



T.C.

TOKAT GAZİOSMANPAŞA ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

GÖRÜNTÜ İŞLEME TEKNİKLERİ İLE CAM ÜRÜNLERDE HATA TESPİTİ

ÇETİN CEM BÜKÜCÜ

DOKTORA TEZİ

MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Dr.Öğrt Üyesi Levent GÖKREM

2021

Her hakkı saklıdır

T.C.
TOKAT GAZİOSMANPAŞA ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DOKTORA TEZİ

GÖRÜNTÜ İŞLEME TEKNİKLERİ İLE CAM ÜRÜNLERDE HATA TESPİTİ

ÇETİN CEM BÜKÜCÜ

TOKAT
2021

Her hakkı saklıdır

TEZ BEYANI

Tez yazım kurallarına uygun olarak hazırlanan bu tezin yazılmasında bilimsel ahlak kurallarına uyulduğunu, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezin içerdiği yenilik ve sonuçların başka bir yerden alınmadığını, kullanılan verilerde herhangi bir tahrifat yapılmadığını, tezin herhangi bir kısmının bu üniversite veya başka bir üniversitedeki başka bir tez çalışması olarak sunulmadığını beyan ederim.

ÇETİN CEM BÜKÜCÜ

13 Ocak 2021

ÖZET

DOKTORA TEZİ

GÖRÜNTÜ İŞLEME TEKNİKLERİ İLE CAM ÜRÜNLERDE HATA TESPİTİ

Kalite kontrol süreçlerinde ürün çıktısındaki hataları tespit etmede ekonomik, ergonomik ve işlem gücü yeteneklerinden dolayı gömülü sistemlere dayalı özgün tasarımlar ve yazılım geliştirme uygulamaları her geçen gün yaygınlaşmaktadır. Bu çalışmada, kalite kontrol işlemini gözle manuel olarak gerçekleştiren bir cam fabrikasından alınan ürünlerin yüzeylerinde oluşan hataları tespit etmek için gömülü sisteme dayalı otomatik kontrol sistemi gerçekleştirilmiştir. Bu sistemin gerçekleştirilmesi için konveyör bant, mikro-sürücü ve kameralı gömülü sistemden oluşan bir prototip tasarlanmıştır. Gömülü sistem açık kaynak kodlu, morfolojik görüntü işleme teknikleriyle çalışan, Gaussian metodu ile sınır tespiti yapan bir yazılıma sahiptir. Sistemin başarı oranı “Destek Vektör Makinası”, “Kuadratik Diskriminant Analizi” ve “Orta Ölçekli Ağaç” sınıflayıcılar kullanılarak bulunmuştur. Çalışmada elde edilen sonuçlarda %93.2 başarı oranı elde edilmiştir. Sistemin uygulaması bir cam fabrikasında test edilmiş, test işlemi sonucunda sistem cam ürünlerdeki hata tespitini yüksek başarı oranıyla gerçekleştirmiştir.

2021, 94 SAYFA

ANAHTAR KELİMELER: Cam Hataları, Morfolojik Görüntü İşleme, Sınıflandırma, Gömülü Sistemler, Cam Ürünlerde Hata Tespit Prototipi

ABSTRACT

DOCTORATE THESIS

DEFECT DETECTION IN GLASS PRODUCTS USING IMAGE PROCESSING TECHNIQUES

Due to their economical, ergonomic and processing power capabilities, unique designs and software development applications based on embedded systems are becoming more common every day in detecting defects in product output in quality control processes. In this study, an automated control system based on embedded system was performed to detect defects on the surface of products purchased from a glass factory that performs quality control manually by eye. A prototype consisting of the conveyor band and micro drive and camera embedded system was designed for the realization of this system. The embedded system has an open source software that works with morphological image processing techniques and makes boundary determination by gaussian method. The success rate of the system was found by classifying it with Support Vector Machine, Quadratics Discriminant and Medium Tree classifiers. The success rate is %93.2 in the models which are used to determine the relationship between experimental data. The application of the system was tested in a glass factory, and as a result of the test process, the system achieved a high success rate of defect detection in glass products.

2021, 94 PAGE

KEYWORDS: Glass Defects; Morphological Image Processing; Classification; Defect; Embedded Systems; Detection Prototype in Glass Products.

TEŞEKKÜR

Bu çalışmanın yürütülmesinde verdikleri destek, gösterdikleri anlayış ve ilgiden dolayı danışman hocam sayın Dr. Öğr.Üyesi Levent GÖKREM'e;

Tez izleme komitesinde yer alan ve değeri katkılarıyla çalışmamda destek veren Sayın Prof. Dr. Mahmut HEKiM ve Dr. Öğr. Üyesi Bülent TURAN'a;

Doktora tez ve ders süresince desteğini bizden esirgemeyen Prof. Dr. Sefa TARHAN'a;

Bu sürecin en başında beni yüreklendiren yüksek lisans tez danışmanım Prof. Dr. Hüseyin UZUN BOYLU'ya;

Doktora sürecinin sonuna kadar her aşamada beni destekleyen Dr. Öğr. Üyesi Kemal ADEM'e;

Çalışmamda tez yazım süresinde her zaman desteğini hissettiğim çalışma arkadaşım Öğr. Gör. Haydar YANIK'a;

Doktora çalışmamda ve yayın süresince tüm çevirilerimde bana inanılmaz destek veren öğretmen arkadaşım Recep TOPUZ'a;

Arkadaşım Öğr. Gör. Ali AKTAŞ'a;

Desteklerinden dolayı teşekkür ederim.

Annem'e ;

Çetin Cem BÜKÜCÜ

13 Ocak 2021

İÇİNDEKİLER

ÖZET	1
ABSTRACT.....	2
TEŞEKKÜR	3
İÇİNDEKİLER	4
SİMGELER VE KISALTMALAR	6
ŞEKİL LİSTESİ.....	8
TABLO LİSTESİ.....	10
1. GİRİŞ	11
2. GENEL BİLGİLER.....	14
2.1 Görüntü İşleme	14
2.2 Açık Kaynak.....	19
2.2.1 Açık kaynak programlama dili PYTHON	19
2.2.2 Açık kaynak kütüphanesi	19
2.2.3 Morfolojik görüntü işleme.....	22
2.3 Sınıflayıcılar	27
2.3.1 SVM sınıflandırıcısı	28
2.3.2 Veri akış madenciliği.....	31
2.3.3 NB sınıflandırıcısı	31
2.3.4 DT sınıflayıcı.....	33
2.3.5 Güçlendirilmiş ağaç sınıflayıcı.....	36
2.3.6 LDA ve QDA	37
2.3.7 K-NN sınıflayıcı	40
2.3.8 LRA sınıflayıcı	41
2.3.9 YSA sınıflayıcı	43
3. MATERYAL VE YÖNTEM.....	45
3.1 Materyal.....	45
3.2 Yöntem.....	59
3.2.1 Morfolojik görüntü işleme, kodlama ve sınıflandırma.....	60
3.2.2 Sınıflandırma işlemi	64
3.2.3 Geçerlilik kriteri	67
4. BULGULAR VE TARTIŞMA	68
5. SONUÇ.....	75
6. KAYNAKLAR	77
7. EKLER.....	81
7.1 EK-1 Matlab kodu	81

7.2 EK-3 Çalışmada kullanılan “BATIK” cam örnekleri.....	86
7.3 EK-4 Çalışmada kullanılan “ÇİZİK” cam örnekleri.....	87
7.4 Ek 18 Özgeçmiş	88



SİMGELER VE KISALTMALAR

Simgeler

S
A, B
f
t
se
w, a, b, c, x, y, x_1 , x_2
 Δ
H
A
V
Wp
Z
 α
D
f
h

Açıklama

Veri sınıfı
Veri kümeleri
Fonksiyon
Zaman
İşlenmiş görüntü verisi
Eşitlik değişkenleri
Delta
Entropi değişkeni
Akım
Gerilim
Kamera çözünürlüğü
Kamera uzaklığı
Kamera görüş açısı
Mesafe değişimi
Odak uzaklığı
Yükseklik

Kısaltmalar

CART
PCA
KNN
2-B
3-B
TI
Tc
BO
BS
CS
SU
TH
TP
GUI
EP, PP
IT
PyPI
OpenCV
SVM
DA
RPM
Na₂CO₃
SATA
WLAN

Açıklama

Classification and Regression Tree
Principal Component Analysis
K-En Yakın Komşu
iki-boyutlu
üç-boyutlu
Etiketlendirilmemiş ve sınıflandırılmış veri
Etiketlendirilmemiş ve sınıflandırılmamış veri
Toplam batık hatasının tüm yüzeyde kapladığı alan
Batık hatası
Toplam çizik hatası
Sağlam ürün
Toplam Hata
Toplam yüzey piksel sayısı
Grafik kullanıcı arayüzü
Konveyör bant çözümleri
İnternet Teknolojileri
Python Paket Endeksinde
Açık kaynak görüntü işleme kütüphanesi
Destek vektör makinası
Doğru akım
Dakikadaki devir sayısı
Sodyum Karbonat
Serial veri yolu
Kablosuz bölgesel ağ

USB	Universal seri port
FoV	Görüş Alanı
PPM	Metre başına düşen piksel
IoT	Nesnelerin interneti
GPIO	Genel giriş çıkış birimi
VFoV	Yatay görüntüleme alanı
HFoV	Dikey görüntüleme alanı



ŞEKİL LİSTESİ

Şekil	Sayfa
Şekil 2.1 Görüntü işleme algoritması (Toraman, 2006)	17
Şekil 2.2 Yapısal eleman örnekleri	23
Şekil 2.3. Açma işlemi (Acar ve Bayram, 2009)	24
Şekil 2.4. Kapama işlemi (Acar ve Bayram, 2009)	25
Şekil 2.5 Aşındırma işlemi (Boztoprak ve ark., 2007)	26
Şekil 2.6 Genleşme işlemi (Boztoprak ve ark., 2007)	27
Şekil 2.7 NLSVM sınıflayıcı (Osuna, 1998)	29
Şekil 2.8 LSVM sınıflandırılmayan veri kümesi (Osuna, 1998)	30
Şekil 2.9 NLSVM tarafından sınıflandırılmış veri kümesi (Hasan, 2017)	30
Şekil 2.10 Veri akış madenciliği	31
Şekil 2.11 NB sınıflayıcı (Yang, 2019)	32
Şekil 2.12 DT'nin genel yapısı (Catanzaro, 2013)	34
Şekil 2.13 DT deneme seti (Catanzaro, 2013)	35
Şekil 2.14 DT test verisi (Catanzaro, 2013)	36
Şekil 2.15 Güçlendirilmiş ağaçla sınıflayıcı (Liu ve ark., 2017)	37
Şekil 2.16 LDA ve QDA analizi (Hastie, 2009)	38
Şekil 2.17 K-NN sınıflayıcı (Catanzaro, 2013)	41
Şekil 2.18 LRA veri (Şentürk, 2011)	42
Şekil 2.19 LRA sınıflandırılmış veri (Şentürk, 2011)	42
Şekil 2.20 LRA sınıflandırma sonucu (Şentürk, 2011)	43
Şekil 2.21 YSA sınıflayıcı (Fırat ve ark, 2004)	44
Şekil 3.1 Kullanılan prototipin konstrüksiyon yapısı	45
Şekil 3. 2 Gerçek zamanlı hata tespiti yapan yeni bir prototip	46
Şekil 3. 3 Konveyör bant motoru	47
Şekil 3. 4 Sistemin eşik değerinin üstündeki hızda alınan ekran görüntüsü	48
Şekil 3.5 Sistemin eşik değerindeki hızda alınan ekran görüntüsü	48
Şekil 3.6 Aydınlatma düzeneği	50
Şekil 3.7 Kamera görüş alanı (Schöning ve Heidemann, 2016)	51
Şekil 3.8 Kameraya ait FoV alanı	52
Şekil 3.9 Tercih edilen kamera	52
Şekil 3.10 h=45 cm'den alınan görüntü	54
Şekil 3.11 h=50 cm'den alınan görüntü	54
Şekil 3.12 h=52 cm'den alınan görüntü	55
Şekil 3.13 FoV ve PPM değerine göre kamera mesafesi	56
Şekil 3.14 Görüntü Alma Merkezi	56
Şekil 3.15 Prototipte kullanılan mikro sürücü	57
Şekil 3.16 Görüntü alma merkezi ve gömülü sistem	58
Şekil 3.17 Gömülü sistemin platformu	59
Şekil 3.18 Akış diyagramı	60
Şekil 3.19 Orjinal cam görüntüsü	62
Şekil 3.20 Hata tespiti yapılmış ekran görüntüsü	62
Şekil 3.21 Prototipten alınan gerçek zamanlı ekran çıktıları	63
Şekil 3.22 DT sınıflayıcı (Hatipoğlu, 2018)	66

Şekil 4.1 Hata dağılım grafiği.....	68
Şekil 4.2 Karar ağaçları ile sınıflandırılma	69
Şekil 4.3 SVM.....	70
Şekil 4. 4 QDA.....	71
Şekil 4. 5 ROC eğrisi	74



TABLO LİSTESİ

<u>Tablo</u>	<u>Sayfa</u>
Tablo 1.1 Cam ürünlerde oluşan hata tipleri.....	12
Tablo 4.1 FFA-DT	72
Tablo 4.2 FFA-SVM.....	72
Tablo 4.3 FFA-QDA.....	72
Tablo 4.3 QDA istatistiksel sonuçları.....	72



1. GİRİŞ

Cam ve cam tabanlı ürünler sağlık açısından uygun özelliklere sahip olmaları ve pek çok alanda kullanılabilirlikleri dolayısıyla hayatımızın her alanında önemli bir yer tutmaktadır. Dünya ve Avrupa pazarlarında rakipleriyle yoğun bir rekabet içerisinde olan Türk cam sanayisi her geçen yıl büyümesini sürdürerek performansını artırmaktadır (Anonim, 2017; Anonim, 2018; Anonim, 2019).

2019 yılı T.C. Bilim, Sanayi ve Teknoloji Bakanlığı cam sektörü raporuna göre; Türkiye düz cam üretiminde; Avrupa'da 1. dünyada 5. sırada, cam ev eşyasında; Avrupa'da 2., dünya da 3., cam ambalaj üretiminde; Avrupa da ve dünya da 5. sırada, soda külü (Na_2CO_3) üretiminde Avrupa'da 5., dünya da ise 9. sıradadır (Anonim, 2019).

Cam ürünleri hammadde olarak doğal kaynakları kullanmaktadır. Kullanmış olduğu doğal kaynakların doğru kullanımı önemlidir. Doğal kaynakların israfına sebep vermemek için üretim esnasındaki hatayı en az seviyeye indirilmesi gerekir. Kalite kontrol sürecinde en erken aşamada hatanın tespiti ve geri dönüşümü, doğal kaynakların israfını önleyecektir. Böylece hatalı ürünün iadesi sebebiyle meydana gelecek olumsuz etkiler ortadan kalkacaktır. Çünkü geri iade süreci üretici açısından azımsanamayacak bir maliyete ve enerji tüketimine sebep olmaktadır.

Günümüzde cam üretimindeki hatalar çoğunlukla insan kaynaklıdır. Bu durum cam üretim aşamasının çok uzun ve zor bir süreci kapsamasından kaynaklanmaktadır. İnsan kaynaklı hata tespiti, ortamdaki ışık, toz, göz yorulması gibi etmenlerden dolayı yanılma payına açıktır. Özellikle kaplama yapılmış camlarda bu tür hata tespiti kaplamaya bağlı olarak işin zorluğunu artırmaktadır. Camın üretimi, işlenmesi, nakliyesi ve işçilik sürecindeki hatalar maliyetin de artmasına neden olmaktadır.

Çok yönlü bir madde olan cam; şeffaflığı, şeklinin bozulmaması, koku ve tat vermemesi nedeniyle birçok ürünün ambalajlanmasında kullanıldığı gibi pek çok üründe de tamamlayıcı aksesuar olarak kullanılmaktadır. Cam üretiminde kullanılan hammaddelerin çıkarılması doğal kaynakların tüketimine sebep olduğu için çevreye

zarar vermektedir. Bu durum cam üretiminde ve işlenmesinde hata payını en aza indirmeyi zorunlu kılmaktadır. Bu yüzden oluşabilecek olan hataların piyasaya sürülmesinden önce tespit edilmesi büyük önem taşımaktadır (Nishu ve Agrawal, 2011).

Cam üzerinde meydana gelen hatalar birçok nedene bağlı olabilir. Aynı kusur şekline sahip olsalar da hataların meydana geliş sebepleri farklı olabilmektedir (Singh vd., 2013). Cam ürünlerde oluşan hata tipleri ise Tablo 1.1’de görülmektedir.

Tablo 1.1 Cam ürünlerde oluşan hata tipleri

Hatalar	Hata Tanımları
Yabancı Maddeler	Cam içerisinde gömülü olarak bulunan yabancı maddeler
Kirli bölge(pusu)	Arka plandan kısmen ayrılabilen rengi bozuk bölgeler
Çizikler ve batıklar	Cam yüzeylerde belirlenmiş dairesel ve dikdörtgen formundaki hatalar
Kabarcıklar	Cam içerisinde oluşmuş hava kabarcıkları
Delik ve kir	Özellikler üretim aşamasında yüzeyde oluşan delik ve kir

Cam ürünlerde oluşan hatalar daha çok geleneksel yöntem ile (gözle) tespit edilmektedir. Bu yöntem; yorgunluğa bağlı olarak görmedeki hassasiyet azalımı ve düşük kontrast, toz, kirli zemin gibi çevresel ortamdaki olumsuzluklar sebebiyle hata tespitini direkt etkilemektedir. Bunun sonucunda zaman, hammadde, enerji kaybına ve buna bağlı olarak da ürün kalitesinin düşmesine sebep olmaktadır.

Hataları bulmak ve doğru değerlendirmek otomatik kalite kontrol sisteminin hedefidir. Otomatik kalite kontrol sistemi gibi birçok endüstriyel sistemde, hataların saptanması ve giderilmesinde kullanılmıştır (Spinola ve ark., 2011; Qu, Zou ve ark., 2016; Chen ve ark., 2016; Cabral ve Araújo, 2012; Keser ve ark., 2010; Hocenski ve ark., 2006; Elbehriy ve ark., 2005; Chao ve ark., 2006; Liu ve Chen ,2011). Cam yüzeyler yansıtıcı olduğundan sistemin hatayı bulması çok da kolay değildir (Engineering, 2013). Bunun için özel yöntemler kullanılmaktadır. Bu sistemlerde kameralardan alınan görüntüler sayesinde hataları tespit edilebilmektedir (Öztürk ve Akdemir, 2018).

Makine görme sistemleri; endüstriyel üretim sistemlerinde hata tespitini anlık olarak gerçekleştirebilen, kullanıldığı sistemin mimarisini sadeleştiren ve basitleştiren sistemlerdir. Bu sistemlerde hata tespiti, konveyör bant üzerindeki ürünlerin resimleri

kamera yardımıyla başlar. Ardından çekilen görüntü, görüntü işleme algoritmalarıyla işlenir. Son olarak da ürünün kalitesi belirlenir. Makine görme sistemlerinde hata tespiti için, renk uzayı dönüşümü, kenar tespiti, Fourier analizi, eşik, kümeleme, anisotropik yayılma modeli ve Sobel operatörü metotları gibi pek çok metot kullanılır (Nishu ve Agrawal, 2011; Singh ve ark, 2013; Öztürk ve Akdemir,2018; Öztürk ve Akdemir,2015; Keser ve ark., 2010; Chao ve ark ,2006; Cabral ve Araújo,2012; Engineering, 2013).

Bu tez çalışmasında; cam ve cam ürünler üzerindeki hatalar morfolojik görüntü işleme tekniklerini kullanarak tespit edilmiştir. Çalışmada hata tipleri genel olarak batık yüzey hataları ve çizik yüzey hataları olarak iki sınıfa ayrılmıştır. Batık yüzey hataları dairesel formda, çizik yüzey hataları ise dikdörtgen formunda, çizik görünümlü hatalardır. Batık ve çizik hatalarını gerçek zamanlı olarak tespit eden gömülü sistem tasarımı yapılmıştır. “QDA (Quadratic Discriminant Analysis-Kuadratik Diskriminant Analizi)”, “SVM (Support Vector Machine-Destek Vektör Makinesi)” ve “DT (Decision Tree-Karar Ağacı)” sınıflayıcılar kullanılarak yapılan sınıflandırma sonucunda yüksek başarı oranı elde edilmiştir.

