ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

BASHAR SAAD FALIH AL-SAFFAR

IMPLEMENTATION AND PERFORMANCE EVALUATION OF CLASSIFIERS SVM, CNN AND ANN IN VINEYARD ESTIMATION

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADANA-2018

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

IMPLEMENTATION AND PERFORMANCE EVALUATION OF CLASSIFIERS SVM, CNN AND ANN IN VINEYARD ESTIMATION

BASHAR SAAD FALIH AL-SAFFAR

MSc THESIS

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of degree of the Master of Science by the board of jury on 30.12.2019.

………………………………………..	……………………………………………..
Assoc. Prof. Dr. Sami ARICA	Assoc. Prof. Dr. Turgay İBRİKÇİ
SUPERVISOR	MEMBER

…………………………………………...…….
Asst. Prof. Dr.Emine AVŞAR AYDIN
MEMBER

This MSc Thesis is written at the Department of Institute of Natural And Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

ABSTRACT

MSc. THESIS

IMPLEMENTATION AND PERFORMANCE EVALUATION OF CLASSIFIERS SVM, CNN AND ANN IN VINEYARD ESTIMATION

BASHAR SAAD FALIH AL-SAFFAR

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Supervisor : Assoc. Prof. Dr.Sami ARICA
Year: 2019, Pages: 69
Jury : Assoc. Prof. Dr. Sami ARICA
: Assoc. Prof. Dr. Turgay İBRİKÇİ
: Asst. Prof. Dr. Emine AVŞAR AYDIN

Grape detection, harvesting, spraying, and yield estimation are difficult, time consuming, and requiring considerable manpower. Therefore, harvesting with the aid of smart robots is now being explored and is currently a frontline problem in agriculture.

This thesis presented a grape bunch and berry recognition by employing Convolution Neural Network (CNN), Artificial Neural Network (ANN) and Linear Support-Vector-Machine (SVM). Histogram of Oriented Gradients (HOG), the Local Binary Pattern (LBP), and combination have been used as feature vectors. The data contain berry and non-berry classes. ANN classifier with HOG+LBP features show better results in single berry recognition with an average of accuracy, precision and recall are 99.64±0.073 %, 99.70±0.094%, and 99.59±0.117% respectively. The proposed detection method uses a Fast-Radial Symmetry Transform (FRST) as interest point detector. Feature extraction and classification are computed in each interest point approach. Afterwards, a DBSCAN method defines the number of grape bunches. Each cluster's spatial distribution and needs to close bunch separation for an improved bunch detection. The average accuracy of grape bunch detection 91.72±2.88% and 89.82±2.58% for berry detection. Results show a good performance based on the shape and texture information, with no need for special illumination and color features.

Keywords: Grape detection, grape bunch detection, HOG, LBP, DBSCAN, SVM, ANN and CNN

ÖZ

YÜKSEK LİSANS TEZİ

ÜZÜM SALKIMI MEYVELERİNİN TANINMASI AMACIYLA DVM, ESA VE YSA SINIFLAYICILARININ GERÇEKLEŞTİRİLMESİ VE BAŞARILARININ BELİRLENMESİ

BASHAR SAAD FALIH AL-SAFFAR

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Doç. Dr.Sami ARICA
Yıl: 2019, Sayfa: 69
Jüri : Doç. Dr. Sami ARICA
: Doç. Dr. Turgay İBRİKÇİ
: Dr. Öğr. Üyesi. Emine AVŞAR AYDIN

Üzüm tespiti, hasat, ilaçlama ve verim tahmini çiftçiler için zor faaliyetlerdir. Bu işlem zaman alıcıdır ve işgücüne ihtiyaç duymaktadır ve üstelik her zaman yeterince başarılı değildir.

Bu tezde, makine öğrenme yöntemlerinden Evrişimsel Sinir Ağı (ESA), Yapay Sinir Ağı (YSA) ve Destek-Vektör-Makine- (DVM) kullanılarak bir üzüm salkımı ve meyvesinin tanınması üzerine bir çalışma sunulmuştur. Yerel İkili Örüntü (YİÖ), Yönlendirilmiş Gradyentlerin Histogramı (YGH) ve bileşimleri özellik vektörleri olarak kullanılmıştır. Kullanıma açık İzlanda veri seti, tek bir üzüm tanesinin algılama ve sınıflandırma eğitimi için uygulanmıştır. Veriler üzüm tanesi olan ve olmayan iki sınıf içerir. Üzüm tanesi tanıma işleminde YGH+YİÖ kombinasyonu özellikleri kullanan YSA sınıflandırıcısının ortalama doğruluk, hassasiyet ve duyarlılık ölçütleri sırasıyla %99,64 ± 0.073, %99.70 ± 0,094 ve %99,59 ± 0,117'dir ve elde edilen en başarılı sonuçlardır. Üzüm salkımı ve meyve saptama için önerilen yöntem, Hızlı Radyal Simetri Dönüşümü üzün tanesi merkezini bulmak için kullanır. Önce özellik çıkarımı ve sınıflandırma, her bir ilgi noktasında çok ölçekli bir yaklaşımla hesaplanmaktadır. Daha sonra, DBSCAN metodu üzüm demetlerini kümeler. Her kümenin uzamsal dağılımı için uzaklık dönüşümünden faydalanılır. Üzüm salkımı doğruluğu % 91,72 ± 2,88 ve tanesi % 89,82 ± 2,58 ortala doğrulukla bulunmuştur.

Anahtar Kelimeler: Üzüm Tespiti, Üzüm Salkımı Tespiti, YGH, YİÖ, DBSCAN, DVM, YSA ve ESA.

EXTENDED SUMMARY

The main objective of this study is to detect grape berries and bunches in the images. The recognition and counting of grape in real sight of the images are one of the main issues should be solved in the viticulture.

The world in grape production is facing many problems due to the lack of workers, costs, and the increment of costs of employment which effects on grape quality, productivity, crop tracking and harvesting time.

This study presents an alternative solution trying to reduce these problems. The use of new technology in the wine industry and grape has been an important research subject due to the relevance of precision viticulture. This area of study searches for the tracking of relevant variables and data to obtain repeatable process, besides real-time information for rapid response and improving the quality of grapes, while also reducing the environmental impact and operational costs. The data obtained by Red, Green and Blue (RGB) sensors may help growers make more informed decisions and manage their field in a better way, handling the high variability in their crops.

The use of robotic systems for the automation of multi wine and grape related processes in the field is one of these technologies that could help us to increase the accuracy of detection. This will include crop monitoring, management, and help in solving the existing difficulties of a worker shortage and increasing worker costs. The robotic in agricultural field applications include three main parts: Hardware (mapping and guidance), mobility algorithms, effector action stage, and end perception system. This study focuses on the grape bunch's detection and berry recognition for any robotics. The grape bunches detection in yield has been a very useful challenge that until now have not been solved completely yet.

Related works have shown that segmentation using features depending on the colors is not accurate for detecting white grapes. Although the features that depend on the colors yield good accuracy in red berry types. This is applicable near harvesting only after the fruit has been changed color and matured, also limiting the number of tasks in which an autonomous unit could operate. Most of the related research, including machine vision applications in viticulture, is aimed to solve the difficulties of vineyard estimation. Color features in different color spaces (RGB, HSV, L*a*b*) have been used to detect white grapes and red grapes. The detection based on the color feature typically shows better results for red grape when compared with white grape because of the color similarity between grapes background and leaves in white grape. This study tries to adopt a new method or improving such one that may be capable to contribute in grape bunches detection and berry recognitions using lateral digital image processing and computer vision techniques under varying illumination, different foliar density and various occlusion in images. To this end, the different algorithm will use to improve the grape bunches detection and berry recognition with the highest true positive detection rates with low false positives.

Automatic grape bunch detection and berry recognition are becoming a major area of researches in agricultural production. The research highlighted on grape bunch detection and berry recognition strategies involving machine-learning such as Artificial Neural Network (ANN), Support Vector Machine (SVM)-linear as a classifier with HOG, LBP and the combination of both (HOG+LBP) as a feature descriptor. Additionally, the Convolutional Neural Networks (CNN) will extract the feature from the image automatically and classify them. With the recent success of Deep Learning and the availability of better hardware and big data-sets, deploying deep learning to object detection is being widely researched, and many notable solutions have evolved over the last few years. The chosen algorithm for grape detection must also be feasible for real-time detection for applications including grape berry detection. In this algorithm when huge training-set gave us better performance. 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. The results of single grape recognition show that recognition of accuracy is 94.66%, 99.64%, and 99.03% with CNN, ANN with HOG+LBP, and SVM-linear with HOG+LBP respectively. So, the results of ANN with a combination of both HOG and LBP have been chosen for grape berry detection because gave better performance and lower time-consumption. The approach has been implemented in the freely available Iceland dataset that has a total number of images 6880 images half of them positive and another half negative, the experiment results recorded with 65% and 35% amount of train and test respectively, with hundred runs.

The validation images were chosen in this experiment collected from (Cukurova university-Agriculture facility (Red grapes), Iceland (White grapes) and Tarsus (Red and White grapes)) the pictures are picked in natural illumination to record real and practical results.

The first stage of the approach consists pre-processing of the images to convert the RGB image to grayscale image and Histogram Equalization will be used to equalize the intensity values, to improve the feature extraction of grape berry and in the interest points detection. The output of equalized images will be the input to the down-sampling, or images resized to speed up the salient point detection on a bunch of grapes. The second stage interest-points detection is performed by utilizing the Fast-Radial Symmetry Transform (FRST) to find the interest points in images that have circular shapes and image smoothing in order to select the correct center of the berry by Non-Maximum Separation (NMS). The ANN classifier with (HOG+LBP) will be used to separate them as berry or non-berry. The output of the classifier will decide each cropped image as a grape berry and non-grape berry. Next, to reduce the misclassification or classifier error by the aid of DBSCAN clustering algorithm, to select the bunches of grapes that have more than 5 berries of grape with distance among them two times of grape berry diameter else the DBSCAN clustering will remove them or the results will be as a noise. The last stage is the separation of Neighbored bunches, if more than one grape bunch close together or within Epsilon, the DBSCAN clustering may cluster them as a one grape bunch and this situation will affect negatively on the accuracy of estimation, so the neighbor grape bunch separation is needed, here the approach have been used to convert the detected mask to the binary and the detected area will be a logic (one) as a grape and background will be a logic zero, by finding the pixel distance, thresholding the image, labeling the white area it will be the number of bunch in this detected bunch will by use in K-means clustering.

The average accuracy of grape bunch detection 91.72±2.88% and 89.82±2.58% for berry detection. Grape bunch detection and a comparison of berry and non-berry pixels was studied, using ground-truth the labels are made manually. Four different colors of grape images, with diverse illumination and acquisition protocol were tested, with an average of correct rate, sensitivity, and specificity are 96.97±0.95%, 97.57±0.969%, and 89.28±3.83% respectively in pixels of berry classification. Results show good performance with no need of special illumination, not require color feature used which allows recognition for white and red grapes as well as the bunch detection scheme working in varying scenarios.

The approach presented performs a multi-scale 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 (FP) on isolated locations, thus improving further the accuracy, sensitivity and precision performance indices. 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.

GENIŞLETILMIŞ ÖZET

Bu çalışmanın temel amacı, resimlerdeki üzüm meyvelerini ve salkımları tespit etmektir. Resimlerde üzümlerin tanınması ve sayılması bağcılıkta çözülmesi gereken ana sorunlardan biridir.

Dünya üzüm üretimi işçi kıtlığı, maliyetler, üzüm kalitesini, verimini, mahsul takibini ve hasat süresini etkileyen işçilik maliyetlerinin artışından dolayı birçok sorunla karşı karşıyadır.

Bu çalışma, bu sorunları azaltmaya çalışan alternatif bir çözüm sunmaktadır. Şarap endüstrisinde ve üzümde yeni teknolojilerin kullanılması ile hassas bağcılık yapılması önemli bir araştırma konusu olmuştur. Bu çalışma, çevresel etkiyi ve işletme maliyetlerini en aza indirirken, gerçek zamanda okuma ve analiz yapması ve ürün takibi ile verimliliği ve kalitesi artırılmış üzüm üretimini desteklemektedir. Kırmızı, yeşil ve mavi (KYM) spektral bandındaki görüntülerin analizi ile elde edilen sonuçlar, üreticilerin mahsullerindeki yüksek değişkenliği izleyerek daha bilinçli kararlar almalarına ve tarlalarını daha iyi yönetmelerine yardımcı olabilir.

Renkli resimlerden üzüm sayımı ve verimliliğin belirlenmesi tarlada üzümle ilgili işlemlerin otomasyonu için robotik sistemlerde kullanılabilir ve algılama doğruluğunu arttırmamıza yardımcı olabilecek teknolojilerden biridir. Bu, mahsul izlemesini, yönetimini ve bir işçi temininindeki zorluklarının çözülmesine ve işçi maliyetlerinin azalmasına yardımcı olacaktır. Tarımsal alan uygulamalarında bir robot üç ana bölümden oluşur: donanım (haritalama ve yönlendirme), hareketlilik algoritmaları, sonuç alan eylem aşaması ve son algılama sistemi. Bu çalışma, üzüm salkımının herhangi bir robotik için algılama ve meyvenin tanınmasına odaklanmaktadır. Üzüm salkımlarının tespiti önemli bir işlevdir ve şimdiye dek henüz tam olarak çözülmemiştir.

Renklere bağlı olan özellikler kırmızı üzüm çeşitlerinde iyi bir doğruluk sağlasa da, ilgili çalışmalar, renklere bağlı özellikleri kullanarak bölütlemenin beyaz üzümleri tespit etmek için yeterli olmadığını göstermiştir. Bu yöntem, hasat yapmaya yakın zamanda meyvenin rengini değiştirdikten ve olgunlaştıktan sonra otomasyon sistemin işlediği sınırlı alanlarda uygulanabilir. Bağcılıkta yapay görme uygulamaları da dâhil olmak üzere ilgili araştırmaların çoğu üzüm verimliliğinin tahmininin zorluklarını çözmeyi amaçlanmaktadır. Beyaz üzümleri ve kırmızı üzümleri tespit etmek için farklı renk uzaylarındaki renk özellikleri (KYM, HSV, L * a * b *) kullanılmıştır. Renk özelliğine dayalı tespit tipik olarak,beyaz üzümün arka planı ile yaprakları arasındaki renk benzerliği nedeniyle, kırmızı üzümde daha iyi sonuçlar verir. Bu araştırma, farklı aydınlatma, farklı yaprak yoğunluğu ve görüntülerde çeşitli nesne ötüşmeleri koşullarında gerek duyulan dijital görüntü işleme ve bilgisayarlı görme teknikleri kullanılarak üzüm salkımlarının algılanmasına ve meyvelerinin tanınmasına katkıda bulunabilecek yeni bir yöntem geliştirmeye ve iyileştirmeye çalışmaktadır. Bu maksatla, farklı bir algoritma, üzüm salkımlarının algılanmasını iyileştirmek ve hatalı okuma oranını azaltmak ve doğru okuma oranını arttırmak için geliştirilecektir.

Otomatik üzüm salkımı tespiti ve meyve tanıma, tarımsal üretimde önemli bir araştırma alanı haline gelmektedir. Bu araştırma, yerel ikili örüntü (YİÖ), yönlendirilmiş gradyentlarin histogramı (YGH) ve her ikisinin (YGH + YİÖ) kombinasyonunu özellik tanımlayıcısı olarak ve üzüm tanesi tanıma yöntemi olarak yapay sinir ağı (YSA) gibi makine öğrenmesi metodu, doğrusal sınıflayıcı olarak YGH ve YİÖ özeliklerini ayrı ayrı ve (YGH+YİÖ) kombinasyonunu kullanan destek vektör makinesi (DVM üzerinde yoğunlaşmıştır. Bunlara ek olarak, evrişimli sinir ağları (ESA), özelliği otomatik olarak görüntüden çıkaracak ve sınıflandıracaktır. Derin öğrenmenin son zamanlardaki başarısı ve daha iyi donanım ve büyük veri setlerinin mevcudiyeti ile derin öğrenmenin nesne algılamak için kullanılması yaygın bir şekilde araştırılmaktadır ve son birkaç yılda çok sayıda kayda değer çözüm geliştirilmiştir. Üzüm saptama için seçilen algoritma, üzüm tanesi belirlemeyi de içeren uygulamalar için gerçekleştirilebilir ve gerçek zamanlı olmalıdır. Bu algoritmada büyük eğitim seti bize daha iyi performans vermektedir. Bu nedenle, araştırmalar, yetiştiricilere çok faydalı olabilecek, doğal ortamda alınmış renkli görüntüleri kullanarak üzüm demetlerinin tespitinde hala geliştirme için yer olduğunu göstermektedir. Tek üzüm tanesi tanıma sonuçları, sırasıyla ESA için %94,66, (YGH + YİÖ) ile YSA için %99,64 ve (YGH + YİÖ) ile DVM için %99,03 olduğunu görülmüştür. . Bu nedenle, YSA'nın YGH ve YİÖ bileşiminin elde ettiği sonuçlar üzüm meyvesi tespiti için önerilmektedir, çünkü en düşük hesaplama zamanı en iyi doğruluk yüzdesi sağlamaktadır. Bu yaklaşım, herkese açık İzlanda veri setinde bulunan yarısı üzüm tanesi, diğer yarısı üzüm tanesi olmayan toplam 6880 görüntüde uygulanmıştır. Deney sonuçları toplam görüntünün %65 i eğitim ve %35 ini test grubu olarak kullanma suretiyle elde edilmiştir. Doğal ortamda çekimi yapılmış olan, Çukurova Üniversitesi Ziraat Fakültesi Bahçe Bitkileri bölümü tarafından sağlanan kırmızı üzüm ve Tarsus çevrisinden alınan kırmızı ve beyaz üzüm görüntüleri ile İzlanda veri seti (beyaz üzüm) bu deneyde, gerçek ve pratik sonuçları elde etmek için kullanılmıştır.

