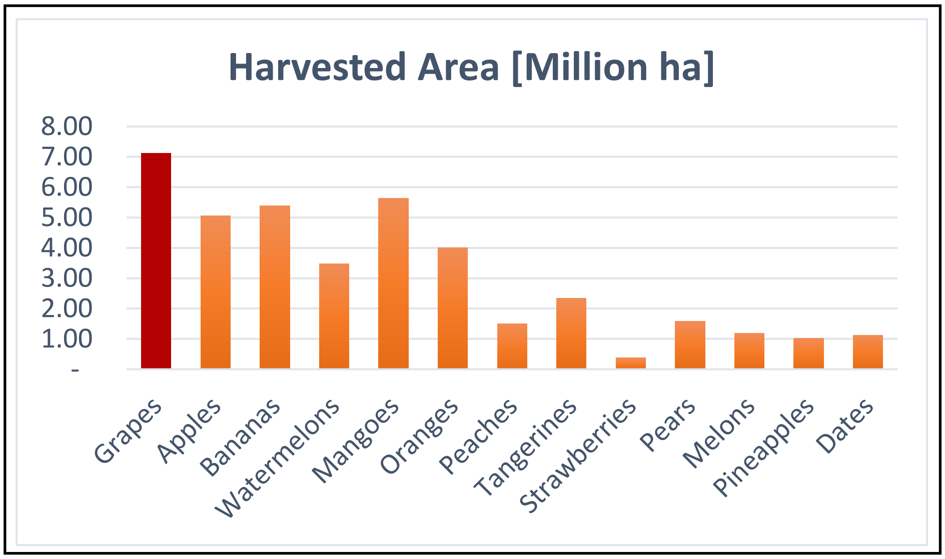


Figure 2. Top fruit harvested area. Source: (FAO, 2017)

The use of robotic systems for the automation of several grape and wine related processes on field is one of these technologies that may help increase the efficiency in crop monitoring, management, and assist in solving the existing problems of labor shortage and increasing labor costs. Robotic units in agricultural applications involve three main components: mobility algorithms and hardware (guidance and mapping), perception system, and end effector action stage (Auat Cheein & Carelli, 2013; Bachche, 2015). This research focuses on the perception or detection and recognition of grape bunches on field. For any robotic or automation process, the sensing stage is crucial for the correct performance of the unit. The detection of grape bunches in vineyards has been an important challenge that still has not been completely solved.

Related studies have shown that segmentation using color features is not robust for detecting white berries. Although color based features yield good results in red grape varieties (Liu, Whitty, & Cossell, 2015b), this is only applicable near harvesting, after the fruit has matured and changed color, limiting the number of tasks in which an autonomous unit could operate. Depending on light reflection for grape recognition (Grossetête et al., 2012; Nuske et al., 2014) may produce different results because of the varying pruinescence some grapes present (Diago et al., 2014) or different light reflections caused by weather, rain over the fruit, or spraying, generating multiple reflection points. To tackle problems caused by changes in natural illumination conditions some methods employ specialized illumination hardware to operate at nighttime (Nuske et al., 2014), which may imply higher operational costs. Skrabanek and Runarsson (2015) suggest the use of a sliding window that sweeps the entire image. The problem with this approach is the computational time needed to cover an entire high-resolution image, multiple times for every scale, with a small sliding window, and extracting the features and recognizing each window as grape or not. Therefore, studies show that there is still room for improvement in the detection of grape bunches using color images with natural illumination, which could be very useful to growers.

1.2 Potential Applications of Automatic Grape Detection

The automatic recognition of grapes and grape bunches could be employed to automate, manage and optimize current agricultural tasks such as harvesting (Luo, Tang, Zou, Ye, et al., 2016), spraying (Berenstein, Shahar, Shapiro, & Edan, 2010), grape and pixel counting for yield estimation models (Diago et al., 2014; Dunn & Martin, 2004; Grossetête et al., 2012; Liu, Marden, & Whitty, 2013; Liu et al., 2015b; Nuske, Achar, Bates, Narasimham, & Singh, 2011; Nuske, et al., 2014), evaluating grape quality, size and grapevine phenotyping (Cubero et al., 2014; Kicherer et al., 2015; Klodt, Herzog, Töpfer, & Cremers, 2015; Roscher et al., 2014), detecting disease in clusters, predicting harvest time, quantifying and standardizing crop thinning and basal leaf removal tasks, all by using non-destructive visible spectrum cameras. The current industry practices are discussed in the following sections, were state-of-the-art of machine vision could be employed in these tasks.

1.2.1 Harvesting

Nowadays grape harvesting is done using man power or by harvesting machines that strike the vine; see Figure 3. The problem with manual harvesting is the large number of workers and time needed, in addition to the rising costs of labor and the fact that many seasonal fieldworkers are moving to longer term position in other industries. Thus, farmers must face frequent problems to harvest on time.

On the other hand, machine harvesting is not suitable for all type of grapes, especially harvesting of table grapes and champagne grapes were the fruit damage causes accelerated oxidation, affecting the final product's quality (Chamelat et al., 2006; Reis et al., 2012). Machine harvesting also collects excessive unwanted materials that wine makers must remove before the crush and press stage. Moreover, machine harvesting is not able to inspect the quality of the grape, ripeness or damage, where sorting is a key activity in the production of champagne and some premium wines.

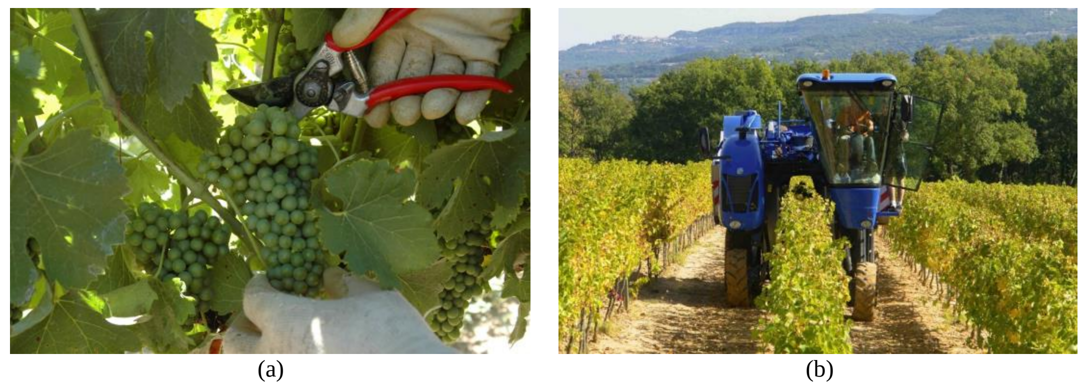


Figure 3. Example of methods of grape harvesting. (a) Manual Harvesting. (b) Mechanical Harvesting Source: (Benito Saez, 2010; Manfull, 2012)

1.2.2 Spraying

Spraying of nutrients, hormones, pesticides and fertilizers is an important task that must be done during the different growth stages of the plant, and must be done on time to protect the crop from pest, diseases and to feed them with the necessary nutrients for ensuring high quality grape. Nowadays, mechanical spraying is done in a homogenous way along all the rows, a method that generates a high amount of wasted resources, which can be reduced by targeted spraying bunches of grapes or the canopy as needed (Berenstein et al., 2010). Alternatively, hand spraying is done by fieldworkers carrying on their backs the equipment. Hand spraying is highly time consuming and less efficient. In addition, hand spraying is dangerous for the fieldworkers because of possible contact with toxic chemical agents.

1.2.3 Yield Estimation

Accurate and early yield estimation is a relevant task in agriculture because it is the input for other logistic operations and production processes, such as harvest time, storage, transport and sales (Grossetête et al., 2012; Liu et al., 2013, 2015b). Accurate yield estimation by automated means is still not a fully solved problem, which has a relevant economic impact on most parts of the value chain (Dunn & Martin, 2004; Herrero-Huerta, González-Aguilera, Rodriguez-Gonzalvez, & Hernández-López, 2015). Nowadays counting is done manually for estimation and forecast, counting grapes on selected grape bunches of random rows, and then extrapolating the results to the whole field (Nuske et al., 2011). The accuracy of the results depends on the number of samples taken in the field, which are usually insufficient to precisely quantify the variation that takes place on the vineyard. Growers also use historical data to obtain a more accurate yield prediction, taking into account such variables as temperature, watering, and historical production. Despite the additional variables considered, the approach is not completely accurate, and is also time consuming and labor demanding in data acquisition (Liu et al., 2013).

1.2.4 Crop Thinning and Leaf Removal

Other important practices include crop thinning, which is the removal of grape bunches from the grapevine in order to improve or control grape quality and yield (Creasy & Creasy, 2009; Vance, Reeve, & Skinkis, 2013). Thinning allows growers to control the leaf area to fruit ratio, which is important to ensure that grapes ripe with a better distribution of sugar contents along with giving open space and light to the clusters to develop properly. In addition, crop thinning allows to increase the grape size, phenolic and aroma compounds, variables essential for grape quality. Nowadays this labor is done analyzing manually some rows and trying to repeat the results to the entire vineyard. The problem is that no easy measurement can be done through the whole field, and since every vineyard is different due to weather, grape variety, water availability, sun light orientation and *terroir*, no standard rule applies.

In order to control grape quality, leaf removal is also a labor done by growers. The removal of leaf near the grape cluster region helps the vine increase air circulation, better spraying, pest control and expose fruit to sunlight (Pence & Grieshop, 1991; Vance et al., 2013). The amount of leaf removal depends on the vineyard practices and climate, but no standard measure is used, only guided by the grape grower opinion and experience, making this process difficult to reproduce through the entire field.

1.3 Objectives

The main objective of this research is to develop an algorithm capable of detecting grape bunches using lateral digital images of the vine and pattern recognition techniques under various occlusion, different foliar density and varying illumination. To this end, a SVM-based classifier will be trained and tested using different datasets and feature descriptors in order to determine the configuration that yields highest detection rates with low misdetections and low false positives.

1.4 Contributions

The contributions of this research can be summarized in:

i. An experimental and comparative study of grape recognition feature descriptors with no use of color information.

ii. A novel approach for detecting grape bunches in outdoor images, tested in four different vineyard image datasets with varying levels of occlusion and foliar density for white and red grapes, capable of recognizing neighbor grape clusters.

1.5 Thesis Outline

Chapter 2 presents a review of the literature and discusses the related work. Chapter 3 introduces the theoretical background. Chapter 4 presents the proposed method for grape bunch detection explaining the features, algorithms and classifiers used. Chapter 5 states the experimental methodology along with the results and analysis. Finally, in Chapter 6 the conclusions and future work are state

2 RELATED WORKS

Most of the related research involving machine vision applications in viticulture is aimed at solving the problem of yield estimation. Color features in different color spaces (RGB, HSV, L*a*b*) have been used to detect red grapes and white grapes. Recognition using color descriptors typically shows better results for red grape near harvest than white grape because of the color similarity white grapes have with leaves and background.

Color thresholding of the RGB channels is employed in Dunn & Martin (2004) for yield estimation of Cabernet Sauvignon grapes, correlating the pixel area with kilograms of grape. The authors use a white screen placed behind the canopy to help in the segmentation. Cabernet Sauvignon grapes, stems, leaves and background pixels are classified using color and multispectral images from a natural environment with K-means clustering in Fernández et al.,(2013).

Classification of different parts of the Tempranillo grapes is presented in Diago et al. (2012) using RGB features and the Mahalanobis distance for classification. In this research, different classes including grape, wood, background and leaves were classified from an image with a white screen as background. The user must provide reference input pixels for each class. A similar approach for yield estimation is presented in Font et al. (2015) using artificial illumination, high resolution images and a Bayesian classifier. Red and white grapes are detected in (Reis et al., 2012) using color limits and morphological image operations.

A more sophisticated method is proposed in Chamelat et al., (2006) where HSV channel information along with Zernike moments are used to describe grape shapes and train a Support Vector Machine classifier in order to identify grapes in images using a sliding windows. The purpose of automatic detecting grapes in this research is to implement a robotic harvesting system.

The novel approach by Nuske *et al.* (2014) for counting grapes and yield estimation, introduces a calibration model that relates number of individual grapes with kilograms of harvested crop. The proposed approach extracts interest points in images acquired in the fields and compares performance of Invariant Maximal Detector (IMD) and Fast Radial Symmetry Transform (FRST) (Loy & Zelinsky, 2003) as key-point detectors. Patches around the points of interest are analyzed extracting color features in RGB and L*a*b* color space, Gabor filter responses in 6 orientations and 4 scales, along with SIFT and FREAK descriptors. After the extraction of the features is done, each patch is classified as grape or not, using a previously trained KD-forest algorithm. The grape key-points recognition metrics varies for each dataset due to differences in image illumination and variability of the field. Key-point detection has recall rates of 61% for IMD and 79% for FRST. The SIFT descriptor yields the highest detection rates across all the datasets. In Grossetête et al. (2012) a similar analysis is done looking for pixels with light peaks in order to count grapes and have an early yield estimation. The novelty of that research is the use of a smartphone for the image processing.

Liu *et al* (2013) analyze the relationship between several variables such as pixel area, perimeter, grape number and size, and their correlation with the actual weight of the grape bunch, in order to determine which metric has the best yield prediction. Later the same authors proposed a method for detecting bunches of red grapes employing a color segmentation in the HSV space (Liu et al., 2015b). Finally, a feature vector containing information of bunch location, texture and bounding box pixel distribution is passed to a previously trained SVM classifier. A similar study proposed by Luo, Tang, Zou, Wang, & Zhang (2016) present a grape cluster detection based in color and an AdaBoost framework for classification. They present an adjoining cluster separator based on the calculation of the barycenter of the binary detection mask.

Vineyard yield estimation by 3D bunch reconstruction is done in Herrero-Huerta et al., (2015), considering the volume, mass and number of grapes in each bunch. The approach employs images from five different angles to reconstruct the grape bunch. A similar study is presented in Liu, Whitty, & Cossell, (2015a), where the reconstruction of a grape bunch is done using only one image, under laboratory conditions. Here color segmentation is used to identify the bunch from the background and circular Hough Transform is used to identify individual grapes and start creating the 3D model. In Ivorra, Sánchez, Camarasa, Diago, & Tardaguila, (2015) a 3D surface is obtained using stereo cameras under ideal conditions in order to assess grape bunch components related to yield, mainly compactness (which affects the quality of the grapes that do not receive enough sunlight in the interior of the bunch).

Škrabánek and Runarsson (2015) designed a white grape recognition method using a sliding window algorithm on photos of vineyard rows. Four classification strategies combining linear SVM or RBF SVM with HOG or pixel intensities. Results show that HOG feature and SVM-RBF combination show better results.