2. GENEL BİLGİLER

2.1 Görüntü İşleme

Sinyal işleme, elektrik mühendisliği ve matematikte analog ve dijital sinyallerin analizi ve işlenmesiyle ilgilenen ve sinyaller üzerinde depolama, filtreleme ve diğer işlemlerle ilgilenen bir disiplindir. Bu sinyaller arasında iletim sinyalleri, ses veya ses sinyalleri, görüntü sinyalleri ve diğer sinyaller yer alır (Goldberger ve Ng, 2010). Tüm bu sinyallerde girdiler ve çıktılar görüntülerdir. Bu tür sinyallerin işlenmesi görüntü işlemenin alanıdır. Adından da anlaşılacağı gibi, görüntülerin işlenmesiyle ilgilenir. Görüntü işlemenin temel aldığı üç aşama vardır. İlk olarak görüntünün elde edilmesi, daha sonra alınan görüntünün incelenmesi ve işlenmesi, en son aşamada ise görüntü analizine dayanan görüntünün değiştirilebileceği veridir. Görüntü işleme ayrıca analog görüntü işleme ve dijital görüntü işleme olarak ikiye ayrılmıştır. Analog görüntü işleme, analog sinyallerde yapılır. İki boyutlu analog sinyaller üzerinde işlemeyi içerir. Bu tür işlemede, görüntüler elektrik sinyalini değiştirerek elektriksel yollarla manipüle edilir. Yaygın örnek, televizyon görüntüsüdür (Ansari ve Borse, 2013).

Kamera teknolojisindeki gelişmeler dijital görüntü işlemenin daha fazla kullanılmasına sebep olmuş bu durum analog görüntü işlemenin yerini dijital görüntü işlemenin almasına sebep olmuştur. Dijital görüntü işleme, dijital bir görüntü üzerinde işlemler gerçekleştiren bir dijital sistem geliştirmeyi amaçlar (Gonzalez ve Woods, 2000).

Dijital işleme uygun hale gelmek için bir görüntü fonksiyonu $f(x, y)$ hem uzamsal hem de büyüklükte sayısallaştırılmalıdır. Tipik olarak, analog video sinyalini örneklemek ve nicelendirmek için bir çerçeve yakalayıcı veya sayısallaştırıcı kullanılır. Bu nedenle dijital bir görüntü oluşturmak için sürekli verileri dijital forma dönüştürülmesi gerekir. Yapıldığı iki adım vardır:

- Örnekleme
- Niceleme

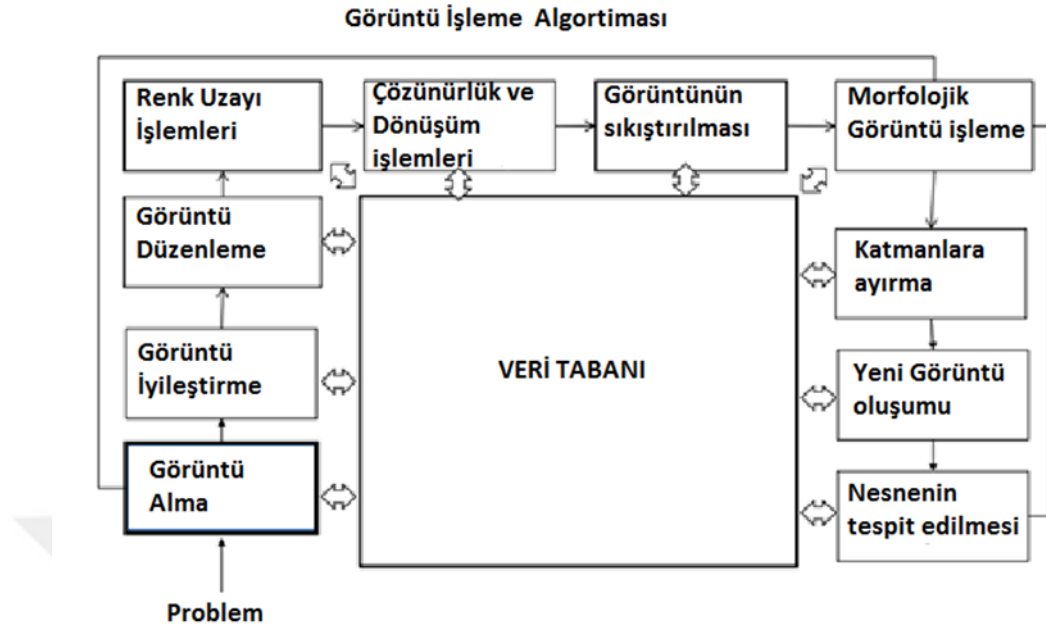
Örnekleme oranı, sayısallaştırılmış görüntünün uzamsal çözünürlüğünü belirlerken, niceleme seviyesi sayısallaştırılmış görüntüdeki gri seviyelerin sayısını belirler. Örneklenen görüntünün büyüklüğü, görüntü işlemede dijital bir değer olarak ifade edilir. Görüntü fonksiyonunun sürekli değerleri ile dijital eşdeğeri arasındaki geçiş niceleme olarak adlandırılır. Niceleme düzeylerinin sayısı, görüntüdeki ince gölgeleme ayrıntılarının insan tarafından algılanması için yeterince yüksek olmalıdır. Yetersiz parlaklık seviyeleri ile nicelleştirilmiş görüntüde yanlış konturların oluşması ana problemidir (Niu ve ark., 2016).

Görüntü enterpolasyonu, görüntünün bir piksel ızgarasından diğerine yeniden boyutlandırıldığında veya deforme edildiğinde gerçekleşir. Görüntüyü yeniden boyutlandırma, toplam piksel sayısını artırmak veya azaltmak gerektiğinde gereklidir, oysa mercek distorsiyonunu (şeklen bozulması) düzeltirken veya bir görüntüyü döndürürken yeniden eşleme yapılabilir. Yakınlaştırma piksel miktarını artırmak anlamına gelir, böylece bir görüntü yakınlaştırıldığında daha fazla ayrıntı görünür. Enterpolasyon, bilinmeyen noktalardaki değerleri tahmin etmek için bilinen verileri kullanarak çalışır. Görüntü enterpolasyonu iki yönde çalışır ve çevreleyen piksellerdeki değerlere dayalı olarak bir pikselin yoğunluğunun en iyi tahminini elde etmeye çalışır. Ortak enterpolasyon algoritmaları iki kategoriye ayrılabilir. Bunlar uyarlanabilir ve uyarlanabilir olmayan yöntemlerdir. Uyarlanabilir yöntemler, enterpolasyon yaptıklarına bağlı olarak değişir, oysa uyarlanabilir olmayan yöntemler tüm pikselleri eşit olarak ele alır (Thévenaz ve ark., 2009).

Bilgisayar algoritmaları, dijital görüntü işlemede çok önemli bir rol oynar. Geliştiriciler, dijital görüntü algılama, analiz, yeniden yapılandırma, geri yükleme, görüntü veri sıkıştırma, görüntü geliştirme, görüntü tahmini ve görüntü spektral tahmini gibi farklı görevleri çözmek için birden çok algoritma kullanır.

Dijital görüntü işlemenin başlıca teknikleri aşağıdaki gibidir:

1. Görüntü düzenleme; temelde dijital görüntülerin grafik yazılım araçlarıyla değiştirilmesi anlamına gelir.
2. Görüntü restorasyonu; kaybolan bilgileri geri almak için alınan bozuk görüntüden temiz bir orijinal görüntünün tahminini ifade eder.
3. Bağımsız bileşen analizi; çok değişkenli bir sinyali hesaplamalı olarak ek alt bileşenlere ayırır.
4. Anisotropik difüzyon; genellikle Perona-Malik Difüzyon olarak bilinir, mümkün görüntünün önemli bölümlerini kaldırmak zorunda kalmadan görüntü gürültüsünü azaltmak için yapar.
5. Doğrusal filtreleme; zamanla değişen giriş sinyallerini işlemek ve doğrusallık kısıtlamasına tabi olan çıkış sinyalleri üretmek anlamına gelen başka bir dijital görüntü işleme tekniğidir.
6. Sinir ağları; çeşitli görevleri çözmek için makine öğreniminde yaygın olarak kullanılan hesaplama modellerinden biridir.
7. Pikselleştirme; genellikle basılı görüntüleri sayısallaştırılmış görüntülere (GIF gibi) dönüştürmeyi ifade eder.
8. Temel bileşenler analizi; özellik çıkarımı için kullanılabilen bir dijital görüntü işleme tekniğidir.
9. Kısmi diferansiyel denklemler; gürültünün giderilmesinde kullanılan bir tekniktir.
10. Gizli Markov modelleri; iki boyutlu görüntü analizi için kullanılan bir tekniktir.
11. Dalgacıklar; görüntü sıkıştırma için kullanılan matematiksel bir işlevi ifade eder.
12. Kendi kendini organize eden haritalar, görüntüleri birkaç sınıfa ayırmak için dijital bir görüntü işleme tekniğidir.
13. Morfolojik görüntü işleme tekniği, kenar belirleme, bölümleme, görüntü sıkıştırma gibi teknikleri kullanmaktadır (Duman, 2019).



Şekil 2.1 Görüntü işleme algoritması (Toraman, 2006)

Şekil 2.1’de görüntü işleme algoritması verilmiştir. Görüntü edinme, dijital görüntü işlemenin temel adımlarının ilk adımı veya işlemidir. Görüntü edinimi, halihazırda dijital biçimde olan bir görüntü vermek kadar basit olabilir. Genel olarak, görüntü elde etme aşaması, ölçekleme vb. gibi ön işlemeyi içerir. Görüntü geliştirme, dijital görüntü işlemenin en basit ve en çekici alanları arasındadır. Temel olarak geliştirme tekniklerinin arkasındaki fikir, gizlenmiş ayrıntıları ortaya çıkarmak veya basitçe bir görüntüdeki ilgi çekici belirli özellikleri vurgulamaktır. Parlaklık ve kontrastın değiştirilmesi gibi özelliklerdir (Demirbaş ve ark., 2007).

Görüntü düzenleme, bir görüntünün görünümünü iyileştirmekle de ilgilenen bir alandır. Bununla birlikte, öznel olan iyileştirmenin aksine, görüntü restorasyonu nesneldir, çünkü restorasyon teknikleri, görüntü bozulmasının matematiksel veya olasılıksal modellerine dayanma eğilimindedir.

Renk uzay işlemleri, dijital görüntülerin internet üzerinden kullanımının önemli ölçüde artması nedeniyle önem kazanan bir alandır. Bu, dijital bir alanda renk modelleme ve işlemeyi içerebilir.

Çözünürlük ve dönüşüm işlemleri, görüntüleri çeşitli çözünürlük derecelerinde temsil etmenin temelidir. Görüntüler, veri sıkıştırma ve piramidal gösterim için art arda daha küçük bölgelere bölünür.

Görüntünün sıkıştırılması, bir görüntüyü kaydetmek için gereken depolamayı veya onu iletmek için bant genişliğini azaltma teknikleriyle ilgilenir. Özellikle internet kullanımında verilerin sıkıştırılması çok gereklidir.

Morfolojik görüntü işleme, şeklin gösterimi ve tanımlanmasında yararlı olan görüntü bileşenlerini çıkarmak için araçlarla ilgilenir (Gonzalez ve Woods, 2000).

Katmanlara ayırma, bir görüntüyü kurucu parçalarına veya nesnelere böler. Genel olarak, otonom bölümlendirme, dijital görüntü işlemede en zor görevlerden biridir. Sağlam bir bölümlendirme prosedürü, süreci, nesnelerin ayrı ayrı tanımlanmasını gerektiren görüntüleme sorunlarının başarılı bir şekilde çözülmesi için uzun bir yol sağlar. Temsil ve açıklama, hemen hemen her zaman, bir bölgenin sınırını veya bölgenin kendisindeki tüm noktaları oluşturan, genellikle ham piksel verileri olan bir bölümlendirme aşamasının çıktısını takip eder. Bir temsilin seçilmesi, ham verileri sonraki bilgisayar işlemesi için uygun bir biçime dönüştürmek için çözümün yalnızca bir parçasıdır. Açıklama, ilgi konusu bazı nicel bilgilerle sonuçlanan veya bir nesne sınıfını diğerinden ayırt etmek için temel olan özniteliklerin çıkarılmasıyla ilgilenir (Ansari ve Borse, 2013).

Nesnenin tespit edilmesi, tanımlayıcılarına göre bir nesneye "araç" gibi bir etiket atayan süreçtir. Yüksek seviyeli görüntü işleme grubuna giren bu aşamada, çeşitli karar verme mekanizmaları (yapay zeka algoritmaları gibi) ile görüntüdeki arka plandan çıkarılmış nesnelerin veya bölgelerin etiketlenmesi, sınıflandırılması yapılır. Görüntüden elde edilen ham bilgilerin, ilgilenilen ayrıntıların ön plana çıkarılmasıdır. Yani aranan özellikli alanların arka plandan ve birbirinden ayırt edilmesidir (Vyas ve ark., 2018).

2.2 Açık Kaynak

Açık kaynak, kamu lisansına sahip Linux çekirdeği ve destekçileri tarafından üretilmiş işletim sistemleridir. Mikrodenetleyiciler açık kaynak işletim sistemi ve açık kaynak kütüphaneleri kullanarak görüntü işleme yapılabilirler.

2.2.1 Açık kaynak programlama dili PYTHON

Açık kaynak işletim sistemleri ve Windows üzerinde çalışabilen, diğer platformların kütüphanelerine destek veren bir yapıdadır. Bu eklentilere verdiği destek sayesinde en fazla kodlama yapılan diller arasında yer almaktadır. Özellikle görüntü işleme teknikleri açısından en uygun yapıya sahip programlama dilidir. Açık kaynak yapıda olmasından dolayı derleme hızı açısından düşük konfigürasyonlara sahip mikrodenetleyiciler için uygun kodlama editörüdür (Chityala ve Pudipeddi, 2020).

2.2.2 Açık kaynak kütüphanesi

OpenCv kütüphanesi, açık kaynak koduna sahip görüntü işleme teknikleri üzerine çalışan bir kütüphanedir. Günümüzde pek çok programlama diline destek veren OpenCv kütüphanesi görüntü işlemenin yaygınlaşmasına önemli katkı sağlamaktadır. İçerisinde bulunan birçok birleşen barındırır. Bunlardan “core” bileşeni temel yapıları içerisinde bulundururken çizim nesneleri XML yapısı tarafından oluşturulur. “HighGui” birleşeni resim görüntüleme, okuma ve grafik arayüz gibi öğelerin üzerinde çalışmayı sağlayan metotları bünyesinde barındırır. “imgproc()” bileşeni görüntü işleme tekniklerinden kenar bulma, nesne tanımlama, renk uzayı ve renk uzayı yönetimi işlemlerini gerçekleştirir. “imcodec()” bileşeni ise dosyada bulunan video ve resimlere erişim sağlayarak onların okuma–yazma işlemlerini yerine getirir. Video bileşeni ise gerçek zamanlı görüntü işleme tekniklerini ve video görüntüleri üzerindeki görüntü işleme tekniklerini yerine getirmektedir. OpenCV, gerçek zamanlı görüntü işleme tekniklerine çok basit bir ara yüz sağlar. Bir videoyu işlemek; video akışının, kareler dizisi olarak ele alınması ve her bir görüntü karesinin ayrı ayrı işlenmesi ile yapılır. Video işlemede veri,

görüntü karelerinden ibarettir. Her kare, $f(x,y)$ resim boyutunda olan (x,y) piksel dizisi olarak temsil edilebilen bir görüntüdür. Başka bir deyişle bir görüntü, herhangi bir noktada matematiksel fonksiyon $f(x,y)$ ile tanımlanan iki boyutlu bir matristen (renkli görüntüler durumunda 3-D), herhangi bir noktada piksel değerini vermekten başka bir şey değildir. Piksel değeri, pikselin ne kadar parlak olduğunu ve ne renk olması gerektiğini tanımlar (Howse, 2013).

Video işlemede kaydedilmiş bir video dosyasından veya gerçek zamanlı kameradan alınan bir görüntü olması işlem basamaklarını değiştirmez. Değişen sadece kamera veya dosya dizininin adı olacaktır. Video işleme için genel işlem basamakları şu şekilde olmalıdır (Joshi, 2015).

```
import numpy as np
import cv2
```

- Pytonda çok boyutlu bir dizi ve matris veri yapıları içeren trigonometrik, istatistiksel ve cebirsel rutinler gibi diziler üzerinde bir dizi matematiksel işlem gerçekleştirmek için kullanılabilen çokça matematiksel, cebirsel ve dönüştürme işlemlerini barındıran “numpy” kütüphanesi çağırılır ve kod satırında “np” olarak kısaltma kullanılmasına olanak sağlar.

- Bir başka görüntü işleme kütüphanesi olan “cv2” kütüphanesi sistemin içine aktarılır. Bu kütüphane görüntü işlemede kullanılacak temel fonksiyonları içerir. Bu fonksiyonlar görüntü okuma, yazma, kaydetme, filtreleme, boyutlama gibi fonksiyonlardır.

```
cap = cv2.VideoCapture(0)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi',fourcc, 20.0, (640,480))
```

- “cap = cv2.VideoCapture(0)”; Bu fonksiyon, görüntü karelerinin sisteme bağlı olan bir kameradan mı, yoksa haricen bağlı bir kameradan mı alınacağını gösteren, “cv2” kütüphanesinin bir bileşenidir. “cv2.VideoCapture(0)” fonksiyonunda kamera seçiminin sıfır (0) olması bilgisayarınızdaki ilk Web kameranın kullanıldığı anlamını taşır.

- “cv2.VideoWriter”, “cv2.VideoWriter_fourcc” ve “cv2.cv.FOURCC” OpenCV ile video dosyaları oluşturmak, yeniden kaydetmek ve saniyede alınması gereken görüntü kare sayısını belirten cv2 kütüphanesini kullanılan bileşenlerdir. Ayrıca FourCC, video codec bileşenini belirtmek için kullanılan 4 baytlık bir koddur.

```
while(True):  
    ret, frame = cap.read()
```

- Bu kod, ret ve frame'in cap.read () olarak tanımlandığı sonsuz bir döngü başlatır (daha sonra bir break ifadesiyle kırılır). Temel olarak, ret bir dönüş olup olmadığına ilişkin bir mantıksal değerdir, video görüntüsünü karelere bölen bir döngüdür.

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

- Burada, griye dönüştürülen çerçeve olarak yeni bir gri değişkeni tanımlanır. Bunun yazılımına dikkat edilmelidir BGR2GRAY. OpenCV' nin renkleri BGR (Mavi, Yeşil, Kırmızı) olarak okuduğunu ve çoğu bilgisayar uygulamasının RGB (Kırmızı, Yeşil, Mavi) olarak okuduğunu unutmamak önemlidir.

```
cv2.imshow('frame',gray)
```

- “imshow” yeni oluşturulan ve renk dönüşümü yapılan görüntünün gösterilmesini sağlar. Burada işlem öncesi kullandığımız ve işlem sonucunda oluşan görüntünün aynı anda gösterilmesi istenilirse her iki görüntü için ayrı ayrı “imshow” yapılmalıdır.

```
if cv2.waitKey(1) & 0xFF == ord('q'):  
    break
```

- Bu ifade çerçeve başına yalnız bir kere çalışır. Eğer bir çıkış verisi alınırsa veya “q” tuşuna basılırsa “while” döngüsünden kodu kopartıp (break) çıkar.

```
cap.release()  
cv2.destroyAllWindows()
```

- Görüntü veya video alımını durdurur ve açık olan bütün görüntü ekranlarını (imshow) kapatır.

Genel bir video işleme kodu aşağıda verilmiştir.

```
import numpy as np
import cv2

cap = cv2.VideoCapture(1)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi',fourcc, 20.0, (640,480))

while(True):
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    out.write(frame)
    cv2.imshow('frame',gray)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
out.release()
cv2.destroyAllWindows()
```

2.2.3 Morfolojik görüntü işleme

Morfolojik görüntü işleme, elimizde var olan resimlerin fiziksel yapısı üzerinde çalışarak resimden arka plan nesnelerini ayırt etmemize ve onları kendi aralarında gruplamamıza imkân sağlar. Bu metot gri seviye görüntüleri üzerinde de uygulama yapılmasına olanak sağlar ama genellikle ikili (binary) siyah-beyaz görüntüler üzerinde kullanılır. Morfolojik görüntü işleme tekniklerinde kullanılan filtreler genelde iki temel işlemden ortaya çıkmıştır. Bunlar aşındırma işlemi ve genişletme işlemi işlemleridir. Aşındırma işleminde ikili bir görüntüde arka plandan ayrılan nesnelerin boyutunu

belirlenen bir yapısal elemente (Structer element) bağlı olarak küçültme işlemi yaparken, genişletme işleminde aynı yapısal element nesnenin kapladığı alanını artırmaktadır. Bu işlemlerden türeyen “erosion” işleminde ise birbirine ince bir gürültü ile bağlanmış birden fazla veya daha fazla nesneyi belli bir sınır değerine göre bir birinden ayırmak için kullanılırken, genişletme işlemi ise aynı nesnenin bir gürültü ile belli bir sınır çizilerek ince bir şekilde bölünerek farklı iki şekil gibi görünmesine engel olmaktadır. Özellikle plaka okuma ya da tanıma, parmak izinin okunması, harflerin tanınması gibi uygulamalarda kullanılarak oluşacak olan ya da var olan gürültülerin etkisini düşürür (Howse, 2013).