Yaklaşımın ilk aşaması, ilgi noktası ve özellik çıkarma işlemlerini daha iyi hale getirmek için, KYM görüntüsü gri tonlamalı görüntüye dönüştürme ve piksel büyüklüğü dağılımını histogram eşitleme yöntemiyle eşit dağılımlı hale dönüştürme işlemlerinden oluşmaktadır. Bir üzüm salkımımdaki belirgin noktaları hesaplama işlemini hızlandırmak için histogramı eşitlenmiş görüntülere örnek azaltma yâda boyut küçültme işlemi uygulanmaktadır. İkinci aşamadaki üzüm tanelerinin merkez noktalarını belirlemek için hızlı radyal simetri dönüşümü uygulanmakta, yumuşatma işleminden sonra maksimum olmayanı ayırma (MOA) yöntemiyle doğru merkez belirlenmektedir. (YGH + YİÖ) özellik vektörü ile YSA sınıflandırıcısı bu merkez noktaları temel alan ve önceden belirlenmiş çaptaki/genişlikteki yamayı meyveli veya meyvesiz olarak ayırmak için kullanılmaktadır. Sınıflandırıcın, kırpılmış her görüntünün bir üzüm tanesi olup olmadığına beş yâda daha fazla üzüm tanesi içeren grubu bir üzüm salkımı, küme aralarında iki üzüm tanesi çapı uzaklığı olan grupların ayrı olduğuna DBSCAN algoritması aracılığı ile karar verilmektedir. DBSCAN kümeleme algoritması aralarında iki üzüm salkımımın çakışması halinde aralarındaki uzaklığa bakmakta ve iki üzüm tanesi çapından daha büyük bir mesafe sahip olması halinde ayrı iki salkım olarak değerlendirmekledir aksi durumda tek bir üzüm salkımı olarak algılamaktadır ve buda başarıyı olumsuz etkilemektedir. Bu durumu önlemek için resim ikili görüntüye dönüştürülerek uzaklık dönüşümü uygulanmakta eşileme sunucu oluşan ayrık bağlantılı bölgelerin sayısı üzüm, salkımımı sayısını vermektedir. Bu sayı k-ortalamalar kümelemesinde grup sayısı olarak alınmakta ve kümeleme yapılmaktadır. Bu kümeleme sonucunda üzüm salkımları tespit edilmiş olmaktadır.

Üzüm salgılaması sayısının belirmesi ortalama %91,72 ± 2,88 doğrulukla ve üzüm tanesi tespiti ortala %89,82 ± 2,58 doğrulukla bulunmuştur. Tespit edilen üzüm ve elle işaretleme ile belirlenmiş üzüm piksellerinin çakıştırılması/karşılaştırılmasıyla da başarı hesaplanmıştır. Farklı okuma ve ışık koşullarında alınmış dört farklı renkteki üzüm görüntüleri analiz edilmiş ve piksel temelli başarı ölçütüne göre ortalama doğruluk, hassasiyet ve seçicilik sırasıyla %96,97 ± 0.95, % 97,57 ± 0,969 ve % 89,28 ± 3,83 bulunmuştur. Sonuçlar bu yöntemim, farklı ışık koşullarında ve şartlarda alınan beyaz ve kırmızı üzüm tespitinde başarılı olduğunu göstermektedir.

Önerilen yaklaşım farklı ölçeklerdeki alınmış görüntüler içeren verilerde çalışmaktadır. İşlem süresindeki bedele karşılık olarak bu yöntemi kamera ile asmalar arasındaki mesafe, üzüm salkımı ve tanesi büyüklüğü, ışık şartları ve kamera donanımı farklılıklarına karşı dayanıklı yapmaktadır. Ayrıca, DBSCAN kullanılan kümeleme aşamasında yalıtılmış konumlardaki hatalı bir şekilde üzüm olarak tespit edilen belgeler filtrelenmekte ve böylece doğruluk, hassasiyet ve kesinlik ölçütlerini iyileştirmektedir. performans endekslerini daha da geliştirir. Bu yöntem robotik hasat, yaprak temizleme, zararlı bitki temizleme ve inceltme veya seçici ilaçlama uygulamalarının geliştirilmesinde ve ayrıca etkin bağ yönetimi için önemli olan verim tahminlerinin geliştirilmesinde çok faydalı olabilir.

ACKNOWLEDGEMENTS

I want to express my heartfelt thanks to Asst. Prof. Dr. Sami ARICA for his guidance, extremely useful suggestions, supervision, encouragements and continuous confidence in me during my study in this thesis. It has been great respect to have Asst. Prof. Dr. Sami ARICA as a supervisor.

I owe my deepest thanks to Çukurova University Engineering-Architecture Faculty, Electrical and Electronic Engineering Department that enabled me to benefit from all facilities of the department during my thesis study.

Finally, I would like to express my deepest gratitude to my family and my friends (Especially to Mahmood Abed, who helped me apply to this prestigious university), for patience, encouragement and continuous moral support.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

LIST OF ABBREVIATIONS

FAOSTAT	: Food and Agriculture Organization of the United Nations
IOV	: International Organization of Vine and Wine
ANN	: Artificial Neural Network
CNN	: Convolution Neural Network
SVM	: Support Vector Machine
DBSCAN	: Dense-Based Spatial Clustering of Applications with Noise
NMS	: Non-Maximum Separation
HOG	: Histogram of Oriented Gradients
LBP	: Local Binary Pattern
RGB	: Red, Green and Blue
HSV	: Hue, Saturation and Value
HE	: Histogram Equalization
DL	: Deep Learning
ReLU	: Rectified Linear Unit
FRST	: Fast Radial Symmetry Transform
DNN	: Deep Neural Network
Minpts	: Minimum Number of Points
Eps	: Epselon
Hz	: Hertz
RAM	: Read And Write Memory
CPU	: Center Processing Unit
TP	: True positives
TN	: True negatives
FP	: False positives
FN	: False negatives
Acc	: Accuracy
Sens	: Sensitivity

Rec	: Recall
TPR	: True Positive Rate
FPR	: False Positive Rate
Sec	: Second Precision
ROC	: Receiver Operational Characteristic
Prec	: Precision
DL	: Deep Learning
DNN	: Deep Neural Networks

LIST OF SYMBOLS

Im	:	Digital image
N	:	Columns number in image
M	:	Rows number in image
b	:	Number of bits
Ω	:	Image domain
k	:	Gray level
n_k	:	The number of occurrences of pixel with gray level k
CDF	:	The cumulative distribution function
d_i	:	Symmetry of radial at a distance
g	:	Image gradient
P_{+ve}	:	Positive affected
P_{-ve}	:	Negative affected
O_d	:	The image orientation matrix
M_d	:	Image magnitude matrix
α	:	Radial strictness
K_d	:	Scaling factor for distance d
ξ_i	:	misclassification
w	:	Weight
b	:	Bias
Eps	:	Epsilon or radius of points
$Minpts$	:	Minimum Number of Points

1. INTRODUCTION AND BACKGROUND

1.1. Introduction

The main objective of the study is to detect berries and recognize bunches in images. The recognition and counting of visual grape bunches or grape berries in real sight of the images is one of the main issues should be solved in the viticulture. Food and Agriculture Organization of the United Nations. (FAOSTAT), reported that the highest productive fruit in Turkey is grape. Figure 1.1. It shows the production of fruit in Turkey and Figure 1.2. It shows the harvested area of fruit in Turkey (FAOSTAT, 2017), as well as the increase of the grape production in all the world annually. The International Organization of Vine and Wine (IOV), reported that the grape production is higher than 76.7 million tons of grapes with harvested surface area of 7.4 million hectares in the world. 3.65 million tons of grape production with harvested surface area 0.496 million hectares with exports grapes 1.75 million tones (47.9% out of the total production) in the Turkey (IOV, 2015), 4 million tons of grape production with harvested surface area of 0.467 million hectares with exports grapes 1.73 million tones (43.3% out of the total production) in Turkey (IOV, 2016). Turkey is one country contribute about 5.3% out of the world in grape production. That shows the production of fresh grapes, dried grapes, and wine grapes represent the biggest economic activity in all the world including in Turkey.

The world in grape production is facing many problems related to a worker shortage, costs, and an increase in costs of employment (Quackenbush et al., 2017), (Subercaseaux, 2013), which on the affecting grape quality, productivity, crop tracking and harvesting time. In addition, the manual harvesting more expensive, destructive, consuming of time, bias by labors and inaccurate (Grossetête, 2012), (Nuske, 2014). Because of these difficulties, the agricultural field needs to implement a new technology to help us in the production and the different aspects such as time, accuracy, quality and productivity.

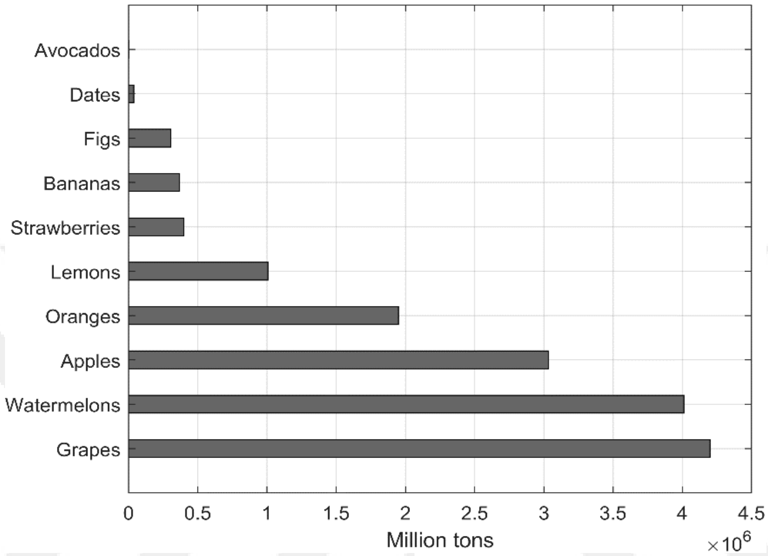


Figure 1.1. The Fruit Production in Turkey (FAOSTAT, 2017)

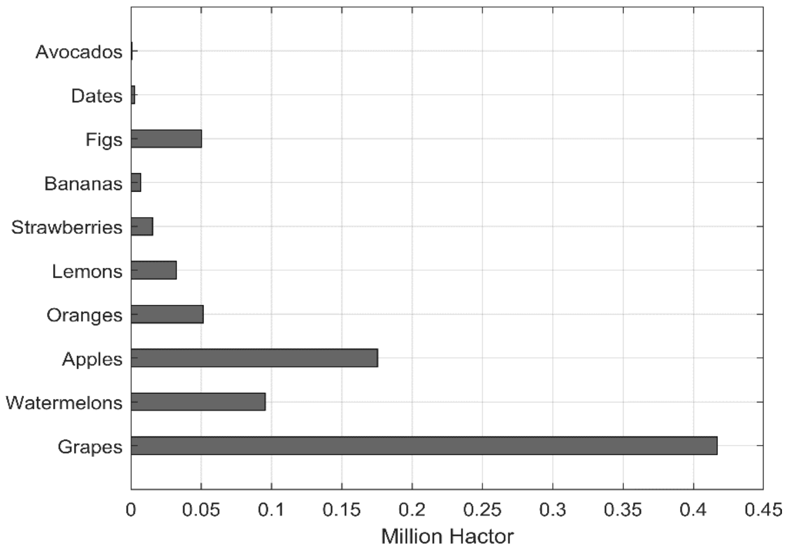


Figure 1.2. The Harvested area in Turkey (FAOSTAT, 2017)

That require to find an alternative solution trying to reduce these problems. The use of new technology in the wine industry and grape has been an important research subject due to the relevance of precision viticulture. This area of study searches for the tracking of relevant variables and data in order to obtain repeatable process, in addition to real-time information for rapid response and improving the quality of grapes, while also reducing the environmental impact and operational costs (Fernández et al., 2013). The data obtained by Red, Green and Blue (RGB) sensors may help growers make more informed decisions and manage their field in a better way, handling the high variability in their crops.

The use of robotic systems for the automation of multi wine and grape related processes in the field is one of these technologies that could help us to increase the accuracy of detection. This will include crop monitoring, management, and help in solving the existing difficulties of a worker shortage and increasing worker costs. The robotic in agricultural field applications include three main parts: Hardware (mapping and guidance), mobility algorithms, effector action stage, and end perception system (Auat Cheein et al., 2013), (Bachche, 2015). This study focuses on the visual grape bunches detection and berry recognition for any robotics. The grape bunches detection in yield has been a very useful challenge that until now have not been solved completely yet.

1.2. Related Works

Related works have shown that segmentation using features depending on the colors is not accurate for detecting white grapes. Although the features that depend on the colors yield good accuracy in red berry types (Liu et al., 2015b). This is applicable near harvesting only after the fruit has been changed color and matured, also limiting the number of tasks in which an autonomous unit could operate.

Depending on illumination reflection for berry recognition (Grossetête et al., 2012), (Nuske et al., 2014), this may produce different results because of the varying in luminescence some grapes present (Diago et al., 2014) or different light reflections caused by weather conditions such as, rain over the fruit, spraying, generating multiple reflection points. To tackle problems caused by changes in natural illumination conditions, some methods employ specialized illumination hardware to operate at night time (Nuske et al., 2014), which may imply higher operational costs.

Most of the related research, including machine vision applications in viticulture is aimed to solve the difficulties of vineyard estimation. Color features in different color spaces (RGB, HSV, L*a*b*) have been used to detect white grapes and red grapes. The visual grape detection based on the color descriptors typically shows better results for red grape when compare it with white grape because of the similarity of white grapes color with background and leaves.

(Liu et al., 2013) analyzed the relationship between several variables such as perimeter, pixel area, size and grape number. There is a relation to the actual grape bunch weight, in order to determine the best estimation of the vineyard. Later the same authors proposed another method for detecting grape bunches of red color by employing segmentation in the HSV color space, (Liu et al., 2015b). At vector of features containing the information of bunch location, texture and bounding box pixel distribution is passed to a previously trained SVM classifier. Another similar study proposed by (Luo et al., 2016) presented 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.

A more sophisticated algorithm is proposed by (Chamelat et al., 2006) where HSV channel information along with Zernike moments are used to describe grape shapes and train a Support Vector Machine (SVM) classifier in order to identify grapes in images using a sliding window. Aim of automatic detecting grapes in this research is to implement a robotic system for harvesting, spraying, and yield estimation.

(Škrabánek, 2015) designed a white grape detection method using a sliding window technique over the image of yield rows. A Support Vector Machine (SVM) with radial basis function (rbf) have been used as a classifier. Results show that the combination of Histogram of Oriented Gradients (HOG) and Local Binary Pattern (LBP) as a feature vector show better results.

Vineyard estimation by 3D reconstruction of the bunch is done by (Herrero-Huerta et al., 2015), considering the volume, mass and number of berries in each bunch. The approach employs images from five different angles to reconstruct the bunch of grapes. A similar study is presented by (Liu et al., 2015a), where the reconstruction of 10 bunches of grapes is done using one image only, under laboratory conditions. Here they used color segmentation to extract the grape berry from the background and applied circular Hough Transform to identify individual grapes and start creating the 3D model. In (Ivorra et al., 2015), a 3D surface is obtained using stereo cameras under ideal conditions in order to assess a bunch of grape parts related to vineyard, mainly compactness (which affects the quality of the grapes that do not receive enough sunlight in the interior of the bunch). An approach presented by (Rodrigo Pérez-Zavala et al., 2018), to grape bunch and berry detection, is uses the extract of the interest points by Fast Radial Symmetry Transform (FRST) technique. Feature extraction done by different types like (DSIFT, DAISY, HOG, LBP and combination with them) with SVM and Support Vector Data Descriptor (SVDD) as a classifier. The results show that the combination of HOG+LBP features with SVM-rbf give better results has been utilized for classification. Another authors (P. Śkrabánek, 2018), have used deep convolutional network with automatic feature extraction to classify singe white grape berry from full color (RGB) low-resolution image.

1.3. Automatic Grape Detection applications

There are many applications of automatic grape detection, they are briefly explained below:

1.3.1. Grape Harvesting

Harvesting of grape is done by laborers or by harvesting machines that strike the vine as shown in Figure 1.3. The problems with harvesting manually are labors, effort, need a number of workers. Besides, time consuming. In addition, the rising costs of worker and the fact that many seasonal field workers are moving to longer term position in other industries. Thus, farmers must face frequent problems to harvest on time and grape quality.