Other studies involving grapevine detection include grape recognition and measuring fruit diameter, in images of grapevine, for precision phenotyping for future plant breeding. Roscher, et al. (2014) use the circular Hough transform for detecting points of interest and then extracting features around the center that include color, HOG and gist, which are later passed to a conditional random field classifier. Ripeness and sizing are important variables when monitoring the grapevine. For measuring these variables vision cameras are used in some studies (Rodriguez-Pulido et al., 2012; Zeng, Liu, Miao, Fei, & Wang, 2008).

Vision algorithms that detect grape and foliage are presented in (Berenstein et al., 2010) for selective spraying of hormones and pesticides. Three grape bunch detection algorithms are evaluated, examining high density edge areas, color channels in RGB and HSV and comparing an individual grape mask using a 2D convolution response. Discrimination of plant elements is done in (Correa, Valero, & Barreiro, 2012; Fernández et al., 2013) using color and clustering algorithms.

The above literature review shows that the critical and common step between the methods discussed is the correct segmentation and detection of the grape bunches, as this is the basis for different applications. This research contributes in the study of different feature descriptors and two types of support vector classifiers for grape recognition, proposing a feature fusion of HOG+LBP with a SVM-RBF classifier.

Several studies have used color information to initially segment the potential grape bunch area, showing better results in red grapes after veraison. Segmenting white grapes using color information still shows results correlating with leaves. This study presents a color independent approach which results in a robust approach independent of illumination or external factors possibly seen in a variable scenario as a vineyard, tested in both red and white grapes.

Subsequently a proposed method for robust grape bunch detection under different illumination and foliar conditions is presented, capable of detecting bunches and separating neighbor grape bunches that are closely together. The proposed method does not need special illumination hardware, capable of working during daylight. This novel method may be used for different application on vineyards, giving growers a powerful tool for standardizing their tasks.

3 PRELIMINARY NOTIONS

3.1 Digital Images and Color Spaces

A digital image I is a quantized and sampled numerical representation of the light emitted by a scene. An image is defined as a mapping that associates coordinates (i,j) of a discrete optical plane Ω with discrete intensity values corresponding to the quantization of light photons received by the detecting element.

$$I\colon (i,j) \,\epsilon\, \Omega \,\subset\, \mathbb{N}^2 \,\rightarrow\, I(i,j) \,\epsilon\, L \,\subset\, \mathbb{Z}_+ \quad ,$$

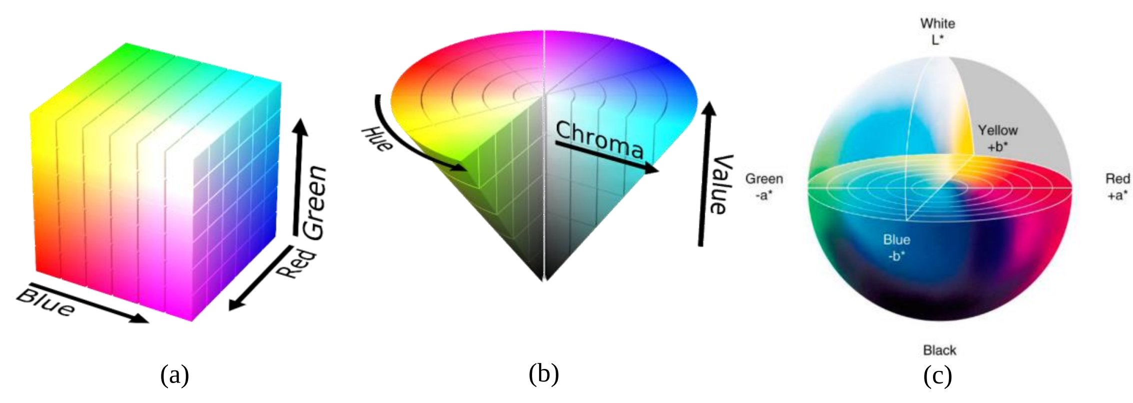
The image domain Ω is a set of $M \cdot N$ coordinates, where M is the number of rows and N is the number of columns of the imaging sensor. The intersection of any row with a column, $\Omega(i,j)$, is denominated a picture element, or better known as pixel. The pixel intensity values lie in a range $L_{range} = [0,\ 2^b - 1]$, where $b \,\epsilon\, \mathbb{N}$ is the number of bits employed to quantize the intensity into 2^b levels.

$$\Omega = \{(i,j) \,\epsilon\, \mathbb{N}^2 \colon 1 \leq i \leq M, 1 \leq j \leq N\}$$

Typically, $b = 8$ bits, representing 256 different levels of intensity. Color images are formed by measuring the light intensity in three wavelengths corresponding to the three primary colors, red, green and blue. These images are formed using band-pass filters that only allows one of the primary colors component to be sampled by the light detection element. Commonly the so-called Bayer filter arrangement is employed with a single detector array to obtain an $MxNx3$ image representation, containing the intensity values in the three primary color channels, hence the name RGB image. The combination of the three channels and the 2^b levels, results in $2^{3 \cdot b}$. For $b = 8$ bits, this corresponds to more than 16.7 million colors. RGB images is just one of several color spaces that may be used to represent digital images; see Figure 4a.

Another commonly used color space is the Hue, Saturation and Value or HSV model. This representation originates from psychophysical theories of color perception by humans (Pratt, 2007) and thus is more intuitive in the domain of the arts and graphic design, because hue is related to color wavelength, saturation is related to color intensity relative to a colorless light (white-gray-black) and value is associated to the overall intensity, therefore is easier to interpret than a color formed by the RGB model. The HSV space is defined by a non-linear model in terms of a cylindrical coordinate system; see Figure 4b. Hue or color component, is represented as the angular coordinate from [0, 360], the Saturation or Chroma is defined by the radial distance from [0, 1] representing the pureness of the color. Value is defined as the height of the coordinate system, representing the brightness ranging from [0, 1]. This model has the advantage that a color region can be easily defined with only the Hue parameter regardless of the color brightness or purity levels.

A third color model usually used is the CIELab or L*a*b* color space representation. This model can be visualized as a sphere, where L* represents the lightness raging from [0, 100]. The a* component represents the green/red plane and b* moves from blue/yellow axis both variables raging from [-128, +127]; see figure 4c.



(a) (b) (c)

Figure 4. Example of common color spaces. (a) Red, Green, Blue (RGB). (b) Hue, Saturation, Value (HSV). (c) L*a*b*. Source:(International Virag, 2016; SharkD, 2010, 2015)

3.2 Histogram Equalization

Often luminous intensity information needs to be normalized to exploit the full range of values and make posterior analysis independent of scene illumination. Some processes such as edge extraction, shape and texture analysis rely on grayscale image with adequate contrast and do not need color information. One popular technique to improve image contrast is to perform an equalization of the intensity histogram to ensure all intensity levels are equally represented and thus reduce the possibility that intensity values in an image are concentrated in a small portion of the range L. The equalization of the intensity histogram is achieved by linearizing the cumulative distribution function (CDF) of the original image. Given an image I, the probability density function for gray level $i,$ with i in the range of $0 \leq i < L$, is defined as

$$h_I[i] = \frac{n_i}{\sum_0^{L-1} n_i} \qquad \forall\, i \in L, \tag{3.2}$$

with n_i the number of occurrences of pixel with gray level i. The CDF is then defined as

$$H_I[j] = \sum_{i=0}^{j} h_I[i] \qquad \forall\, j \in L, \tag{3.3}$$

which ranges in the values from 0 to 1. To linearize the CDF, i.e. obtain a uniform probability density function for the luminous intensity of the transformed image, a pixel with intensity j_{in} has to be mapped to a pixel with intensity j_{out} given by (Pratt, 2007):

$$j_{out} = (j_{max} - j_{min}) \cdot H_I[j_{in}] + j_{min} \ , \tag{3.4}$$

where j_{min} and j_{max} are the minimum and maximum values of the intensity range. An example of histogram equalization is shown in Figure 5.

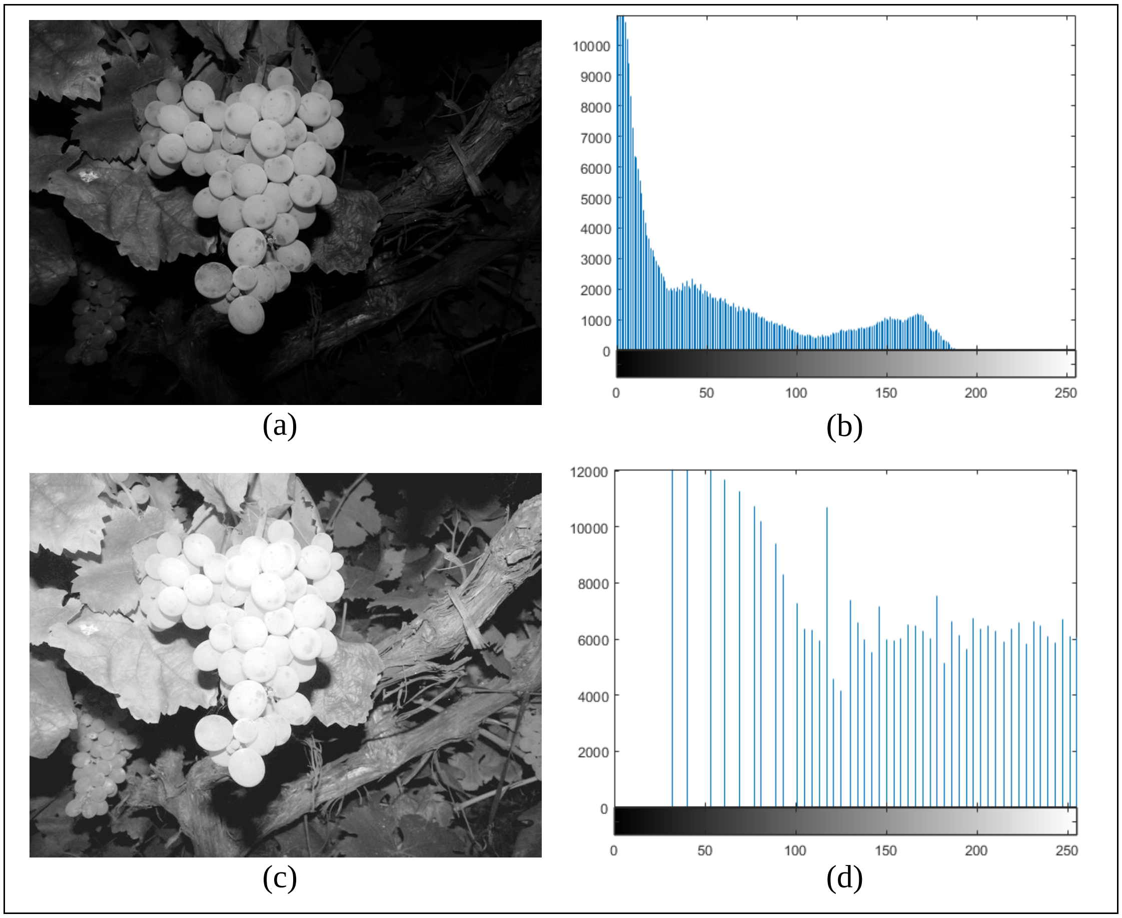


Figure 5. Example of histogram equalization for contrast enchantment. (a) Original grayscale image Portugal Dataset. (b) Original Image Histogram. (c) Image after Histrogram Equalization. (d) Output of Histogram Equalization.

3.3 Pattern Recognition

Pattern recognition are the set of tools and algorithms for classification of feature descriptors computed from measurements into groups or classes that share some characteristics. Two key aspects of the pattern recognitions process will be discussed next. First the construction of a vector of features which jointly constitute a descriptor of a given class of object. The second important part is the construction of a model that defines the rules for categorizing or labeling the features data.

3.3.1 Feature Vectors

A feature vector is a vector $\boldsymbol{x_p} = [x_1, \dots, x_n]^T$ containing scalar values $x_i,\ i = 1, .., n$, which represent a measure of some property of an image location or region $p = (i, j)$ such as color, texture or shape among others, which can be computed for objects of a given class. These vector is a numerical representation of an object or class, which is used in the learning and prediction/classification stage of a pattern recognition strategy.

3.3.2 Classifiers

Let $\mathcal{C} = \{c_1, \dots, c_m\}$ denote a set of class labels of object categories. Given a feature vector $\boldsymbol{x}_p$ the classification problem consists in finding a function or process $g: \boldsymbol{x_p} \rightarrow c_p$ that returns a correct class label $c_p \,\epsilon\, \mathcal{C}$ of the image element of region p. In board terms, the pattern recognition process involves estimating density functions from samples of descriptor vectors $\boldsymbol{x_{i,c}}$ for the different classes $c \,\epsilon\, \mathcal{C}$ in a high-dimensional space (typically $n \gg m$) and obtaining a so-called discriminant function that divides the space into categories or classes. The analysis of the distribution of $\mathcal{X}_{\boldsymbol{i,c}}$ for each category $c \,\epsilon\, \mathcal{C}$ and finding a discriminant function to design the classifier is called learning or training; see (Duda, Hart, & Stork, 2001; Fukunaga, 1990)

Supervised learning is a type of pattern recognition system that has been trained from labeled data, explicitly been tough through experience. Unsupervised learning, also known as clustering, is a machine learning system that has the objective of finding similarities to form and label groups or clusters, given a set of feature vectors without their corresponding label or class. Since there is no label data, there is no training stage.

4 PROPOSED METHOD

The proposed grape bunch detection approach is shown in the block diagram of Figure 6. The first step consists in preprocessing the images to convert the RGB color image into grayscale images and equalizing the histogram of intensity values, as explained in section 3.2, to improve the extraction of salient-points and grape recognition.

The equalized images are down-sampled to speed up the detection of salient points on grape bunches. Image areas around salient points are analyzed at larger resolution scales. The detection of salient-points is implemented using the Fast Radial Symmetry Transform (FRST) due to the results shown in previous studies for finding circular shaped points of interest with a less computational time than the Circular Hough Transform.

Descriptor vector containing HOG and LBP features are computed at different resolution scales centered at the salient point and evaluated with a SVM-RBF classifier to sort them as grape or non-grape. The cloud of points identified as grapes are then grouped using the DBSCAN clustering algorithm, allowing to number the grape bunches. This method, contrary to the popular k-means, does not need as input the number of clusters to find, allowing any number of grape bunch on one image.