İşlemleri matematiksel olarak ifade etmek ve anlamak için iki tane terime dikkat etmek gerekir. Bunlardan birincisi, yapısal elementin elemanlarından oluşan veri kümesi ile üzerinde resmin piksellerinin bire bir olarak örtüşmesi durumudur. Bu durum “Fit” olarak adlandırılır. Mantıksal olarak da “ve” işlemine benzetilmektedir. İkincisi ise, yapısal elemanın veri seti içerisinde bulunan piksellerden en az bir tanesinin üzerinde çalıştığı piksel ile eşleşmesine denir. Bu durum “Hit” olarak adlandırılır. Mantıksal olarak “veya” işlemine benzetilmektedir. Örneğin aşındırma ve silme işlemi için yapısal elementteki pikselleri, karşılıklı piksellerle “Fit” olması gerekirken, genişleme (Dilation) işlemi ise “Hit” olması kafi (Elbehery ve ark., 2007).

1	1	1
1	1	1
1	1	1

0	1	0
1	1	1
0	1	0

0	1	0
0	1	1
0	0	0

Şekil 2.2 Yapısal eleman örnekleri

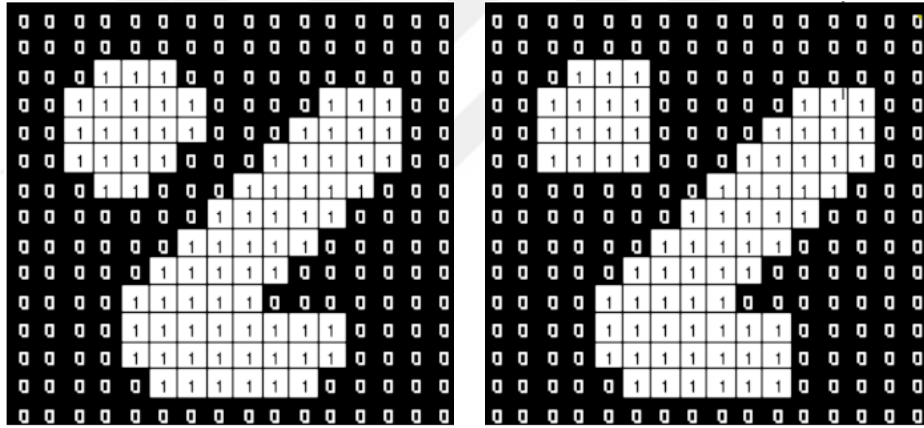
Şekil 2.2’de morfolojik görüntü işlemede yapısal elemanın çalışması verilmiştir. Morfolojik görüntü işleme yönteminde, görüntüler üzerinde 4 farklı işlem uygulamaktadır. Birinci işlem olarak görüntü üzerinde açma işlemidir. Bu açma işlemi genel olarak “imopen()” diye adlandırılan fonksiyon ile görüntü üzerine uygulanmaktadır. Görüntülerin tümü “f” isimli değişkene atanmaktadır. Daha sonraki adımda (50×50) piksel boyutlarında yapısal element oluşturulmalıdır. Oluşturulan bu

yapısal elementin görüntünün pikselleri üzerinde gezdirilerek var olan hatalar ortadan kaldırılmaktadır.

“imopen()” fonksiyonun genel olarak çalışmasında, görüntü üzerinde (50X50) piksellik “t” yapısal elementinin gezdirilmesiyle “açma işlemi” yapılmaktadır. Hareket ettirilirken (50×50) den küçük olan görüntü üzerindeki kare matriste var olan görüntü hataları açma işlemine tabii tutularak temizlenir.

$$A \oplus B = \{Z \mid (\hat{B})_Z \cap A \neq \emptyset\} \quad (2.1)$$

“Eşitlik 2.1’ de” ‘imopen()’ fonksiyonuna ait matematiksel model verilmiştir. Bu fonksiyonda A; görüntüyü, B; yapısal elemanı temsil etmektedir.



(a)Orijinal görüntü

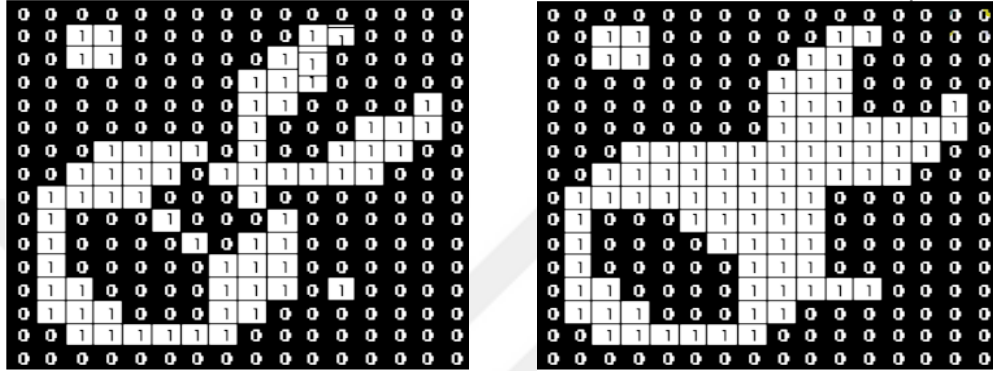
(b) Açma işlemi

Şekil 2.3. Açma işlemi (Acar ve Bayram, 2009)

Şekil 2.3’de “imopen()” komutunun çalışması (a) orijinal görüntü ve (b) açma işleminden sonraki görüntü verilmiştir.

Morfolojik görüntü işleme tekniklerinde ikinci işlem olarak “görüntü kapama” işlemi yapılmaktadır. Bu işlem “imclose()” fonksiyonu tarafından yapılır. Kapama işleminde elimizde bulunan görüntüler üzerinde gene (50X50) boyutunda “t” yapısal elementi gezdirilerek eşleşme yapılmaktadır. 50×50’lik türünde bir kare matris olan “t” yapısal elementi ile görüntünün beyaz kısmı (arka plandan ayrılan ve şekli belirleyen) merkez noktada olacak biçimde elimizde bulunan resmin pikselleri genişletilmektedir. Bu işlem

sayesinde resmin iç kısımdaki bulunan tüm boşluklar temizlenip piksel sayıları artırıldığında şekil yayılmış olacaktır. Fonksiyonun çalıştırılmasından sonra görüntü büyütülmüştür. Bu işlemin tam tersi işlem yapıldığında ise şekil küçülecektir bu işlemede “küçültme işlemi” denilmektedir. Yani görüntü ilk haline tekrar dönüşmektedir.

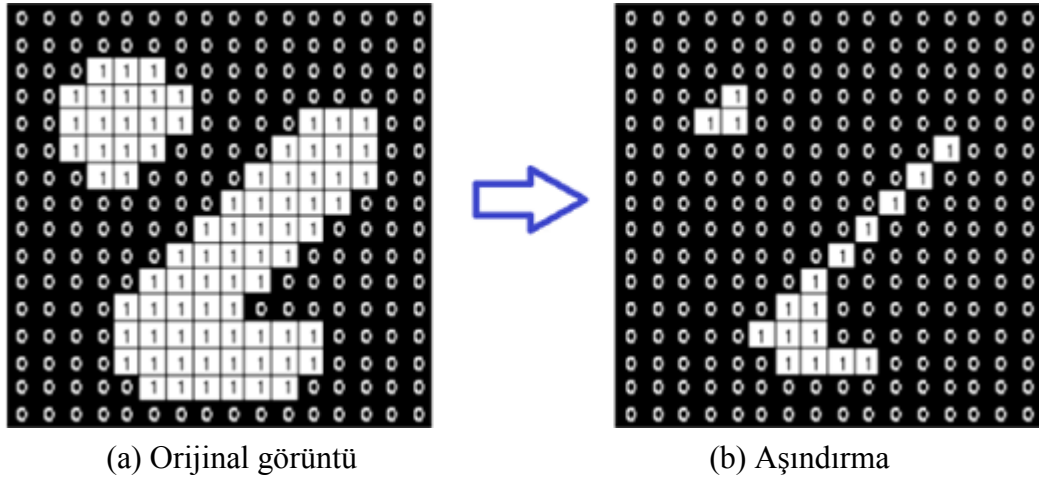


(a) Orijinal görüntü

(b) Kapama işlemi

Şekil 2.4. Kapama işlemi (Acar ve Bayram, 2009)

Şekil 2.4’de “imclose()” komutunun çalışması (a) orijinal görüntü ve (b) kapama işleminden sonraki görüntü verilmiştir. Morfolojik görüntü işleme tekniklerin üçüncü işlem türü olarak da görüntü üzerinde “aşındırma işlemi” uygulanmaktadır. “imrode()” fonksiyonu tarafından yapılan bu işlemde yine yapısal “t” elementi (50X50) boyutundadır. Bu yapısal element resmin pikselleri üzerinde gezinerek aşındırma işlemi denilen bir işlem yapmaktadır. Piksellerin bir kısmını silen bu işlem resmin daralmasına sebep olmaktadır. Eğer resim genişletilme işlemine tabi tutulmuş ise ilk haline geri döner. Yani bu işlem kısaca resmin daralmasına sebep olmaktadır (Gonzalez ve Woods, 2000).



Şekil 2.5 Aşındırma işlemi (Boztoprak ve ark., 2007)

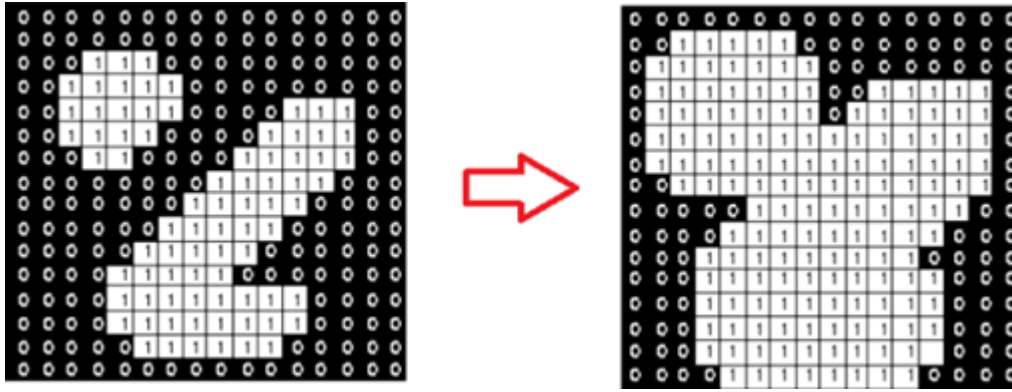
Şekil 2.5’de “imrode()” komutunun çalışması (a) orijinal görüntü ve (b) açma işleminden sonraki görüntüsü verilmiştir. Morfolojik görüntü işleme tekniklerinin dördüncü ve son adımında görüntüde üzerinde genleşme işlemi yapılmaktadır. Bu işlemde “imdilate()” fonksiyonu kullanılmaktadır. Bu fonksiyon ile görüntü üzerindeki piksellerin üzerinde yapısal element gezdirilerek, görüntü üzerindeki piksel sayıları artırılmaktadır. Yapılan bu artırma işlemi sayesinde genleşme işlemi gerçekleştirilmektedir.

$$A \circ B = (A \ominus B) \oplus B \quad (2.2)$$

“Eşitlik 2.2’de” yapılan işlemde görülmekte olan işlem sırası ile tepe noktaları düzleştirir, birebirinden kopuk olan ve uzun ince pikselleri birleştirir ve küçük delikleri ya da diğer bir deyişle noktasal bölgeleri yok eder.

$$A \cdot B = (A \oplus B) \ominus B \quad (2.3)$$

“Eşitlik 2.3’de” görülmekte olan matematiksel ifade genleşme işlemini gerçekleştirir.



(a) Orjinal görüntü

(b) Genişleme işlemi

Şekil 2.6 Genleşme işlemi (Boztoprak ve ark., 2007)

Şekil 2.6'da imdilate() komutunun çalışması (a) orjinal görüntü ve (b) genişleme işleminden sonraki görüntüsü verilmiştir.

2.3 Sınıflayıcılar

Sınıflandırma, fikirleri ve nesneleri önceden belirlenmiş kategorilere veya alt özelliklere göre tanıma, anlama ve gruplama sürecidir. Sınıflayıcılar önceden kategorize edilmiş eğitim veri kümelerini makine öğrenimi programları kullanarak, gelecekteki veri kümelerini gruplara ayırmak için farklı algoritmalar kullanırlar. Makine öğrenimindeki sınıflandırma algoritmaları, sonraki verilerin önceden belirlenmiş kategorilerden birine düşme olasılığını tahmin etmek için eğitim verilerini girdi olarak kullanıp hangi sınıfa dahil olup olmadığını belirlerler (Charu ve Aggarwal, 2015).

En çok kullanılan sınıflayıcı metodları şunlardır;

- SVM sınıflandırıcısı
- Veri Akış Madenciliği (Data Stream Mining)
- NB (Naive Bayes Sınıflandırıcısı- Naive Bayes Classifier)
- Karar Ağacı Öğrenmesi (Decision Tree Learning):
- Güçlendirilmiş ağaç sınıflayıcı (Boosted Trees Sınıflayıcı)
- LDA(Doğrusal Diskriminant Analizi-Linear Discriminant Analysis) ve QDA

- KNN (Yakın Komşular Sınıflandırması)
- LRA (Logistic Regression Analizi)
- YSA (Yapay Sinir Ağları)

2.3.1 SVM sınıflandırıcısı

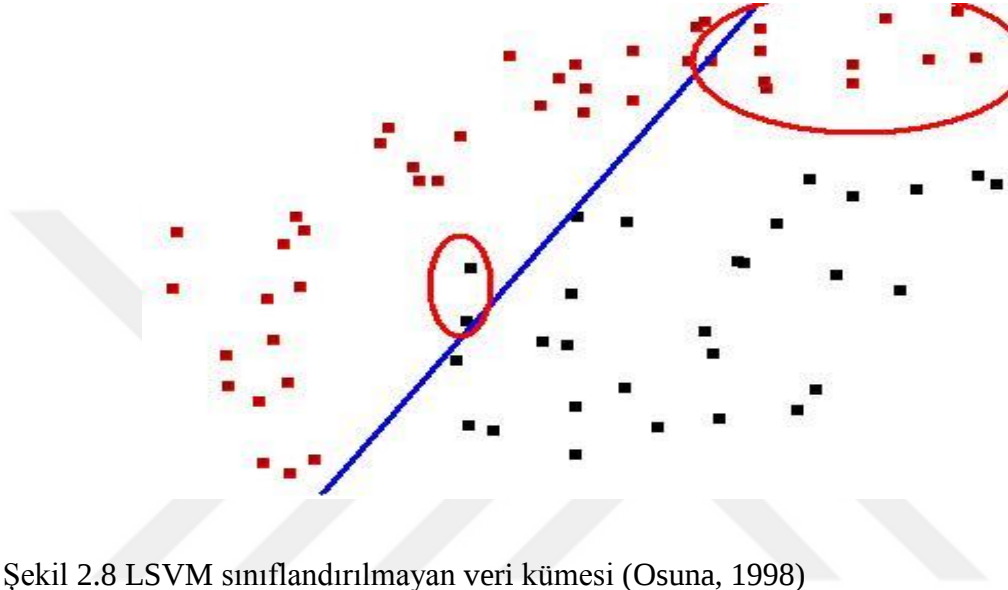
SVM temelde iki sınıflı veri setlerini birbirinden ayırmak için kullanılan en iyi sınıflayıcıdır. Bir hiper düzlem oluşturularak veri kümesi içerisindeki ikili verileri birbirinden ayırır. SVM sınıflayıcı iki türdür;

- LSVM (Doğrusal SVM-Linear Support Vector Machine,)
- NLSVM (Doğrusal Olmayan SVM-Non-Linear Support Vector Machine)

LSVM, başlangıçta ikili sınıflandırma için oldukça genelleştirilebilir sınıflandırıcı oluşturmak üzere tasarlanmıştır. LSVM, orijinal özellik uzayını kullanıcı tanımlı bir çekirdek işlevine göre daha yüksek boyutlu bir alana dönüştürür ve ardından iki sınıf arasındaki ayrımı (marjı) en üst düzeye çıkarmak için destek vektörleri bulur. LSVM ilk önce her iki sınıfı da ayıran bir hiper düzleme yaklaşır. Buna göre, LSVM, hiper düzleme en yakın olan destek vektörleri olarak adlandırılan her iki sınıftan da örnekler seçer. Hiper düzlem ve destek vektörleri arasındaki toplam ayırım, kenar boşluğu olarak adlandırılır. LSVM daha sonra hiper düzlemi yinelemeli olarak optimize eder ve marjı maksimize etmek için vektörleri destekler, böylece en genelleştirilebilir karar sınırlarını bulur. Veri kümesi doğrusal olmayan sınırla ayrılabilirdiğinde, özellik uzayını uygun şekilde dönüştürmek için LSVM' de belirli çekirdekler uygulanır. Kolayca ayrılamayan bir veri kümesi için, karar sınırları etrafındaki sınıflandırma hatalarına daha az ağırlık vererek aşırı uyumu önlemek için yumuşak kenar boşluğu kullanılır (Kahraman ve ark., 2004).

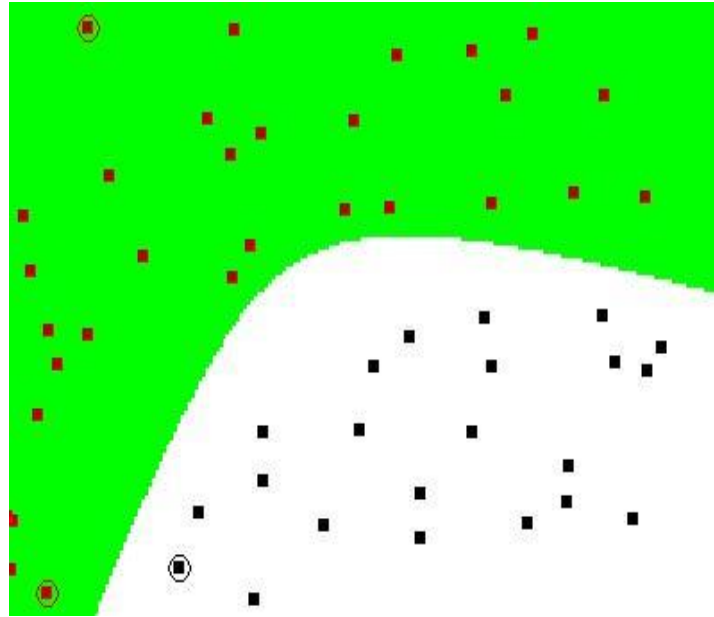
LSVM çalışırken oluşturulan hiper düzlemde üzerine veri kümesi yerleştirilir. Veri kümesi içerisindeki veriler, belirlenen eşik değerine göre birbirinden ayrılarak ikili sınıflar oluşturmaktadır. Belirlenen eşik değerinin hassasiyetinin yüksek olması

(tölerans) değerine göre de sınıflandırma işlemi yapar. Bu tölerans değerinin değişimine göre sınır çizgisinin şekli değişecektir ve en doğru sınıflandırmayı yapana kadar, iterasyon devam edecektir. LSVM'deki kadar başarı oranının yüksek olması beklenmemelidir. Sınıflandırma yapılacak veriler oldukça karmaşık ve birbirine benzer yapıda olması sınıflandırma işlemini zorlaştırırken başarı oranını da düşürmektedir.



Şekil 2.8 LSVM sınıflandırılmayan veri kümesi (Osuna, 1998)

Şekil 2.8' de doğrusal düzlem üzerinde yerleştirilmiş, fakat sınıflandırılmayan veriler verilmiştir.

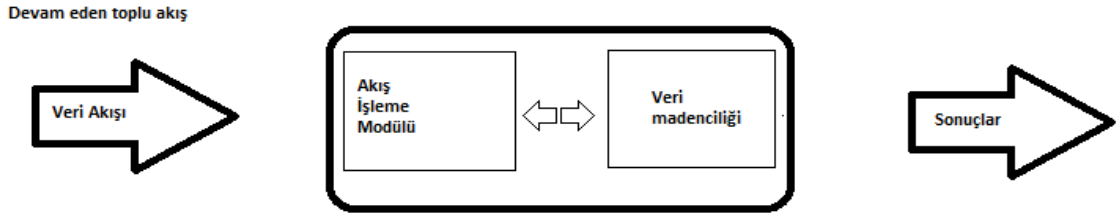


Şekil 2.9 NLSVM tarafından sınıflandırılmış veri kümesi (Hasan, 2017)

Şekil 2.9’da doğrusal düzlem üzerinde yerleştirilmiş, sınıflandırılan veri kümesi verilmiştir.