Further, harvesting by use machine is not suitable for all kinds of grapes, especially harvesting of table grapes and champagne grapes were the fruit damage causes accelerated oxidation, affecting the final product's grape quality (Chamelat, 2006 et al.,), (Reis et al., 2012).

(a)

(b)

Figure 1.3. Grape harvesting (Michael James, 2018). (a) Harvesting by machines. (b) Harvesting by the worker.

1.3.2. Vineyard Estimation

Accurate and early the estimation of vineyards is very important in agriculture, specially to the owner or farmers because it is used in many things such as; input for other logistic operations, storage time of harvesting, production processes, sales, irrigation and transportation of it. Growers also use historical data like materials used in previous time to improve production according to previously produced quantities.

1.3.3. Spraying of the Grape Bunches

Spraying of fertilizers and nutrients, pesticides, hormones is a very important task that must be done during the different stages of growth of the plant, and must be done on time to protect the crop from pests, diseases and to supply them with the necessary nutrients for ensuring higher quality of grapes. Spraying by hand is need long time and inaccurate. Beside of hand spraying is dangerous for the workers because of possible contact with toxic chemical agents. The spraying by using helicopters has need a high cost (expensive), low accuracy and loss of pesticides due to volatilization in the air. At this time, the spraying by mechanical machines 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 grape bunches spraying of or the canopy as needed. Figure 1.4. Represents the spraying types.

(a)

(b)

(c)

Figure 1.4. The spraying type (Gallery images and information: Pesticides In Water Pollution) (a) The spraying by helicopter (b) Spraying by the mechanical machine and (c) Spraying by the worker.

1.4. Goals of Study

To adopt a new or improve a method that may capable contribute in visual grape berries and bunches detection using lateral digital image processing and computer vision techniques under varying illumination, different foliar density and various occlusion in images. To this end, different algorithm will use to improve the grape bunches detection and berry recognition with highest True Positive (TP) detection rates with low False Positives (FP).

1.5. Thesis Organization

Automatic visual grape berry bunches and detection are becoming a major area of researches in the agricultural production. The research highlighted on grape bunch detection and berry recognition strategies involving machine learning such as Artificial neural network (ANN), Support Vector Machine (SVM) as a classifier with Histogram of Oriented Gradients (HOG), and Local Binary Pattern (LBP) as a feature descriptor. Additionally, the Convolutional Neural Networks (CNN) will extract the feature from the image automatically and classify them. With the recent success of Deep Learning (DL), and the availability of better hardware and big data-sets, deploying deep learning to object detection is being widely researched, and many notable solutions have evolved over the last few years in agriculture field. The chosen the algorithm for visible grape detection must also be feasible for real time detection for applications including grape berry and bunch detection and counting for yield estimation. Therefore, studies show that there is still room for improvement in the detection of grape bunches using full color images with natural illumination, which could be very useful to growers.

1.6. Thesis Outline

Chapter 2. Highlight on the digital image processing, feature extraction, classifiers type and clustering.

Chapter 3. Highlights on the proposed method, including image processing and machine learning.

Chapter 4. Implementation the experiment, dataset description, results and discussion.

Chapter 5. Conclusion and future work.

2. IMAGE PROCESSING

Solving berry detection problem requires beat on some main problems, like differing dimensions of the image and qualities, different light levels, and applying a sufficient number of images in each method. That is why, it is important to apply pre-processing stage on the images before processing it. In this work, Histogram Equalization (HE) technique and alignment of the dimensions (image resizing) have been employed to give us better performance in object detection. The employed techniques have been summarized below.

2.1. Digital Image processing

A digital image Im 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. See the mathematical equation (2.1).

$$Im\colon (\mathrm{i},\mathrm{j}) \in \Omega \subset \mathbb{N}^2 \rightarrow Im(\mathrm{i},\mathrm{j}) \in \mathrm{L} \subset \mathbb{Z}_+ \quad (2.1)$$

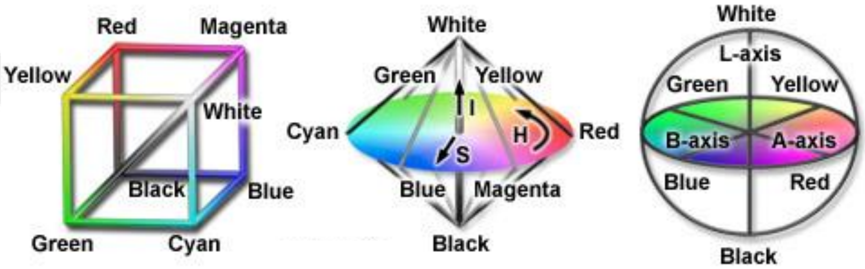
The image domain Ω is a set of $M{\cdot}N$ coordinates, where N is the columns number and M is the rows number of the imaging sensor. The intersection of any row with a column, Ω (i, j), is denominated a picture element, or better known as pixels The pixel intensity values lie in a range $L_{range} =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

$$\Omega = \{\ (\mathrm{i},\mathrm{j}) \in \mathbb{N}^2 : 1\ \leq \mathrm{i} \leq \mathrm{M}, 1 \leq \mathrm{j} \leq \mathrm{N}\ \} \quad (2.2)$$

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 allow one of the primary color components 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*b}. For b=8 bits, this corresponds to more than 16.7 million colors. RGB images have just one of several color spaces that may be used to represent digital images; see Figure 2.2.

2.1.1. Color Spaces

The red, green and blue (RGB) model of color is a standard design in computer graphics systems for all of its applications. The RGB color parts are highly correlated. This makes it is difficult to implement the image processing. So, that require to find an alternative. A different color processing technique depend on the intensity component of an image only. A color space Hue, Saturation and Value (HSV). This representation originates from psychophysical color theories perception by humans (Pratt, 2007) and thus is more intuitive in the domain of the graphic design and the arts, because the hue is related to the color of the wavelength, saturation is related to color of intensity relative to a color-less white-gray-black light and value is associated with the overall intensity, therefore is easier to interpret than a color formed by the RGB model. The HSV color space is defined by a nonlinear model in terms of a cylindrical coordinate system or color component, is represented as the angular coordinate from [0, 360], the Saturation 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 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 ranging from [-128, +127]. Figure 2.2 that displayed the type of color space.

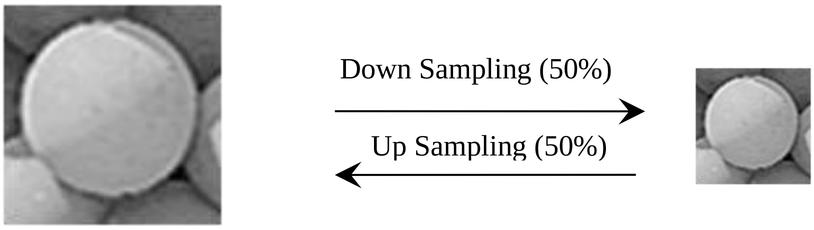


(a) (b) (c)

Figure 2.1. The type of color space (John C. Russ, 2006). (a) RGB type. (b) HSV type, (c) L*a*b* type

2.1.2. Image Resizing

Dimensions alignment is meant to change the size of the input image from size [x, y] to [X, Y] to make it bigger size or smaller size. However, in our method, used different image sizes of, each image has a different size for Training-set that used [40×40] in machine learning. Because that, we employed the image resizing to increase the image size or decrease, and make all image sizes equal (for all cropped images = [40×40] size) to be the same the size of training images and test. This will help us in the feature extractor algorithm to extract the feature with the same length for all images. In addition to when the reducing size of image (Down Sampling) the speed of feature extraction and other processing will be faster.



(a) (b)

Figure 2.2. Down and up sampling example. (a) image with size [80,80] pixels. (b) result of down sampling algorithm with size [40,40] pixels.

2.2.3. Histogram Equalization (HE)

Often luminous intensity information needs to be normalized to exploit the full range of values and make posterior analysis independent of scene illumination by (Dey et al., 2013). Some processes such as edge extraction, shape and texture analysis rely on the 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 Im, the probability density function for gray level k, with k in the range of $0 \leq \mathrm{k} < L$, is defined as

$$\mathrm{P_r}(\mathrm{r_k}) = \frac{\mathrm{n_k}}{\text{Length X Width}} = \mathrm{n_{nk}} \tag{2.3}$$

Which can write in terms of mathematical transformation:

$$\begin{cases} [0, L-1] \rightarrow \mathrm{N} & \rightarrow \\ x \rightarrow Card(x) & \rightarrow \end{cases} \qquad \begin{cases} [0, L-1] \rightarrow [0,1] \\ x \rightarrow pdf(x) = \dfrac{Card(x)}{\sum_{i=0}^{L-1} Card(x)} \end{cases}$$

With n_{nk} the number of occurrences of pixel with gray level k. The cumulative distribution function (CDF) defined in equation 2.4.

$$\mathrm{CDF(x)} = \sum_{k=-\infty}^{x} \mathrm{P(k)} \tag{2.4}$$

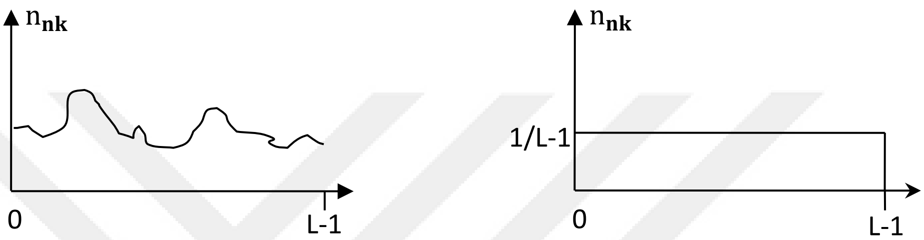


Figure 2.3. Performance of the HE technique.

$$\mathrm{S_k} = (\mathrm{L}-1).\mathrm{CDF(x)} \tag{2.5}$$

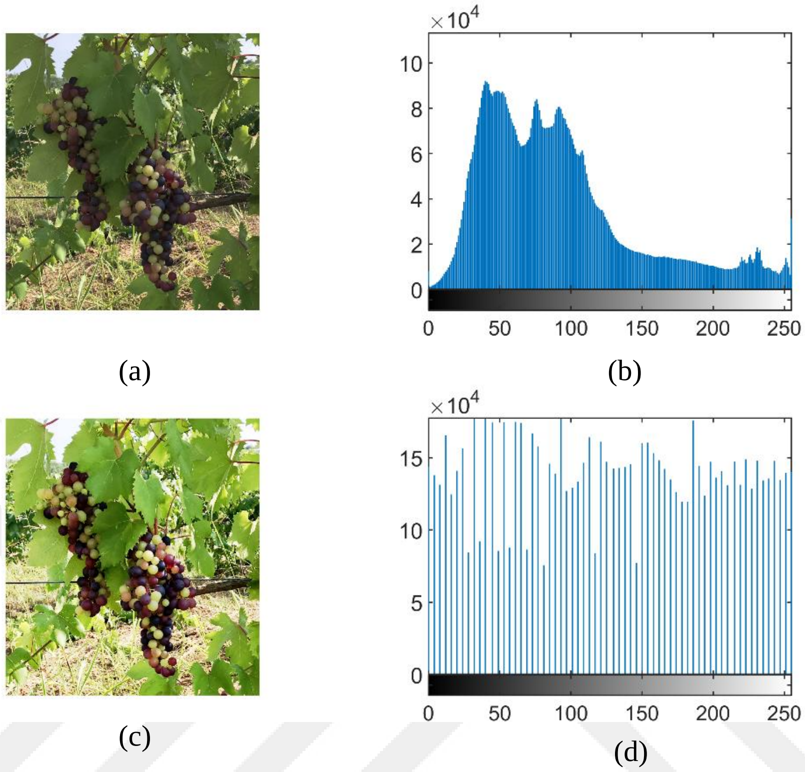


Figure 2.4. Example of HE. (a) Input image. (b) Histogram of input image (c) The results of HE image. (d) Histogram of Equalized image.

2.1.4. Image Smoothing and Non-Maximal Suppression (NMS)

The output image of FRST will be gray image and has some noises and not smooth in order to detect the center of berry we need to find the maximum peak value for each detected interest points so the average filter will be employed with size 3x3 to make smoothing and neighbor pixels suppuration, then Non-Maximum Separation (NMS) will be used to select the local maximum values by the aid of sliding window over the image and will compare with neighbor pixels that’s above threshold value and the pixels with less than threshold value will be zeros. 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 average filter window size has been set [3,3] and this window will slid over the image to smooth the output of FRST. Figure 2.5. Shows the window of the average filter size (3*3). Figure 2.6. Shows an example of FRST, smoothing and salient point detection.

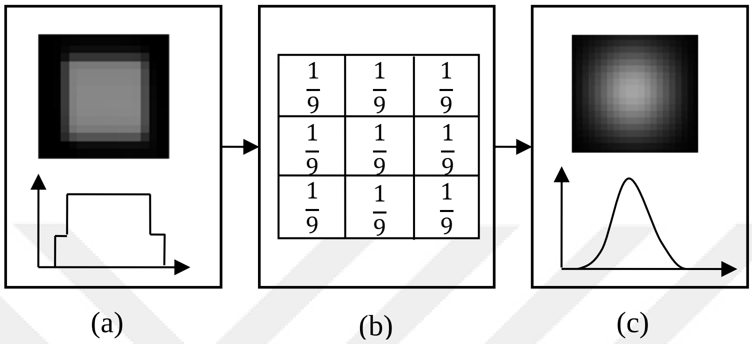


Figure 2.5. An example of average filter (a) The output of FRST (b) the 3x3 structure element (c) Smoothed image.

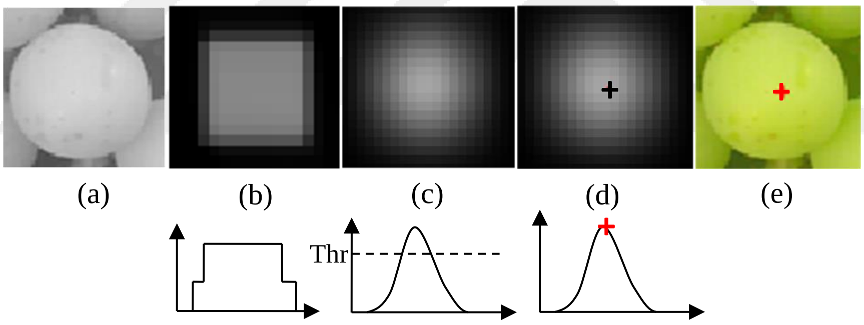


Figure 2.6. An example of salient point detection. (a) Input image. (b) The output of FRST. (c) After the average filter. (d) Local maximum detection after thresholding. (e) Final detected image.

2.2. Pattern Recognition

Pattern recognition task deals with detecting instances of images and real-world targets, while handling variation with labels to the targets' shape and orientation. Target recognition algorithms typically take in a raw image as input and output the bounding boxes coordinates for the recognized targets. Target detection plays a major activity in agriculture robotic, in this chapter, a classical machine learning approaches using; SVM, ANN and CNN used for classifying images and will do the berry recognition mission in the next will explain each method. Two key parts of the pattern recognition process will be explained in the next. The first one is the construction of a feature vector which jointly shape a description of a given class of object. The second important aspects are the construction of a model that defines the rules for labeling the features data.

2.2.1. Target Recognition Challenges

Target recognition combines the tasks of localization, classification and outputs the target along with same pixel coordinates in the input image. This poses a few challenges such as: -

2.2.1.1. Varying Target Number

The target number detected in an input image is not predefined, and due to this the output of the detector cannot be a vector of a fixed size.

2.2.1.2. Object Size

In image classification, mission, the biggest target in the image is classified into one of the known labels of the target. As opposed to this, target of varying sizes must be recognized in a recognition mission. The length of test feature vectors should be the same the length of trained features, this is depending on the image size. In CNN should resize the test image to same size of input layer, so this step is very important in object detection.

2.2.1.3. Architecture

A target recognition mission combines object localization and classification. Thus, the architecture used for detecting objects must be involved, the requirements for both of these missions must take into consideration, and combined model must be created.

2.3. Feature Vectors

A vector of feature is $x_p = (x_1, x_2 \ldots, x_n)^T$ involving values of the scalar x_i, i is a real number, which represent a measure of some property of an image location or region p=(i,j) like color, texture or shape among others, which can be calculated for the objects of a given class. These vectors are a numerical impersonation of a class or group, which is used in the detection and learning in the classification stage of a pattern recognition strategy. And in Convolution Neural Network the feature vector will extract automatically.

The different feature vectors have been used to improve the grape berry detection strategy, which involved Histogram of Local Binary Pattern (LBP), Oriented Gradients (HOG), a combination of HOG and LBP and CNN will extract the feature automatically by convolutions layers, are briefly explained in the next sections.