The boundary of a grape bunch is finally obtained using the alpha-shape algorithm for non-convex shapes. This method allows to detect the non-convex shapes of the grape bunch boundary, which will be later analyzed to correct the grape bunch counting and detection. Finally, the grape bunch’s shape and spatial distribution is analyzed in order to improve the grape bunch detection, by computing a neighboring bunch separation method. The next sections discuss in further detail the salient point detection, feature computations and classification, grape clustering, bunch detection and grape mask creation using the alpha-shape method.

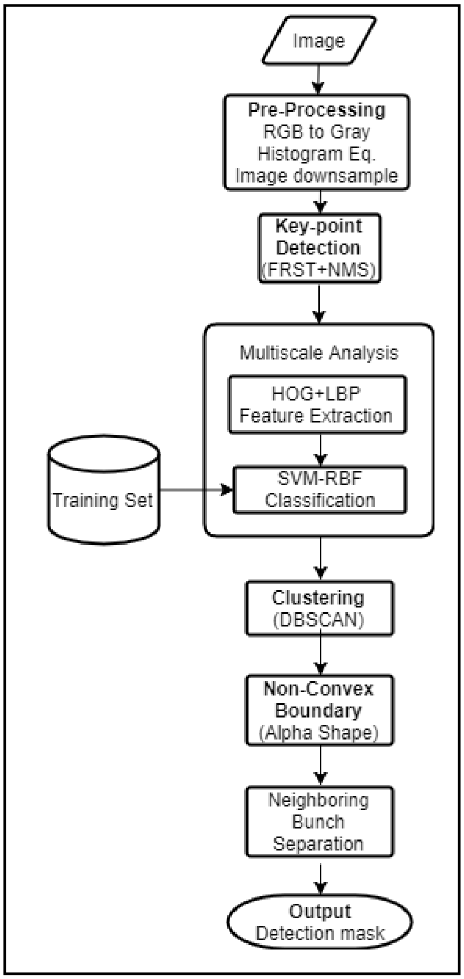


Figure 6. Proposed Method Framework

4.1 Salient Point Detection

Several authors have proposed the use of symmetric measures for the detection of points of interest (Nuske et al., 2014; Rahman & Hellicar, 2014; Roscher et al., 2014). Here the Fast Radial Symmetry Transform (FRST) proposed in (Loy & Zelinsky, 2003) and employed by (S. Nuske et al., 2014) was chosen for salient point detection. However unlike (S. Nuske et al., 2014) here FRST is applied on a down-sampled image to gain computational speed. This transform detects radially symmetric salient points, which are good grape candidates. FRST gives each pixel a score for radial symmetry at a distance $r_i \epsilon \mathcal{R} = \{r_1, .., r_n\}$, where $\mathcal{R}$ is a set of possible radius values. The score in each pixel is computed by first using the gradients information at a radius r from the center pixel of an image I of size MxN. At each pixel $p = (i,j),\ 0 \leq i \leq M, 0 \leq j \leq N,$ of the gradient image $g(p) = \nabla I(p)$, a positive affected pixel, p_+, and negative affected pixel, p_-, at a distance r from p is defined; see equation (4.1) and figure 7.

$$p_\pm(p) = p \pm nint\left(\frac{g(p)}{\|g(p)\|} r\right) \tag{4.1}$$

Using this information, the orientation image matrix, O_r, and magnitude image matrix, M_r, at a radius r, are created using a recursion loop, where for each pixel $p(i,j)$: $1 \leq n \leq (M \cdot N)$, the related $p_+(p_-)$ is increased (decreased) by 1 and $\|g(p)\|$, throughout all pixels; see equation (4.2) and (4.3) (Loy & Zelinsky, 2003; Ni, Singh, & Bahlmann, 2003). Starting values of the gradient image matrixs are defined as $O_r^0 = \vec{0}$, $M_r^0 = \vec{0}$.

$$O_r^n(p_\pm(p)) = O_r^{n-1}(p_\pm(p)) \pm 1 \tag{4.2}$$

$$M_r^n(p_\pm(p)) = M_r^{n-1}(p_\pm(p)) \pm \|g(p)\| \tag{4.3}$$

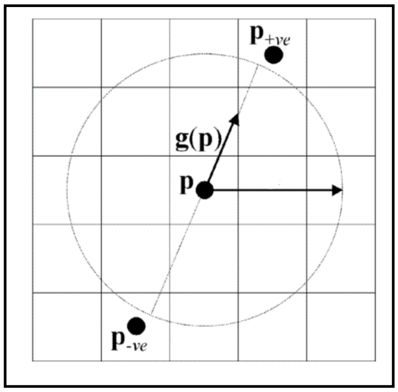


Figure 7. FRST affected pixels. Source (Loy & Zelinsky, 2003)

Using the orientation image matrix O_r and magnitude image matrix M_r along with two parameters α and κ_r, radial strictness and a scaling factor for distance r respectively, the matrix F_r is defined; see equation (4.4). Finally, the symmetry transform matrix at radius r, S_r is obtained by convoluting F_r with a two-dimensional Gaussian function A_r. See equations (4.5).

$$F_r(p) = \frac{M_r(p)}{\kappa_r}\left(\frac{abs(O_r(p))}{\kappa_r}\right)^{\alpha} \tag{4.4}$$

$$S_r(p) = F_r(p) * A_r \tag{4.5}$$

The absolute value of S_r will give a transform image with each pixel a symmetry score, independently of its orientation, pixel symmetry from dark to light or vice versa. To consider all the radius $r_i \in \mathcal{R}$, the final transform image S is defined as the average of all S_r; see equation (4.6).

$$S = \frac{1}{|\mathcal{R}|}\sum_{r\,\epsilon\,\mathcal{R}} S_r \tag{4.6}$$

In order to keep the relevant scores and then define the possible grape center, a non-maximal suppression algorithm is run through the score matrix S. An application example of FRST is shown in Figure 8, which represents a grape in Figure 8 (a), the score image matrix S in Figure 8 (b), and the maximum score selected after non-maximal suppression in Figure 8 (c). In the NMS stage, the image S with the score of the FRST is analyzed using a sliding window to find the local maximum value in each window, that is above a minimum threshold. Pixels with scores below the threshold are set to zero even if the score is a local maximum value within the window. Therefore, the role of this step is to carry out the so called non-maximal suppression to remove spurious information from the image and keep the best salient points that potentially match the center of the grapes. The size of the sliding window is set to the half of the size of the average grape size found in the image given the camera parameters (focal length, resolution, working distance).

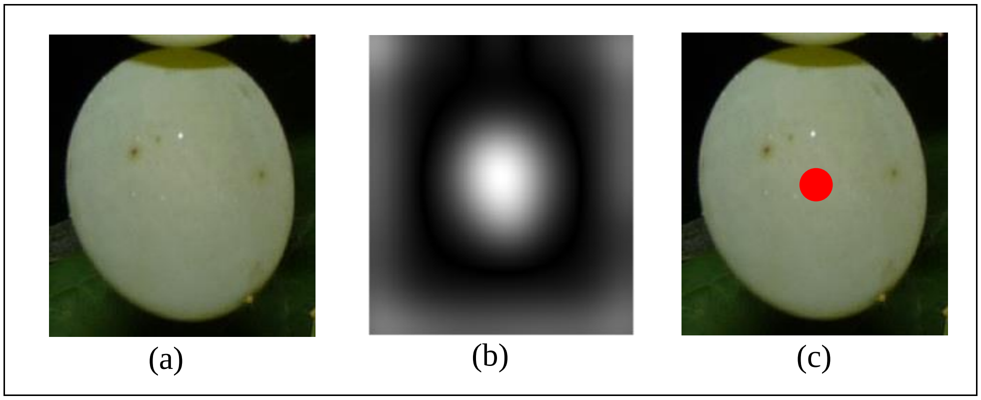


Figure 8. Fast Radial Symmetry Transform and Non-maximal suppression example. (a) Original Image. (b) FRST output. (c)NMS output. Source of Original Grape: Cropped from (Reis et al., 2012)

4.2 Grape Recognition (Feature and Classifier Selection)

Once salient points have been detected it is necessary to remove many points that are not part of grape centers. Thus, the neighborhood around each salient point is analyzed using a shape and texture descriptor built from HOG and LBP features together with a SVM in order to classify the descriptor as grape or non-grape. The implementation of the SVM requires a training dataset containing pairs of feature vectors and labels in a two-class set (grape and non-grape), $\mathcal{T} = \{(x_1, y_1), \dots (x_m, y_m)\}$. The training dataset was created by computing the feature vectors on hand-labeled regions of the grapevine.

The descriptors considered for the grape recognition process are the Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), DAISY and Dense Scale of Invariant Feature Transforms (DSIFT). The relevant aspects of these descriptors are discussed next. The performance of two classifiers, a Support Vector Machines (SVM) classifier and a Support Vector Data Descriptor (SVDD), using the different feature vectors is presented in the next chapter to identify the most accurate and reliable grape detector.

A multiscale analysis is used, which allows the detection of several sizes of berries by analyzing the patches around salient points in different scales. The multiscale analysis consists of creating a dense image pyramid, which means resizing and blurring the image several times. To keep the edges well defined, only the subsample of the image in a range considering the size of possible grapes is computed. This allows the usage of scale variant feature detectors.

4.2.1 Features

The different feature descriptors evaluated for the development of the grape recognition strategy, which include Histogram of Oriented Gradients, Local Binary Pattern, Dense Scale Invariant Feature Transform and DAISY, are briefly explained in the next sections.

4.2.1.1 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) is a feature descriptor used to define shape and appearance based on gradient orientation of its intensity values, where the number of occurrences are counted in a locally spaced histogram (Dalal & Triggs, 2005). To compute the feature vector, first the image gradient is calculated by convolving a derivative mask, $[-1, 0, +1]$ and $[-1, 0, +1]^T$ with the image in grayscale. The image is divided into groups of cells or bins (see Figure 8) where the gradient orientation at each pixel is counted in one of the orientation bins or added to the nearest bins weighted by the corresponding magnitude (called soft binning). A group of cells defines a block, on which histogram normalization and spatial binning is done.

Finally, histograms from overlapping blocks are concatenated to define the feature vector of an image. This work employs a variant of the original Dalal Triggs HOG feature called UoCTTI HOG proposed in (Felzenszwalb, Girshick, Mcallester, & Ramanan 2010). The UoCTTI HOG incorporates a strategy for reducing the dimensionality of the descriptor vector. A visualization of this feature is presented in Figure 9, where the local histograms are presented in the image, showing the dominant orientations bins.

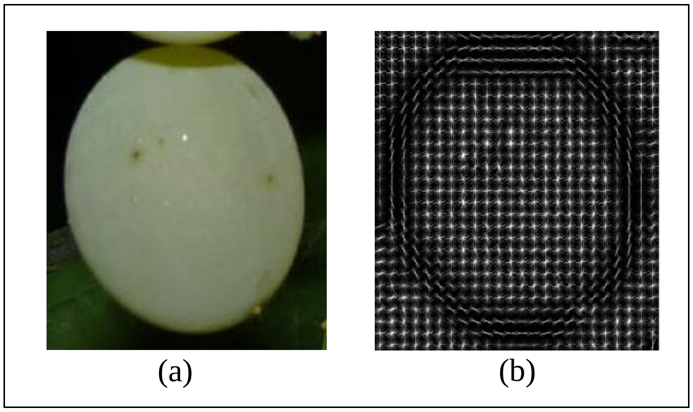
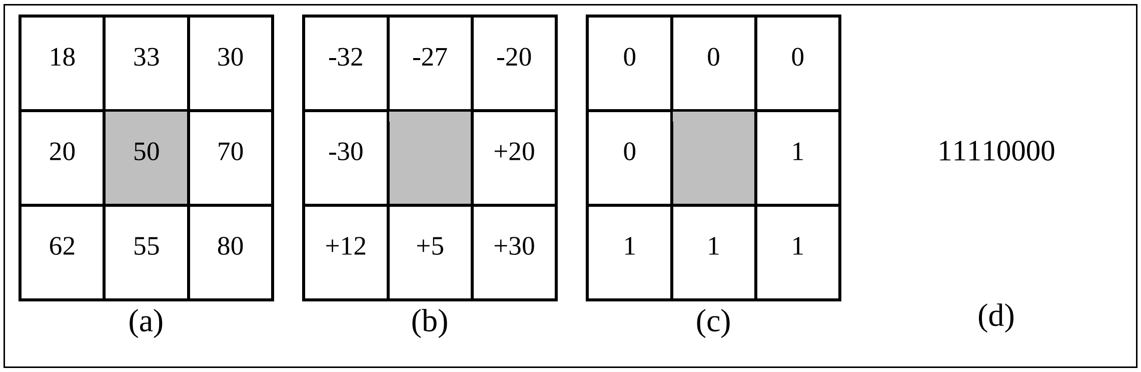

(a)
(b)

Figure 9. Example of Histogram of Oriented Gradients.
(a) Input Image. (b) A visualization of HOG descriptor

4.2.1.2 Local Binary Patterns

Local Binary Patterns or LBP descriptor is useful for texture classification. Each input image is divided in cells of *n*x*n* pixels, each pixel in a cell is compared with a central neighbor in a *m*x*m* *m*<*n* arrange, typically *m*=3. If the center pixel value is smaller than the neighbor value, the neighbor is coded as a 1, otherwise 0. The binary pattern is computed by comparing the central pixel with its neighbors within a cell. More specifically, following the neighborhood clockwise or counter-clockwise, a binary code is created by appending 0 or 1 to the code whenever the central pixel is greater or smaller than the neighbor pixel, respectively (Guo, Zhang, & Zhang, 2010). This process is repeated for each *n*x*n* pixel cells within the image. Using 8 neighbors yields an 8-bit number; see Figure 10, which allows to compute a 256 bins histogram by counting the occurrences of each binary number in a cell. Finally, each histogram is concatenated in order to form the feature vector of a whole image (Ahonen, Hadid, & Pietika, 2006).

An extension of LBP called Uniform LBP allows to reduce the 256 possible values to a subset of 59 values, by counting only the patterns that have at most two transitions (from 0 to 1 or viceversa) in the binary code. This form of lossy encoding is based on the fact that some binary patterns occur more frequently than others and that uniform patterns account for about 90% of the patterns when using 8 neighbors (Ojala, Pietikäinen, & Mäenpää, 2002; Pietikäinen, 2010).