2.3.2 Veri akış madenciliği

Veri Akışı Madenciliği, sürekli, hızlı veri kayıtlarından bilgi yapılarının çıkarılması işlemidir. Veri madenciliği bir yerde depolanmış büyük bilgi toplulukları ve bu veri setinin sürekli olarak büyümeye devam etmesi sonucu oluşan sınıflandırma problemlerine çözüm olan yaklaşımdır. Veri akış madenciliğini diğer sınıflandırma metodlarından ayıran en önemli özelliği sabit bir sınıfının olmamasıdır. Ayrıca veri kümesi içerisine sürekli yeni veri eklenmesi, sınıflandırmanın tekrardan al baştan yapılması bu tür sınıflandırmanın dinamik ve güncel kalmasına sebep olmaktadır (Miller ve ark., 2011).



Şekil 2.10 Veri akış madenciliği

Şekil 2.10’da veri madenciliği ile sınıflandırma tekniği verilmiştir. Veri akış madenciliği, çok büyük kaydedilmiş verilere, sürekli olarak yeni veri ekleyerek oluşan yeni veri setini sınıflandırır.

2.3.3 NB sınıflandırıcısı

Standart dışı veri kümelerinde de çalışan bir öğrenme algoritması olarak da adlandırılır. Olasılık temelli bir sınıflandırma yapmaktadır. Var olan olayda her bir durumun olma

olasılığını hesaplar, sonuçta ortaya çıkan en büyük olasılığa göre sınıflandırma yapmaktadır (Choi ve ark., 2016).

$$P(A|B) = (P(B|A).P(A))/P(B) \quad (2.4)$$

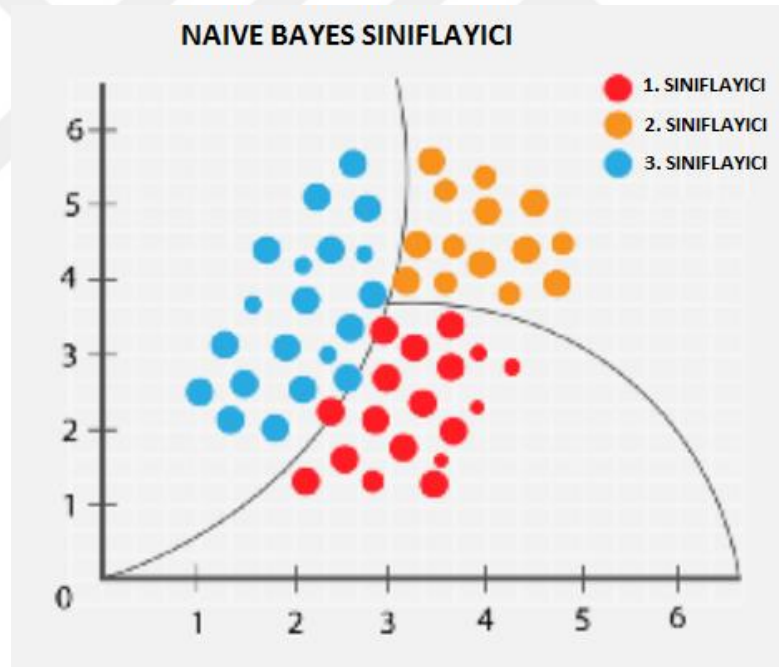
Eşitlik 2.4’de;

$P(A|B)$ = B olayı gerçekleştiğinde A olayının gerçekleşme olasılığı

$P(A)$ = A olayının gerçekleşme olasılığı

$P(B|A)$ = A olayı gerçekleştiğinde B olayının gerçekleşme olasılığı

$P(B)$ = B olayının gerçekleşme olasılığı



Şekil 2.11 NB sınıflayıcı (Yang, 2019)

Olasılık temelli çalışan NB sınıflandırıcısı en büyük olasılığa göre sınıflandırma yapmıştır. Şekil 2.11’de bu sınıflandırmayı göstermektedir. NB Text sınıflandırıcı, olayların olasılıklarına göre sınıflandıran basit bir sınıflandırıcıdır. Genellikle metin sınıflandırmasına uygulanır. Basit bir algoritma olmasına rağmen, birçok metin sınıflandırma probleminde iyi performans gösterir (Ting ve ark., 2011).

Diğer sınıflandırma metotlarına göre daha az eğitim süresi ve daha az eğitim verisi içerir. Yani, daha az işlemci üzerindeki bellek kullanımı. Herhangi bir makine öğrenimi modelinde olduğu gibi, her kategori (sınıf) için mevcut bir örnek setine (eğitim seti) ihtiyaç duymaktadır.

$$P (A | B) = (P(A \cap B)) / (P (B)) \quad (2.5)$$

Eşitlik 2.5’de

$P (A | B)$ = B olayı gerçekleştiğinde A olayının gerçekleşme olasılığı

$P (A)$ = A olayının gerçekleşme olasılığı

$P(A \cap B)$ = A olayı gerçekleştiğinde B olayının gerçekleşme olasılığı

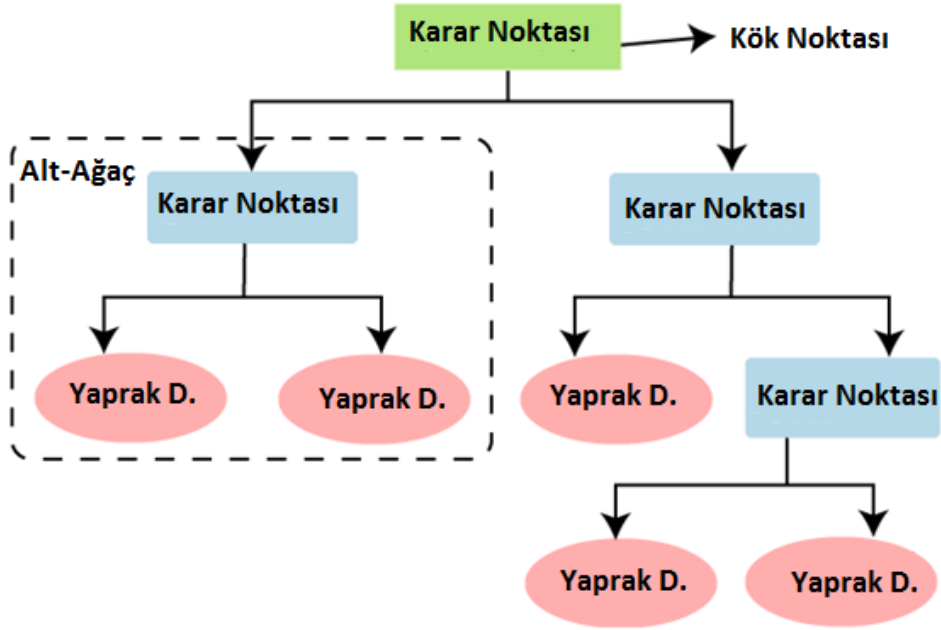
$P (B)$ = B olayının gerçekleşme olasılığı

“NB Text” sınıflandırıcı da olasılık temelleri sınıflandırıcıdır. En büyük olasılığa göre sınıflandırma yapmaktadır.

2.3.4 DT sınıflayıcı

DT, hem sınıflandırma hem de regresyon problemleri için kullanılabilen, ancak daha çok sınıflandırma problemlerini çözmek için tercih edilen bir denetimli öğrenme tekniğidir. Dâhili düğümlerin bir veri kümesinin özelliklerini, dalların karar kurallarını ve her yaprak düğümün sonucu temsil ettiği ağaç yapılı bir sınıflandırıcıdır. Bir DT sınıflayıcısında, karar düğümü ve yaprak düğümü olmak üzere iki düğüm vardır. Karar düğümleri herhangi bir karar vermek için kullanılır ve birden çok şubeye sahipken, yaprak düğümleri bu kararların çıktısıdır ve başka dallar içermez. Kararlar veya test, verilen veri setinin özelliklerine göre gerçekleştirilir. Verilen koşullara göre bir soruna / karara tüm olası çözümleri elde etmek için grafiksel bir sunumdur. DT olarak adlandırılmasının sebebi bir ağaca benzer şekilde, daha fazla dalda genişleyen ve ağaç benzeri bir yapı oluşturan kök düğüm ile başlamasıdır. Bir ağaç oluşturmak için, sınıflandırma ve regresyon ağacı algoritmasının kısaltması olan “CART” algoritmasını

kullanılmaktadır. Bir DT basitçe bir soru sorar ve cevaba (Evet / Hayır) bağlı olarak ağacı alt ağaçlara ayırır (Aitkenhead, 2008).



Şekil 2.12 DT'nin genel yapısı (Catanzaro, 2013)

Şekil 2.12'de DT'nin genel yapısı verilmiştir. DT terminolojileri genel yapısı aşağıda belirtildiği gibidir.

1. Kök düğüm, DT'nin başladığı yerdir. Ayrıca iki veya daha fazla homojen kümeye bölünen tüm veri kümesini temsil eder.
2. Yaprak düğümü, son çıkış düğümleridir ve ağaç, bir yaprak düğümü aldıktan sonra daha fazla ayrılamaz.
3. Bölme, karar düğümünü / kök düğümünü verilen koşullara göre alt düğümlere ayırma işlemidir.
4. Dal / alt Ağaç, ağacın bölünmesiyle oluşan ağaçtır.
5. Budama, ağaçtan istenmeyen dalların uzaklaştırılması işlemidir.
6. Ana / alt düğüm, ağacın kök düğümüne ana düğüm, diğer düğümlere alt düğümler denir.

DT algoritması aşağıda verilen adımlardan oluşmaktadır.

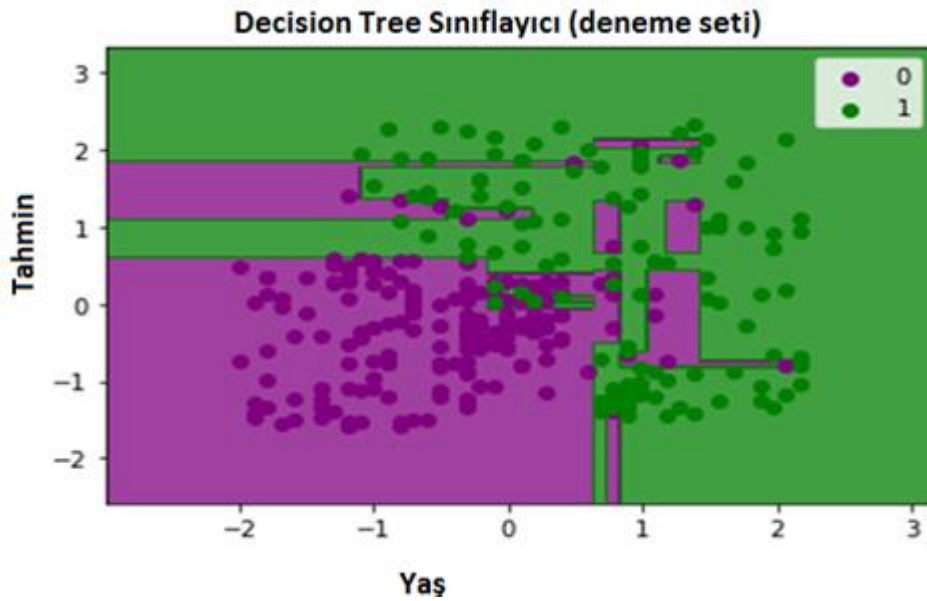
Adım-1: Ağaca kök düğüm ile başlanır. “S” tüm veri kümesini içerir.

Adım-2: Öznitelik seçimi ölçüsünü (ASM) kullanarak veri kümesindeki en iyi özniteliği bulunur.

Adım-3: S'yi en iyi öznitelikler için olası değerleri içeren alt kümelere bölünür.

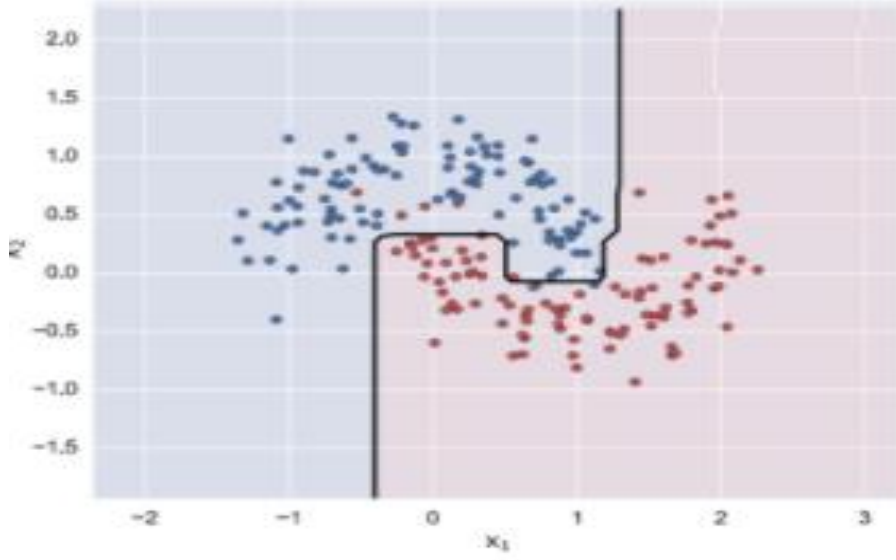
Adım-4: En iyi özniteliği içeren DT düğümünü oluşturulur.

Adım 5: Adım 3' te oluşturulan veri kümesinin alt kümelerini kullanarak yinelemeli olarak yeni karar ağaçları oluşturulur. Düğümler daha fazla sınıflandırma olmayana ve son düğüm yaprak düğüm olarak adlandırılana kadar bu işlem devam etmektedir.



Şekil 2.13 DT deneme seti (Catanzaro, 2013)

Şekil 2.13’de DT sınıflandırılma işlemine tabii tutulan test verisine ait görüntü verilmiştir.

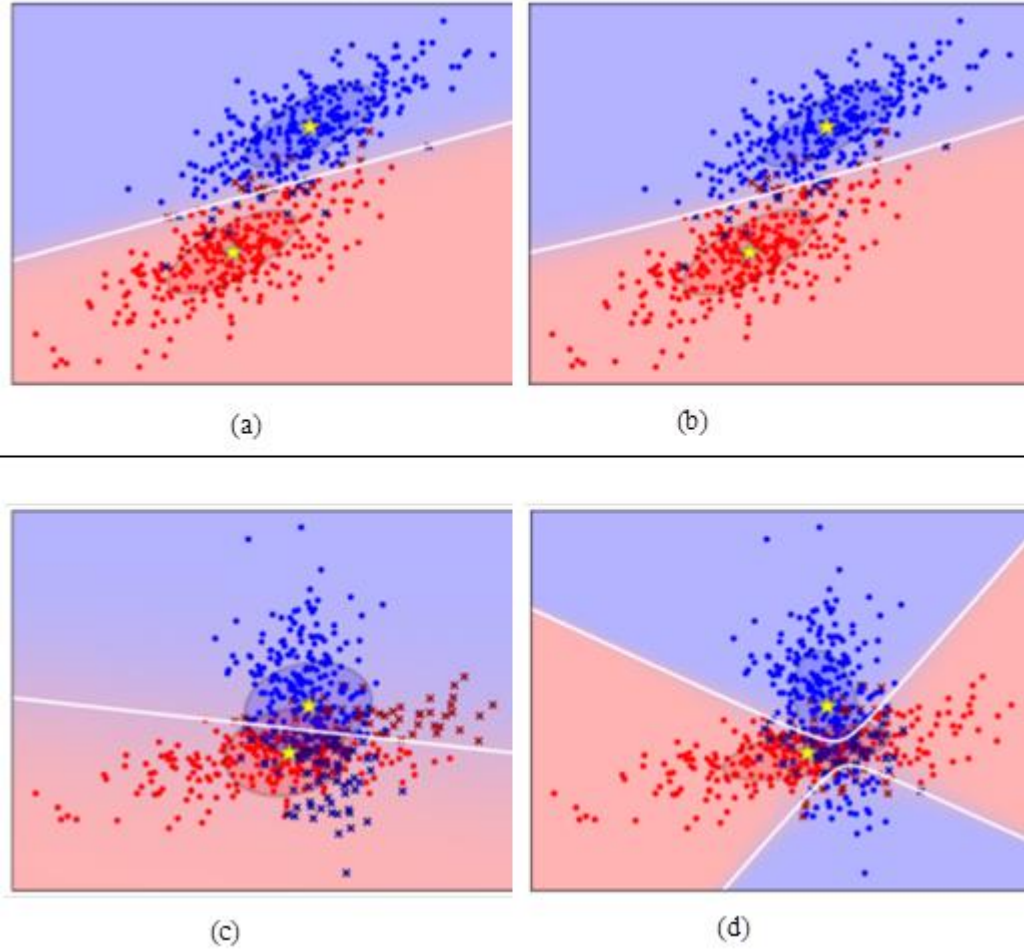


Şekil 2.14 DT test verisi (Catanzaro, 2013)

Şekil 2.14’de DT sınıflandırmasına işlemine tabii tutulan test verisine ait sonuç görüntüsü verilmiştir.

2.3.5 Güçlendirilmiş ağaç sınıflayıcı

Güçlendirilmiş ağaç modelleri, hem regresyon hem de sınıflandırma için en popüler ve etkili makine öğrenimi yaklaşımları arasındadır. Birkaç ağaç modellerinden gelen tahminleri birleştiren bir topluluk tekniğidir.



Şekil 2.16 LDA ve QDA analizi (Hastie, 2009)

Şekil 2.16’da doğrusal ayırım analizi ve kuadratik ayırım analizi için karar sınırlarını gösterir. (a) doğrusal bir denklem de LDA sınıflayıcı, (b) doğrusal bir denklem de QDA sınıflayıcı, (c) İkinci dereceden bir denklemde LDA sınıflayıcı, (d) İkinci dereceden bir denklemde QDA sınıflayıcı göstermektedir. Kuadratik sınıflayıcının ayırım analizinin ikinci dereceden sınırları öğrenebileceğini ve bu nedenle daha esnek olduğunu gösterir.

Hem LDA hem de QDA, verilerin sınıf koşullu dağılımını modelleyen basit olasılıklı modellerden türetilir. $P(X|y=k)$ her sınıf için k. tahminler, daha sonra her eğitim örneği için Bayes kuralı kullanılarak elde edilebilir.

$x \in \mathbb{R}^d$:

$$P(y = k|x) = \frac{P(x|y=k)P(y=k)}{P(x)} = \frac{P(x|y=k)P(y=k)}{\sum_i P(x|y=i)P(y=i)} \quad (2.4)$$

“Eşitlik 2.4’de” “k” sınıf seçimi yapılır, bu da olasılığı en üst düzeye çıkarır. Daha net olarak, doğrusal ve ikinci dereceden ayırt edici analiz için, $P(x | y)$ yoğunluklu çok değişkenli bir Gauss dağılımı olarak modellenir.

$$P(x|y = k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2} (x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k)\right) \quad (2.5)$$

“d” özelliklerin sayısıdır.

QDA formülü;

“Eşitlik 2.5’ e göre”;

$$\begin{aligned} \log P(y = k|x) &= \log P(x|y = k) + \log P(y = k) + Cst = \\ &= -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k) + \log P(y = k) + Cst, \end{aligned} \quad (2.6)$$

“Eşitlik 2.6’da” $P(x)$ Gauss'un diğer sabit terimlerine ek olarak, sabit terim “Cst” paydaya karşılık gelir. Öngörülen sınıf, bu log-posterior'u maksimize eden sınıftır.

LSVM, her sınıf için Gauss ile aynı kovaryans matrisini paylaştığı varsayılan özel bir QDA durumudur.

$$\begin{aligned} \log P(y = k|x) &= -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k) + \\ &= \log P(y = k) + Cst, \end{aligned} \quad (2.7)$$

“Eşitlik 2.7’de” $(x - \mu_k)^t \sum -1(x - \mu_k)$ verilen örnek arasındaki Mahalanobis mesafesine karşılık gelir. “ x ” ve ortalama “ μ_k ” Mahalanobis mesafesine ne kadar yakın olduğunu söyler, “ μ_k ” aynı zamanda her özelliğin varyansını da hesaba katar. Böylece SVM’yi atama olarak yorumlanır. Ortalama Mahalanobis mesafesi açısından en yakın olan sınıfa, aynı zamanda sınıf önceki olasılıklarını da hesaba katar (Siqueira ve ark., 2017).