2.3.1. Histogram of Oriented Gradients (HOG)

A feature descriptor is a representation of an image or an image patch that simplifies the image by extracting useful information and encoding them in a way that they are unique for each image. There are several feature descriptors the most popular one is HOG (Histogram of Oriented Gradients) presented by (Dalal et al., 2005). HOG features are chosen here, as they capture the shape of the object better than other features. Since the features chosen for this approach are sensitive to the aspect ratio of the object to be detected, a separate detector must be trained for each orientation of the object. In the HOG feature descriptor, the distribution of directions of oriented gradients is used as features to represent images. Around the edges and corners of an image, the magnitude of intensity changes is large. These intensity changes reveal a lot about the shape and orientation of the objects, and hence are extracted as useful gradient features. To compute the HOG features, the image is broken into smaller windows, and in each window, every pixel is compared with its neighboring pixels, to compare the variation in brightness across the pixels. The gradient arrow is drawn along the direction in which the pixel intensity increases. The gradient direction that is the maximum in that window is then assigned as the window’s gradient direction. This process is repeated for the entire image, until all the windows are replaced by gradient arrows, which show the variation in pixel intensities from bright to dark. In contrast to analyzing the pixel intensities directly, computing the gradients is better as it is brightness invariant, and so Histogram of Oriented Gradients feature is chosen for berry detection. Figure 2.7. Shows the feature extraction in the block diagram.

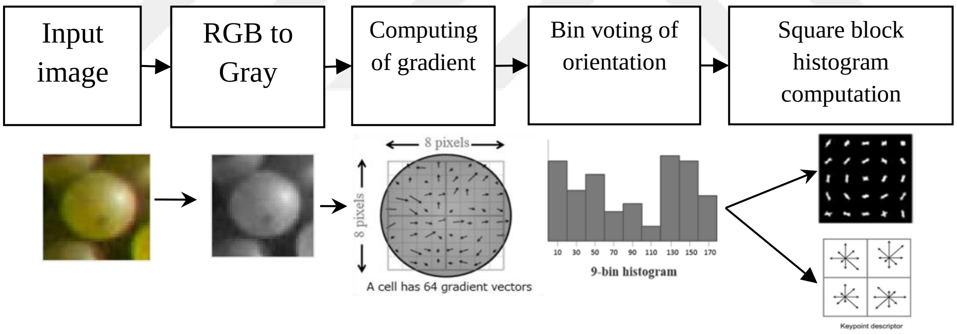


Figure 2.7. An example of HOG Feature extraction

2.3.2. Local Binary Patterns (LBP)

The LBP, presented by (Ojala et al., 2002), is a powerful and efficient texture information that is openly used in computer vision areas and image processing as a feature extractor. LBP algorithm has the ability to capture the body shape in image by searching to each pixel neighbor. The input image is divided into cells of n x n pixels, the center pixel will compare neighbor pixels m x m, m <

n arrange, normally m =3. If the pixel at the center value of it is bigger than the value of neighbor, the neighbor is generated as zero, otherwise one. The binary examples are evaluated by comparing the pixel at the center with its neighbor pixels within a cell. For more explanations, following counter-clockwise, or the neighborhood clockwise, binary code is will generate by appending” 1” or ” 0” to the code whenever the pixel at the center is bigger it will label 0 code or lower than the neighbor pixels it will label 1 (Guo et-al., 2010). the neighbor pixels it will label 1 (Guo et-al., 2010). This-process is-repeated for each mask with size *n* x *n* pixel cells on the gray image. By aid of 8 neighbor pixels yields an 8-bit with 28 making 256 levels (colors) by counting each value of a color by a binary-number in a cell. At the end, each one histogram is concatenated in order to form the feature-vectors of a-whole image (Ahonen et-al., 2006).

This pattern will reduce from 256 color levels to 59 color levels, by using counting for the patterns that have at most two transitions (from zero to one) in binary code only by (Ojala et al., 1996). Figure 2.8. Is represented the LBP feature extraction.

29	125	65
46	45	35
17	119	205

(a)

0	1	1
1		0
0	1	1

(b)

1	2	4
128		8
64	32	16

(c)

Pattern = 01101101, **Decimal** = 2+4+16+32+128 = 182

Figure 2.8. LBP feature extraction. (a) Example the red color value represents the less than the value of the center pixel, and the green color are the bigger than the center pixel. (b) Thresholded. (c)Weight.

2.3.3. The Radial Symmetry Transform

After the pre-processing the image in order to improve the berry detection and reducing the consumption time, we have to find the circular or interest points in THE image in order to make pre-berry detection. There are many authors are used Radial Symmetry Transform (Lakehal Elkhamssa et al., 2014), (Nuske et al., 2014), (Roscher et al., 2014) and (Pérez-Zavala et al., 2018). Fast Radial Symmetry Transform (FRST) is presented by (Loy et al., 2003) and used by (Nuske et al., 2014) was chosen for detection of salient point. However, not same (Nuske et al., 2014) in this work Fast Radial Symmetry Transform (FRST) is applied a down sampling on the image to gain to process them faster. The FRST will use to find the interest point that has radial symmetry with high value; these are the centers of circular points. FRST gives each pixel a score for symmetry of radial at a distance $di \in D = (d1, d2,., dn)$, D is extremely reduce value. The score in each pixel is determined by the aid of gradients information firstly at a radius d from the pixel at the center of an image Im 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) = $\nabla \mathrm{Im}(p)$, a positive-affected pixel called $\mathrm{P}_{+\mathrm{ve}}(\mathrm{p})$ and negative affected pixel called $\mathrm{P}_{-\mathrm{ve}}(\mathrm{p})$ at the distance d from p is defined; see mathematical Equation(2.6 and 2.7) below, and Figure 2.9.

Positive affected mathematical equation: -

$$\mathrm{P}_{+\mathrm{ve}} = \mathrm{p} + \mathrm{int}(\frac{g(p) * d}{\|\mathrm{g(p)}\|}) \tag{2.6}$$

Negative affected mathematical equation: -

$$\mathrm{P}_{-\mathrm{ve}} = \mathrm{p} - \mathrm{int}(\frac{g(p) * d}{\|\mathrm{g(p)}\|}) \tag{2.7}$$

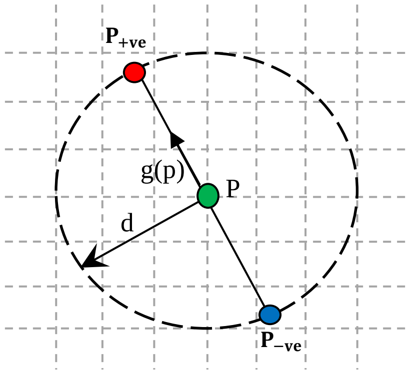


Figure 2.9. The demonstration of FRST method.

Using this information, the image orientation matrix, O_d, and image magnitude matrix, M_d, at a radius d, are created using a recursion loop, where for each pixel $p(i,j)$:$1 \leq n \leq (M \cdot N)$, the related P_{+ve} or P_{-ve} is increased or decreased by 1 and ‖g(p)‖, throughout all pixels; see equation (2.8 to 2.11), (Loy & Zelinsky, 2003; Ni, Singh, & Bahlmann, 2003; Lakehal Elkhamssa,2014). Starting values of the gradient image matrixes are defined as

$$O(P_{+ve}(p)) = O(P_{+ve}(p)) + 1 \tag{2.8}$$

$$O(P_{-ve}(p)) = O(P_{-ve}(p)) - 1 \tag{2.9}$$

$$M(P_{+ve}(p)) = M(P_{+ve}(p)) + ||\, g(p) \,|| \tag{2.10}$$

$$M(P_{-ve}(p)) = M(P_{-ve}(p)) - ||\, g(p) \,|| \tag{2.11}$$

By using orientation matrix images O_d and image magnitude matrix M_d along with two parameters α and K_d, radial strictness and a scaling factor for distance d, respectively, the matrix F_d is defined; see equation (2.13). Finally, the symmetry transform matrix at radius d, S_d is obtained by convoluting Fd with a two-dimensional Gaussian function A_d. See equations (2.12). Radial symmetry contribution at radius d is called as convolution:

$$S_d(p) = F_d(p) * A_d \tag{2.12}$$

$$F_d(p) = \frac{M_d(p)}{K_d}\left(\frac{|Od(p)|}{K_d}\right)^{\alpha} \tag{2.13}$$

The absolute value of $S_d$will makes the transform image with each pixel a symmetry score, independently of its orientation, pixel symmetry from dark to light. To consider all the radius di ϵ D, the final transform image S is defined as the average of all S_d; See equation (2.14).

$$S = (|D|)^{-1} + \sum_{d \in D} S_d \tag{2.14}$$

2.4. Classifiers

When we are deals with statistical and other fields where huge data-set are collected and resolved, it is often usefully to classify the features of data-set into class name. This task is a very difficult for a human, who often cannot recognize to which label or class a data point belongs due to a large amount of data contained in each case in data. Instead, the classifier is used. There are sundry different techniques to create a classifier, and two of the most common are explained in the next sections. It is common information about all classifiers who work through supervised machine learning, where a classifier trains data-set with a known labels or classes name and then uses it on data of the same type and feature length, allowing the classifier use its knowledge of data in training-set to classify new test-set data. It is useful that the classifier can generalize and separate data-set that has never been found before in training-set, depending on the labels that most closely resembles.

2.4.1. Deep Learning

The main ideas behind deep learning are the feature vector will extract automatically from the input images. The aim of deep learning learns automatically from features with noisy data this is a big challenge for traditional machine learning and the accuracy of a machine learning algorithm drops significantly with noisy data. A deep learning network is usually made up of an input layer, many hidden layers that learn the high-level features using some activation functions called Rectified Linear Unit (ReLU), followed by an output layer. While training, values of random are assigned as the weights of the network, and the network will learn in the optimal weights using an algorithm called the backpropagation. In the simplest sense, backpropagation algorithm propagates the errors from the output back to the input and calculates the partial derivatives with respect to each weight to compute steps in the forward direction (making the weight update to reducing the error). As more and more hidden layers are added to the network, the update of weight process becomes harder with decreasing input signal strength. Deep learning helps alleviate this issue by using many algorithms that use for feature extraction.

2.4.1.1. Convolutional Neural Network (CNN)

A 15-layer Convolutional Neural Network (CNN) are used for the detection of single grape berry with input layer size is used [145X145X1]. Three convolutional layers and Max-pooling are used to extract the features and down-sample the image pyramid, it down-samples, from the input. Stride is the number of pixels by which the filter matrix slides over the input matrix. ReLU (Rectified Linear Unit) is used as the activation function, to introduce non-linearity in the output of a neuron in the network. This is done since the real-world data is mostly non-linear, and the network should learn the non-linear representations to perform well on real data. Table 2.1. Shows the architecture of the Deep Neural Network (DNN) used in this study, consisting of Convolution layers and ReLU activation functions, Max-pooling to down-sampling again with size [4x4]. The network consists of down-sampling and feature extraction layers and processes the one channel of the gray as an input image.

Table 2.1. The CNN Architecture.

Layer	Layer Name	Layer parameters
1	Image Input	145x145x1 images with 'zerocenter' normalization
2	Convolution	N0.Filters 8, 3x3 convolutions with stride [1 1] and padding [1 1 1 1]
3	Batch Normalization	Batch normalization
4	ReLU	ReLU
5	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
6	Convolution	N0.Filters 16, 3x3 convolutions with stride [1 1] and padding [1 1 1 1]
7	Batch Normalization	Batch normalization
8	ReLU	ReLU
9	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
10	Convolution	N0.Filters 32, 3x3 convolutions with stride [1 1] and padding [1 1 1 1]
11	Batch Normalization	Batch normalization
12	ReLU	ReLU
13	Fully Connected	2 fully connected layer
14	Softmax	softmax
15	Classification Output	crossentropyex

As described above, a simple CNN is built from many layers, and each layer of a CNN transforms one volume of activations to another through a differentiable function. This experiment utilized five main types of layers to made CNN architectures: Convolutional Layer, Batch Normalization Layer, Max-Pooling Layer and Fully-Connected Layer (exactly as seen in regular Neural Networks) each layer described below. The combination of these layers is making CNN architecture, each layer will describe below.

- Convolutional: The convolution in the first layer of CNN to extract the useful features from input image.
- Batch Normalization: The batch normalization layer amid to normalizes each input channel across a mini-batch. To hurry up the training of CNN and it will reduce the sensitivity of the network initialization, utilize batch normalization layers between convolutional layers and ReLU layers.
- ReLU: Stands for ReLU it for a non-linear operation. The output is will be only for positive values else will be zero. $f(x) = max(0,x)$.
- Max-Pooling: It is a function aimed to reduce the spatial size of the representation to reduce the amount of features and computation in the CNN, also to control overfitting, down-sampling to select in each window the maximum value.
- Softmax: To evaluate the probability of each class from each input.
- Fully-Connected (FC): The neurons at fully connected layer have fully connected with all activations function ReLU at previous layer, as seen in Table2.1. These activations may hence be calculating with a matrix multiplication followed by an offset of the bias.

2.4.2. Artificial Neural Network (ANN)

The Artificial Neural Network (ANN) is a supervised machine learning algorithm works as a classifier based on the way the human brain works, which is very different from human brain how the computer code is usually written. A human brain involves a huge number of nerve cells, neurons. Each of these cells is connected to many other like cells, creating a very complex signal transmission network. Each cell collects inputs from all other neuronal cells to which it is connected, and if it reaches a reliable threshold of activation function, it will send the signals to all the cells to which are connected. Figure 2.10 demonstrates the example of graphical perceptron.

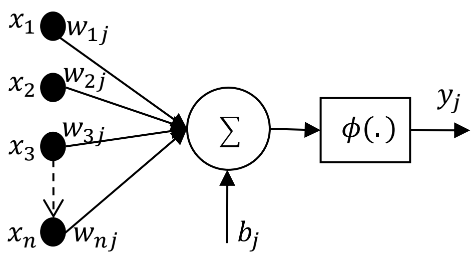


Figure 2.10. The graphical example of a perceptron. n represent of input, w_i represent the weight, ϕ activation function and y represent the network output.

When writing an ANN, this is simulated utilizing a "perceptron" as the basic unit instead of the neuron. The perceptron can take sundry weighted inputs and summarize them if the output before the activation function more than the threshold value the output of neural will be one else zero. Since the derivative of the activation function is often used in network training, it is convenient if the derivative It can be expressed in terms of the value of the original function, since few additional calculations are needed to calculate the derivative in this case. The equation for a perceptron can be written as:

The equation for a perceptron can be written in equation (2.15).

$$y_j = \phi \times \sum_{i=1}^{n} x_{ij}.w_i + b_j \tag{2.15}$$

The main layers of ANN are three which; input layer equal number of examples in input layer (n), Hidden layer need to solve complex problem and output layer same the number of class. Figure.2.11 represent a sample ANN connection.

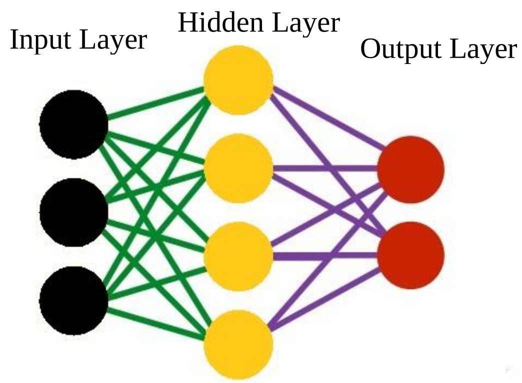


Figure 2.11. A graphical representation of ANN.

In artificial neural networks, this is achieved by updating the weights associated with the connections between the layers. There are several ways to do this, most of which involve initializing weights and providing examples for the network. Errors generated by the network at the exit will be calculated and feedback through a process called "backpropagation." In the simplest way, backpropagation is an algorithm propagates the errors from output back to the input and calculates the partial derivatives with respect to each weight to compute steps in the forward direction (making the weight update to reducing the error). As more and more hidden layers are added to the network, the update of weight process becomes harder with decreasing input signal strength. This structure of ANN results with better accuracy and decreases the number of connections between hidden layer and output layer. the fully connected neurons exist only between the hidden neurons of cores and output neurons of the cores. The reduced number of connections between the hidden neurons and the output neurons decreases the latency of prediction. The nntool box of Matlab is used to train.

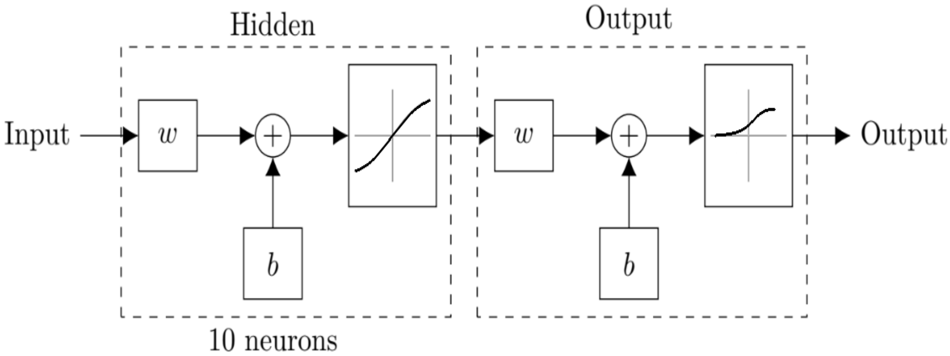


Figure 2.12. The ANN- Architecture.