18	33	30
20	50	70
62	55	80

(a)

-32	-27	-20
-30		+20
+12	+5	+30

(b)

0	0	0
0		1
1	1	1

(c)

11110000

(d)

Figure 10. Example of a Local Binary Pattern. (a) The 3x3 arange. (b) Comparison with the central pixel. (c) Codification. (d) Binary Pattern

4.2.1.3 Dense Scale Invariant Feature Transform

In its original form, Scale Invariant Feature Transform (SIFT) is a descriptor for image matching which was composed of two main parts, an interest point detector and an image descriptor (Lowe, 2004). One of its variants, which omits the interest point detector, and instead densely samples the image, constructs a descriptor called Dense SIFT (DSIFT). This variant has been used for image classification and object recognition. Differently from SIFT, DSIFT computes on one scale and does not include the dominant orientation on its descriptor, meaning that it is not scale invariant or orientation invariant. On each point, image gradient, including magnitude and orientation, are computed on a 16x16 pixel neighborhood. This neighborhood is divided in a 4x4 block, where histogram of 8 orientations, weighted by magnitude and a spatial Gaussian, is created for each spatial bin, resulting in 16 histograms of 8 orientations bins, forming a 128-dimension feature vector, for each interest point in an image (Vedaldi & Fulkerson, 2010). A visualization of this feature is presented in Figure 11.

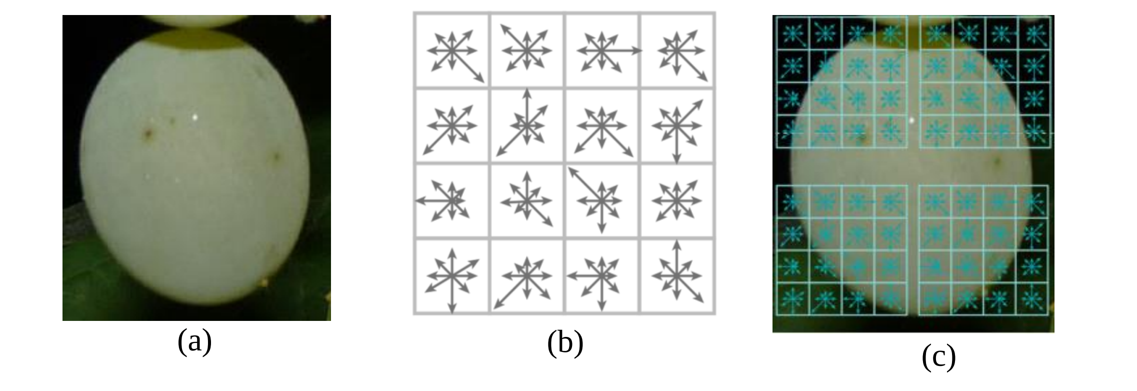


Figure 11. Visualization of DSIFT. (a) Input Image. (b) DSIFT feature geometry. (c) DSIFT feature grid.

4.2.1.4 DAISY

The DAISY also employs histograms of gradients like SIFT, but can be computed densely and faster (Tola, Lepetit, & Fua, 2010). Its original purpose was to estimate depth maps from image pairs, but this descriptor has shown good performance in classification and detection problems (Velardo & Dugelay, 2010). Unlike HOG or SIFT, the DAISY descriptor computes the gradients orientations on concentric circles (see Figure 12), for which the radiusm number of rings, number of histograms per ring, and number of bins in the histogram define the geometry and size of the descriptor in each point. The descriptor vector is built with the values resulting from convolutions between the gradient of the input image along a set of directions and Gaussian kernel with different standard deviations σ on each circular region. Even though the descriptor is meant to be densely applied to the entire image, it can also be used on a grid of fixed points as shown in Figure 12.

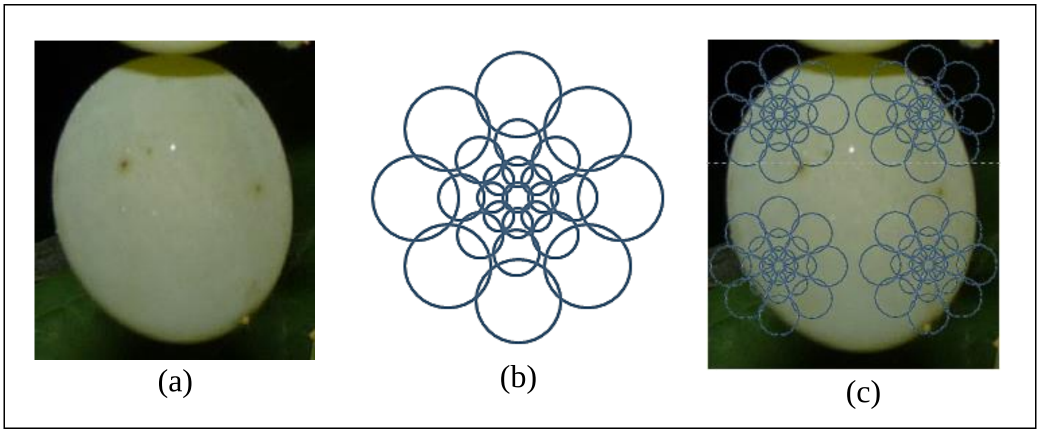


Figure 12. Daisy feature vector geometry in spatial grid. (a) Input Image. (b) DAISY feature geometry (c) DAISY feature on a 2x2 grid. Source: (Tola et al., 2010)

4.2.2 Classifiers

Once the feature vector is computed for grape and non-grape images, the following step is to determine whether a region contains a grape bunch or not by evaluating the feature vector in a discriminant function, i.e. comparing the features against the model. To this end, two classifiers are considered in the implementation of the grape detection scheme. The first approach is a standard Support Vector Machine (SVM). The second classifier strategy is a one-class SVM known as Support Vector Data Descriptor (SVDD). A support vector approach was chosen because of its fewer parameters selection to tune compared to other classifiers such as Artificial Neural Networks. Also, this classifier is capable of handling highly dimensional feature vectors differently from decision trees. Additionally, support vector classifiers do not show problems with decision boundaries, avoiding overfitting. Furthermore, the computational cost of support vectors is lower than deep learning classifiers. The fundamentals of support vector classifiers are explained next.

4.2.2.1 Support Vector Machine

SVM is a supervised learning classifier that builds an optimal hyperplane from a set of features vectors corresponding to elements of a labeled dataset. Let $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_k\}$, denote a set of feature vectors $\boldsymbol{x}_i \epsilon\ \mathbb{R}^n$, corresponding to images $\boldsymbol{I_i}$, $i\ \epsilon\ [1, k]$ and let $\mathcal{Y} = \{y_1, \dots, y_k\}$ denote the set of class labels. The problem is to find an optimal linear hyperplane $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$, where $\boldsymbol{w}$ is the normal to the hyperplane and b the translation constant, that bests separates the classes in $\mathcal{Y}$ i.e. is such that maximizes the distance of the feature vector to the dividing hyperplane and minimizes the number elements incorrectly assigned to the wrong class (incorrect side of the dividing hyperplane). In the case that data is not linearly separable, a cost parameter C and a misclassification variable ξ_i

$$\xi_i = \max\big(0, 1 - y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} + b)\big) \geq 0, \tag{4.7}$$

which represents the distance to the correct region.

This allows to relax the problem when data is not strictly separable by introducing a misclassification penalty to the sought hyperplane, see Figure 13. Considering the misclassification penalty, and that in the context of grape detection $y_i\ \epsilon\ \{1, -1\}$, because images can be either of class grape or non-grape, the classifier training problem can be formulated as that of finding $\boldsymbol{w}$ and b such that:

$$\boldsymbol{w} \cdot \boldsymbol{x_i} + b \geq 1 - \xi_i\ , \tag{4.8}$$

$$\boldsymbol{w} \cdot \boldsymbol{x_i} + b \leq -1 + \xi_i\ , \tag{4.9}$$

$$\xi_i \geq 0 \quad \forall i\ , \tag{4.10}$$

for $y_i = 1$ and $y_i = -1$ respectively. Using the class labels, these inequalities can be combined into one equation:

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) - 1 + \xi_i \geq 0 \quad \forall i \tag{4.11}$$

The objective function is defined by finding the maximum distance between the parallel hyperplanes (see Figure 13), on which the restrictions are active and the penalization error is minimum. The distance between both hyperplanes is given by $2/\|\boldsymbol{w}\|$, then the optimal distance is found by maximizing $2/\|\boldsymbol{w}\|$, which is equivalent to minimizing $\frac{1}{2}\|\boldsymbol{w}\|^2$. Finally, the optimization problem can by written:

$$min_{w,,b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i}^{k}\xi_i \tag{4.12}$$

$$s.t. \quad y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) - 1 + \xi_i \geq 0 \quad \forall i \tag{4.13}$$

$$\xi_i \geq 0 \quad \forall i \tag{4.14}$$

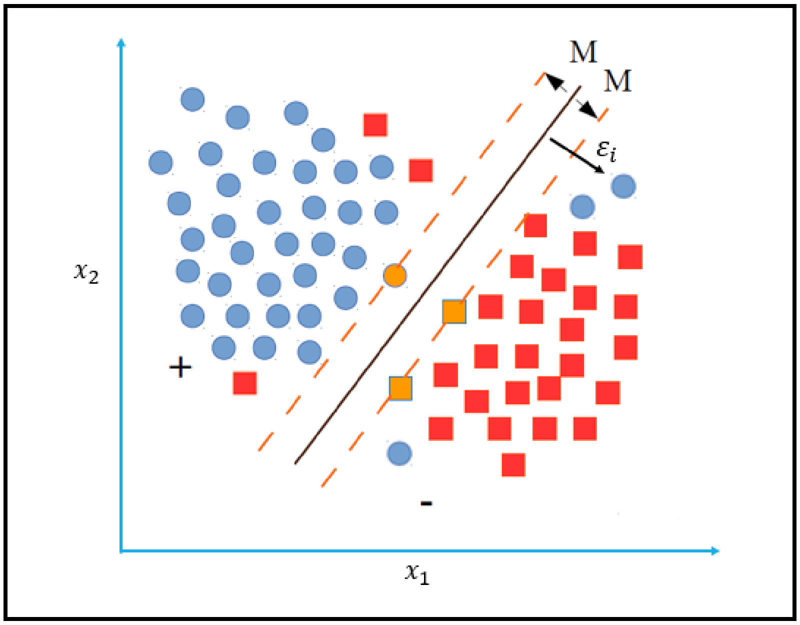


Figure 13. Support Vector Machine Classifier. Source: (Saavedra, 2015)

To solve the optimization problem stated above, it is convenient to use the Lagrange multipliers approach and solve the dual problem. This allows the problem to be computed more efficiently (Lin, 2006; Burges, 1997). The reformulated problem is expressed as:

$$max_{\alpha} \quad \sum_{i}^{k} \alpha_i - \frac{1}{2} \sum_{i}^{k} \sum_{j}^{k} \alpha_i \alpha_j y_i y_j (\boldsymbol{x_i} \cdot \boldsymbol{x_j}) \tag{4.15}$$

$$s.t. \quad 0 \leq \alpha_i \leq C \quad \forall i \tag{4.16}$$

$$\sum_{i}^{k} y_i \alpha_i = 0 \tag{4.17}$$

where α_i the Lagrange multipliers. Using a linear discriminant function with data that is a not linearly separable will inevitably result in misclassification of samples. In order to achieve a better decision boundary, a non-linear kernel function can be used. The main idea of this is to map the *n*-dimensional feature vector onto a higher dimensional space where the data is linearly separable, see Figure 14. The advantage is that the mapping $\Phi: \mathbb{R}^n \to \mathbb{R}^{>n}$ in the solution presented above, only depends on the inner product of the training data, so given a kernel function such that $\Phi(\boldsymbol{x_i}) \cdot \Phi(\boldsymbol{x_j}) = k(\boldsymbol{x_i}, \boldsymbol{x_j})$, it would only be necessary to replace $\boldsymbol{x_i} \cdot \boldsymbol{x_j}$ with $k(\boldsymbol{x_i}, \boldsymbol{x_j})$, without any need to compute the data in the higher dimensional space, but rather compute the inner product between the feature vectors in the feature space (Burges, n.d.). To this end, a Radial Basis Function is employed as kernel:

$$\Phi(\boldsymbol{x_i}) \cdot \Phi(\boldsymbol{x_j}) = k(\boldsymbol{x_i}, \boldsymbol{x_j}) = e^{-\frac{1}{2\sigma^2}\|x_i - x_j\|^2} = e^{-\gamma\|x_i - x_j\|^2} \tag{4.18}$$

For the model to be complete and perform well, adequate values of C and the parameter γ must selected. The correct selection is performed by *k*-fold cross-validation. This technique allows to estimate how accurate the prediction model is, by using the training data. The main idea is to divide the training dataset into *k* equal subsets. One of these subsets is used for validation, while the k-1 reaming is used for training. This process is repeated k times, in each iteration using a different subset. Afterwards, the results are averaged. A grid search is performed to find the correct C and γ parameters. Each parameter is tested in a range of values and then k-fold cross-validation takes place on each update of the parameters.

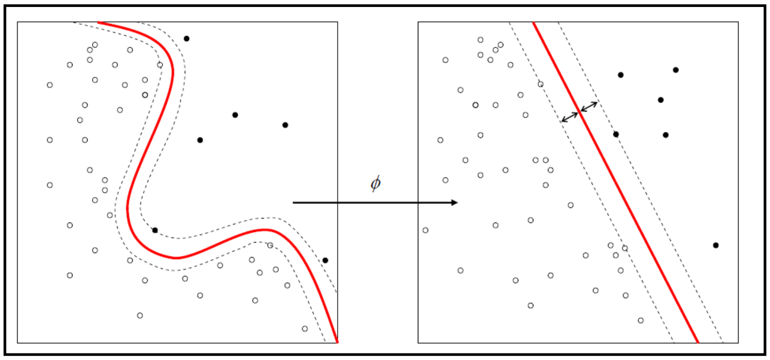


Figure 14. RBF kernel trick. Source: (Alisneaky, 2011)

4.2.2.2 Support Vector Data Descriptor

Another method of classification is the Support Vector Data Description (SVDD), which is a one-class support vector classifier. Classic SVM tries to label new descriptor vectors according to two or more known classes. However, when there is only one class, assigning the label to objects that are of the class and rejecting everything else is complicated, as in this case in which the decision has to be made between grape and everything else that is not a grape. The initial objective of SVDD classifiers was to detect outliers in a process, where obtaining normal state training data is simple, but gathering non-normal state training data might be practically impossible. The one class classification scheme was developed to employ only positive class data in order to find an optimal decision boundary.