2.3.7 K-NN sınıflayıcı

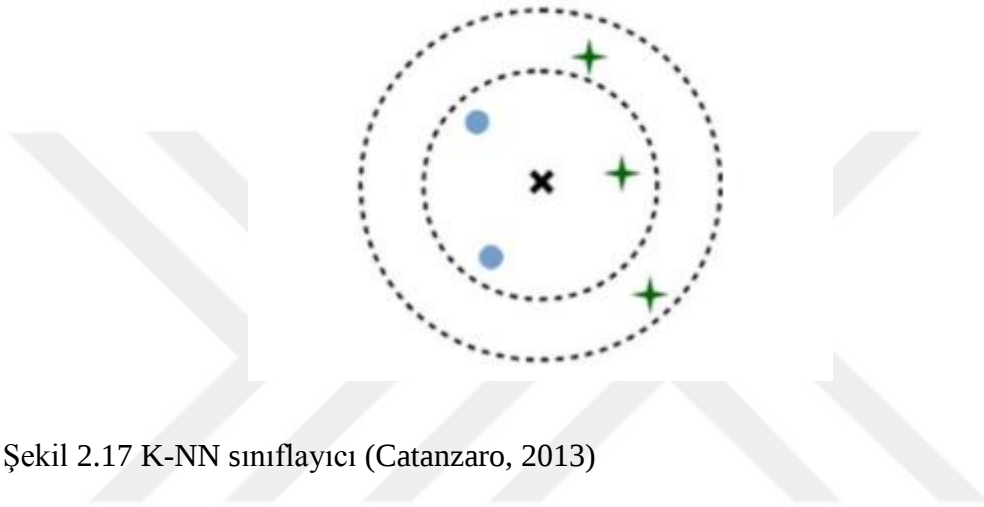
K-NN, mevcut tüm vakaları saklayan ve yeni vakaları bir benzerlik ölçüsüne (örneğin, mesafe fonksiyonları) göre sınıflandıran basit bir algoritmadır.

K-NN algoritması en basit ve en çok kullanılan sınıflandırma algoritmasından biridir. K-NN parametrik olmayan, tembel bir öğrenme algoritmasıdır. Tembel kavramını anlamaya çalışırsak öğrenmenin aksine, bir eğitim aşaması yoktur. Eğitim verilerini öğrenmez, bunun yerine eğitim veri kümesini ezberler. Bir tahmin yapmak istenildiğinde, tüm veri setinde en yakın komşuları arar (Deng, 2016).

Algoritmanın çalışmasında bir “K” değeri belirlenir. Bu “K” değerinin anlamı bakılacak eleman sayısıdır. Bir değer geldiğinde en yakın “K” kadar eleman alınarak gelen değer arasındaki uzaklık hesaplanır. Uzaklık hesaplama işleminde genelde “Öklid” fonksiyonu kullanılır. Öklid fonksiyonuna alternatif olarak “Manhattan”, “Minkowski” veya “Hamming” fonksiyonları da kullanılabilir. Uzaklık hesaplandıktan sonra sıralanır ve gelen değer uygun olan sınıfa atanır.

K-NN sınıflayıcı toplamada iki değer üzerinde sınıflandırma yapmaktadır. Bunlar uzaklık ve komşuluk sayısı olarak adlandırılır. Mesafe, “Minkowski” ile sınıflandırmada tahmin edilecek noktanın sınıflandırılmamış başka noktalara uzaklığı hesaplamasıdır. Komşu sayısı, sınıflandırma yapılacak olan veriye en yakın noktaların hesaplanması gerekmektedir. Bu değere komşu sayısı denir. Buradaki değişkeni sonucu belirleyen faktördür. “K” değerinin “1” değeri alması durumunda aşırı uyma olasılığı çok yüksek olacaktır. Aşırı uyma durumunda sistem öğrenme yükünü

tamamlayacağından yeni veri öğrenmeyecektir. Çok büyük olursa da çok genel sonuçlar verecektir. Bundan dolayı standart bir “K” değeri bulmak zorundadır. “K” değerinin önemini aşağıda verilen örnekte açıklanmaktadır. “K”=3 değeri seçilirse, sınıflandırma sonrasında “?” işareti ile belirtilen noktayı, kırmızı üçgenin sınıfında görecektir. “K”=5 seçilmesi durumunda sınıflandırmanın sonucunda “?” işareti ile belirlenen nokta mavi kare sınıfına dâhil olarak gösterecektir.

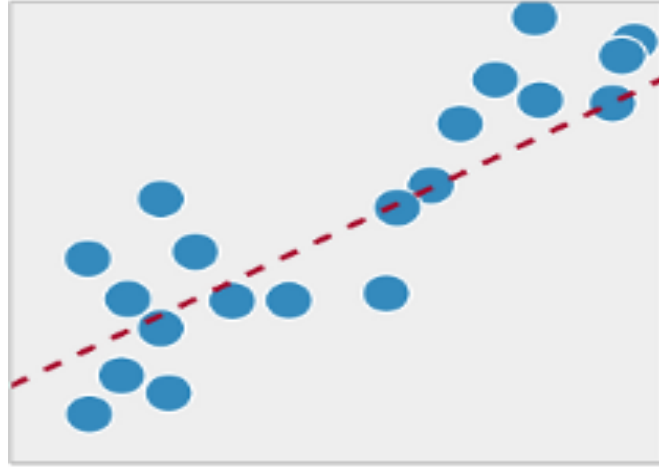


Şekil 2.17 K-NN sınıflayıcı (Catanzaro, 2013)

Şekil 2.17’de K-NN sınıflayıcının çalışması verilmiştir.

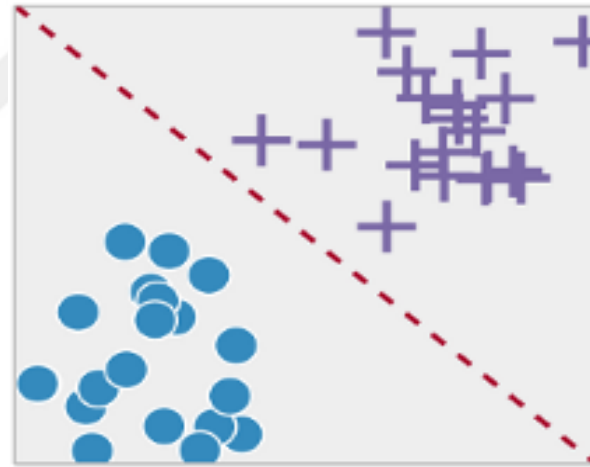
2.3.8 LRA sınıflayıcı

Makine öğrenmesinde sayısal verilerin tahmin edilmesi için tahmin algoritmaları kullanılır. Sayısal olmayan yani kategorik verilerin tahmini için ise sınıflandırma kullanılır.



Şekil 2.18 LRA veri (Şentürk, 2011)

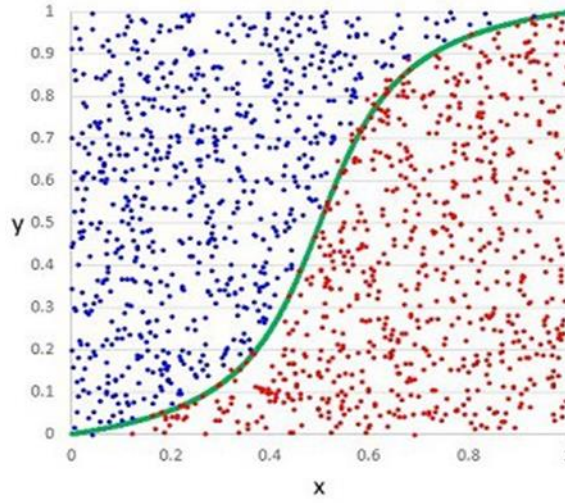
Şekil 2.18’de LRA ait sınıflandırma ait sınıflandırılmış veriler verilmiştir.



Şekil 2.19 LRA sınıflandırılmış veri (Şentürk, 2011)

Şekil 2.18’de LRA’ya ait sınıflandırma veriler, Şekil 2.19’da ise LRA’ya göre sınıflandırma işlemi sonrası işlem verilmiştir.

Sınıflandırma işlemi yapmaya yarayan bir regresyon yöntemidir. Kategorik veya sayısal verilerin sınıflandırılmasında kullanılır. Bağımlı değişkenin yani sonucun sadece iki farklı değer alabilmesi durumunda çalışır. Doğrusal sınıflandırma problemlerinde (Evet / Hayır, Erkek / Kadın, Şişman / Zayıf vs.) yaygın bir biçimde kullanılır.

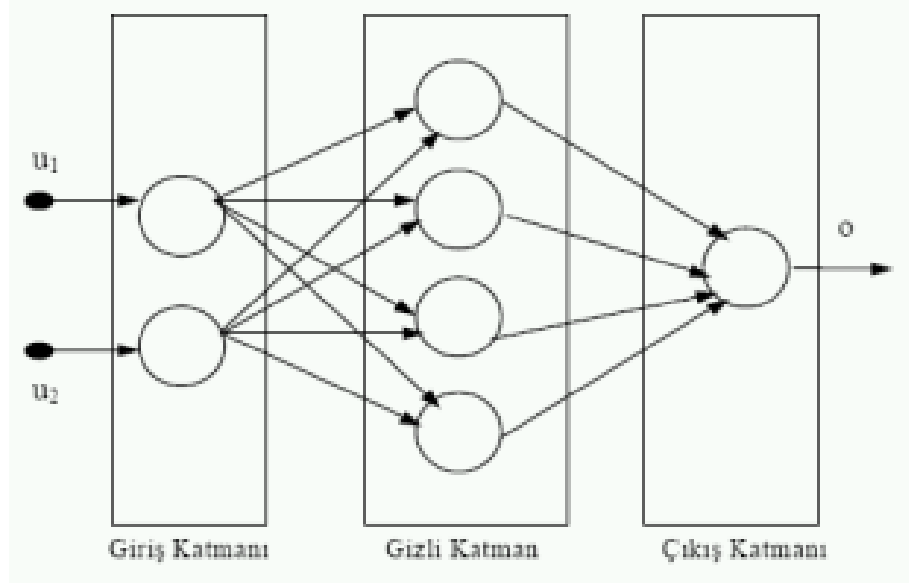


Şekil 2.20 LRA sınıflandırma sonucu (Şentürk, 2011)

Şekil 2.20’de LRA sonuç görüntüsü verilmiştir. LRA sınıflandırılmasında yeşil bölge sınır değerini (eşik), kırmızı bölge yanlış sınıflandırılmış veriyi, mavi bölge ise doğru sınıflandırılmış veriyi göstermektedir.

2.3.9 YSA sınıflayıcı

YSA başlamadan önce beynin temel yapı taşı olan sinir hücrelerinin çalışma prensibinden basitçe bahsetmekte fayda vardır. Biyolojik bir sinir hücresi temelde 4 kısımdan oluşmaktadır. Bunlar; çekirdek, gövde, akson ve dentritlerdir. Sinir hücrelerinin ucunda bulunan dentritlerin görevi, diğer nöronlardan gelen sinyali çekirdeğe iletmektir. Çekirdek ise dentritlerin ilettikleri sinyalleri toplar ve aksonlara iletir. Toplanan sinyaller, akson tarafından işlenerek sinapslara gönderilir. Sinaps, akson ve dentritlerin birleştiği noktalardır (Foody, 1995).

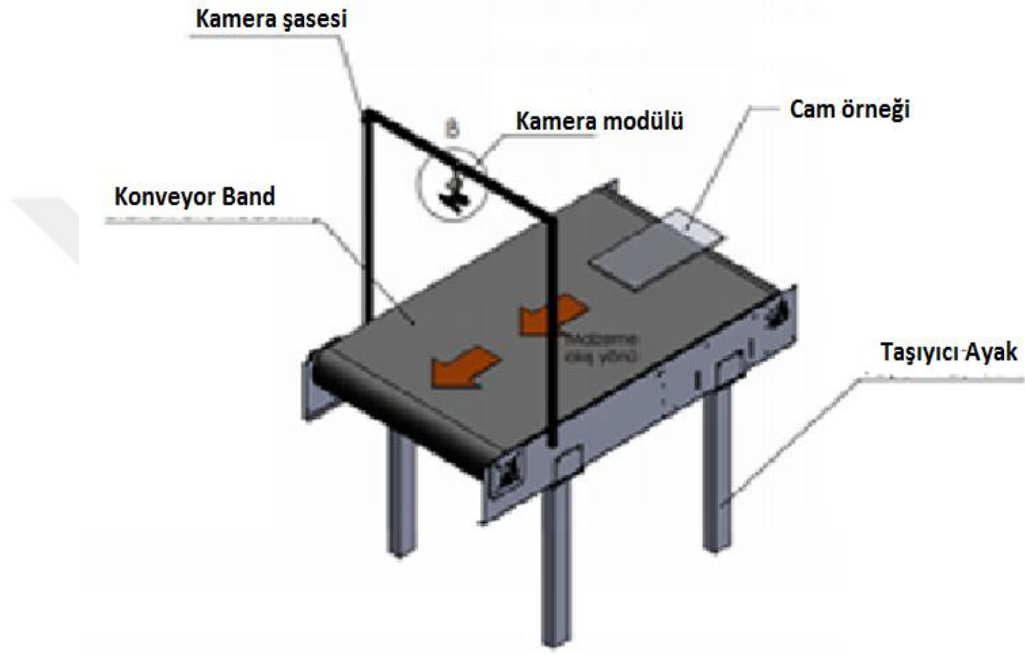


Şekil 2.21 YSA sınıflayıcı (Fırat ve ark, 2004)

Şekil 2.21’de YSA sınıflayıcısının çalışma şekli verilmiştir.

3. MATERYAL VE YÖNTEM

3.1 Materyal



Şekil 3.1 Kullanılan prototipin konstrüksiyon yapısı

Şekil 3.1’de cam ürünlerinde gerçek zamanlı hata tespiti yapan yeni bir prototipe ait tasarım verilmektedir. Konveyör sistemleri, malzemeyi minimum çabayla taşıyan mekanik cihazlar veya tertibatlardır. Pek çok farklı tipte konveyör sistemi varken, bunlar genellikle materyallerin bir yerden diğerine hareket ettiği makaraları, tekerlekleri veya bir bandı destekleyen bir çerçeveden oluşur. Bir motorla, yerçekimi ile veya manuel olarak çalıştırılabilirler. Bu malzeme taşıma sistemleri, taşınması gereken farklı ürünlere veya malzemelere uyacak şekilde birçok farklı çeşitte gelir. Maksimum yük kapasitesi (birim uzunluğa düşen yük ve konveyör hız ile ilişkisi), konveyör bant sistemi hızı / anma hızı (bant hızının zaman ile değişimi) ve sürücü sistemleri temel bir konveyör bant için önemli özelliklerdir. Konveyör bandın yapacağı iş/görev ve taşıyacağı yük bileşenleri birlikte değerlendirilmelidir. Konveyör bant sisteminde tahrik

elemanı; kontrol ünitesi ile iletişim özelliğine sahip ve bandın taşıyacağı fiziksel özellik ve hız ile uyumlu, kullanılan sistem ile eşzamanlı olması tercih edilmelidir.



Şekil 3. 2 Gerçek zamanlı hata tespiti yapan yeni bir prototip

Şekil 3.2’de cam ürünlerinde gerçek zamanlı hata tespiti yapan yeni bir prototipe ait gerçek görünüş verilmiştir. Konveyör tahrik sistemi olarak 12 V’luk DA motoru tercih edilmiştir. Motorun boşta 2 A’lık akım çekmektedir. Bu değer yük durumuna göre 10 A kadar çıkmaktadır. Aynı zamanda zorlanma akımı ile yük torku ise 45 kg-cm’dir. Mil Çapı 10 mm ve şaft uzunluğu 60 cm olarak tercih edilmiştir. Motor uzun süreli çalışmalarda oluşabilecek ısınma veya aşırı akım çekimine karşı korumalıdır. Konveyör tahrik sistemine uygun bağlantı noktalarına ve yeterli uzunlukta şafta sahiptir.



Şekil 3. 3 Konveyör bant motoru

Şekil 3.3’de konveyör bant motoru verilmiştir. Çalışmada hatalı veya sağlam cam ürün görüntülerinin, sistem ile eşzamanlı olarak alınması ve alınan görüntülerin sistem ile uyumlu olarak işlenip sonuç üretmesi gerekmektedir. Motorun hareketi ile konveyör bant ve kameranın saniyedeki görüntü karesi alma hızı eşzamanlı olmalıdır. Bu nedenle motorun dakikadaki devir sayısı 50-70 RPM olarak belirlenmiştir.

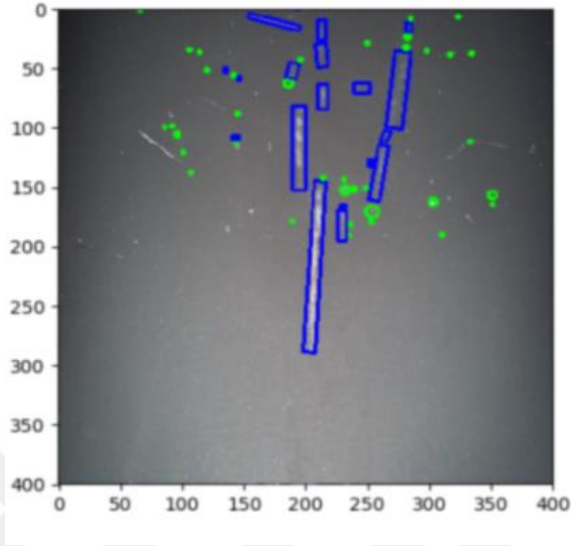
Saniyede 30 görüntü karesi alabilme özelliğine sahip sabit bir kamera ile, gerçek zamanlı görüntü işlemek için ortamdaki görüntü değişimini zamana bağlı olarak alınması gerekmektedir. Bu iş için en temel yol, arka arkaya yakalanan resimlerin mutlak farkının, resmi oluşturan toplam piksel sayısına oranının belirli bir sınır değeri geçip geçmemesine bakılır. Matematiksel olarak;

$$OÇF(t) = \frac{1}{w*h} \sum_{x=1}^w \sum_{y=1}^h |I_t(x, y) - I_{t-1}(x, y)| > E_1 \quad (3.1)$$

“Eşitlik 3.1’de verilen ifade de;” OÇF, ortalama çerçeve farkını; I_t , t zamanının da yakalanan çerçeveyi; I_{t-1} , bir önceki örneklemede yakalanan çerçeveyi; w, yakalanan imgenin yatay piksel boyutunu; h, düşey piksel boyutunu; (x, y), ilgili pikselin konumunu ve E_1 ise eşik değerini ifade etmektedir (Coşar ve ark, 2004).

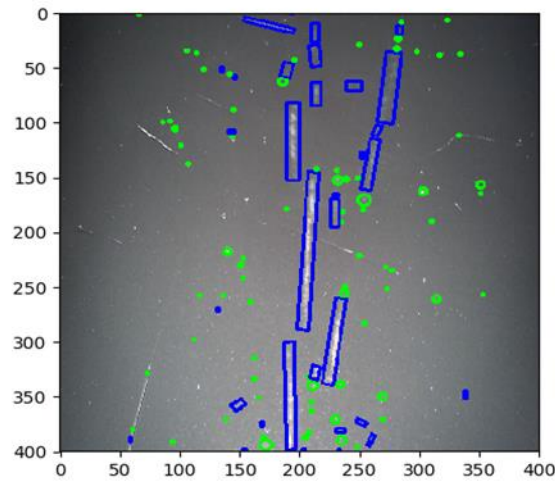
Eşitlik 3.1’de verilen fonksiyondaki eşik değeri konveyör bandın dönüş hızını temsil etmektedir. Bu değer çalışmamızda 5.75 cm/sn olarak belirlenmiştir. E_1 değerinin

belirlenen değerin altında ve üstünde kalması alınan örneklemelerde sistem kararlılığını olumsuz etkilediği, hem teorik hesaplamada hem de uygulama örneklerinde teyit edilmiştir.