Figure 2.12 shows the trained ANN structure. These ten hidden layer neurons were selected by a number of trials, resulting for the best accuracy and low latency. There were a number of experiments done using different kinds of activations function but the best accuracy was produced by using the sigmoid activation demonstrated in equation (2.16) function for hidden layer and Tanh demonstrated in equation (2.17), activation function for the output layer.

$$f(\lambda) = \frac{1}{1 + e^{-\lambda}} \tag{2.16}$$

$$f(\lambda) = \frac{e^{+\lambda} - e^{-\lambda}}{e^{+\lambda} + e^{-\lambda}} \tag{2.17}$$

The biggest problem with artificial neural network algorithm is data over-fitting, which happens when the classifier becomes inefficient in classifying the different test examples, at the expense of not being able to classify general input data. It can avoid by cross-validation, where the ANN is trained by one data-set, and then evaluated results of different test-set. If the error is larger in the validation data, the ANN might be overfitting. If the previous design is saved, the ANN can go back to which one that produced more accuracy (Haykin, S. 2009).

2.4.3. Support Vector Machine (SVM)

Support Vector Machine is a supervised machine learning classifier that builds a best hyperplane from a set of vectors of features corresponding kinds of a class label dataset. Let $\mathcal{X} = (x_1, x_2 \dots, x_k)$, indicate a feature vector set $x_i \epsilon \mathrm{R}^n$, corresponding to the images I_i ,$i \epsilon$ [1,k] and let $\mathrm{y} = (y_1, y_2 \dots, y_k)$ indicate the set of class manes. The problem here is to find to the optimal linear hyperplane $\boldsymbol{w} \cdot \boldsymbol{x} + b$ = 0, where $\boldsymbol{w}$ is the normal to hyperplane and b translation constant, that better isolated the classes in $\mathcal{Y}$ is such that increases the distance of the vector of feature to the dividing hyperplane and minimizes the number elements negative assigned to the wrong labels (incorrect side of the dividing hyperplane). In case the data is non-linearly can be separate, a cost parameter C and error (mis-classification) variableξ_i:

$$\xi_i = \mathrm{maximum}\big(0, 1 - y_i(\mathrm{w} \cdot \mathrm{x_i}, + \mathrm{b})\big) \geq 0 \tag{2.18}$$

Which appears the distance to the correct place?

This allows to be easy the problem when data is not accurately can be separated by introducing the misclassification (error) penalty to the sought hyperplane, Figure 2.12 is displayed the example of leaner SVM.

$$\mathrm{w} \cdot x_i + \mathrm{b} \geq 1 - \xi_i \tag{2.19}$$

$$\mathrm{w} \cdot x_i + \mathrm{b} \leq -1 + \xi_i \tag{2.20}$$

$$\xi_i \ \geq 0 \ \forall \tag{2.21}$$

for y_i = 1 and y_i = −1 respectively. Using the label of the classes these inequalities can be made it in one equation:

$$(\mathrm{w} \cdot x_i + \mathrm{b}) - 1 + \xi_\mathrm{i} \geq 0 \ \forall_\mathrm{i} \tag{2.22}$$

The function of an object is defined by finding the maximum distance between the parallel hyperplanes. See Figure 2.13. On which the restrictions are the penalization error is minimum and active. The distance between both hyperplanes is given by $\frac{2}{\|\mathrm{w}\|}$, then the optimal distance is found by maximizing $\frac{2}{\|\mathrm{w}\|}$, which is equivalent to minimizing $\frac{1}{2}\|\mathrm{w}\|^2$. At the end, the optimization problem mathematical equation can by written:

$$\mathrm{Minimum}_{\mathrm{w},b} \quad \frac{1}{2}\|\mathrm{w}\|^2 + \mathrm{C}\sum_{i}^{k} \xi_\mathrm{i} \tag{2.23}$$

$$\mathrm{y_i}(\mathrm{w}.\,\mathrm{x_i} + \mathrm{b}) - 1 + \xi_\mathrm{i} \geq 0\forall_\mathrm{i} \tag{2.24}$$

$$\xi_\mathrm{i} \geq 0\forall_\mathrm{i} \tag{2.25}$$

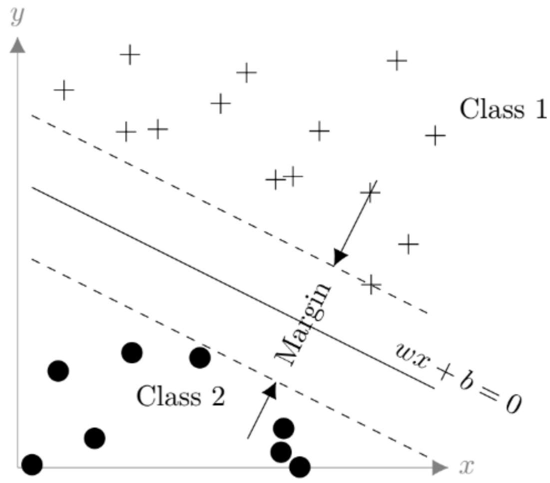


Figure 2.13. A Leaner-SVM example of separation.

To solve the optimization problem stated above, it is convenient to use to solve the dual problem and Lagrange multiplier algorithm. This allows the problem to be computed more efficiently (Lin, 2006) and (Burges, 1997).

2.4. Clustering Technique

Clustering is an unsupervised machine learning approach, also known as clustering, is a machine learning system that has the objective of looking for similarity in pattern characteristics to form and label groups, given a set of feature vectors without their corresponding class or group. Since there are no label data, there is no training stage. first, it has to set the number of cluster or groups, the aim of the algorithm will sort feature by looking to the similarity of the information and the similar class data will group together. Clustering the similar clusters together help profile the different patterns to the clusters. Clustering technique will contribute to reduce the data dimensionality when you are dealing with the hugely variables. At the next step in order to detect the grape bunches over the images so we need a clustering algorithm called Dense Based Spatial Clustering of Applications with Noise (DBSCAN) Presented by (Ester M et al.,1996). This algorithm will help us to find the clusters with high density and the number of clusters will select automatically not like the traditional clustering like K-mean, as well as the DBSCAN algorithm will isolate points as noise or out-layer. Both advantages are helping us in the bunch of grape recognition problem, given the variability of the number of clusters and possible shapes due to occlusion. There are many authors use this algorithm in agriculture field (Díaz, C. A et al., 2018), (Latha G, 2018) and (B M Sagar, 2018). There are two main parameters will lead this algorithm the first one is, the radius of points called epsilon (Eps), Second Minimum Number of Points (Minpts) is representing the minimum clustered points are closers within (Eps) if the number of clustered points less than the set of Minpts the group will release all. The core points if it has more than a specified number of Minpts within Eps. Border point has less than Minpts within Eps, but is in close of a core point. Noise point any point that is not Core and not a Border it will be out the Eps distance. This approach also will play overlap grape bunch pre-separation, improve the accuracy of grape berry detection that have an error in classification. See Figure 2.14. That shows DBSCAN algorithm work.

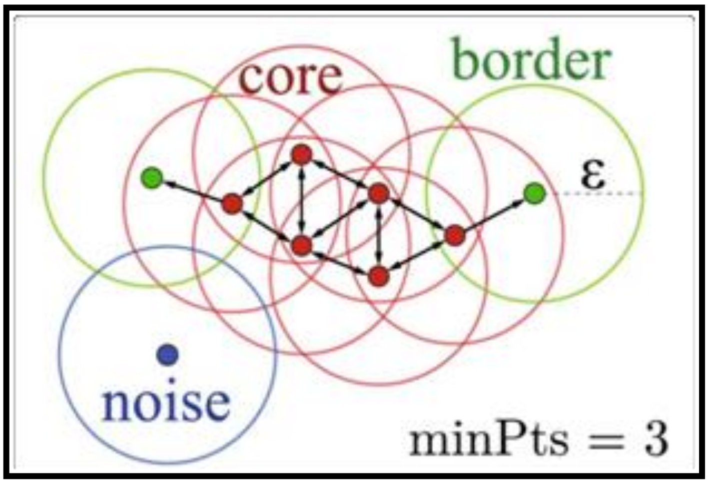


Figure 2.14. Example of DBSCAN Clustring (Mehle, 2017).

3. PROPOSED METHODS

3.1. Algorithm

This research presents an approach for visible recognition of the grape berries and bunches, the algorithm is displayed in Figure 3.1. The proposed method does not use color feature amid use it in all the color of grapes and under natural illumination. This research presents an approach for visible recognition of the grape berries and bunches, the algorithm is displayed in Figure 3.1. The first step consists pre-processing stage to process an image such as; convert the RGB image to grayscale image, Histogram Equalizing (HE) will be used to equalize the intensity values, as explained in previous chapter to improve the extraction of grape berry and the interest points detection. The output of equalized images will be input to the images resize to hurry the interest point detection on a bunch of grapes. The interest-points detection is performed by utilizing the Fast-Radial Symmetry Transform (FRST) due to the results shown in previous work studies for finding circular shaped points of interest and faster than Circular Hough Transform. Feature vector involving HOG, LBP and combination of both features are calculated in different resolution scales centered at the interest point and evaluated with Support Vector Machine SVM-Linear, Artificial Neural Network (ANN) and Convolution Neural Network (CNN) will extract the feature from the images automatically. The classifier will be used to separate them as berry or non-berry. The approach will test the classifier and which one will be more accurate it will choose. After the investigation part output of the classifier will be separate the images of berry and non-berry. To improve the performance of classifier, method has used density clustering (DBSCAN) algorithm, allowing to pre-number of 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 the grape bunches on one image. DBSCAN technique will play to remove the noise (misclassification) and improve the accuracy of detection. At the end method has used a technique that may can solve this issue called close bunch separation to improve the accuracy of bunch detection and counting for estimation, which will explain clearly in the next section.

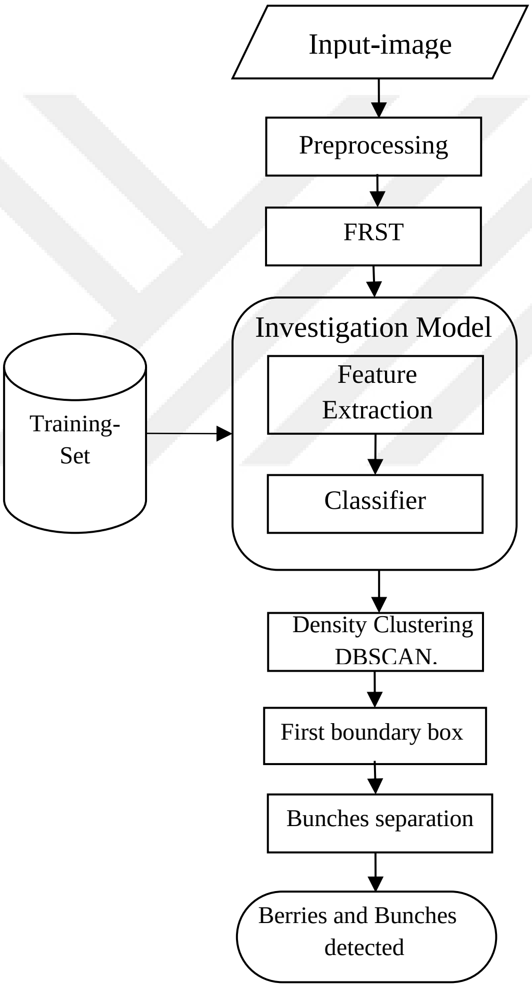
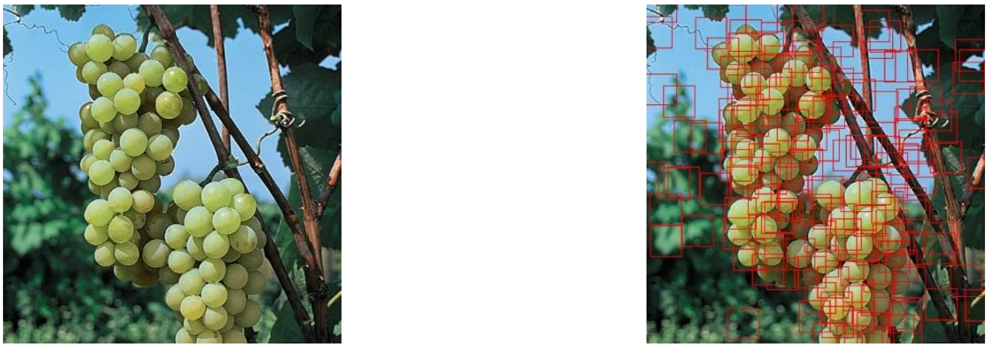


Figure 3.1. Flow chart of proposed Methods.

3.2. Neighborhood Grape Bunches

Neighborhood grape bunch separation first converts detected mask of the grape bunch to a binary image involving the original cluster of pixels that may make one or more overlapping grape bunches. if more than one grape bunch close together or within Epsilon, the DBSCAN clustering may cluster them as a one grape bunch and this situation will affect negatively on the accuracy of estimation, so the neighbor grape bunch separation is needed as shown in Figure 3.2 (d). In order to improve performance for yield estimation and reducing the classifier error. The K-means (pixel positions) presented by (Thakare et al., 2015), will be used to re-cluster the berries coordinate but in this approach the number of cluster (bunch) is required. Each detected bunch will process separately; Convert the detected mask to binary. Morphological operation will be used such as; Morphological filter to remove unconnected components and Image inverse will be used. Next, applying distance transform on binary cleaned image. Furthermore, thresholding the image by value 75\% out of maximum pixel value in image, the result will be binary image the number of connected areas (connected components) will represent the number of cluster(K) in detected bunch. Finally, re-clustering the berries coordinate by K-means with number of connected components as (k) and the iterations value was set 14. Figure 3.2. Represents the neighbors grape bunch separation example. Figure 3.3 is displayed the flow work of proposed algorithm clearly.



(a) (b)

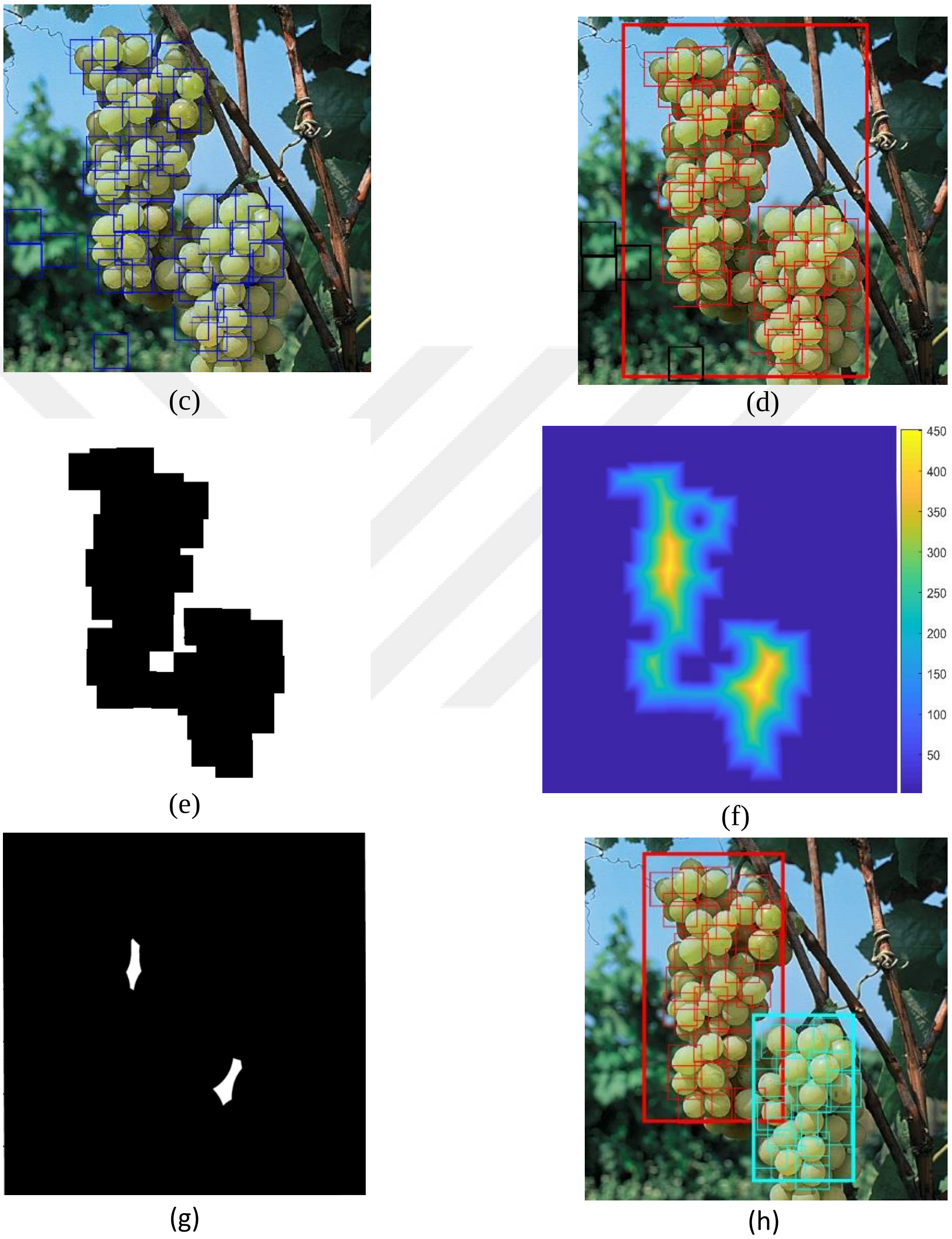


Figure 3.2. Neighbor grape bunch separation example (Golden Muscat Grape, 2016). (a) Input image. (b)Interest point detection by FRST. (c)Results of the classification(d)first bounded box and noise rejection by DBSCAN in black colors (e) Binary image mask of detected area. (f)

Distance transform (g) thresholded image. (h) Results of K-means clustering.