SVDD proposed by Tax & Duin (2004) searches for the optimal hypersphere that encloses the descriptor vectors of all positive data, while trying to minimize the volume of the decision boundary in order to not include outliers. Given $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_k\}$, as the set of feature vector of k positive data images, with $\boldsymbol{x}_i$ the i-th feature vector of n-dimensions, $\boldsymbol{x}_i \epsilon\ \mathbb{R}^n$, the center $\boldsymbol{a}$ and the radius $\boldsymbol{R} > 0$ is defined, which characterizes the optimal hypersphere. A new query is classified as positive if the distance from the data point $\boldsymbol{x}_i$ to the center $\boldsymbol{a}$ is smaller than $\boldsymbol{R}$**,** see Figure 15. To obtain a soft margin boundary, some outliers can be accepted by introducing a penalty parameter C and an error variable $\xi_i \geq 0$. The optimization problem can be written as:

$$min_{a,R}\ \boldsymbol{R^2} + C \sum_{i=1}^{k} \xi_i \tag{4.19}$$

$$\xi_i = \max(0, \|\boldsymbol{x}_i - \boldsymbol{a}\|^2 - \boldsymbol{R^2}) \tag{4.20}$$

$$s.t.\quad \|\boldsymbol{x}_i - \boldsymbol{a}\|^2 \leq\ \boldsymbol{R^2} + \xi_i \qquad \forall i \tag{4.21}$$

$$\xi_i \geq 0 \qquad \forall i \tag{4.22}$$

$$\boldsymbol{R} > 0 \tag{4.23}$$

The problem above can be solve using Lagrange multipliers and the dual formulation. The kernel trick can also be applied here to achieve a better decision boundary. Finding the optimal parameters is done by a grid search, since no cross-validation can be done using only positive example training data.

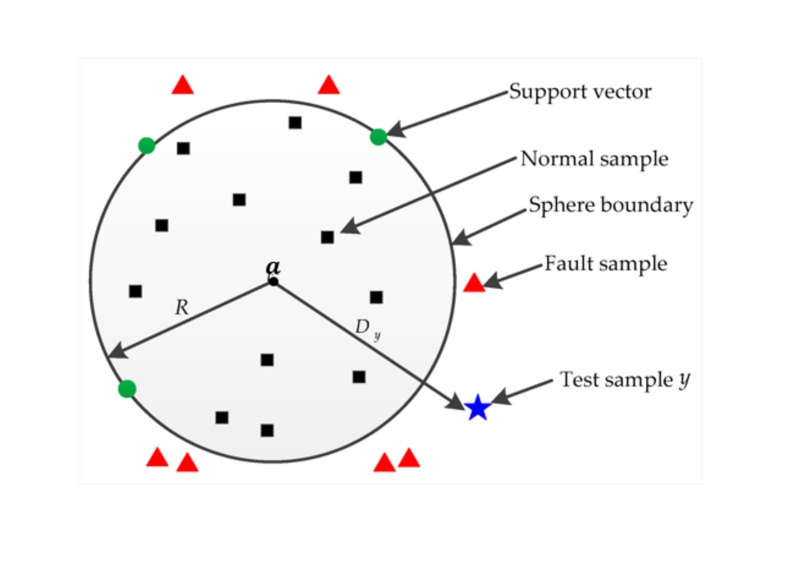


Figure 15. Support Vector Data Descriptor. Source: (Huang et al., 2016)

Once all the features and classifiers are compared, the best performing combination is chosen to create the model to be used to detect grape bunches in vineyards. In order to recognize the grape berries, patches of the same size as the training dataset must be sampled to extract the feature vector and to be classified.

After grape recognition is done in several scales, all results are transform to the original image size. This results in an image with several patches containing grape berries and some false positive results. Since some patches overlap, and they include a classification score given by SVM, a non-maximal suppression algorithm is used only in the classification output patches, eliminating redundant information.

4.3 Clustering

Next, in order to identify each grape bunch in an image, a clustering technique called Dense-Based Spatial Clustering of Applications with Noise, DBSCAN, is used. DBSCAN algorithm is capable of clustering points based on density, not needing as an input parameter the number of clusters like k-means. Also, this method classifies isolated points as outliers, handling false positive from the classification stage. Both advantages help in the grape bunch detection problem, given the variability of the number of clusters and possible shapes due to occlusion. The main objective of this algorithm is to group near grape berry patches as a cluster.

Given a set of points in a space, a point q is considered part of a cluster if a minimum number of points are reachable at a distance ε. An edge or border point is considered part of the cluster if it is reachable but cannot reach more points. All points that are not reachable and do not fulfil the minimum number of points restriction are consider outliers. Both distance ε and minimum number of points are parameters that depend on the camera and distance to the object, defined by the average berry size in pixels.

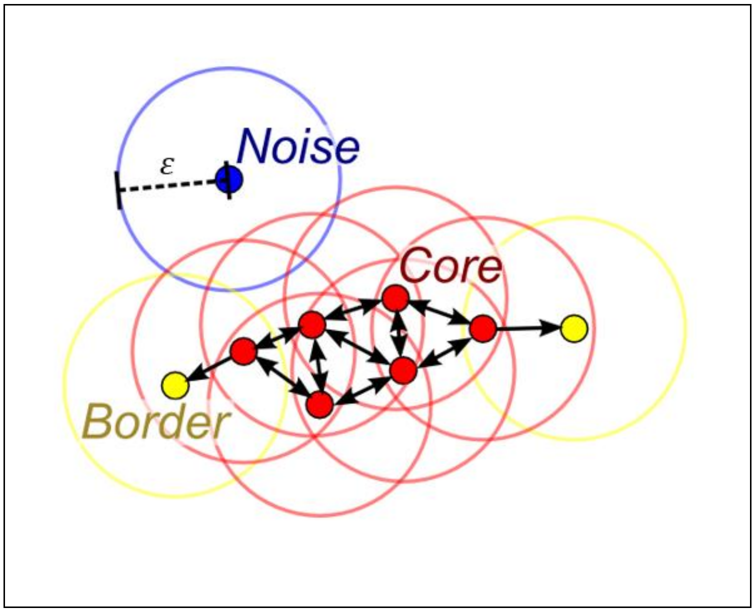


Figure 16. DBSCAN algorithm example. Source: (Chire, 2011)

4.4 Cluster Boundary

Once the x, y coordinates of grapes and their belonging cluster are obtained, it is needed to define the boundary of the cluster and surround the inner points, assuming those pixels belong to grapes pixels. Several researcher propose using a convex hull approach (Herrero-Huerta et al., 2015; Liu et al., 2015b; Nuske et al., 2014). The problem with this approach is that the convex area created may include non-berry pixels and spatial information might be lost, counting several grape bunches as one. An alternative method would be to create a non-convex area, using the boundary of the identified berries. Using the alpha shape method, the shape or boundary of the grape bunch can be defined using the finite set of grape centers for each cluster.

A formal definition for alpha shape method is: given a group of points $\mathcal{S}$, two points $p, q \in \mathcal{S}$ are boundary points, if a disk of radius α, with p and q on its circumference, lies in the space not containing any other point from $\mathcal{S}$. If the condition is true, an edge segment is computed connecting p and q. The α-shape of $\mathcal{S}$ is the group of straight-lines connecting all the boundary points (Kirkpatrick & Seidel, 1983). Then, the binary mask is created from all the pixels inside α-boundary, defining the area and location of the grape cluster.

4.5 Neighboring Grape Bunch Separation

If two or more clusters are close together, the DBSCAN algorithm may classify them as one grape bunch, affecting the detection performance. To improve the detection, a novel method that analyses the shape and spatial distribution of each cluster in the binary mask is proposed. The proposed method divides the cluster into smaller two new groups if an area of a cluster has enough inter-separation in the vertical and horizontal axis. To achieve this, using the binary image, the distance between inner cluster separation is measured, by studying the number of pixels were transitions from 1 to 0 and 0 to 1 occur in a row. If more than one low to high edge occurs in a row, a separation exists. If the separation in the row and column is bigger than the size of two berries, we assume another grape bunch exists, creating a new detection. The novel method is presented in Algorithm 1. An example of the complete proposed process is presented in Figure 17. This approach, compared to (Liu et al., 2015b) and (Luo, Tang, Zou, Wang, et al., 2016), is independent of color, being able to detect grape bunches of red and white grapes, and is able to separate two or more adjoining grape bunches since the method searches for changes in the binary image boundary and not the barycenter.

Algorithm 1: Neighboring Grape Bunch Separation

```
Let $BW$ be the binary image of the grape bunch after DBSCAN
Let $r$ be the radius in pixels of the grape
$y_{min}$ ← minimum row from $BW$
$y_{max}$ ← maximal row from $BW$
for $y$ between $y_{min}$ and $y_{max}$
    $edge_{raising}$ ← find rising edges in the row $y$ of $BW$
    $edge_{falling}$ ← find falling edges in the row $y$ of $BW$
    if length of $edge_{raising} < 2$
        Continue to next $y$
    Else
        for $i = 1$ to length $edge_{raising} - 1$
            $x_{sepaation}[i] \leftarrow edge_{falling}[i] - edge_{raising}[i+1]$
            if $x_{sepaation}[i] > 4 * r$
                $x_{new}[i] \leftarrow \left(edge_{falling}[i] + edge_{raising}[i+1]\right)/2$
                $y_{sepaation}[i]$ ← calculate the vertical distance to the cluster
                if $y_{sepaation}[i] > 4 * r$
                    Calculate the new bounding boxes
                end if
            end if
        end for
    end if
end for
```

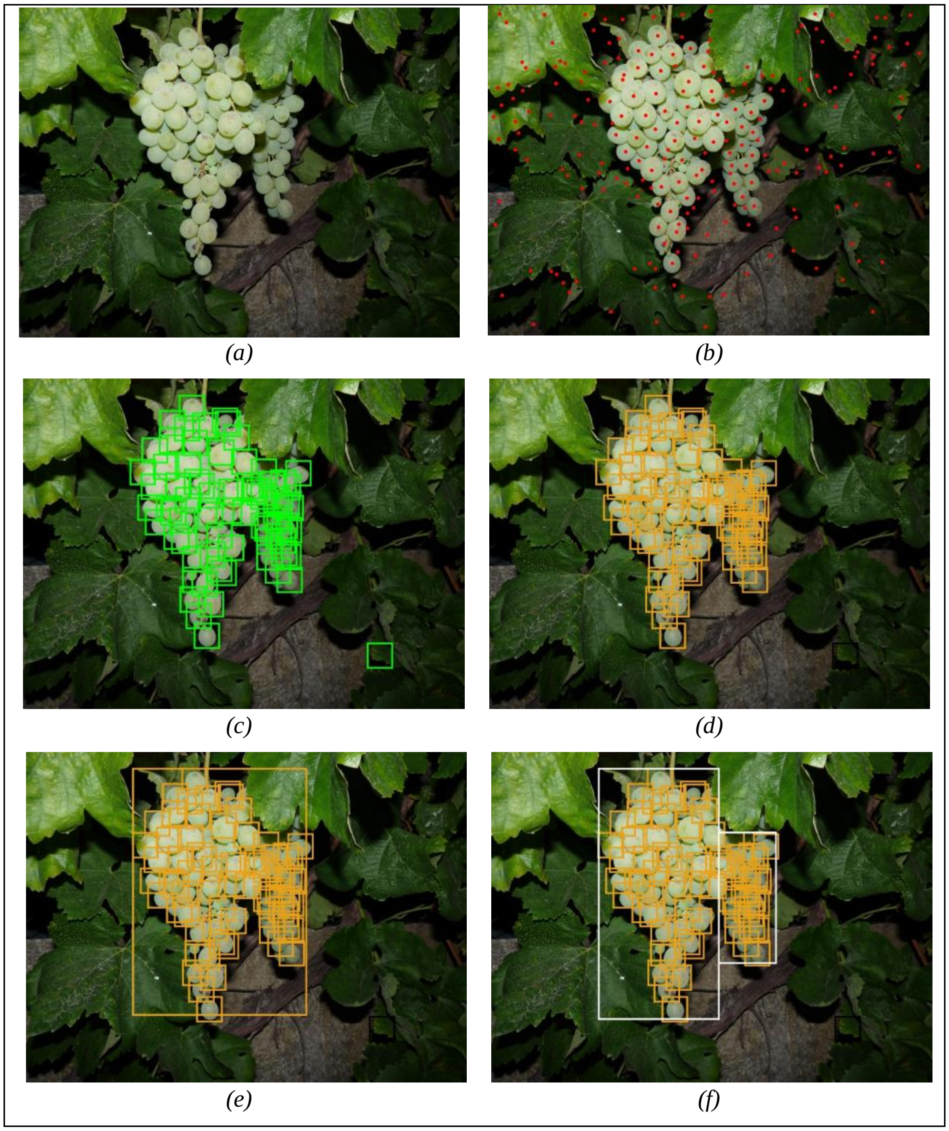


Figure 17. Example from Portugal dataset, images showing the output of the different stages of the framework. (a) Input image. (b) Salient points of potential berries. (c) Output of the multiscale HOG+LBP SVM-RBF classifier. (d) Output of the DBSCAN eliminating isolate berries. (e)Initial bounding box of the cluster. (f) Output of the neighboring grape bunch.

Figure 18 shows the final detection binary mask and its comparison with the hand labeled ground-truth. This metric allows to understand the real percentage of detected pixel area of each identified cluster.

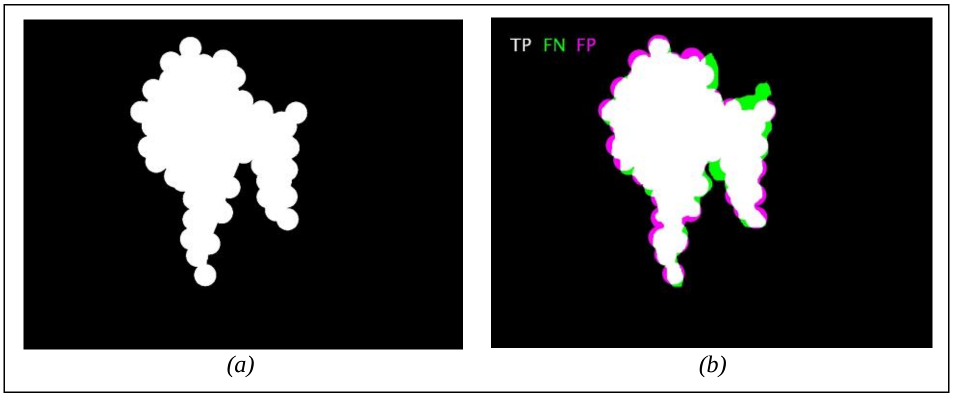


Figure 18. Example of the detection mask to study the spatial distribution and the area of the grape cluster. (a) Binary detection mask of the cluster region after applying the method. (b) Comparison with ground truth, white=TP, black=TN, Green=FN, magenta=FP.