Şekil 3. 4 Sistemin eşik değerinin üstündeki hızda alınan ekran görüntüsü

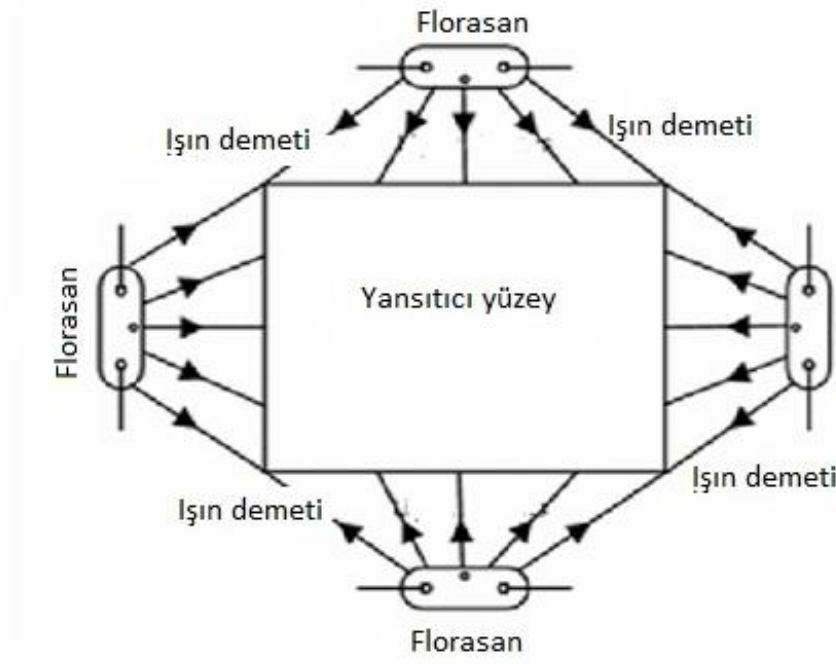
Şekil 3.4’de eşik değerinin üstünde hareket eden bir sistemden alınan görüntü örneği verilmiştir.



Şekil 3.5 Sistemin eşik değerindeki hızda alınan ekran görüntüsü

Şekil 3.5’de eşik değeri hesaplandıktan sonra hareket eden konveyör band üzerindeki ürünlerden alınan ekran görüntüsü verilmiştir. Eşik değerinin altında ve üstünde alınan cam görüntülerinde, sistemin yakalayamadığı bazı cam kusurları görülmüştür.

Görüntü işleme tabanlı cam ürünlerinde gerçek zamanlı hata tespiti yapmak için sistem; çevresel koşullardan etkilenmeyecek şekilde tasarlanmıştır. Tasarımda konveyör band renginin koyu ve metal aksamın ise mat yüzey olmasına dikkat edilmiştir. Görüntü işleme sistemlerinde nesne veya cisimler aktif (yapay) veya pasif (doğal) ışık kaynağı kullanırlar. Aktif ışık kaynaklarında ortam aydınlatması için sistem harici bir ışık kaynağı kullanır. Pasif ışık kaynaklarında sistem harici bir aydınlatma kullanmaz ve doğal ışık kaynaklarını kullanır. Çalışma ortamında pasif ışık kaynağının zamana bağlı olarak değişmesinin sistem kararlılığına etki etmektedir. Bu nedenle sistem sabit bir aydınlanmaya ihtiyaç duymaktadır. Ortam aydınlatması için tercih edilebilecek led, akkor veya floresan aydınlatma kaynakları tercih edilebilir. Ortamdaki istenmeyen parlaklık veya gürültü oluşmasını en aza indirmek için ortamın sabit bir şekilde ışık kaynağından ışık alması gerekmektedir. Cam ürünlerinde gerçek zamanlı hata tespiti yapan prototip cam yüzeylerden yansıyan ışınları, kamera tarafından gürültü olarak algılar. Bu da alınan görüntünün bozulmasına sebep olur (Öztürk ve Akdemir, 2017). Aynı zamanda dış ortamdaki ışık değişikliğinin sistem sonuçlarını etkilememesi, kontrol ve hata tespit ünitesinin dışarıdaki ışık değişiminden bağımsız hale getirilmesi gerekmektedir. Bu sebepten, ortam aydınlatması için floresan ışık kaynağı ve ortamının aydınlığının 100 lümende sabit tutulmuştur.



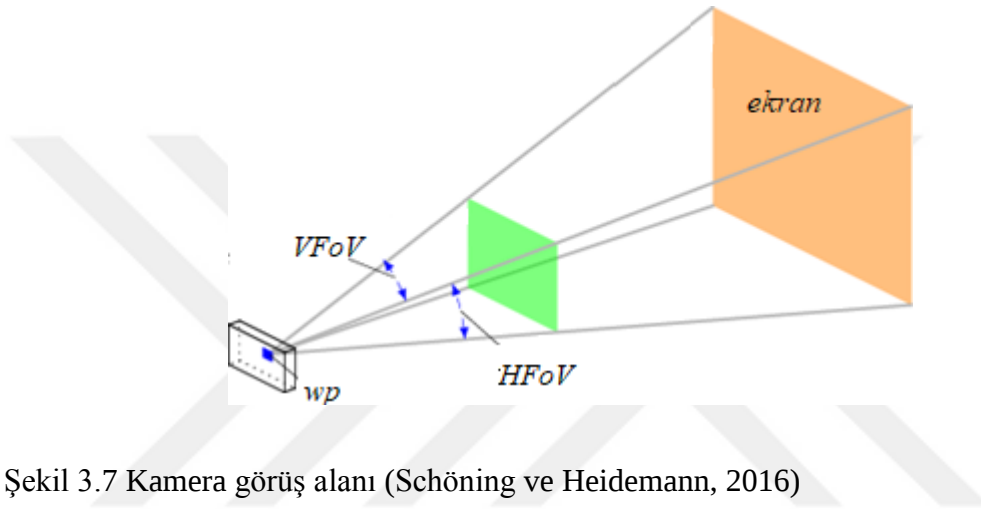
Şekil 3.6 Aydınlatma düzeneği

Şekil 3.6’da cam ürünlerinde gerçek zamanlı hata tespiti yapan prototipimizin görüntü alma merkezini aydınlatmasına ait tasarım verilmiştir. Tasarımda aydınlatma dört ayrı köşeden uygun yerlere yerleştirilen floresan ışık kaynakları ile homojen olarak sağlanmıştır.

Görüntü alma; kontrol sisteminde harici veya dâhili bağlı olan kameralar yardımı ile gerçekleştirilebilir. Raspberry Pi gibi tek kartlı bordlar için dâhili kamera modülleri ve özellikleri bakımından piyasada bulunan birkaç model vardır.

Yatay görüş alanı (HFOV-Horizantal Field Of View), ölçüm nesnesi düzlemindeki ölçüm alanının yatay boyutlarını açıklar. Dedektörün yatay boyutlarına, ölçüm nesnesine olan mesafeye ve merceğin yatay görsel açısına göre tanımlanabilir.

Dikey görüş alanı (VFOV-Vertical Field Of View), nesne düzlemindeki ölçüm alanının dikey boyutlarını açıklar. Dikey görsel açı ve dedektörün dikey boyutlarına bağlı olarak HFOV ile aynı şekilde hesaplanabilir.



Şekil 3.7 Kamera görüş alanı (Schöning ve Heidemann, 2016)

Kameralara ait görüş alanı kameranın görüntü alacağı nesnenin kamera merkezine olan uzaklığına bağlıdır. Bu durum Şekil 3.7’de verilmiştir. Görüş Alanı (FoV-Field Of View); kamera ile hedef nesne arasındaki mesafeye ve kameranın fiziksel özelliğine bağlıdır. Temel bir FoV değeri “Eşitlik 3.2 ve 3.3’de” verilmiştir.

$$FoV = \tan\left(\frac{\alpha}{2}\right) * Z * 2 \quad (3.2)$$

$$PPM = \frac{wp}{FoV} \quad (3.3)$$

“Eşitlik 3.2 ve 3.3’de”;

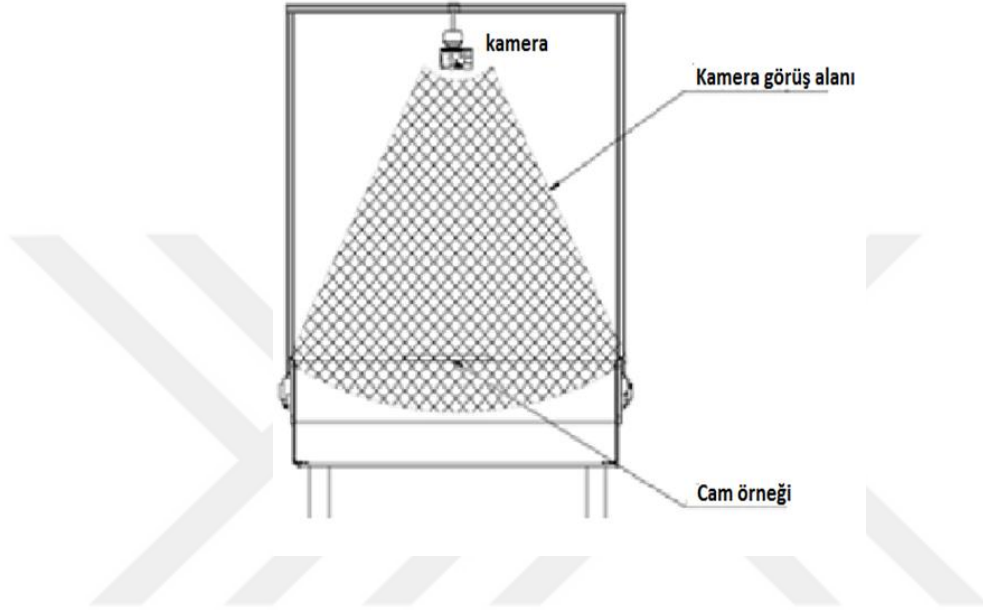
α : Kamera görüş açısı(derece)

Z: Cam ürünlerin kameraya olan uzaklığı (metre)

FoV: Mesafeye bağılı kameranın görüş mesafesi (metre)

PPM : 1 m²'ye düşen piksel sayısı (piksel)

wp: Kamera çözünürlüğü-piksel olarak alınmıştır.



Şekil 3.8 Kameraya ait FoV alanı

Şekil 3.8’de kullanılan kameraya ait FoV alanı verilmiştir. Sabit odak uzaklıklı bir kamerada hedef nesne ne kadar uzaksa kameranın FoV alanı o kadar geniştir. FoV genişliğinin artması PPM (piksel per meter) değerini düşürür. Bu da uzak mesafedeki nesne üzerindeki piksel genişliğini artırır ve görüntü kalitesizleşir.



Şekil 3.9 Tercih edilen kamera

Şekil 3.9’da çalışmada kullanılan kameranın görüntüleri verilmiştir. Cam ürünlerinde gerçek zamanlı hata tespiti yapan prototip de “LOGITECH C270” model kamera kullanılmıştır. Kameraya ait fiziksel özellikler aşağıda verilmiştir.

1 GHz 512 MB veya daha yüksek RAM 200 MB sabit sürücü alanı internet bağlantısı USB 1.1 bağlantı noktası, 2.4 GHz Intel® Core™2 Duo 2 GB RAM 200 MB sabit sürücü alanı USB 2.0 bağlantı noktası 1 Mbps veya daha yüksek karşıya yükleme hızı 1280 x 720 ekran çözünürlüğü, HD görüntülü arama (1280 X 720 piksel) HD görüntü yakalama: En çok 1280 X 720 piksel Logitech Fluid Crystal™ teknolojisi fotoğraflar: 5 megapiksele kadar (yazılım destekli), yüksek hızlı veri iletimi için USB 2.0 sertifikası, yatay kaydırma, eğme ve yakınlaştırma denetimleri görüntü ve fotoğraf yakalama, yüz izleme hareket algılama gibi uygulamalara uygundur.

Kamera, ölçülecek olan cam ürünü tam olarak görebilecek bir konuma yerleştirilmelidir (Adelson ve Wang, 1992). Kameranın konumunu belirlemek için aşağıdaki denklem kullanılmıştır.

$$\frac{1}{d} = \frac{h}{v} \left(\frac{1}{f} - \frac{1}{D} \right) + \frac{1}{D} \quad (3.4)$$

“Eşitlik 3.4’de;”

d: Kamera yüksekliğini,

D: “h” nesnesinin görüntüsündeki değişimi, boşluk mesafesindeki yer değişimini

f: Objektifin odak uzaklığını ifade etmektedir.

“Eşitlik 3.4’e göre” yapılan hesaplamalar da, görüntü alma merkezinin cam örneklerine olan mesafesi hesaplanmıştır. Bu mesafedeki bir kameranın ölçüm hassasiyeti ve buna bağlı numune ölçüleri, gerekli FoV değerine ve “Eşitlik 3.4” göz önüne alınarak hesaplanmıştır. 90x10 cm ve 60x10 cm uzunluklarındaki cam örnekleri için en uygun kamera merkezi uzaklığı 52 cm olarak belirlenmiştir.

45 cm, 50 cm ve 52 cm yüksekliğe yerleştirilen kameralardan aynı numuneye ait görüntüler alınmıştır. Bu görüntüler incelendiğinde 45 cm veya 50 cm uzaklığa yerleştirilen kameralara ait görüntü ekranlarında cam numunesinin bir kısmının görüş alanının dışında kaldığı görülmektedir. 52 cm uzaklıktaki kameradan alınan numune incelendiğinde numunenin tamamının görüntü ekranına girdiği görülmektedir. Şekil 3.10, 3.11 ve 3.12’de 3 farklı mesafeden alınan numune görüntülerinin kamera görüntü ekranları verilmiştir.



Şekil 3.10 h=45 cm’den alınan görüntü

Şekil 3.10’da kamera yüksekliği (h) 45 cm olarak belirlenmiş ve örneklerden alınan kamera görüntüsü verilmiştir.



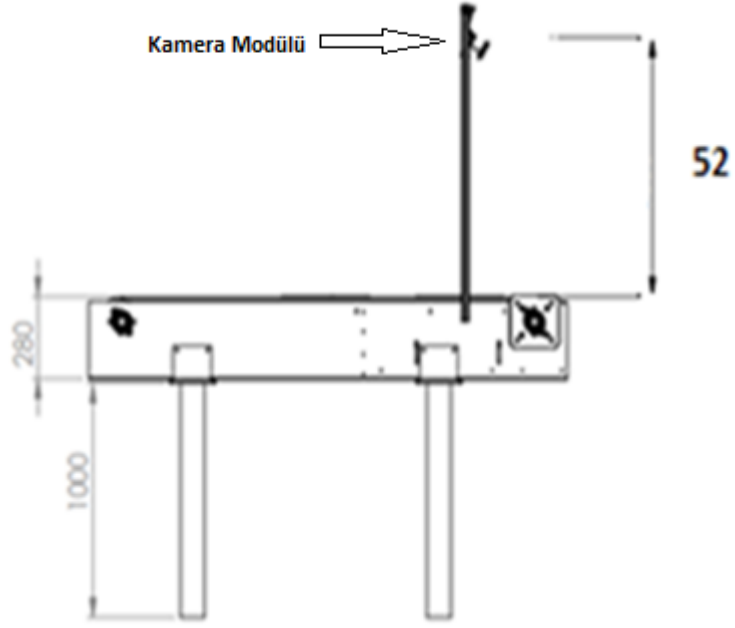
Şekil 3.11 h=50 cm’den alınan görüntü

Şekil 3.11’de kamera yüksekliği (h) 50 cm olarak belirlenmiş ve örneklerden alınan kamera görüntüsü verilmiştir.

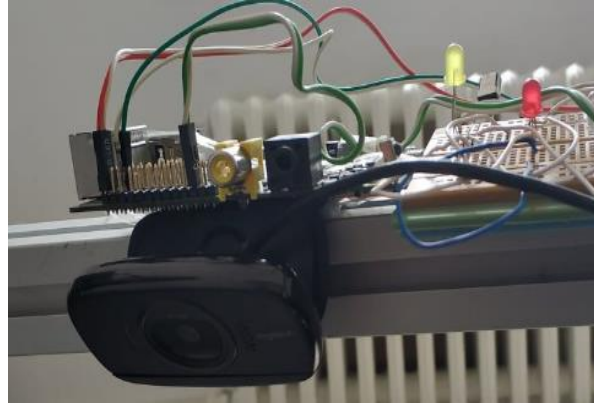


Şekil 3.12 h=52 cm’den alınan görüntü

Şekil 3.12’de kamera yüksekliği (h) 52cm olarak belirlenmiş ve örneklerden alınan kamera görüntüsü verilmiştir. Kamera ile ilgili bir diğer önemli konu ise kameranın görüntü yakalama hızıdır. Gerçek zamanlı sistemlerde görüntü yakalama hızı, konveyör bandın hızından daha hızlı olmalıdır. Bu çalışmada, görüntüdeki bulanıklığı gidermek için kamera saniyede 50 kare görüntü alacak şekilde seçilmiştir. Şekil 3.13’de kamera merkezinin ve konveyör bandına ait teknik resim verilmiştir.



Şekil 3.13 FoV ve PPM değerine göre kamera mesafesi



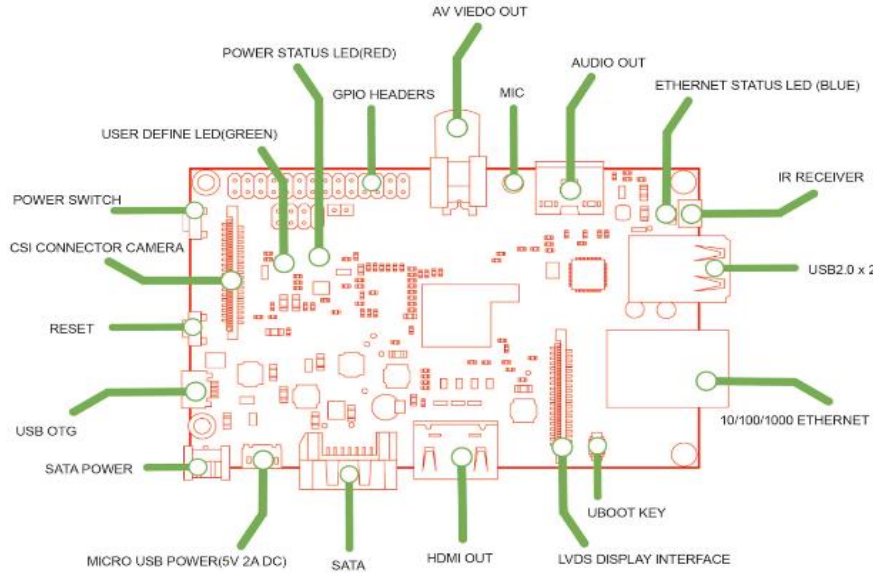
Şekil 3.14 Görüntü Alma Merkezi

Şekil 3.14’de ise görüntü alma merkezimize ait resim verilmiştir. Kameralar sadece bir görüntü karesinin zaman içerisinde değişimini yakalayan cihazlar değil aynı zamanda almış oldukları bu görüntü verilerinin işlenmesi için bir başka sisteme aktaran bilgi dönüştürücüleridir. Bu işlenmiş bilgiler artık bir başka sistem için veri haline dönüşmüştür. Veriler bir mikrodenetleyici için giriş bilgisidir.

Mikroişlemcilere ilave edilen bellek, giriş/çıkış birim donanımlar ile mikrodenetleyiciyi hatta biraz daha gelişmiş donanım olan tek kartlı bilgisayarları meydana getirir.

Tek kartlı bilgisayar (SBC) teknolojisi günümüzde hızla gelişmektedir. Sadece bir kredi kartı büyüklüğündeki bu mikrobilgisayar, insanların günlük aktivitelerinde destekleyici bir araç olarak kullanılabilir. Birçok kişi / geliştirici bu mikro bilgisayarı geliştirmiş oldukları sistemlerin bir parçası olarak kullanmaktadır.

Raspberry Pi, Banana Pi, Beagle Board gibi SBC'ler deney yapma, basit programlanabilme, bir medya oynatıcı, robotik ve ev otomasyonu gibi çeşitli amaçlar için kullanılabilen küçük bilgi işlem cihazlarıdır. SBC' ler ayrıca robotik ve nesnelerin interneti (IoT) gibi alanlarda veya çok çeşitli endüstriyel uygulamalar için gün geçtikçe daha da fazla kullanılmaya başlanılmışlardır. Tek kartlı bilgisayarlar, çok çeşitli mikroişlemciler kullanılarak oluşturulmuştur. Bilgisayar gibi basit tasarımlar için statik RAM ve düşük maliyetli 8 veya 16 bit işlemciler kullanılır.