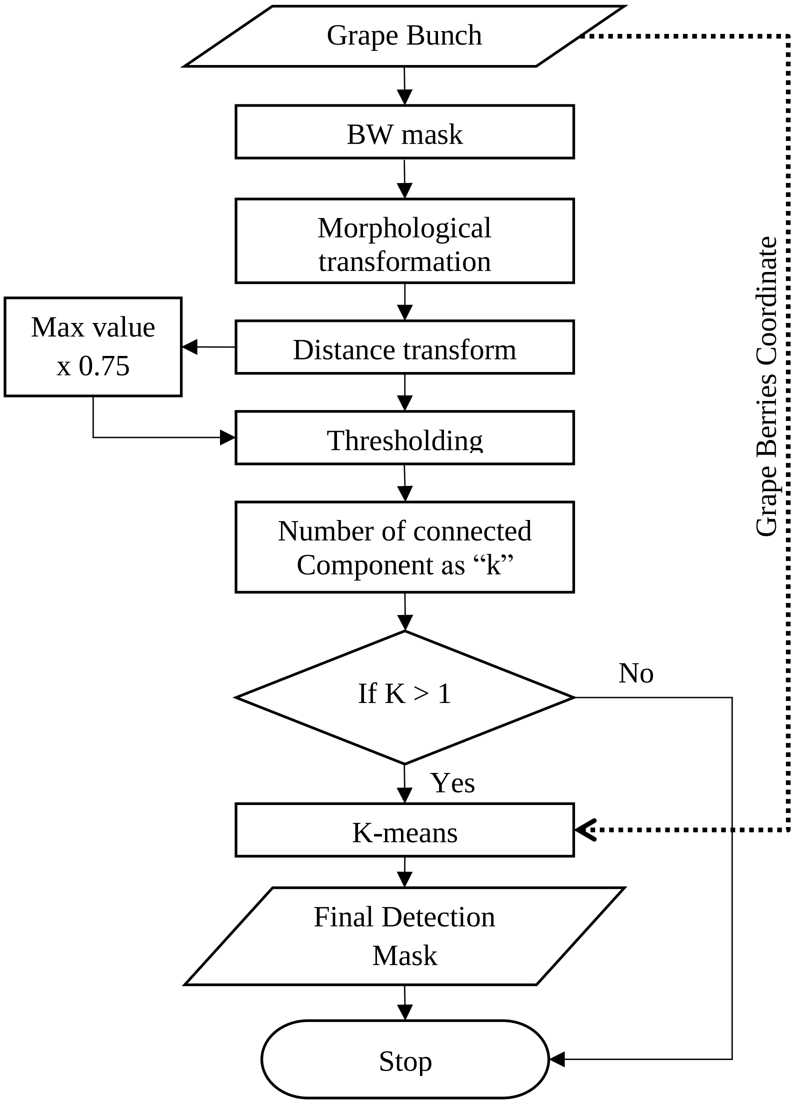


Figure 3.3. The Proposed method of Neighbourhood bunches separation.

4. IMPLEMENTATION AND RESULTS

4.1. Implementation

The different type of classifier and feature technologies are used to implement this experiment done by Matlab™ (2018b) software with Neural Network, Computer Vision, digital image processing and Convolution Neural Network tool boxes to do image preprocessing, extract Patterns and classifications, FRST implemented by (Tola et al., 2010). DBSCAN algorithm implemented by (Yarpiz, 2015). The Computer with Intel(R) Core™ -i7-7500U, CPU (2.9 GHz) with RAM 12GB, it helped us to give a good performance and a good processing time, The Training Dataset that used in our algorithm is Iceland dataset freely available made by (Škrabánek et al., 2015) it has a good number of images for test-set 4000 images and for training-set 2880 images, it made by 90, 180 and 270 degree rotating for each image, different illumination and berry size as displayed in Figure 4.1. In this work we have been mixed the training-set images with test-set images in one file to make 6880 images 50% are non-berry labels and other half represent the berry labels, image sizes 40x40 pixels full color (RGB) and the Bit depth = 24. Figure 4.1. An example of training and test images.

Positive Samples Negative Samples

Figure 4.1. Iceland data-set Examples.

The first challenge in this study is to sort out two huge datasets into classes. Three types of investigation models have been used with various type of feature vectors in order to test which classifier will give better performance, to sort the huge amount of images in the data set. Iceland data-set involved 6880 images, each image in data-set has a different berry size, little contrast in color and light. 35% have been used to test the classifier and 65% are used to train the classifier. The results obtained by a hundred runs and for each run, the test-set and train-set were selected randomly.

4.2. Performance Evaluation

The Accuracy of the algorithm for grape single grape berry detection computed by the aid of the standard confusion matrix as shown in Table 4.1. The accuracy of bunches and grape berry detection calculated by counting manually the number of grape berries and bunches compared with actual number the results are displayed in table 4.2 and 4.3.

Table 4.1. Confusion matrix

		Predicted (output)	
		Berry	Non-Berry
Actual (Input)	Berry	TP	FN
	Non-Berry	FP	TN

- True positives (TP): These are true berries, predicted as a berry. In another way predicted the positive and it is berry (true).
- True negatives (TN): These are true berries, predicted as a non-beery. In another way predicted the negative and it is a non-berry (true).

- False positives (FP): These are non-berries, predicted as a berry. In another way predicted negative and it is berry (false).
- False negatives (FN): These are non-berries, predicted as a non-beery. In another way predicted positive and it is a non-berry(false).

- **Accuracy (Acc)**

Represent overall accuracy, how many times is the classifier correct. The amount that we predicted correctly. It is better to be high as possible. Can be written mathematically as shown equation

$$\text{Accuracy}(\%) = \frac{(TP + TN)\ x\ 100}{(TP + FP + TN + TN)} \tag{4.1}$$

- **Sensitivity (Sens) or Recall (Rec) or TPR**

When it's actually a berry, how often does it predict berry. The output of all the berries classes, the amount that we predicted correctly. It is better to be high as possible. Can be written mathematically as shown equation (4.2).

$$\text{Sensitivity}(\%) = \frac{TP\ x\ 100}{TP + FN} \tag{4.2}$$

- **Precision (Prec)**

When it predicts berry, how many times is it correct. Can be written mathematically as shown equation (4.3).

$$\text{Precision}(\%) = \frac{TP\ x\ 100}{TP + FP} \tag{4.3}$$

- **False Positive Rate (FPR)**

When it's actually non-berry, how often does it predict berry and it is better to be low values as possible.

$$\text{FPR (\%)} = \frac{FP\ x\ 100}{TN + FP} \tag{4.4}$$

The size of images that used to train and classification in ANN and SVM were [40x40x1] pixels with a HOG feature window size [8,8] pixels, the number of orientations 9, Block Size [2,2] and Use Signed Orientation set false were selected, that made feature length 576 variables. LBP feature with linear Interpolation, Cell size [40,40] and the number of neighbors were 8 pixels and the upright was set false that made feature length 10 variables or 10 bins and the combination of the two patterns made length 586 that will improve the accuracy with a little time difference when compare it with HOG feature, are calculated by MATLAB with window size 8x8 in HOG. The size of the images that used to train and test in CNN [145x145x1].

4.3. Results and Discussion

The results of grape berry detection utilizing the various features and classifiers are displayed in Table 4.5 and Figure 4.2. The recorded from multiple runs 100 runs and for each run the input images for training-set and test-set are selected randomly in addition to the amount of data were fixed 35% and 65%, for test and train respectively, to get a more accurate result. Figure 4.3. Summarize the accuracy results obtained with the different amount of images in the training-set and fixed the number of test images 35% out of 6880 images (2408 images) we used various types of features and classifiers to test the performance of the experiment such as ANN, SVM with HOG, LBP, and combination of both as a features and CNN the feature will extract automatically in this approach.-

The results of the experiment show that the amount of training-set was affected positively in overall performance, CNN possible to compare it with SVM and ANN. The accuracy of CNN is less than ANN and SVM classifier in spite of higher time consuming than ANN and SVM. The accuracy of ANN is larger than CNN by 4.98%, and SVM by 0.61%, it has the lowest classification time as well. However, the CNN and SVM have provided acceptable accuracies for single berry classification. Overall a result shows the best performance in the dataset got by the ANN classifier with a combination of HOG+LBP features recorded the highest accuracy, precision and sensitivity as well as faster. These values can explain as that the high change of field affects the decision boundary of a classifier, where the classifier that has samples of non-berry images will give a better result. In summary, the combination of HOG+LBP with ANN classifier introduced the best performance such as, Accuracy, Sensitivity and Precision reached 99.64±0.073%, 99.59±0.117% and 99.70±0.094 respectively, with good False Positive Rate (FPR) 0.293±0.094% with a good time 0.0065 seconds per image (Time is involved converting the image from color image to gray, Histogram equalization, image resize, two kinds of features for extraction and classifications). The example of confusion matrix for each method is displayed in Table 4.2 to 4.4.

Table 4.2. Example of confusion matrix for CNN

		Predicted (output)	
		Berry	Non-Berry
Actual (Input)	Berry	1116	66
	Non-Berry	34	1169

Table 4.3. Example of confusion matrix for SVM with (HOG+LBP) features.

		Predicted (output)	
		Berry	Non-Berry
Actual (Input)	Berry	1184	9
	Non-Berry	11	1204

Table 4.4. Example of confusion matrix for ANN with (HOG+LBP) features.

		Predicted (output)	
		Berry	Non-Berry
Actual (Input)	Berry	1190	8
	Non-Berry	4	1206

Table 4.5. The results of single grape berry recognition for each method.

Classifier	Feature	Accuracy	Precision	Sensitivity	FPR	Average time(Sec.)
SVM	HOG	98.8 ±0.1900	98.8 ±0.280	98.8 ±0.2960	1.19 ±0.280	0.0074
	LBP	73.50 ±0.740	79.76 ±1.39	63.02 ±1.330	15.99 ±1.22	0.0071
	HOG+LBP	99.038 ±0.180	99.035 ±0.28	99.038 ±0.266	0.96 ±0.283	0.0079
ANN	HOG	99.588 ±0.075	99.60 ±0.11	99.57 ±0.132	0.39 ±0.110	0.0058
	LBP	85.68 ±0.490	89.49 ±0.85	80.88 ±0.751	9.50 ±0.880	0.0054
	HOG+LBP	99.64 ±0.073	99.70 ±0.094	99.59 ±0.117	0.293 ±0.04	0.0065
CNN	-------------	94.66 ±1.350	96.05 ±1.68	93.19 ±2.580	3.88 ±1.730	0.0095

The results show that the ANN approach faster than other classifiers. Beside of accuracies. Figure 4.2. Is displayed the accuracy, time consuming and precision for each approach clearly. The improvement in results of single berry detection will also improve the results of bunch detection because in proposed method will recognize the bunches based on single grape berry.

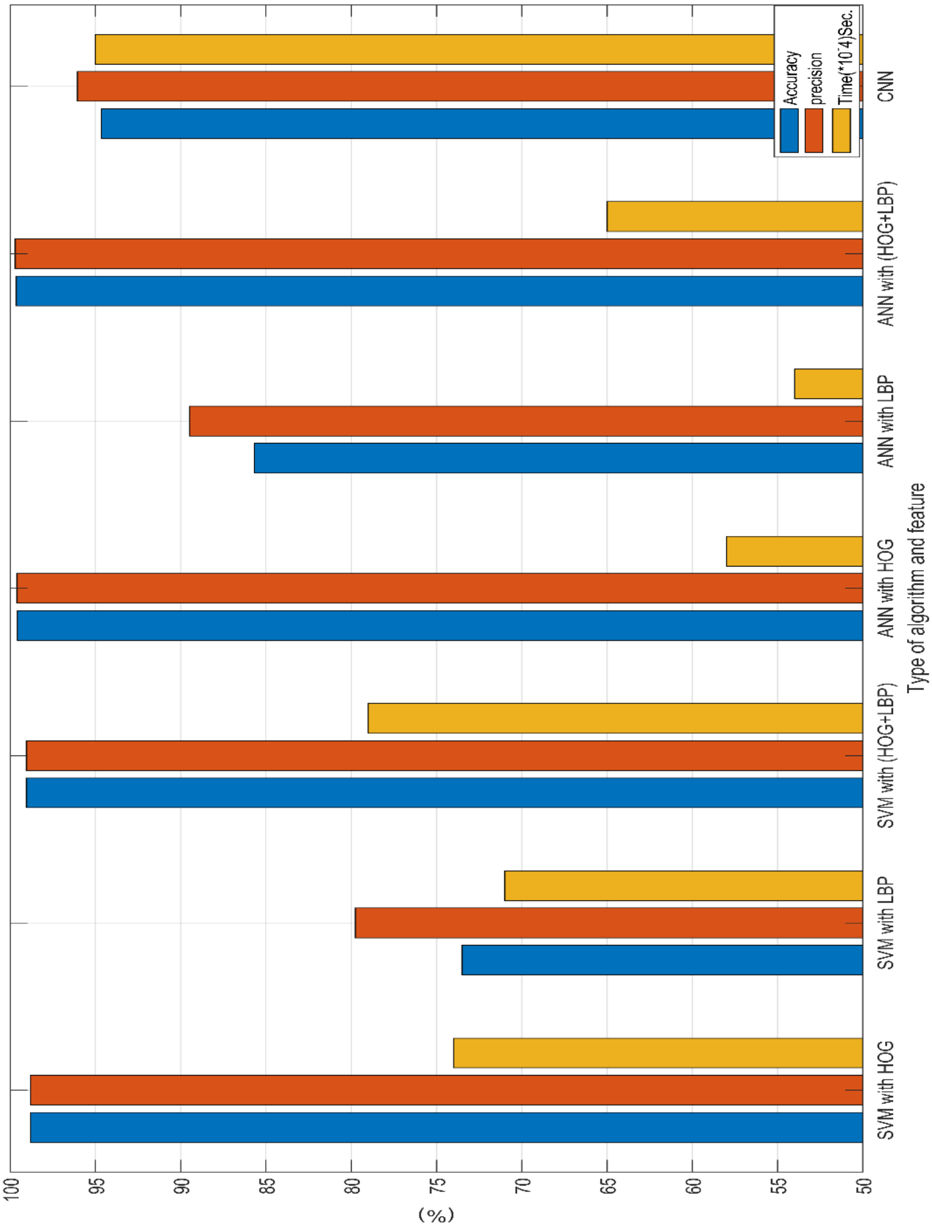


Figure 4.2. Accuracy, time consuming and precision for each approach.

The amount of the training images effects on the results, giving better accuracies in Iceland dataset. Figure 4.3. Represents the influence of the number of images in training-set with accuracies in SVM, ANN and CNN.

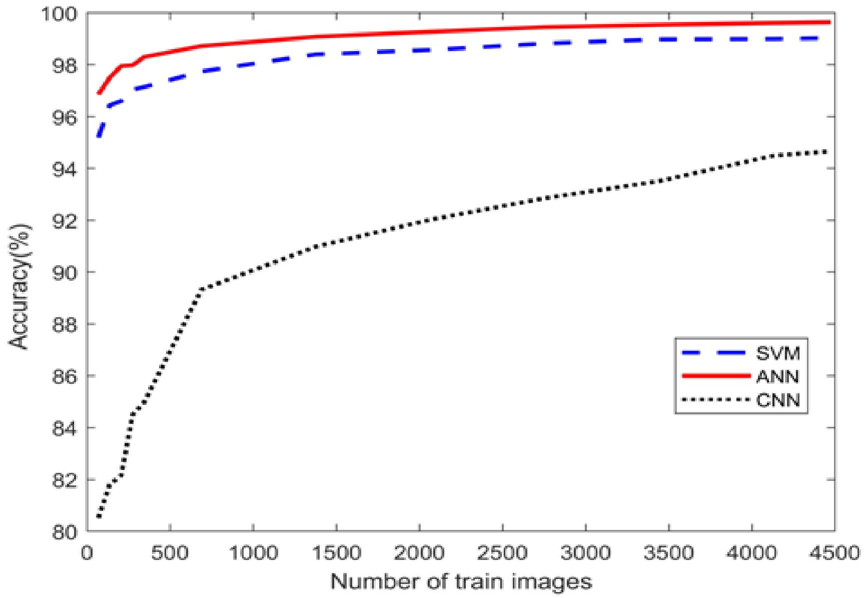


Figure 4.3**.** The accuracy changes with the amount of training images.