5 EXPERIMENTAL METHODOLOGY AND RESULTS

5.1 Methodology

The different descriptors and classification strategies were implemented in MATLAB® using the computer vision and pattern recognition libraries as LibSVM (Chang & Lin, 2013), VLFeat (Vedaldi & Fulkerson, 2010), DBSCAN (Inglese, 2015) and MATLAB® implementations of Fast Radial Symmetry Transform, Non-maximal suppression and DAISY (Kovesi, 2004; Malisiewics, n.d.; Tola et al., 2010). The algorithms and proposed grape detection strategy were executed on a computer with an Intel Core i5-3317U 1.70GHz CPU with 2 cores and 16 GB RAM.

The dataset made available by Pavel & Runarsson (2015) was employed to train and test the berry classifier. This dataset consists of single white grape images and images of background patches, all of them size 40x40 pixels. Datasets T-3 and T-X, made available by Pavel & Runarsson (2015), were used in the training stage. The T-3 training set consists of 576 images (288 non-berries and 288 berry images) while T-X dataset consists 11,332 images created by rotating by 90, 180, and 270 degrees all T-N, N={1,..5} examples available. The amount of true positive and true negatives in the T-X dataset are equal. On the other hand, the testing dataset has a total of 4000 images. Examples of this dataset are shown in Figure 19.

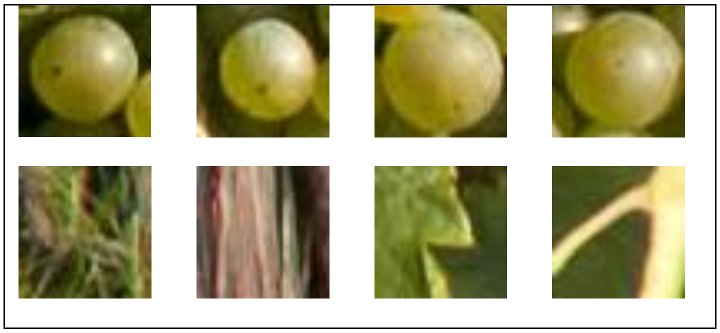


Figure 19. Examples of Pavel & Runarsson (2015) training and testing dataset.

Evaluating the proposed grape detection strategy and testing the different descriptors was done employing four different lateral vid image datasets; Israel Dataset (Berenstein et al., 2010), Iceland Dataset (Pavel & Runarsson, 2015), Portugal Dataset (Reis et al., 2012) and the Chile Dataset. Number of images and image size are shown in Table 6. Only visible spectrum cameras were used. It is to be noted that each dataset was acquired under different illumination and camera-grape distances, listed in Table 6, thus the data covers a representative range of real world situations. Examples of these datasets are shown in Figure 22. The datasets were hand-labeled to obtain the ground truth.

The performance of the approaches for single berry and grape bunch detection is evaluated using the standard confusion matrix summarized in Table 1 and the corresponding true/false positive/negative rates, as well as Accuracy (Acc), Precision (Prc), and Recall metrics indices together with the corresponding Confidence Intervals (CI) with $\alpha = 95\%$.

The HOG is computed here using UoCTTI variant with an 8x8 cell size; 2x2 block size, 9 orientation bins, without bilinear interpolation. The LBP descriptor implemented employs 10x10 sliding blocks with; 3x3 pixel neighborhood, and uniform binning histogram. The parameters of the DAISY descriptor are; 15 pixel radius; 3 rings; 8 histograms in each ring level; 8 bins in each histogram and 16 fixed points. Finally, the DSIFT descriptor uses a bin cover size of 5 pixels, a step of 10 pixels, and a geometry of 4x4 bins with 8 orientations.

		Predicted Condition	
		Grape	Non-Grape
True Condition	Grape	True Positive (TP)	False Negative (FN)
	Non-Grape	False Positive (FP)	True Negative (TN)

Table 1. Confusion Matrix for Grape Recognition

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \cdot 100 \quad (5.1)$$

$$Precision = \frac{TP}{TP + FP} \cdot 100 \quad (5.2)$$

$$Recall = True\ Positive\ Rate = \frac{TP}{TP + FN} \cdot 100 \quad (5.3)$$

$$False\ Positive\ Rate = \frac{FP}{FP + TN} \cdot 100 \quad (5.3)$$

$$CI = \hat{p} \pm Z_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \quad (5.4)$$

5.2 Results

The results for grape recognition using the different datasets and classifiers are presented in Tables 2 to 5. Tables 2 and 3 summarize the results obtained with the different descriptors using the SVM classifier applied to the T-3 and T-X datasets, while Tables 4 and 5 summarize the results for the same datasets, but employing the SVDD classifier. A comparison of the results shows that regardless of the dataset, the HOG+LBP descriptor yields the highest accuracy, precision and recall rates when using the SVM classifier. When the SVDD is employed for the detection of berries in the T-X dataset, the HOG+LBP descriptor outperforms the other. The results show that the HOG feature shows the best results from the three gradient based features. This can be attributed to the size of the feature vector and the feature geometry, where HOG analyses 8x8 windows, while the other features use bigger windows on a fixed grid.

Comparing the results for the SVM classifier with those of the SVDD, it is possible to observe that the latter has 15-25% lower accuracy, precision and recall values. This values can be interpreted as that the high variability on field affects the decision boundary of a classifier, where a classifier that has examples of non-grape images will give a better result. In summary, the combined HOG+LBP descriptor together with the SVM classifier provide a high accuracy of 95.98 ± 0.61%, a very high precision of 99.41 ± 0.23%, and a good recall rate of 92.50 ± 0.82%. These results shown an improvement with respect to other approaches reported in the literature (Pavel & Runarsson, 2015), where a different cell size of hog combined with texture information helps in the hyperplane decision boundary. In terms of computational time, the best descriptor is LBP, which takes 25-35% less time than the HOG+LBP approach. Nonetheless, the HOG+LBP is at least 2 to 3 times faster than DSIFT, DSIFT+LBP or DAISY+LBP. Therefore, the HOG+LBP descriptor and the SVM classifier provide the best berry detection approach in terms of effectiveness and speed.

The Receiver Operational Characteristic (ROC) curves for grape recognition using the SVM classifier are shown in Figure 20. The performance curves of the feature descriptors evaluated above are compared, studying the True Positive Rate or Recall (fraction of correctly labeled positive) against the False Positive Rate (fraction of labeled negatives classified as positive) at several SVM decision thresholds. Figure 21 presents the Precision-Recall (PR) curves for the feature descriptors at various values of SVM decision threshold.

T-3	HOG	LBP	HOG+ LBP	DSY	DSY+ LBP	DSIFT	DSIFT+LBP
SVM Param	C=2 γ=3e-2	C=2 γ=6e-2	C=2 γ=1e-2	C=2 γ=2e-2	C=2 γ=2e-2	C=4 γ=3e-2	C=2 γ=1e-2
Acc. %	91.0 ± 0.87	87.88 ± 1.01	91.57 ± 0.86	86.18 ± 1.07	86.32 ± 1.07	88.53 ± 0.99	89.30 ± 0.96
Prc.. %	98.88 ± 0.33	98.65 ± 0.36	98.99 ± 0.31	96.35 ± 0.58	96.48 ± 0.57	98.49 ± 0.38	98.64 ± 0.34
Rec. %	83.75 ± 1.14	76.80 ± 1.31	84.00 ± 1.13	75.20 ± 1.33	75.40 ± 1.34	78.25 ± 1.28	79.70 ± 1.25
Avg. Time per image	0.0034	0.0025	0.0038	0.0148	0.0154	0.0095	0.0099

Table 2. Grape recognition results using the SVM classifier and the T-3 training set.

T-X	HOG	LBP	HOG+ LBP	DSY	DSY+ LBP	DSIFT	DSIFT+LBP
SVM Param	C=2 γ=3e-2	C=2 γ=6e-2	C=2 γ=1e-2	C=2 γ=2e-2	C=2 γ=2e-2	C=4 γ=3e-2	C=2 γ=1e-2
Acc. %	95.38 ± 0.65	95.03 ± 0.67	95.98 ± 0.61	92.5 ± 0.82	94.13 ± 0.73	93.73 ± 0.75	94.53 ± 0.71
Prc. %	99.62 ± 0.19	99.51 ± 0.22	99.41 ± 0.23	99.30 ± 0.26	99.44 ± 0.23	99.72 ± 0.17	99.50 ± 0.22
Rec. %	91.10 ± 0.88	90.50 ± 0.91	92.50 ± 0.82	85.60 ± 1.09	88.75 ± 0.98	87.70 ± 1.02	89.50 ± 0.95
Avg. Time	0.0039	0.0044	0.0059	0.0194	0.0224	0.0111	0.0129

Table 3. Grape recognition results using the SVM classifier and the T-X training set.

T-3	HOG	LBP	HOG+LBP	DSY	DSY+ LBP	DSIFT	DSIFT + LBP
SVDD Param	C=5e-1 γ=2e-3	C=3e-1 γ=2e-2	C=5e-1 γ=2e-3	C=4e-3 γ=6e-2	C=1e-3 γ=3e-2	C=2e-3 γ=6e-2	C=2e-3 γ=3e-2
Acc. %	72.33±1.39	68.65 ± 1.44	70.13 ± 1.42	59.50 ± 1.52	60.68 ± 1.49	63.93 ± 1.49	65.88 ± 1.47
Prc. %	92.56 ± 0.81	88.45 ± 0.99	96.00 ± 0.61	82.31 ± 1.18	89.69 ± 0.98	90.66 ± 0.90	89.94 ± 0.93
Rec. %	48.55 ± 1.55	42.90 ± 1.54	42.00 ± 1.53	24.20 ± 1.33	24.35 ± 1.41	31.05 ± 1.43	35.75 ± 1.48
Avg. Time	0.0049	0.0025	0.004	0.015	0.0161	0.0089	0.009

Table 4. Grape recognition results using the SVDD classifier and the T-3 training set.

T-X	HOG	LBP	HOG+ LBP	DSY	DSY+ LBP	DSIFT	DSIFT+ LBP
SVDD Param	C=5e-1 γ=2e-3	C=3e-1 γ=2e-2	C=5e-1 γ=2e-3	C=4e-3 γ=6e-2	C=1e-3 γ=3e-2	C=2e-3 γ=6e-2	C=2e-3 γ=3e-2
Acc. %	79.15±1.26	73.83±1.36	81.00±1.21	65.00±1.48	66.75±1.46	74.25 ±1.35	76.54±1.31
Prc. %	93.06±0.79	89.22±0.96	89.95±0.93	87.04±1.04	81.84±1.19	85.56±1.09	84.85±1.11
Rec. %	63.00±1.50	54.20±1.54	69.80±1.42	35.25±1.48	43.05±1.53	58.35±1.53	64.40±1.48
Avg. Time	0.0055	0.0027	0.0069	0.0169	0.0249	0.0107	0.0126

Table 5. Grape recognition results using the SVDD classifier and the T-X training set.

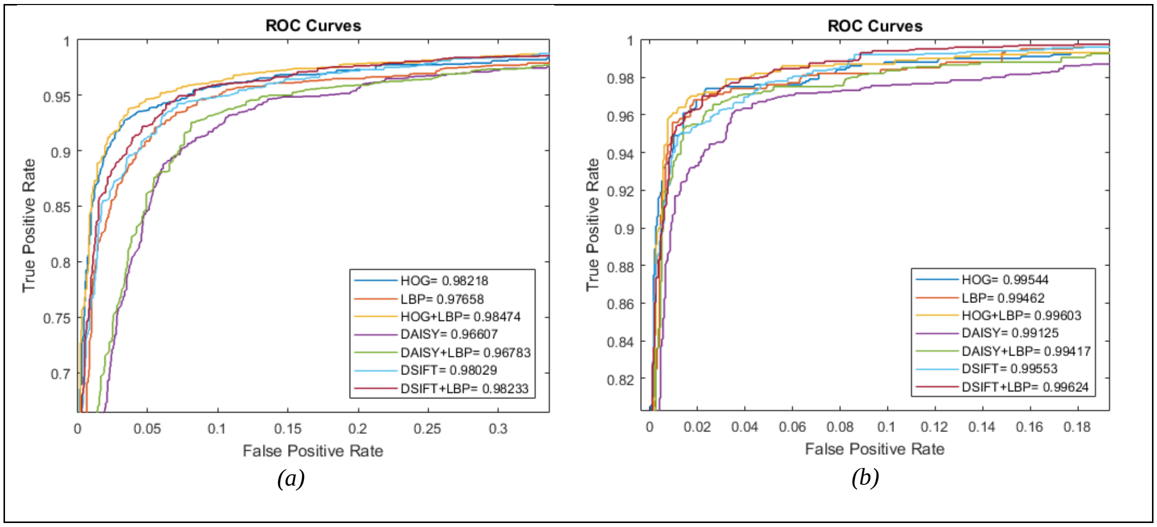


Figure 20. ROC curves and AUC of grape recognition using SVM classifier. (a) Using T-3 training set. (b) Using T-X training set

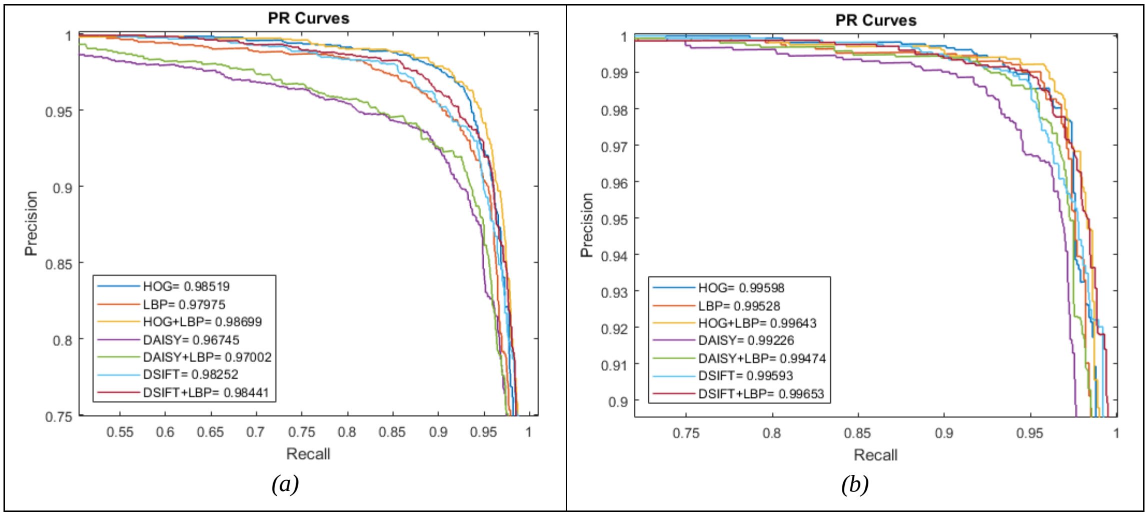


Figure 21. PR curve and AUC of grape recognition using SVM classifier. (a) Using the T-3 training set. (b) Using the T-X training set.