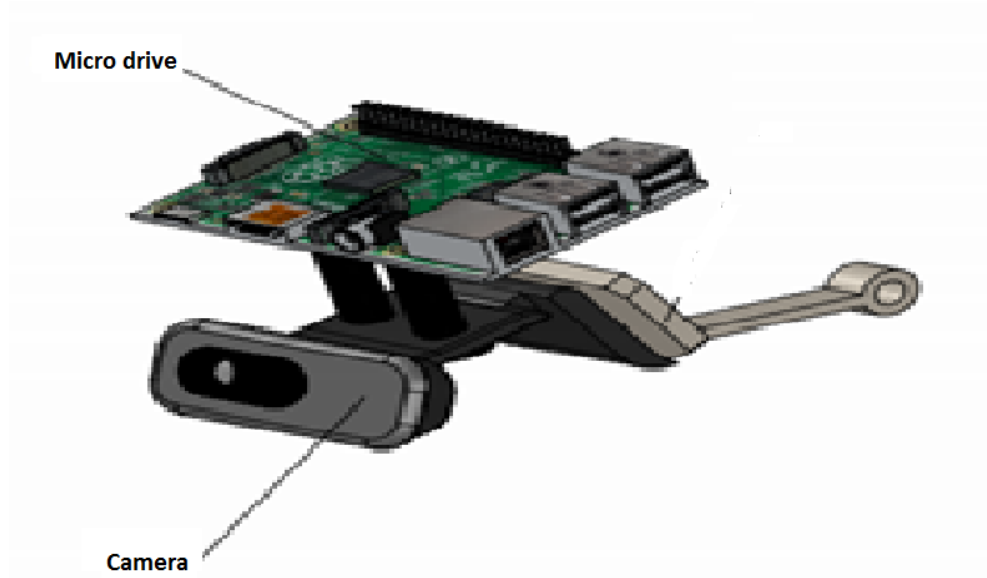
Şekil 3.15 Prototipte kullanılan mikro sürücü

Şekil 3.15’de Banana Pi’ ye ait ön görünüş ve modülleri verilmiştir. Banana Pi, oldukça kompakt bir tasarıma sahip bir mini bilgisayardır. Bir Banana Pi’ de, tam teşekküllü bir bilgisayarın tüm önemli bileşenleri, bir kartvizitten biraz daha büyük olan tek bir devre

kartında bulunur. Merkezi işlemci ve belleğe ek olarak, bu tek kartlı bilgisayarın genişletilebilmesini ve ek donanımların (örneğin monitörler, sabit sürücüler, hafıza kartları ve WLAN antenleri) bağlanabilmesini sağlayan bir USB bağlantı noktası gibi temel ara yüzlere sahiptir.

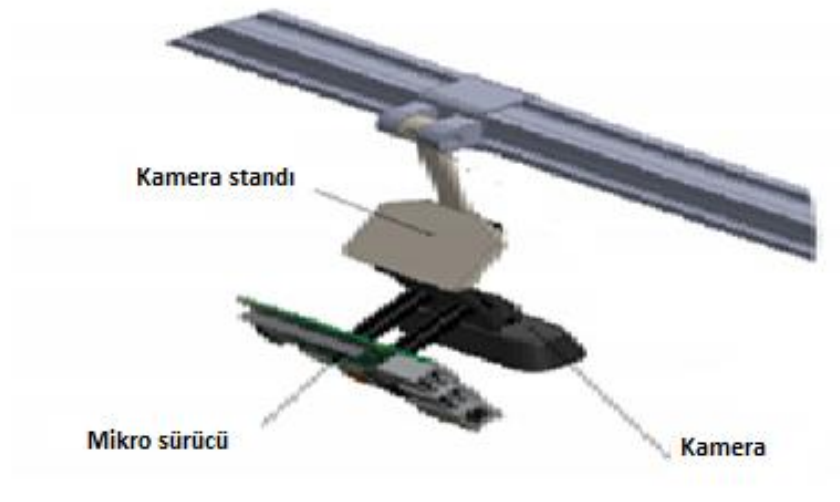
Banana Pi, Raspberry Pi' den çok daha iyi olan çift çekirdekli bir Android 4.2 ürünüdür. Android, Debian Linux, Ubuntu Linux, Opensuse Linux ve Raspberry Pi ve Cubieboard üzerinde çalışan imajlarını çalıştırır. Bir gigabit ethernet bağlantı noktasına ve bir “SATA” soketine sahiptir. Ayrıca 1080P yüksek çözünürlüklü video çıkışını desteklediği için oyunları kolayca çalıştırabilir. GPIO, Raspberry Pi ile uyumludur ve Raspberry Pi görüntülerini doğrudan çalıştırabilir.

A20 ARM Cortex-A7 çift çekirdekli CPU, ARM Mali400MP2 OpenGL ES 2.0 / 1.1 ile uyumlu GPU, 1 GB DDR3 (GPU ile paylaşılır) SDRAM bellek, USB 2.0 bağlantı noktaları, GPIO (2X13) pin: GPIO, UART, I2C veri yolu, iki çip seçimli SPI veri yolu, CAN veri yolu, ADC, PWM, + 3.3v, + 5v, giriş/çıkış kontrol birimleri sistemde kullanılan “Banana Pi” nin sahip olduğu özelliklerdir.



Şekil 3.16 Görüntü alma merkezi ve gömülü sistem

Şekil 3.16’da kamera düzeneği ve gömülü sisteme ait görüntü verilmiştir.



Şekil 3.17 Gömülü sistemin platformu

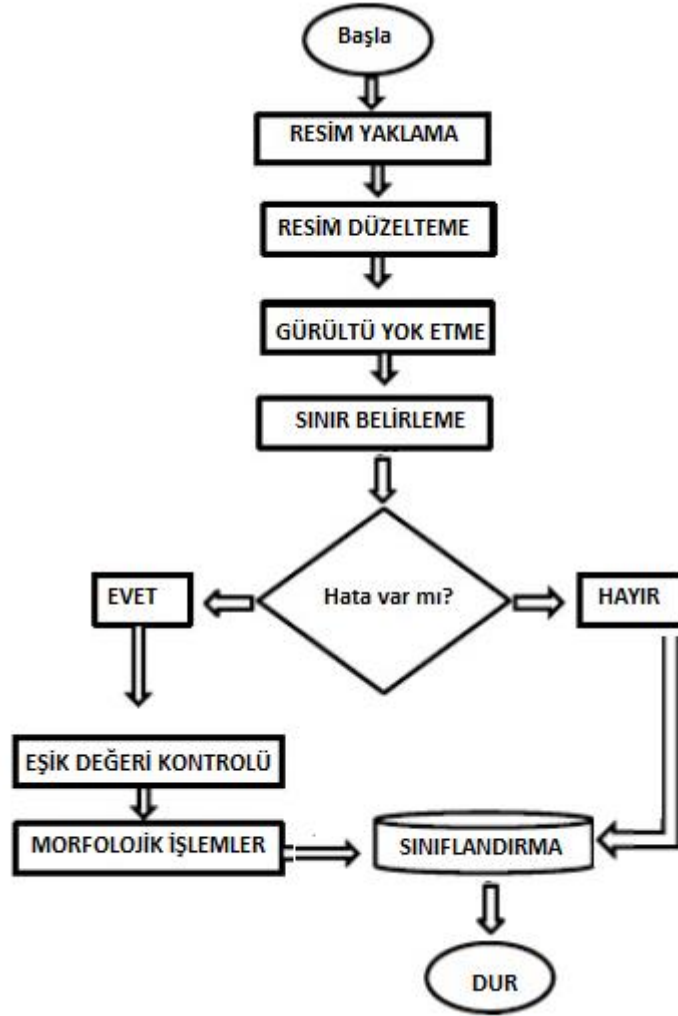
Şekil 3.17’de gömülü sisteme ait platformun görüntüsü verilmiştir.

3.2 Yöntem

Yürütülen tez çalışması aşağıda verilen hata tespit süreçlerini kapsamakta olup tez çalışmasının yürütülmesinde kullanılan yöntem, donanım ve yazılımlar ile ilgili detaylara bu bölüm altında yer verilmiştir. Bu kapsamsa öncelikle hata ayıklamada görüntü işleme yönteminin kullanımı, programın yazılması ve çalışması ile ilgili konular ele alınmıştır. Daha sonra cam ürünlerinden düzenli veri alınmasına olanak sağlayan donanım özellikleri ayrıntı olarak verilmiştir. Oluşturulan prototipte kullanılan görüntü işleme teknikleri ile cam yüzey üzerindeki hataların belirlenmesi işlemi yapılmıştır. Bu işlemde gömülü sistemin tasarımına geçilmesiyle birlikte Python dilinde kodlama yapılarak gömülü sistemin yazılımı tamamlanmıştır. Yazılımda aşağıda belirtilen algoritma kullanılmıştır.

3.2.1 Morfolojik görüntü işleme, kodlama ve sınıflandırma

Tasarlanan prototipin yazılımında “Phyton” programlama dili kullanılmıştır. Phyton dili açık kaynak kodlu bir yazılımdır. Mikro sürücüye yüklenen yazılımın algoritması aşağıdaki Şekil 3.18’de verilmiştir.



Şekil 3.18 Akış diyagramı

Şekil 3.18’de Kullanılan sistemin akış diyagramı verilmiştir. Algoritmanın işlem basamakları şunlardır;

- Resim alma / yakalama
- Resmi gri renk dönüşümü yaparak görüntü sadeleştirme

- Resimde istenmeyen pikselleri yok ederek görüntüyü daha belirgin hale getirme
- Görüntü üzerindeki nesnelerin sınırlarını belirleme
- Hata var ya da yok karşılaştırması
- Hata yok ise hata sınıflandırmasına giderek yazılımı sonlandırma
- Eğer hata tespit edilirse sınır değerine göre hata belirleme
- Belirlenen hatalar morfolojik olarak hangi şekle benziyorsa hata tipini oluşturma
- Hatayı sınıflandırmak ve yazılımı sonlandırma.

Görüntü üzerinde hataların sınır tespiti yapıp, arka plandan ayrılmıştır. Hataların piksel sayıları hesaplanmıştır. Hesaplanan piksel sayılarına göre batık hatası ve çizik hatası ayrımı yapılmıştır. Batık hatalarının 0-5.000 piksel aralığında, çizik hatalarının ise 5.000-16.000 piksel aralığında olduğu tespit edilmiştir. Bu tespitler yapıldıktan sonra analiz yapmak ve sınıflandırmada kullanılmak üzere, toplam batık hatası (BS), toplam çizik hatası (CS), toplam batık hatasının tüm yüzeyde kapladığı alan ($BO=BS/TP$), toplam çizik hatasının tüm yüzeyde kapladığı alan ($CO=CS/TP$) değerleri yazılım tarafından hesaplanmıştır. Burada CO bize çizğin yüzeyde ne kadar alan kapladığını ve hatanın boyutu hakkında bilgi verir, BO ise batığın yüzeyde ne kadar alan kapladığını ve hatanın boyutu hakkında bilgi verir. Diğer BS, CS ve toplam piksel sayıları (TP) hata sınıflandırmasında prototipin ne kadar doğruluk oranıyla çalıştığını tespit etmek için veri setini oluşturur.

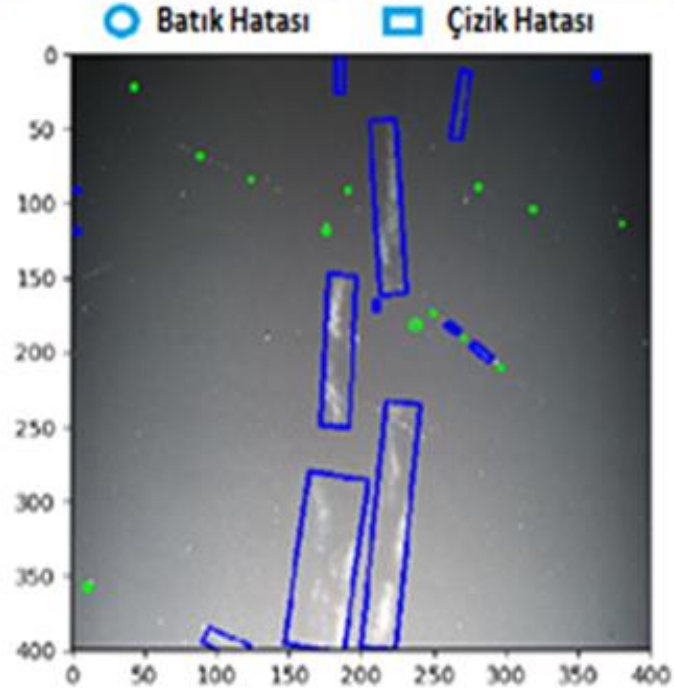
Hataların tespitinden sonra morfolojik görüntü işleme teknikleri uygulanarak (açma, kapama, aşındırma, genleşme) görüntü üzerindeki batık hataları dairesel formda, çizik hataları ise dikdörtgen formda belirlenmiştir.



Şekil 3.19 Orjinal cam görüntüsü

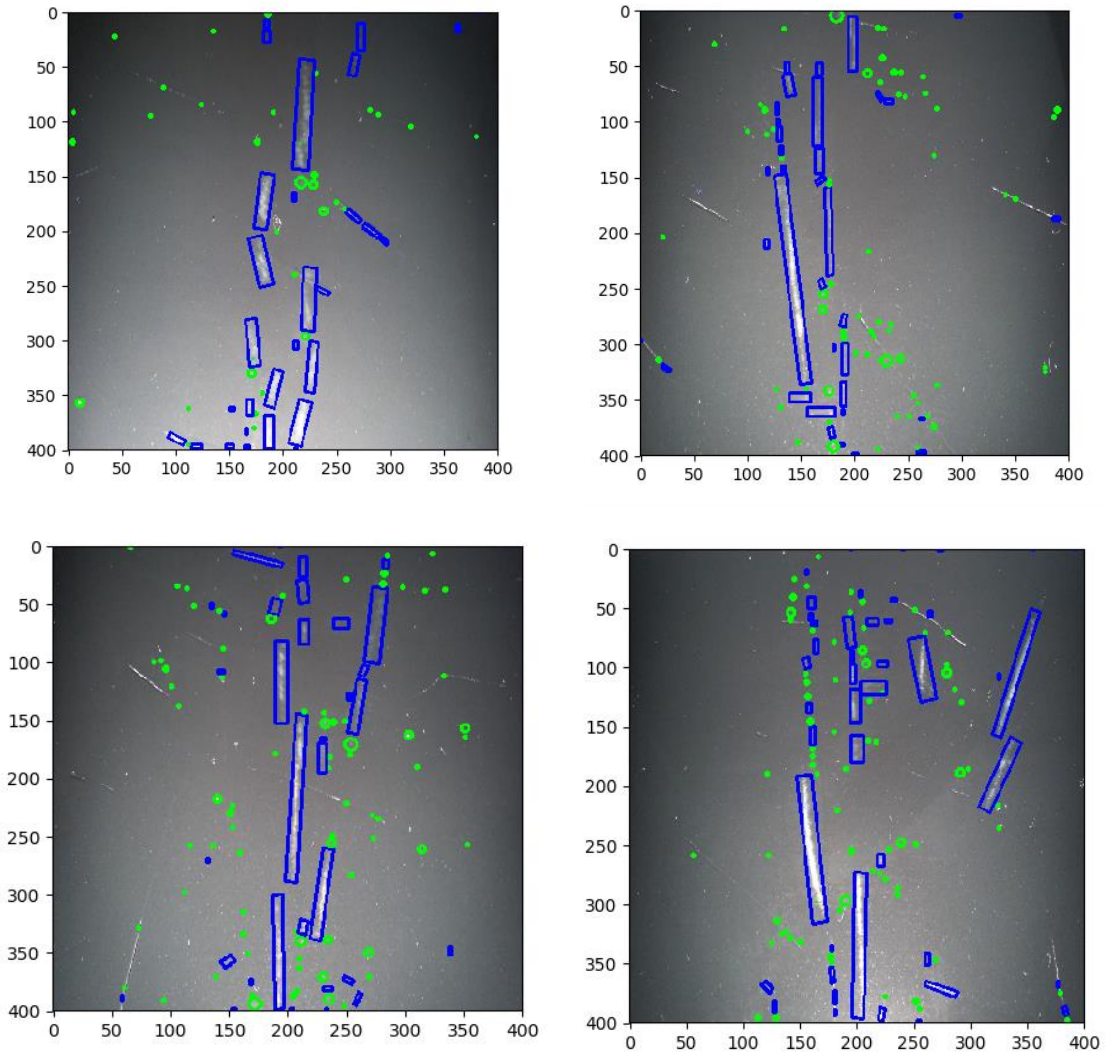
Şekil 3.19’da hataları belirlenmemiş orijinal cam görüntüsüne ait ekran çıktısı verilmiştir.

Çizik sayısı :14 çizik yüzey alanı(%):3.19 Batık sayısı:13 Batık yüzey oranı(%):0.06



Şekil 3.20 Hata tespiti yapılmış ekran görüntüsü

Şekil 3.20’de Cam ürünlerinde gerçek zamanlı hata tespiti yapan prototipin ekran çıktısı verilmiştir. Gerçek zamanlı alınan cam görüntüleri Python dilinde kodlanan gömülü sistem tarafından işlenerek hataları “mavi renk” çizik hatalarını, “yeşil renk” batık hatalarını göstermektedir.



Şekil 3.21 Prototipten alınan gerçek zamanlı ekran çıktıları

Şekil 3.21’de gerçek zamanlı olarak alınan cam görüntüler üzerindeki hataları gösteren ekran çıktıları verilmiştir.

3.2.2 Sınıflandırma işlemi

Sınıflandırma; nedeni ve sonucu bilinen örneklerden yararlanarak; yeni karşılaşılan bir örnek hakkında yorum yapabilmeyi sağlayan bir işlemdir (K. Adem ve U. Orhan, 2013). Bu çalışmada cam üzerindeki hatalar iki tiptedir (batık hataları ve çizik hataları) ve bir düzlem üzerindedir. Hataların düz bir çizgi halinde değil de, cam yüzey üzerine dağılmış bir halde olmasından dolayı 3 tane sınıflayıcı da başarı oranı yüksektir. QDA, NLSVM ve DT (Karar Ağacı -Decision Tree) de ikili değişken ve tek bir düzlem üzerine sınıflayıcılardır.

Sınıflandırmada kullanılmak üzere 30 sağlam cam görüntüsü, 30 batık hatalı cam görüntüsü, 30 adet çizik hatalı cam görüntüsünden oluşan bir veri kümesi kullanılmıştır. SVM’ de %90.0, DT’ de %93.1 ve QDA %93.2 oranında yüksek bir başarı ile hata tespiti yaptığı görülmüştür.

QDA; $a \neq 0$, 2. dereceden 1 bilinmeyenli bir denklemde kökleri bulur. Bu kökler X_1 ve X_2 dir. Kökleri hesaplamak için öncelikle denklemde delta değeri hesaplanır. Δ (delta) şekliyle ifade edilir: Δ değeri hesaplandıktan sonra karşılaştırma işlemine geçilerek kökler hesaplatılır.

$\Delta > 0$ yani Δ pozitif ise, denklemin farklı iki gerçek kökü vardır ve kökler “Eşitlik 3.5’de” belirtildiği şekilde hesaplanır.

$$X_1, X_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (3.5)$$

$\Delta = 0$ yani Δ sıfıra eşit ise, denklemin değerleri birbirleriyle çakışan, yani birbirine eşit, iki gerçek kökü vardır (Eşitlik 3.6).

$$X_1 = X_2 = \frac{-b}{2a} \quad (3.6)$$

$\Delta < 0$ yani Δ negatif ise, denklemin gerçek kökü yoktur yani denklemin çözümü bulunamaz.

SVM; istatistiksel yapıda bir makine öğrenmesi teorisidir. Bu yöntemin amacı en uygun karar fonksiyonunu tahmin etmektir. Üzerinde çalışılan veriyi iki sınıfı ayırt ederek karar fonksiyonu ile en uygun hiper düzlem belirlenir, böylece eğitim verisini ayırt edebilir. SVM'deki destek vektörlerinin denklemleri kullanılacak (Eşitlik 3.7) verilmiştir.

$$\begin{aligned} wx + b &= +1, y = +1 \\ wx + b &= -1, y = -1 \end{aligned} \quad (3.7)$$

Doğrusal olarak farklılaştırılabilen bir ikili sınıflandırma probleminde “y” sınıf etiketi olduğunda, “w” ağırlık vektörüdür ve “b” yaklaşık değerdir. M değerini artırmak için optimum düzlem aralıklarının değerini “Eşitlik 3.8'de gösterildiği gibi” “w” değerini en aza indirmek için gereklidir.

$$m = \frac{2}{\sqrt{ww}}, f_{\min}(w) = \frac{ww}{2} \quad (3.8)$$

Buradan;

$$y_i(wx_i + b) - 1 \geq 0 \quad (3.9)$$

$$L(w, b, a) = \frac{w^2}{s} - \sum_{i=1}^k a_i y_i (wx_i + b) + \sum_{i=1}^k a_i$$

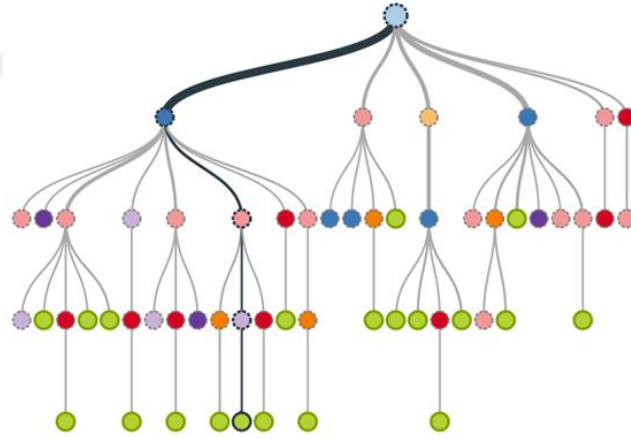
elde edilir.