The ROC (Receiver Operational Characteristic) curves for a single grape berry detection by using the CNN, ANN and SVM classifier are shown in Figure 4.4 and 4.5. The performance curves of the feature descriptors evaluated above are compared, studying with TPR (True Positive Rate) (grape berry detected as a positive class, correctly) with FPR (False Positive Rate) (non-grape berry detected as a positive class, incorrectly) at several types of classifiers such as ANN, SVM and CNN and feature vectors, as shown in Figure 4.4. &4.5. The ROC of each method, obtained by MATLAB software.

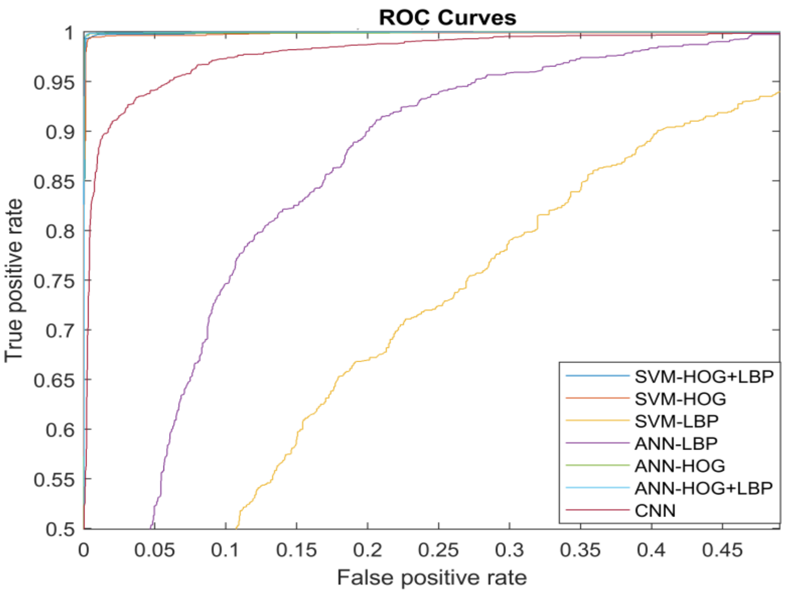


Figure 4.4. The ROC of each method.

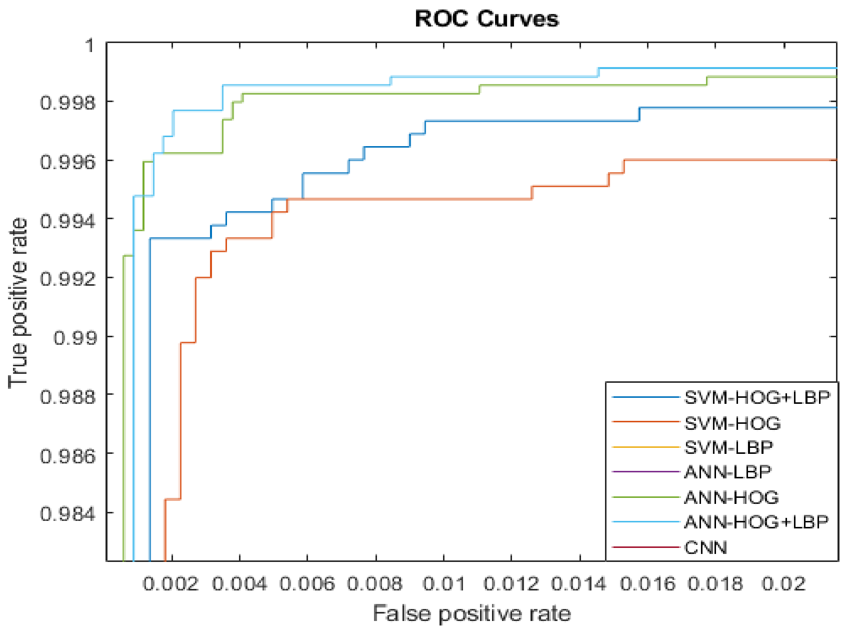


Figure 4.5. The ROC of each method zoomed in.

According to the previous results of single grape berry detection part, the ANN classifier with a combination of two features LBP+HOG was chosen and the classifier trained all the images in the Iceland dataset (6880 images) to perform the grape berry and bunch recognition strategy. The validation results of the proposed algorithm for grape berry and bunch recognition scheme, considers three different datasets such as (Cukurova university-Agriculture facility (Red grapes), Iceland (White grapes) and Tarsus (Red and White grapes)) the pictures are picked in natural illumination in order to record a real and practical results. The results example of each dataset is displayed in Figure 4.6. To 4.9. The results of the proposed algorithm for grape-berry and bunch recognition are summarized in Table 4.6 and 4.7.

The area of grape pixels is also evaluated by ground truth labeler app in MATLAB Software, made by hand manually and compared with detected pixels as shown the results of pixel classification in Table 4.8.

Table 4.6. Grape berry detection accuracy for each data-set

Data-set	Çukurova University	Tarsus	Iceland	Average
No. of images	6	8	3	
Image Size	1600X1200	3024X4032	3888X2592	
Illumination	Natural	Natural	Natural	
Distance	~1.2 m	~0.75 to ~1.5 m	~1.5m	
Accuracy (%)	89.72±1.87	93.06 ± 1.36	86.65±4.50	89.82 ± 2.58

Table 4.7 **.** Grape bunch detection accuracy for each data-set

Data-set	Çukurova University	Tarsus	Iceland	Average (%)
No. of images	6	8	3	
Image Size	1600X1200	3024X4032	3888X2592	
Illumination	Natural	Natural	Natural	
Distance	~1.2 m	~0.75 to ~1.5 m	~1.5m	
Accuracy (%)	89.75±4.78	96.25 ± 2.97	89.22±0.876	91.72 ± 2.88

Table 4.8. Grape berry area detection for each data-set by pixel classification.

Data-set	Çukurova University	Tarsus	Iceland	Average (%)
No. of images	6	8	3	
Image Size	1600X1200	3024X4032	3888X2592	
Illumination	Natural	Natural	Natural	
Distance	~1.2 m	~0.75 to ~1.5 m	~1.5m	
Correct Rate (%)	93.15± 1.541	98.19 ± 1.008	99.59 ± 0.30	96.97 ± 0.950
Sensitivity (%)	94.67± 2.036	98.56 ± 0.66	99.50 ± 0.212	97.57 ± 0.969
Specificity (%)	78.75± 5.374	96.56 ± 3.61	92.55 ± 2.53	89.28 ± 3.830
Precision (%)	96.50± 1.011	99.29 ± 0.99	98.54 ± 1.62	98.11 ± 1.207
ClassifiedRate(%)	97.62± 1.63	98.93 ± 1.174	98.45 ± 1.026	98.33 ± 1.277

The proposed method of the grape bunch and berry recognition approach show results of the average accuracies 91.72±2.88% and 89.82 ± 2.58%, respectively. Results show the correct performance of the proposed method, including the efficient close grape bunch separator. By analyzing the results, it can be noticed that errors may be a consequence of the not detected salient points, miss-classification by ANN or not clustered in the DBSCAN stage. Grape bunch detection shows an accuracy 91.72±2.88% on average by using three different datasets, presenting an efficient method for correctly detecting and separating the close grape bunch. Grape berry pixels or area classification is also studied in order to understand the correct classification and completeness the detection of the grape bunch. Some of mis-detection in grape bunches and single berry detection are gotten from FRST and NMS because the image contrast and shadows this will cause an error and will reduce the accuracy of yield estimation.

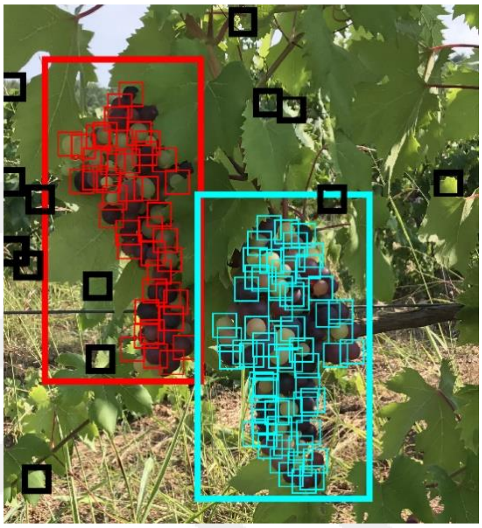
In the current proposed method implementation, the interest point detection by FRST thresholds has been set at a low value, because to detect low contrast variations, though this increases the amount of test-set images and requires more time for processing and it may increase the error value. The results of the suggested algorithm show that clustering step is very efficient in the identity of the bunch of grapes. Only challenge for a succeeded cluster is the highly blocked area where only a few berries of grape can be seen. Since the DBSCAN process eliminates isolated points and the minimum number of individual grape berries detected in the connected area is limited, the proposed method cannot detect that the grapes are blocked. Furthermore, when the bunches of grapes are neighbors together in the vertical direction, the proposed method cannot correctly sort them into different groups since spatial analysis and segmentation are not performed in this direction. The performance of the grape bunch area detection is also shown in Table 4.8. The results show the Correct Rate of the proposed method is on average 96.97±0.950% for grape pixel recognition. However, this large value is in part due to the large number of true negative pixel detection and not only correctly labeled pixels or due to a high rate. The average precision of the proposed method used for various data-sets is 98.11±1.207%. Resolving the position of the FP pixels, most of them come in the neighbor of the grape bunches. Therefore, even if the FP rate is high, due to the location of occurrence of the false positive, the results of grape bunch detection are not affected. Furthermore, the average sensitivity rate by used three different datasets is reached to 97.57±0.969%. The sensitivity rate can be interpreted as the rate of detection. Even if this rate sensitivity is fewer idealistic rates in the range between 95.0-99.0%, it is to be noted that this rate considers the total pixels number within grape bunches that were not correctly labeled as belonging to the corresponding bunch of grape due to DBSCAN errors. It is to be noted that the grape detection results obtained may be improved by training the classifier with a high number of images (datasets). Unfortunately, it is difficult to obtain large datasets from different vineyards and grape varieties. The Iceland dataset only involves white grapes under natural illumination with a fixed camera-plant distance, while the test data-sets includes white, red and various color of grapes under natural illumination. The proposed approach of grape berry and bunch recognition strategy can be utilized for diverse aim in the several vineyard managements missions. Another application can be used for background or leaf removal.

This would allow for the owner of farm to standardize their removal of leaf processes throughout the whole sector. The proposed strategy can be employed for diverse purposes in the various vineyard missions.

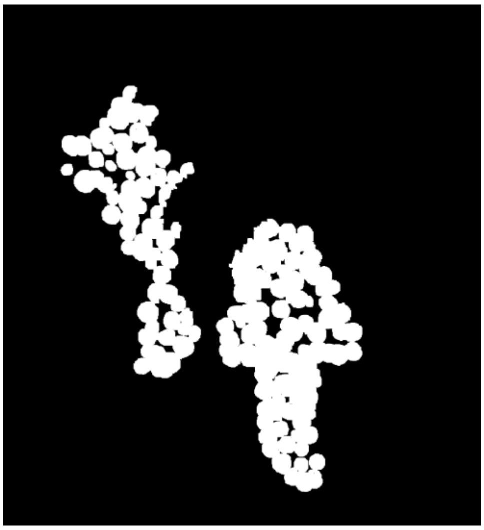
The results obtained in the current study is better than the outcome reported in (Škrabánek, 2018), the improvement in accuracy and precision has been 2.29%, and 3.2%, respectively in single grape berry recognition. Also the comparison of current study with (R. Pérez-Zavala et al., 2018), the results in our study show improvement in single grape detection accuracy, precision and sensitivity 4.27%, 0.7%, and 7.09%, respectively, the grape bunch and berry detection improved by 1.21% and 11.38% respectively, by changing the method of close bunch separation and changed the classifier with parameters of feature vectors.

In (P. Škrabánek et al., 2017), described in this percent of accuracy 87 % of the success is enough for automatic grape berry detection in robotic tasks such as harvesting and spraying, but it is not enough in yield estimation task.

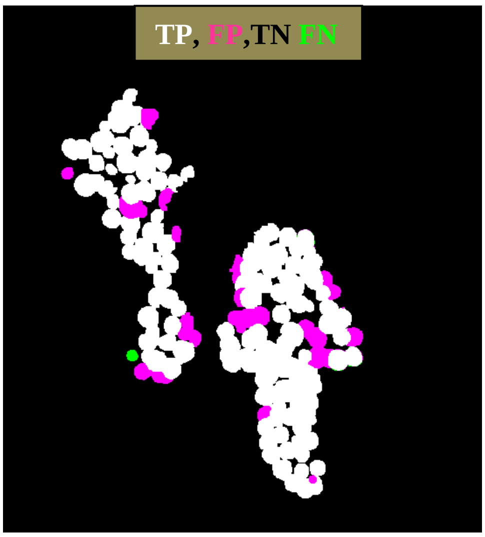
(a)



(b)

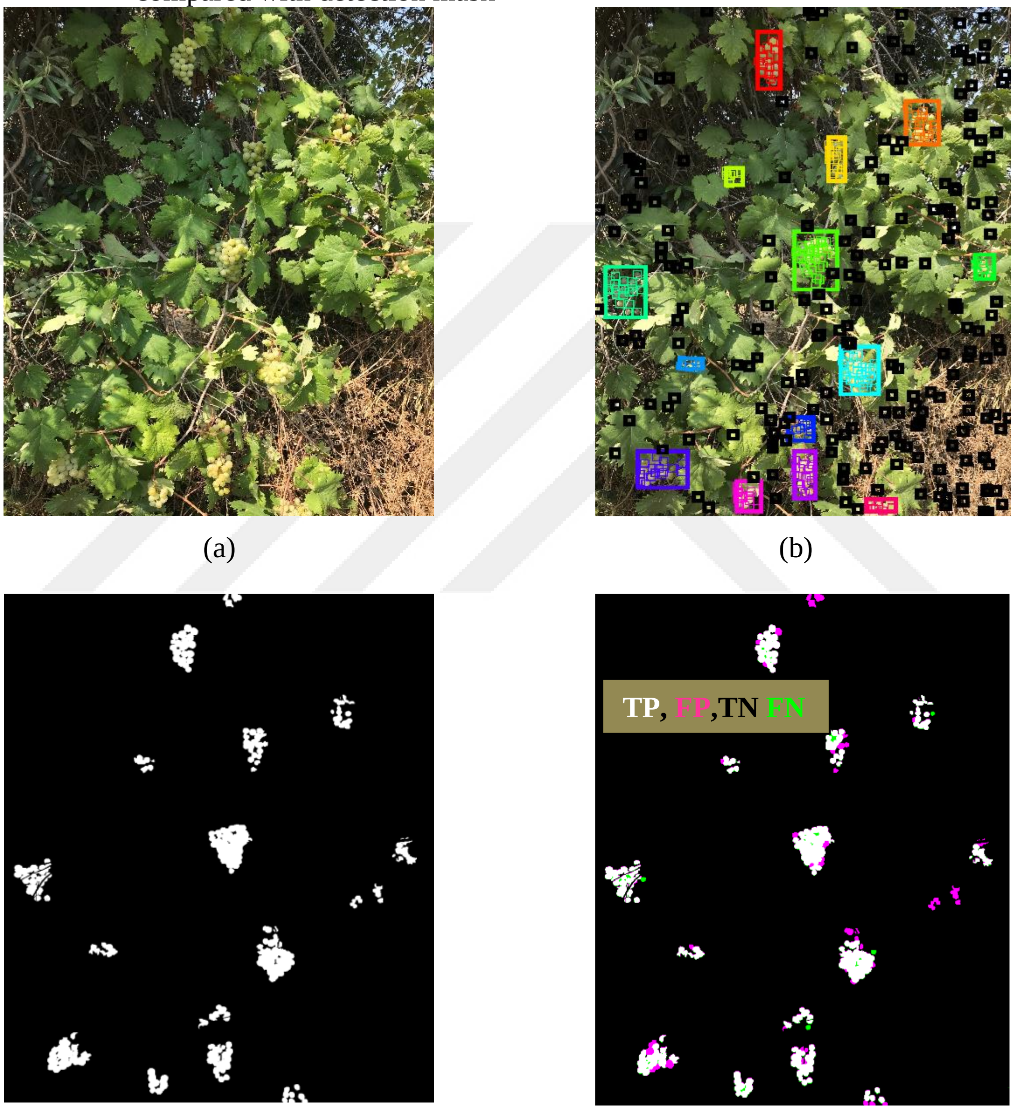


(c)



(d)