Size and variability of the training sets influence on results, showing a better performance the T-X dataset. This is due to the larger number of images that better represent the grape and non-grape classes.

Considering the previous results for the grape recognition stage, a combined HOG+LBP feature descriptor was chosen together with the SVM classifier trained with the T-X dataset to implement the grape bunch detection strategy. The evaluation of the proposed grape bunch detection scheme considers four different datasets. Examples of each dataset are shown in Figure 22. Results of the novel method for grape bunch detection are summarized in table 6. Area classification is also studied comparing hand labeled ground truth against the outcome of the proposed algorithm, results are shown in table 7.

		Israel Dataset	Iceland Dataset	Portugal Dataset	Chile Dataset	
	# of Images	129	3	15	16	
	Illumination	Daytime Artificial Illumination	Daytime Natural Illumination	Nighttime Artificial Illumination	Daytime Natural Illumination	
	Camera-Object Distance [m]	~ 1.0	~ 1.5	~ 0.5	~ 1.0	
	Original Resolution	800 x 600	3888x2592	3648x2736	5472x3648	Average
Grape Bunch Detection	Avg. Prc %	88.46% ±4.34	83.02% ±8.80	96.67% ±7.04	86.28% ±3.95	88.61%
	Avg. Recall %	82.43% ±5.03	81.63% ±8.94	76.22% ±14.52	81.06% ±4.42	80.34%

Table 6. Grape bunch detection results for each dataset.

		Israel Dataset	Iceland Dataset	Portugal Dataset	Chile Dataset	
	# of Images	129	3	15	16	
	Original Resolution	800 x 600	3888x2592	3648x2736	5472x3648	Average
Grape Area Recognition	Avg. Acc. %	92.64% ±0.01	97.40% ±0.01	95.80% ±0.01	95.97% ±0.01	95.45%
	Avg. Prc. %	75.68% ±0.03	79.92% ±0.06	87.93% ±0.01	80.06% ±0.02	80.90%
	Avg. Recall %	64.08% ±0.03	66.65% ±0.08	84.75% ±0.02	67.79% ±0.02	70.82%

Table 7. Grape bunch area results for each dataset.

The results of the proposed grape bunch detection method show an average precision of 88.61% and an average recall of 80.34%. Results show the correct performance of the proposed method, including the novel neighborhood grape bunch separator. By analyzing the results, it can be noticed that errors may be consequence of the not detected salient points, misclassification by SVM or not clustered in the DBSCAN stage.

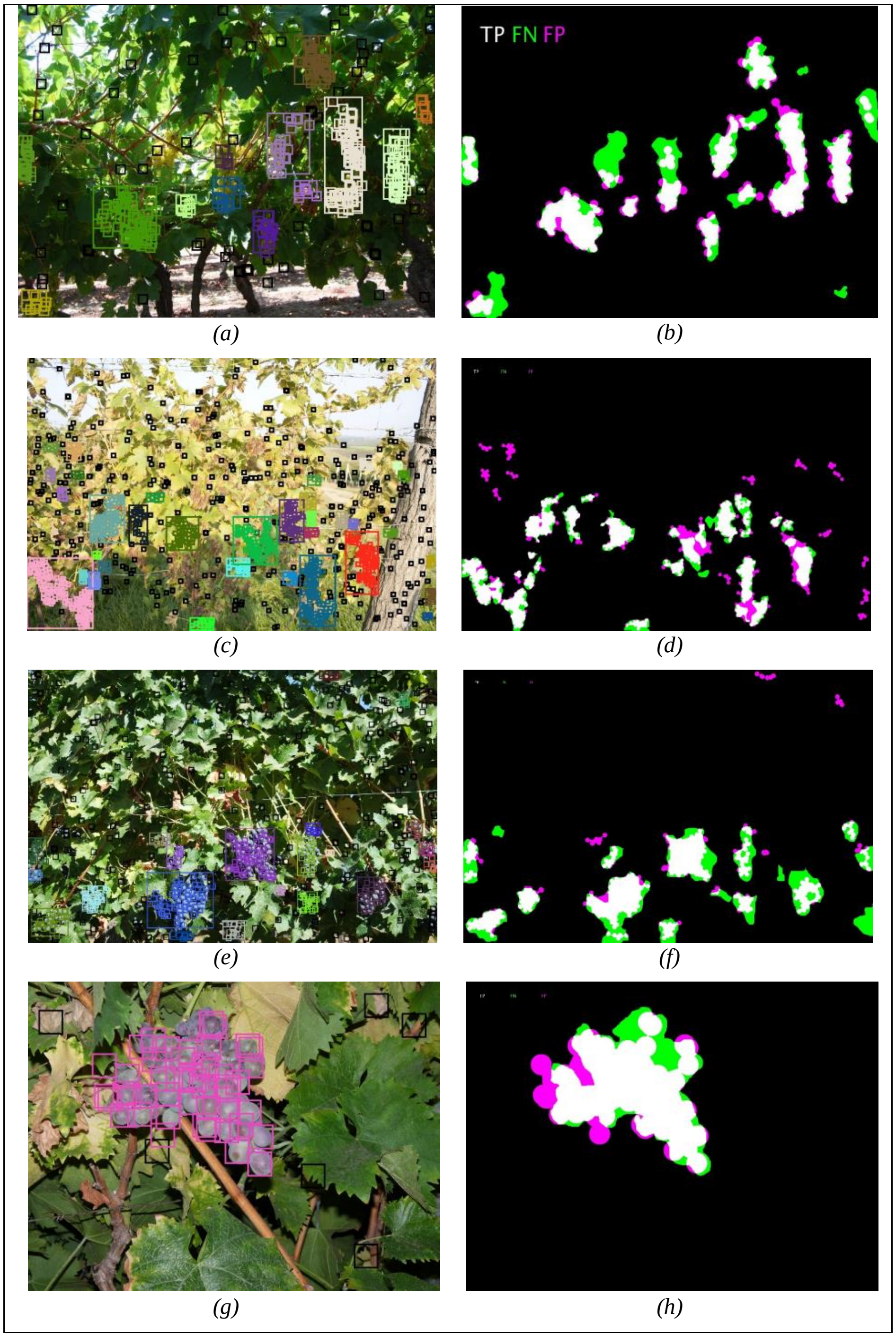
Grape bunch detection shows an 88.6% precision and 80.3% recall on average through different datasets, presenting a novel method for correctly detecting neighboring grape bunches. Pixel or area classification is also studied in order to understand the correct classification and completeness of the grape bunch detection.

Concerning the detection of the centroid of each berry in the grape bunch, the analysis of the images shows that centroid misdetections occurs more often in image regions that contain shadows of where the contrast is insufficient for the FRST+NMS stage to yield good symmetry scores. In the current implementation of the proposed approach, the FRST thresholds were set to permissive levels in order to detect berries with low contrast, although this increases the amount of false positive points and computation time because of the larger number of pixels that have to be evaluated by the classifier. The results show that clustering stage is very effective in the identification of grape bunches. The only challenge to a successful clustering are the highly occluded areas where only a couple of berries can be seen. Since the clustering procedure removes isolated points and there are constraints for the minimum number of single grape detected in connected regions, this occluded grape bunch are not detected by the proposed method. Also, when grape bunches are closed together in a vertical orientation, the proposed method is not able to correctly cluster them in different groups since no spatial analysis and segmentation is done in this orientation.

The performance of the area recognition of the grape bunch detection is also shown in Table 7. The accuracy of the proposed approach is on average 95.5% for grape pixel detection. However, this high value is in part due to the high true negative detection and not only due to a high rate or correctly labeled pixels. The average precision of the approach applied to the different datasets is 80.9±0.03%. Analyzing the location of the false positives pixels, most of them occur in the neighborhood of the grape bunches. Therefore, even if the false positive rate is high, due to the location of occurrence of the false positive, the results of grape bunch detection are not afected. On the other hand, the average recall rate considering the different datasets is 70.8±0.03%. The recall rate can be interpreted as the rate of detection. Even if this rate is below ideal rates in the range 95-99%, it is to be noted that this rate considers the number of pixels within grape bunches that were not labeled correctly as belonging to the corresponding grape bunch due to clustering errors.

It is to be noted that the grape detection results obtained may be improved by training the classifier with a larger number of datasets. Unfortunately, it is difficult to obtain large datasets from different vineyards and grape varieties. The T-X dataset only contains white grapes under natural illumination with a fixed camera-plant distance, while the test datasets includes red and white grapes under natural and artificial illumination.

The proposed grape bunch detection strategy can be employed for diverse purpose in the different vineyard management tasks. For example, FRST can be computed again on the detected grape bunch region to refine the count of single grapes in the bunch and improve yield estimation, as shown in Figure 23. Another application is the measurement of leaf removal and light exposure levels as shown in Figure 24. This would allow growers to standardize their leaf removal processes throughout the whole field. This same information is also useful for controlled crop thinning.



(a) *(b)*

(c) *(d)*

(e) *(f)*

(g) *(h)*

Figure 22. Image examples of the different datasets. In white=TP, green=FN, magenta=FP black=TN. (a) Israel Dataset. (c) Iceland Dataset. (e) Chile Dataset. (f) Portugal Dataset.

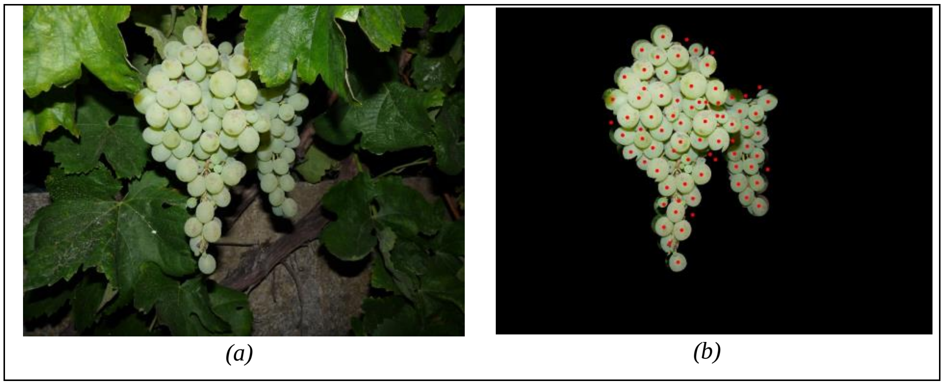


Figure 23. Example of berry counting for yield estimation models.

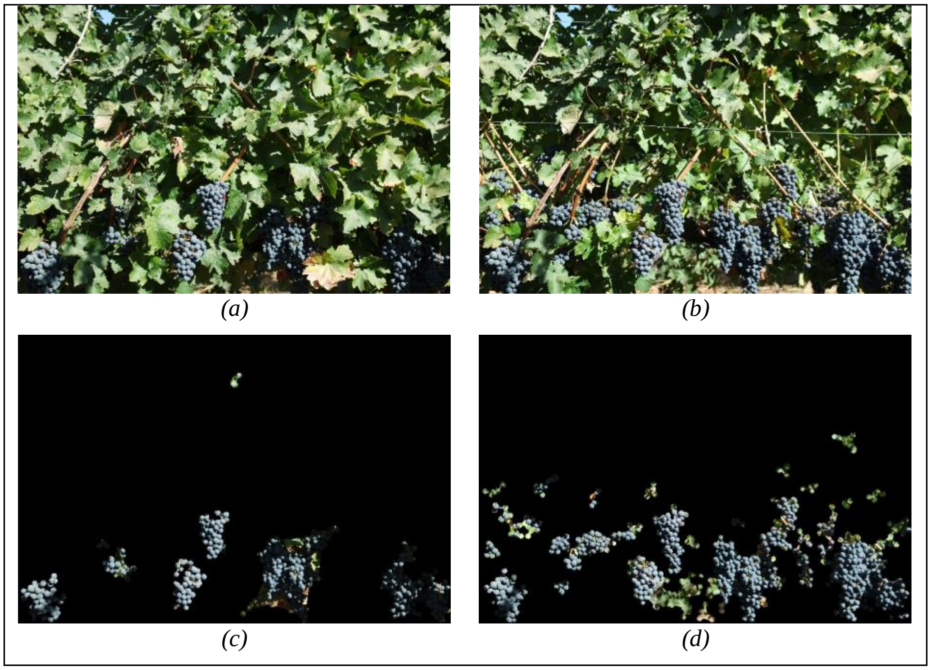


Figure 24. Example of leaf removal application to measure grape exposure level. Images from Chile Dataset. (a) Before leaf removal. (b)After leaf removal. (c) Grape mask from image before leaf removal. (d) Grape mask from image after leaf removal.

6 CONCLUSIONS AND FUTURE WORK

This work presented a grape recognition and grape bunch detection strategy that employs the HOG+LBP descriptor together with a SVM-RBF classifier and the DBSCAN clustering. The proposed approach capable of detecting both red and white berries, under different illumination conditions, levels of occlusion and distances between the camera and the vine. The approach was compared with other detection strategies using alternative descriptors that are among the most effective for different recognition applications such as are DAISY and DSIFT. The use of combined shape and texture information proved to yield better results than the shape or texture information provided individually by HOG and LBP, respectively. Also, an alternative one-class SVDD formulation of the SVM was implemented and compared to the traditional SVM-RBF classifier. However, results show that the traditional SVM-RBF yields a better grape bunch detection.

Grape bunch detection shows an 88.6% precision and 80.3% recall on average through different datasets, presenting a novel method for correctly detecting neighboring grape bunches. Pixel or area classification is also studied in order to understand the correct classification and completeness of the grape bunch detection.

At the pixel level, the SVM classifier with HOG+LBP feature vectors yields an average accuracy of 95.5%, average precision of 80.9% and average recall of 70.8% in grape/non-grape recognition with the SVM trained for white grapes under natural daylight illumination tested on grapes of other varieties and illumination conditions including red grapes with artificial illumination.

Size and variability of training dataset showed to have an influence on the results. The larger datasets allowed better overall performance, increasing all three metrics, but especially accuracy and recall when comparing results trained with T-3 and T-X.