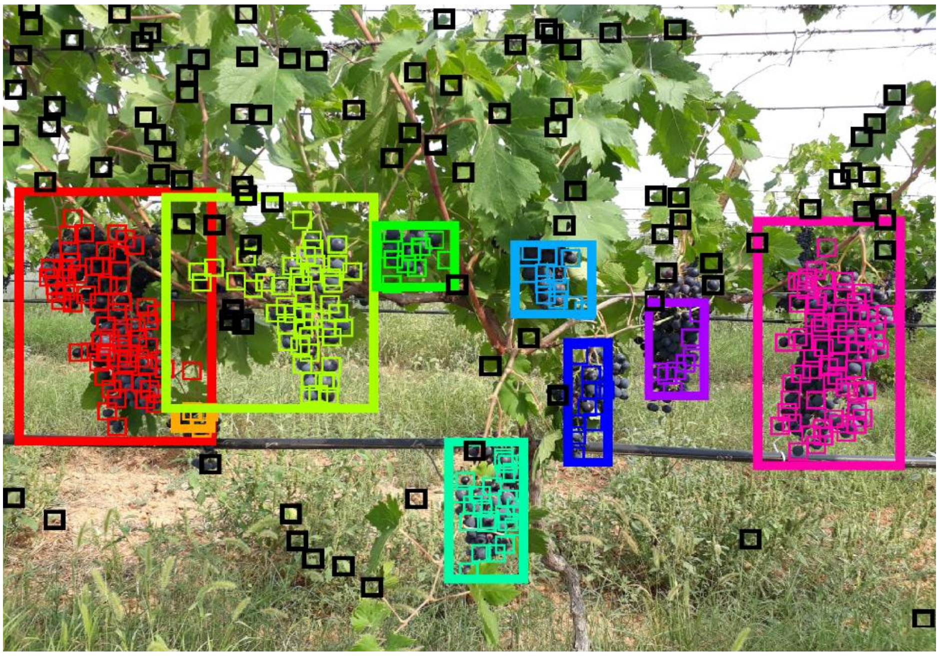
Figure 4.6. Tarsus data-set example of hybrid grape image. (a) Original image (b) results of detection. (c) Ground truth of input image. (d) Ground truth compared with detection mask



(a) (b) (c) (d)

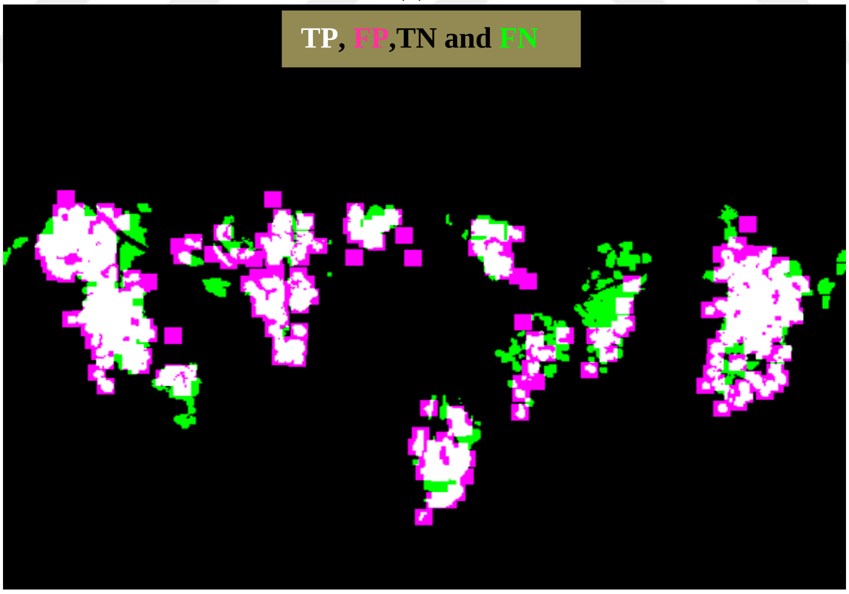
Figure 4.7. Tarsus data-set example of white grape image. (a) Original image (b) results of detection. (c) Ground truth of input image. (d) Ground truth compared with detection mask.

(a)



(b)

(c)

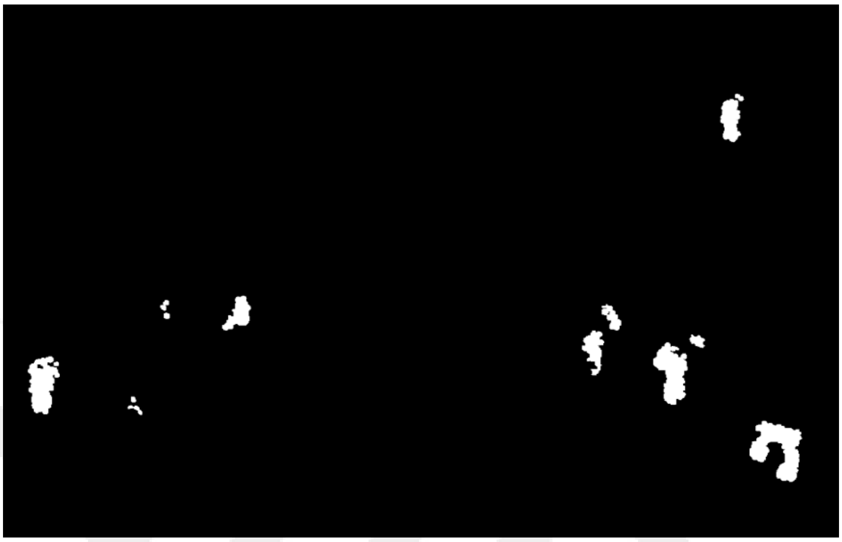


(d)

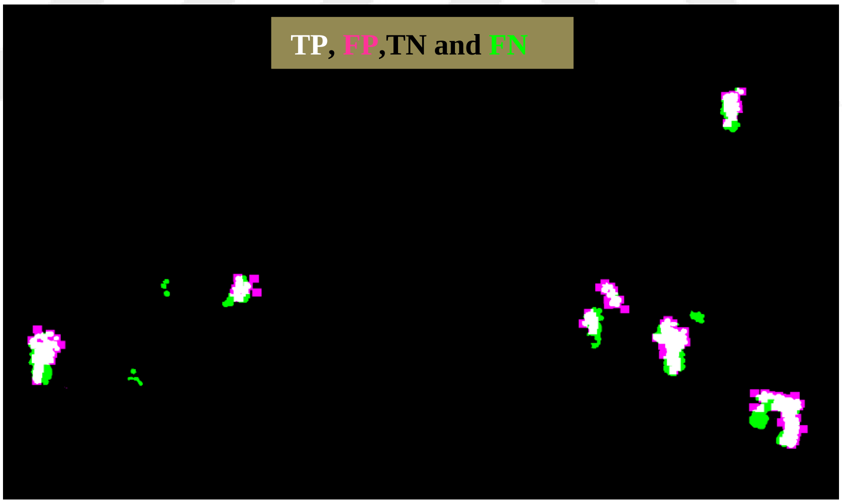
Figure 4.8. An example of Cukurova university grape image. (a) Original image (b) results of detection. (c) Ground truth of input image. (d) Ground truth compared with detection mask.

(a)

(b)

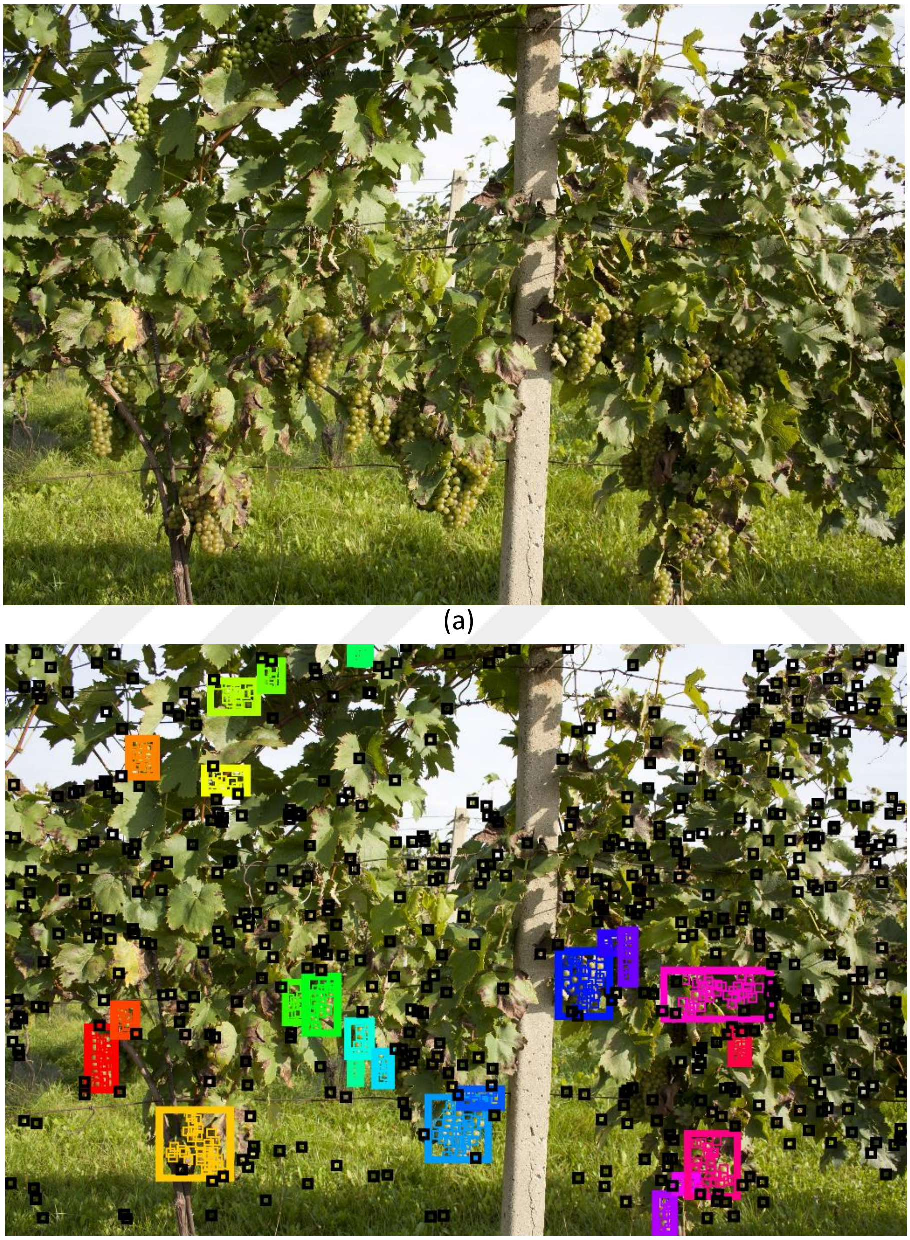


(c)



(d)

Figure 4.9. An example of Iceland data-set grape image. (a) Original image (b) results of detection. (c) Ground truth of input image. (d) Ground truth compared with detection mask.



(a)

(b)

Figure 4.10. Another example of Iceland data-set. (a) Original image. (b) Output image.

5. CONCLUSION AND FUTURE WORK

5.1. Conclusion

This study has presented a visual grape berry and bunches recognition model that uses the combination of both HOG and LBP as a feature vector with ANN classifier and the DBSCAN clustering. The proposed approach capable of recognizing a various color of grapes such as; white and red grapes, under natural light conditions, occlusion levels and distances between the grapes-camera. The approach was compared with other detection strategies using alternative descriptors that are among the most efficient for various classification algorithms such as are ANN and SVM with different type of feature vectors such as LBP, HOG and the combination of both, in CNN the feature will extract automatically from the image. The use of combined shape and texture information (HOG+LBP) with ANN classifier proved to yield better results than the shape or texture information provided individually by HOG and LBP, respectively. Also, an alternative SVM and CNN formulation of the ANN was implemented and compared other classifiers. However, the results show that the traditional ANN with combination of feature yields a better grape bunch detection. The average accuracy of grape bunch detection reached 91.72±2.88%, presenting an efficient method for detecting close 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 ANN classifier with HOG+LBP feature vectors yields an average accuracy of 96.97±0.95%, average precision of 98.11±1.207% and an average sensitivity of 97.57±0.969% in two classes such as a grape berry /non-grape berry pixels detection with the ANN trained for white grapes under natural daylight illumination tested on grapes of other varieties and illumination conditions including red grapes with artificial illumination. The amount of training dataset showed to have an influence on the results. The larger datasets allowed better overall performance. The approach presented performs a multi-scale 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 (FP) on isolated locations, thus improving further the accuracy, sensitivity and precision performance indices. 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.

5.2. Future Work

The planned future work is to optimize the parameters of HOG, LBP and ANN to further improve the recognition performance and reduce processing time. Additionally, another type of features and classifiers are intended to test, 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 centers, and compare this with other interest point selectors. Design a sliding window with an efficient classifier like ANN with Combination of both LBP and HOG feature vectors, that used in the current study may improve the visible grape berry and bunch detection.

REFERENCES

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence, (12), 2037-2041.

Auat Cheein, F. A., & Carelli, R. (2013). Agricultural robotics: Unmanned robotic service units in agricultural tasks. IEEE Industrial Electronics Magazine, 7(3),

Bachche, S. (2015). Deliberation on Design Strategies of Automatic Harvesting Systems: A Survey, 194–222.

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

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.

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.

Díaz, C. A., Pérez, D. S., Miatello, H., & Bromberg, F. (2018). Grapevine buds detection and localization in 3D space based on Structure from Motion and 2D image classification. Computers in Industry, 99, 303-312.

Elkhamssa, Lakehal, and Benmohammed Mohamed. "Robust Content Based Watermarking Algorithm Using Singular Value Decomposition Of Radial Symmetry Maps." Signal & Image Processing 5.1 (2014): 13.

Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise.In Kdd, vol. 96, 1996 ;34: pp. 226-231.

FAOSTAT (Food and Agriculture Organization of the United Nations). (2017). Retrieved from http://www.fao.org/faostat/en/#data Accessed at 2019- 05-24

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.

Gallery images and information: Pesticides In Water Pollution, from the link : http://cerev.info/addpthis-pesticides-in-water-pollution.htm Accessed at 2019- 06-14

Golden Muscat Grape, (2016) available in this link. https://www.starkbros.com/products/berry-plants/grape-vines/golden-muscat-grape accessed at 2019-07-28

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 53

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

Haykin, Simon. "Neural networks and machine learning." *Ontario: Pearson Education* (2009): 147139-9.

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.

IOV (International Organization of Vine and Wine). (2016). Dried Grapes. Retrieved from http://www.oiv.int/en/statistiques/?year=2016&countryCode=TUR accessed at 2019- 06-14

Ivorra, E., Sánchez, A. J., Camarasa, J. G., Diago, M. P., & Tardáguila, J. (2015). Assessment of grape cluster yield components based on 3D descriptors using stereo vision. Food Control, 50, 273-282.

John C. Russ. (2006) Materials Science and Engineering Dept., North Carolina State University, Raleigh, North Carolina, 27695. From the link: https://micro.magnet.fsu.edu/primer/digitalimaging/russ/colorspaces.html Accessed at 2019- 03-05

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.

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.

Loy, G., & Zelinsky, A. (2003). Fast radial symmetry for detecting points of interest. IEEE Transactions on Pattern Analysis and Machine Intelligence,

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.

Mehle, A., Likar, B., & Tomaževič, D. (2017). In-line recognition of agglomerated pharmaceutical pellets with density-based clustering and convolutional neural network. IPSJ Transactions on Computer Vision and Applications, 9(1), 7.

Michael James. (2018) Grape Harvesting-An Overview from the link: https://totalwinesystem.com/grape-harvesting-an-overview/ accessed at 2019- 06-14

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

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.

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.

Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. Pattern recognition, 29(1), 51-59.

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

P. Śkrabánek, Deepgrapes: Precise detection of grapes in low-resolution images, IFAC-PapersOnLine 51(6) (2018) 185–189.

P. Škrabánek, F. Majerík, Detection of grapes in natural environment using hog features in low resolution images, in: Journal of Physics: Conference Series, Vol. 870, IOP Publishing, 2017, p. 012004.

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

Quackenbush, J. (2017). Napa, Sonoma vineyard-worker scarcity sprouts wage growth, alternatives. https://www.northbaybusinessjournal.com/northbay/sonomacounty/6951644-181/napa-sonoma-vineyard-wine-employment

R. Pérez-Zavala, M. Torres-Torriti, F. A. Cheein, G. Troni, (2018) . A pattern recognition strategy for visual grape bunch detection in vineyards, Computers and electronics in agriculture 151.136–149.

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.

Richard OD, Peter EH, David GS. Pattern classification. A Wiley-Interscience. 2001:373-8.

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.

Škrabánek, Pavel & Runarsson, Thomas. (2015). Detection Of Grapes In Natural Environment Using Support Vector Machine Classifier.

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

Thakare, Y. S., & Bagal, S. B. (2015). Performance evaluation of K-means clustering algorithm with various distance metrics. International Journal of Computer Applications, 110 (11), 12-16.

Yarpiz. (2015). Implementation of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) in Matlab from: https://www.mathworks.com/matlabcentral/fileexchange/52905-dbscan-clustering-algorithm Accessed at 2019- 06-14.

CURRICULUM

Bashar AL-SAFFAR Received B.Sc in Electrical Engineering from University of Technology-Baghdad in 2014. Now master student at Department of Electrical and Electronics Engineering at Çukurova University, Adana, Turkey, Under supervision of Asst. Prof. Dr.Sami ARICA