The approach presented performs a multiscale analysis. This makes the approach more robust to variations in grape size and distance variations between the camera and the vine, but at the cost of more processing time. Also, the clustering stage employing DBSCAN filters out false positives on isolated locations, thus improving further the accuracy, recall and precision performance indices. The clustering approach together with the alpha-shape method produce non-convex boundaries, which combined with the proposed grape bunch segmentation algorithm allow the identification of individual grape bunches even when there might be some overlap among some of them. This can be very useful for the development of robotic harvesting, leaf removal, plant thinning or selective spraying, as well as help to improve yield estimations, critical for an efficient vineyard management.

Ongoing research is concerned with a longitudinal study of the impact of automated grape detection on the vineyard production tasks, and compare precision robotic harvesting with manual harvesting by hand or mechanical harvesters. For robotic harvesting, one of the challenges is peduncle detection and adequate tool design.

Other aspects specific to the proposed grape detection method that are part of future studies is the use of adaptive local histogram equalization and its effect on the FRST for better detection of grape centroids, and compare this with other interest point selectors.

7 REFERENCES

Ahonen, T., Hadid, A., & Pietika, M. (2006). Face Description with Local Binary Patterns : Application to Face Recognition, *28*(12), 2037–2041.

Alisneaky. (2011). Kernel Machine. Retrieved from https://commons.wikimedia.org/wiki/File:Kernel_Machine.png

Auat Cheein, F. A., & Carelli, R. (2013). Agricultural robotics: Unmanned robotic service units in agricultural tasks. *IEEE Industrial Electronics Magazine*, *7*(3), 48–58. https://doi.org/10.1109/MIE.2013.2252957

Bachche, S. (2015). Deliberation on Design Strategies of Automatic Harvesting Systems: A Survey, 194–222. https://doi.org/10.3390/robotics4020194

Benito Saez, P. (2010). Urbina Vinos. Retrieved from http://urbinavinos.blogspot.cl/2010/09/aclareo-de-racimos.html

Berenstein, R., Shahar, O. Ben, Shapiro, A., & Edan, Y. (2010). Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer. *Intelligent Service Robotics*, *3*(4), 233–243. https://doi.org/10.1007/s11370-010-0078-z

Burges, C. J. C. (n.d.). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, *2*(2), 121–167.

Canadian Agricultural Human Resource Council. (2016). *Agriculture 2025: How the sector's labour challenges will shape its future*. *Agri-LMI Labour Market Information*.

Chamelat, R., Rosso, E., Choksuriwong, a., Rosenberger, C., Laurent, H., & Bro, P. (2006). Grape Detection By Image Processing. *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, 3–8. https://doi.org/10.1109/IECON.2006.347704

Chang, C., & Lin, C. (2013). LIBSVM : A Library for Support Vector Machines, 1–39.

Chire. (2011). DBSCAN Ilustration. Retrieved March 7, 2017, from https://commons.wikimedia.org/wiki/File:DBSCAN-Illustration.svg

Correa, C., Valero, C., & Barreiro, P. (2012). Characterization of vineyard's canopy through fuzzy clustering and SVM over color images. *International Conference of*

Agricultural Engineering. Retrieved from http://oa.upm.es/13687/

Creasy, G., & Creasy, L. (2009). *Grapes*. CABI.

Cubero, S., Diago, M. P., Blasco, J., Tardaguila, J., Millan, B., & Aleixos, N. (2014). A new method for pedicel / peduncle detection and size assessment of grapevine berries and other fruits by image analysis. *Biosystems Engineering*, 62–72. https://doi.org/10.1016/j.biosystemseng.2013.06.007

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection.

Diago, M. P., Tardaguila, J., Aleixos, N., Millan, B., Prats-Montalban, J., Cubero, S., & Blasco, J. (2014). Assessment of cluster yield components by image analysis. *Journal of the Science of Food and Agriculture*. https://doi.org/10.1002/jsfa.6819

Duda, R., Hart, P., & Stork, D. (2001). *Pattern Classification* (Second Edi).

Dunn, G. M., & Martin, S. R. (2004). Yield prediction from digital image analysis : A technique with potential for vineyard assessments prior to harvest, 196–198.

FAO. (2017). FAO STATS. Retrieved from http://www.fao.org/faostat/en/#data

FAO-OIV. (2016). *Table and Dried Grapes*.

Felzenszwalb, P. F., Girshick, R. B., Mcallester, D., & Ramanan, D. (2010). Object Detection with Discriminatively Trained Part Based Models, 1–20.

Fernández, R., Montes, H., Salinas, C., Sarria, J., & Armada, M. (2013). Combination of RGB and multispectral imagery for discrimination of Cabernet Sauvignon grapevine elements. *Sensors (Switzerland)*, *13*(6), 7838–7859. https://doi.org/10.3390/s130607838

Font, D., Tresanchez, M., Martínez, D., Moreno, J., Clotet, E., & Palacín, J. (2015). Vineyard Yield Estimation Based on the Analysis of High Resolution Images Obtained with Artificial Illumination at Night. *Sensors*, *15*(4), 8284–8301. https://doi.org/10.3390/s150408284

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition* (Second Edi).

Grossetête, M., Berthoumieu, Y., Costa, J., Germain, C., Lavialle, O., & Grenier, G. (2012). Early Estimation of Vineyard Yield: Site Specific Counting of Berries By Using a Smartphone, (September 2015), 1–6. Retrieved from http://cigr.ageng2012.org/images/fotosg/tabla_137_C1915.pdf

Guo, Z., Zhang, L., & Zhang, D. (2010). A Completed Modeling of Local Binary Pattern, *19*(6), 1657–1663.

Herrero-Huerta, M., González-Aguilera, D., Rodriguez-Gonzalvez, P., & Hernández-López, D. (2015). Vineyard yield estimation by automatic 3D bunch modelling in field conditions, *110*, 17–26. https://doi.org/10.1016/j.compag.2014.10.003

Huang, N., Fang, L., Cai, G., Xu, D., Chen, H., & Nie, Y. (2016). Mechanical Fault Diagnosis of High Voltage Circuit Breakers with Unknown Fault Type Using Hybrid Classifier Based on LMD and Time Segmentation Energy Entropy. *Entropy*, *18*(9), 322. https://doi.org/10.3390/e18090322

Inglese, P. (2015). DBSCAN Clustering Algorithm. Retrieved April 2, 2016, from https://www.mathworks.com/matlabcentral/fileexchange/53847-dbscan

International Virag. (2016). Research and Development. Retrieved from http://www.viraginternational.com/home/plant/research-and-development/#!

Ivorra, E., Sánchez, A. J., Camarasa, J. G., Diago, M. P., & Tardaguila, J. (2015). Assessment of grape cluster yield components based on 3D descriptors using stereo vision. *Food Control*, *50*, 273–282. https://doi.org/10.1016/j.foodcont.2014.09.004

Kicherer, A., Herzog, K., Pflanz, M., Wieland, M., Rüger, P., Kecke, S., … Töpfer, R. (2015). An Automated Field Phenotyping Pipeline for Application in Grapevine Research, 4823–4836. https://doi.org/10.3390/s150304823

Kirkpatrick, D. G., & Seidel, R. (1983). On the Shape of a Set of Points in the Plane. *IEEE Transactions on Information Theory*, *29*(4), 551–559. https://doi.org/10.1109/TIT.1983.1056714

Klodt, M., Herzog, K., Töpfer, R., & Cremers, D. (2015). Field phenotyping of grapevine growth using dense stereo reconstruction. *BMC Bioinformatics*, *16*(1), 11. https://doi.org/10.1186/s12859-015-0560-x

Kovesi, P. (2004). MATLAB and Octave Functions for Computer Vision and Image Processing. Retrieved from http://www.peterkovesi.com/matlabfns/

Lin, C. (2006). A Guide to Support Vector Machines.

Liu, S., Marden, S., & Whitty, M. (2013). Towards Automated Yield Estimation in Viticulture. *Conference on Robotics and Automation, 2-4 December 2013 (Australia)*, 9. Retrieved from http://www.araa.asn.au/acra/acra2013/papers/pap163s1-file1.pdf

Liu, S., Whitty, M., & Cossell, S. (2015a). A Lightweight Method for Grape Berry Counting based on Automated 3D Bunch Reconstruction from a Single Image. *Workshop on Robotic Agriculture*, 4.

Liu, S., Whitty, M., & Cossell, S. (2015b). Automatic grape bunch detection in vineyards with an SVM classifier. *International Conference on Machine Vision Applications*, *13*(4), 238–241. https://doi.org/10.1016/j.jal.2015.06.001

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*.

Loy, G., & Zelinsky, A. (2003). Fast radial symmetry for detecting points of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(8), 959–973. https://doi.org/10.1109/TPAMI.2003.1217601

Luo, L., Tang, Y., Zou, X., Wang, C., & Zhang, P. (2016). Robust Grape Cluster Detection in a Vineyard by Combining the AdaBoost Framework and Multiple. https://doi.org/10.3390/s16122098

Luo, L., Tang, Y., Zou, X., Ye, M., Feng, W., & Li, G. (2016). Vision-based extraction of spatial information in grape clusters for harvesting robots. *Biosystems Engineering*, *151*, 90–104. https://doi.org/10.1016/j.biosystemseng.2016.08.026

Malisiewics, T. (n.d.). Non-maximum Suppression. Retrieved from https://github.com/rbgirshick/rcnn/blob/master/nms/nms.m

Manfull, S. (2012). Provence Winezine. Retrieved from http://www.provencewinezine.com/enchanting-evening-in-provence-wine-tasting-at-cave-aureto-and-dinner-at-le-jardin-dans-les-vignes-in-la-coquillade-at-an-enchanting-price-too/

Ni, J., Singh, M. K., & Bahlmann, C. (2003). Fast Radial Symmetry Detection Under Affine Transformations.

Nuske, S., Achar, S., Bates, T., Narasimham, S., & Singh, S. (2011). Yield Estimation in Vineyards by Visual Grape Detection. *International Conference on Intelligent Robots and Systems (IROS '11)*, 7.

Nuske, S., Gupta, K., Narasimhan, S., & Singh, S. (2014). Modeling and calibrating visual yield estimates in vineyards. *Springer Tracts in Advanced Robotics*, *92*, 343–356. https://doi.org/10.1007/978-3-642-40686-7_23

Nuske, S., Wilshusen, K., Achar, S., Yoder, L., Narasimhan, S., & Singh, S. (2014). Automated Visual Yield Estimation in Vineyards. *Journal of Field Robotics*, *24*(5), 421–434. https://doi.org/10.1002/rob

OIV. (2015a). Global Economic Vitiviculture Data, (October), 5–9.

OIV. (2015b). State of the Vitiviniculture World Market. *38th OIV World Congress of Vine and Wine*, (April), 1–14.

Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns, 1–35.

Pavel, Š., & Runarsson, T. P. (2015). DETECTION OF GRAPES IN NATURAL ENVIRONMENT USING SUPPORT VECTOR MACHINE CLASSIFIER, (JUNE 2015).

Pence, R. A., & Grieshop, J. I. (1991). Leaf Removal in Wine Grapes: A Case Study in Extending Research to the Field, *32*(2), 519–527.

Pietikäinen, M. P. (2010). Local Binary Patterns. Retrieved from http://www.scholarpedia.org/article/Local_Binary_Patterns

Pratt, W. (2007). *Digital Image Processing* (Fourth Edi).

Quackenbush, J. (2017). Napa, Sonoma vineyard-worker scarcity sprouts wage growth, alternatives. *North Bay Business Journal*. Retrieved from http://www.northbaybusinessjournal.com/northbay/sonomacounty/6951644-181/napa-sonoma-vineyard-wine-employment?artslide=1

Rahman, A., & Hellicar, A. (2014). Identification of mature grape bunches using image processing and computational intelligence methods. *2014 IEEE Symposium on Computational Intelligence for Multimedia, Signal and Vision Processing*

(CIMSIVP), 1–6. https://doi.org/10.1109/CIMSIVP.2014.7013272

Reis, M. J. C. S., Morais, R., Peres, E., Pereira, C., Contente, O., Soares, S., … Cruz, J. B. (2012). Automatic detection of bunches of grapes in natural environment from color images. *Journal of Applied Logic*, *10*(4), 285–290. https://doi.org/10.1016/j.jal.2012.07.004

Rodriguez-Pulido, Gómez-Robledo, Melgosa, Gordillo, Gonzalez-Miret, & Heredia. (2012). Ripeness estimation of grape berries and seeds by image analysis, (March). https://doi.org/10.1016/j.compag.2012.01.004

Roscher, R., Herzog, K., Kunkel, A., Kicherer, A., Töpfer, R., & Förstner, W. (2014). Automated image analysis framework for high-throughput determination of grapevine berry sizes using conditional random fields. *Computers and Electronics in Agriculture*, *100*, 148–158. https://doi.org/10.1016/j.compag.2013.11.008

Saavedra, J. (2015). Reconocimiento De Patrones. *Notes Onf SVM*. Retrieved from http://galia.fc.uaslp.mx/~fac/pr/reconocimiento_patrones.pdf

SharkD. (2010). RGB Color Model. Retrieved from https://commons.wikimedia.org/wiki/File:RGB_Cube_Show_lowgamma_cutout_a.png

SharkD. (2015). HSL and HSV. Retrieved from https://commons.wikimedia.org/wiki/File:HSV_color_solid_cone.png

Subercaseaux, J. P., & Contreras, M. F. (2013). El gran desafío de la fruticultura, 16–21.

Tax, D. M. J., & Duin, R. P. W. (2004). Support Vector Data Description. *Machine Learning*, *54*(1), 45–66. https://doi.org/10.1023/B:MACH.0000008084.60811.49

Timmins, J. (2009). *Seasonal Employment Patterns in the Horticultural Industry*.

Tola, E., Lepetit, V., & Fua, P. (2010). DAISY : An Efficient Dense Descriptor Applied to Wide-Baseline Stereo, *32*(5), 815–830.

USDA. (2014). Fresh Deciduous Fruit (Apples , Grapes , & Pears): World Markets and Trade. *United States Department of Agriculture*, 9.

Vance, A. J., Reeve, A. L., & Skinkis, P. A. (2013). The Role of Canopy Management in Vine Balance. *Extention Service. Oregon State University*, 1–12.

Vedaldi, A., & Fulkerson, B. (2010). VLFeat - An open and portable library of computer vision algorithms, 1–4.

Velardo, C., & Dugelay, J.-L. (2010). Face Recognition with DAISY Descriptors.

Zeng, Q., Liu, C., Miao, Y., Fei, S., & Wang, S. (2008). A Machine Vision System for Continuous Field Measurement of Grape Fruit Diameter. *2008 Second International Symposium on Intelligent Information Technology Application*, 1064–1068. https://doi.org/10.1109/IITA.2008.274