Son olarak elde edilen denklem (Eşitlik 3.9) Lagrange metoduyla çözülür. Böylelikle ikili sınıflandırma için denklemin karar fonksiyonu elde edilir (Eşitlik 3.10).

$$f(x) = \text{sign}\left(\sum_{i=1}^k a_i y_i(x) + b\right) \quad (3.10)$$

DT sınıflandırması; Bu metod, veri kümesinin elemanlarının gruplara ayrılmasına dayanır. Amaç, grubun bütün elemanları aynı sınıf etiketine sahip olana kadar işlemi devam ettirmektir. Veri kümesinden DT oluşturup belirli durumlarda sistemin ne yapacağını belirlemek için kullanılır (Eşitlik 3.11). DT sınıflandırmasında rassal bir değişkenin belirsizlik ölçütü olarak bilinen entropi, bir süreç için tüm örnekler tarafından içerilen enformasyonun beklenen değeridir. Enformasyon ise rassal bir olayın gerçekleşmesine ilişkin bir bilgi ölçütüdür. Eşit olasılıklı durumlar yüksek belirsizliği temsil eder. Entropi değerinin formülü aşağıda verilmiştir (Eşitlik 3.11).

$$H(X) = \sum_{i=1}^2 p_i \log_2 p_i = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1 \quad (3.11)$$



Şekil 3.22 DT sınıflayıcı (Hatipoğlu, 2018)

Şekil 3.22’de DT sınıflandırması şematik olarak verilmiştir. Entropi (H) adıyla anılan terim, tüm ait durumlarına ait π olasılıklarına bağlı bir değerdir.

3.2.3 Geçerlilik kriteri

Sınıflandırmanın başarısını görmek için aşağıda verilen değerler hesaplatılmıştır.

Batık sayısı (BS): Batık hatası içeren cam ürünlerdeki piksel sayısıdır.

Çizik sayısı (CS): Çizik hatası içeren cam ürünlerdeki piksel sayısıdır.

Toplam yüzey piksel sayısı (TP): Cam ürün yüzeyindeki toplam piksel sayısıdır.

Batık oranı (BO) : Batık hatalarının toplam piksel sayısının, tüm yüzeydeki piksel sayısına oranıdır. Bu oran hatanın yüzeyde ne kadar alan kapladığını bularak, hata toleransı olup olmadığını tespit eder.

Çizik oranı (CO) : Çizik hatalarının her birinin, tüm yüzeye oranıdır. Bu oran hatanın yüzeyde ne kadar alan kapladığını bularak hata toleransı olup olmadığını tespit eder.

Sağlam ürün (SU): Belli bir eşik değeri altındaki hata pikseline sahip üründür.

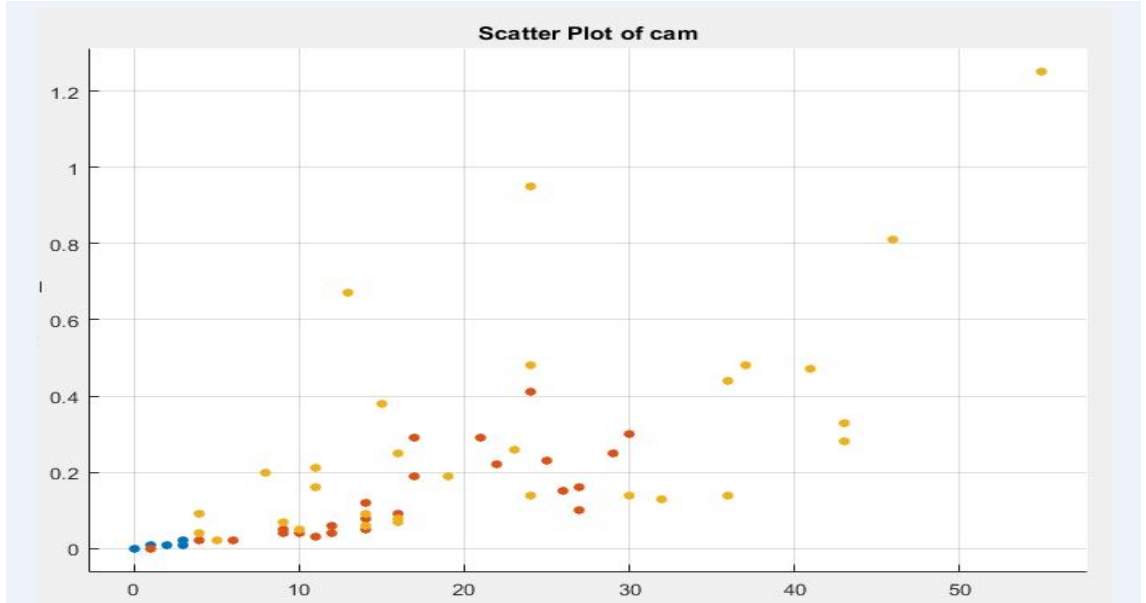
Toplam Hata (TH): Bir ürün üzerindeki hatalı piksel sayısıdır.

Toplam hata parametresi (duyarlılık), hatalı cam ürünlerin ya da sağlam cam ürünlerin birbirinden ayrılması için eşik değeri piksellerinin bulunmasında geçerli olan değerdir. Hatalı sağlam ayırımı yapıldıktan sonra batık oranı belirlenen ürünün batık eşik değerini üzerinde olup olmadığını tespit edilmesini sağlayan değerdir. Çizik hata sayısı (CS), aynı şekilde ikinci tip hata olarak adlandırılan çizik hatasının belirlenmesindeki değerdir. Batık oranı (BO) ve çizik oranı (CO) batık ve çizik hata sayısının toplam piksele oranından çıkan değerdir. Bu da hataların göz ardı edilip edilemeyeceği kararını vermek için kullanılan değerdir (Eşitlik 3.12).

$$TH=(BS+CS)/TP \quad BO=BS/TP \quad CO=CS/TP \quad (3.12)$$

4. BULGULAR VE TARTIŞMA

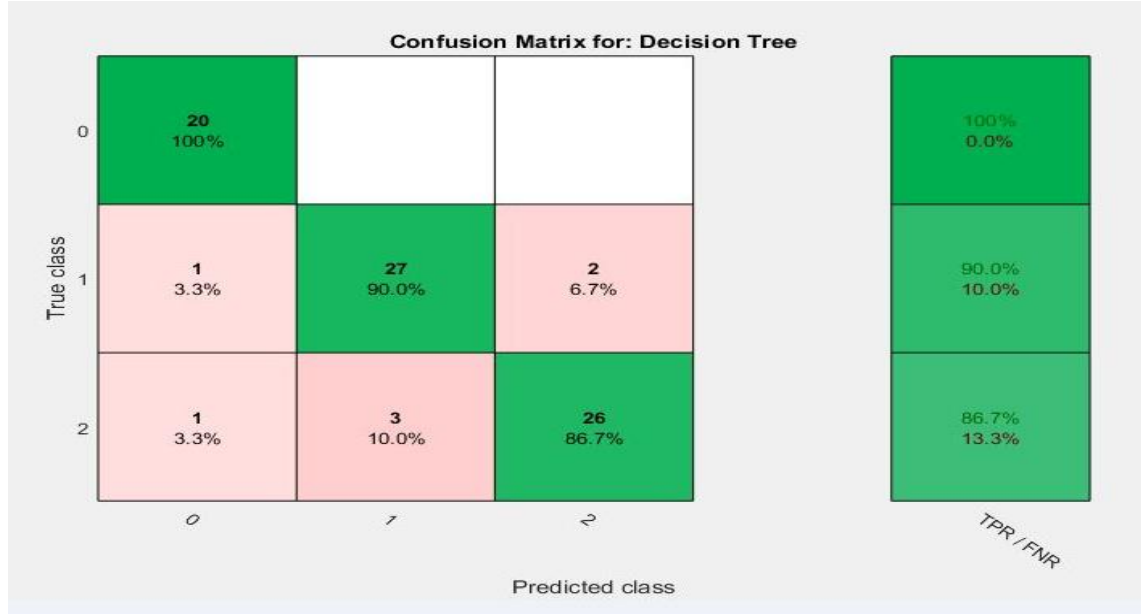
Başarı oranı tespiti için çalışılan tüm sınıflayıcılarda eşik değeri 0.63 olarak belirlenmiştir. Sınıflandırmada veriler 5 (beş) itersayonla sonuca ulaşmıştır. Beş iterasyondan sonra makinenin öğrendiğini tekrar etme süreci başladığından dolayı doğru iterasyon sayısı 5 olarak belirlenmiştir. Sınıflandırma işleminin sonuçlanmasından sonra ortaya çıkan sonuçların grafiksel olarak dağılımı ve Cofusion matrisleri aşağıda verilmiştir.



Şekil 4.1 Hata dağılım grafiği

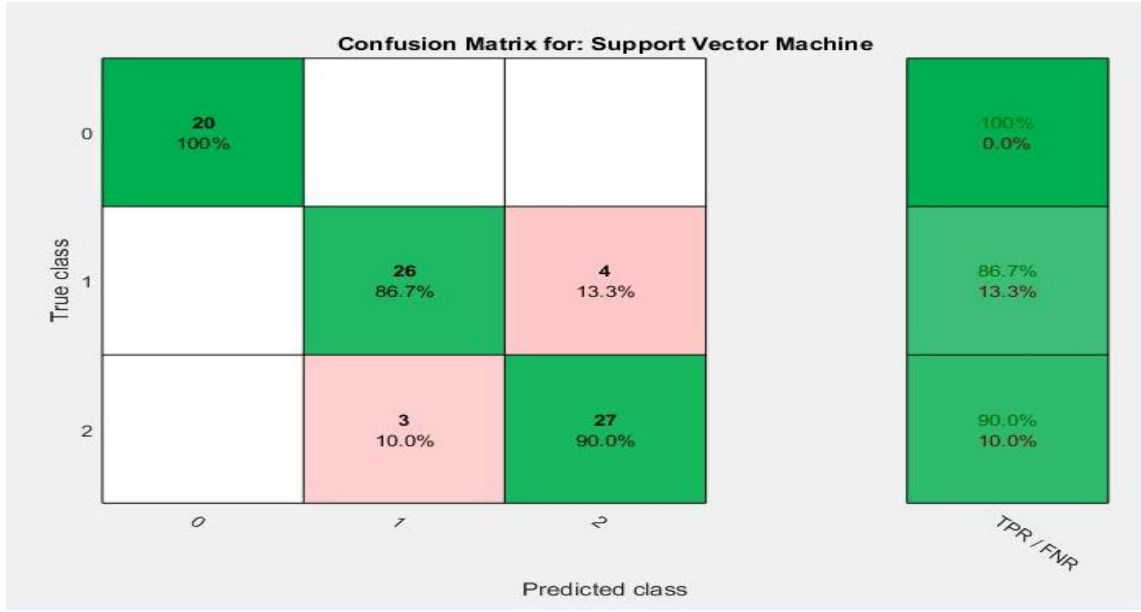
Şekil 4.1’de x ve y ekseninde belirlenen değişkenlerin değişimini gösterir. Aynı zamanda kabarcık grafiği olarak da adlandırılır. Burada “MAVI” sağlam cam ürünlerini, “KIRMIZI” çizik hatalı cam ürünleri, “SARI” ise batık cam hatalarını göstermektedir. Bu grafikte tüm kabarcıklar eşit boyuttadır ve her bir renk farklı bir değişkeni temsil etmektedir. Grafikten anlaşılabacağı üzere 3 tane değişkenden söz etmek mümkündür (batık, çizik ve sağlam olmak üzere). Grafikten anlaşılabacağı üzere çizik hatalarını yoğunluğu batık hatalarının yoğunluğuna göre çok daha az olduğu tespit edilmiştir. Sistemin başarı oranının tespitinde 3 tane ikili sınıflandırma metodu büyük

oranda hata tespiti yapmıştır. Bunların başarı oranlarını gösteren Confusion matrisleri aşağıdaki Şekil 4.2, Şekil 4.3 ve Şekil 4.4’de verilmiştir.



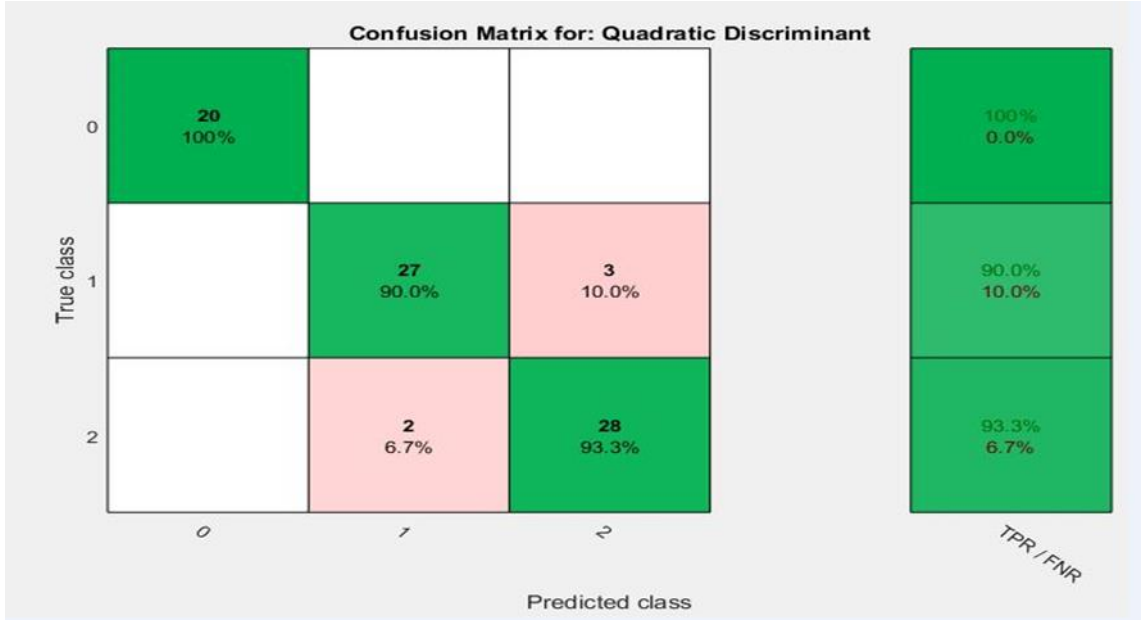
Şekil 4.2 Karar ağaçları ile sınıflandırılma

Şekil 4.2’de DT sınıflayıcıda başarı oranı %93.1’dir. Bu sınıflayıcıda sağlam olan 20 cam ürünün tamamını doğru bulurken, hatalı camlardan batık cam örneklerinde 3 tane, çizik cam ürünlerinde 4 tane cam hatasını yanlış sınıflandırmıştır. Burada batık cam ürünleri üzerindeki piksel sayısının çizik cam ürünleri üzerindeki piksel yakınlığından dolayı başarı oranı düşmüştür. Sağlam cam ürünlerdeki başarı oranı %100, Batık cam ürünler üzerindeki başarı oranı %90 ve çizik cam ürünler üzerindeki başarı oranı da %86.7 olduğu görülmüştür.



Şekil 4.3 SVM

Şekil 4.3’de SVM cam ürünlerin tamamı doğru tespit edilirken DT sınıflayıcıdaki problem burada da ortaya çıkmaktadır. Bazı cam ürünlerdeki hatalı piksel sayının, sağlam piksel sayısına yakın olması sınıflayıcının başarı oranını düşürmüştür. Sınıflandırma işlemi sonucunda sağlam cam ürünlerdeki başarı oranı %100, batık cam ürünler üzerindeki başarı oranı %86.7 ve çizik cam ürünler üzerindeki başarı oranı da %90 olarak belirlenmiştir.



Şekil 4. 4 QDA

Şekil 4.4’de QDA başarı oranı yüksek olan diğer DT ve SVM sınıflayıcı da olduğu gibi sağlam ürünlerin tespitindeki başarı oranı %100 dür. Bu sınıflayıcıda ikili sınıflandırma işlemi sonucunda, diğer sınıflayıcılarda olduğu gibi hatalı piksel değeri yakın olan ürünlerdeki hata tespiti sonucunda başarı oranı düşmüştür. Buna rağmen üç sınıflandırma işlemi sonucunda başarı oranı %90 üzerindedir. QDA da sağlam ürünlerin başarı oranı %100, batık hatasında %90 ve çizik hatasında %93.3 başarı oranı tespit edilmiştir.

Sınıflandırmada kullanılan sınıflayıcılardan “FFA-DT”, “FFA-SVM” ve “FFA-QDA” içeriğinden ortaya çıkan Confusion matrisinin sonuçları aşağıda verilmiştir.

Tablo 4.1 FFA-DT

Sınıf	Batık	Çizik
Batık	26	3
Çizik	3	27

Tablo 4.2 FFA-SVM

Sınıf	Batık	Çizik
Batık	27	3
Çizik	4	26

Tablo 4.3 FFA-QDA

Sınıf	Batık	Çizik
Batık	28	2
Çizik	3	27

Tablo 4.1, 4.2 ve 4.3’de belirtildiği gibi DT, SVM ve QDA tabanlı sınıflayıcılarda yinelemeli eşikleme ile sınıflandırma yapılmıştır. DT sınıflandırmasında toplamda 30 çizik hatası içerisinde 27 çizik, 2 batık hatası şeklinde bir sonuç ortaya çıkmıştır. Toplamda 30 batık hatasında ise 27 batık hatası, 3 çizik hatası biçiminde sınıflandırma yapılmıştır. Batık ve çizik hatalarının birbirine benzerliklerinden çizikleri batık olarak sınıflandırmıştır. Çizik hatalarında 1 tane hatayı piksel sayısına bağlı olarak sağlam olarak sınıflandırmıştır. SVM tabanlı yinelemeli eşikleme yapılan sınıflandırmada ise toplamda 30 çizik hatası içerisinde 26 çizik ve 4 batık hatası biçiminde sınıflandırma yapmış, batık hatalarında toplamda 30 batık içerisinde 27 batık, 3 çizik hatası tespit edilmiştir. Yapılan sınıflandırma hatalarında çizik ve batık hatalarının benzerliği doğruluk yüzdesini düşürmüştür. QDA tabanlı sınıflayıcıda aynı veri topluluğu içerisinde çizik hatalarında 27 çizik hatası ve 2 batık hatası, batık hatalarında ise 28

batık hatası ve 2 çizik hatası olarak sınıflandırma yapılmıştır. Burada da çizik hatalarının batık hatalarına benzerliklerinden dolayı başarı oranını düşürmüştür. Sonuçlar, kullanılan 3 farklı modelde en yüksek başarı oranını göstermektedir. Sınıflandırmaya giren ürünler sağlam, batık hatalı ve çizik hatalı olarak tespit edilen ürünlerdir; bu ürünlerindeki piksel sayıları belirlenerek sınıflandırma yapılmıştır.

Sınıflandırmada 22 tane farklı sınıflayıcı arasından 3 metot belirlenmiştir. Bu metotlar, Lagrange denklemi ile sınıflandırma işlemi gerçekleştirilir ve 2 sınıflı ve tek düzlem üzerinde işlem yapan sınıflayıcılardır. Bu çalışmada kullanılan sınıflayıcılarda başarı oranı %93 ün üzerindedir. Bu başarı oranının yakalanmasının sebebi cam üzerindeki hataların aynı düzlem üzerinde ve 2 sınıflı olmasıdır.

Tablo 4.4 QDA istatistiksel sonuçları

Sınıflayıcı	Çizik			Batık		
QDA sınıflayıcı	Duyarlılık	Özgünlük	TCC	Duyarlılık	Özgünlük	TCC
	0,931	0,900	0,916	0,903	0,933	0,916

Tablo 4.4’de QDA sınıflayıcının istatistiksel hesaplamalar sonucunda ortaya çıkan veriler verilmiştir. Bu tablo aşağıdaki istatistiksel ölçütler kullanılarak hazırlanmıştır.

TP= Doğru sınıflandırılmış batık hatalarının sayısı,

TN=Doğru sınıflandırılmış çizik hatlarının sayısı,

FP=Batık olarak belirlenen fakat yanlış sınıflandırılan batık hatlarının sayısı,

FN=Çizik olarak belirlenen fakat yanlış sınıflandırılan çizik hatalarının sayısı,

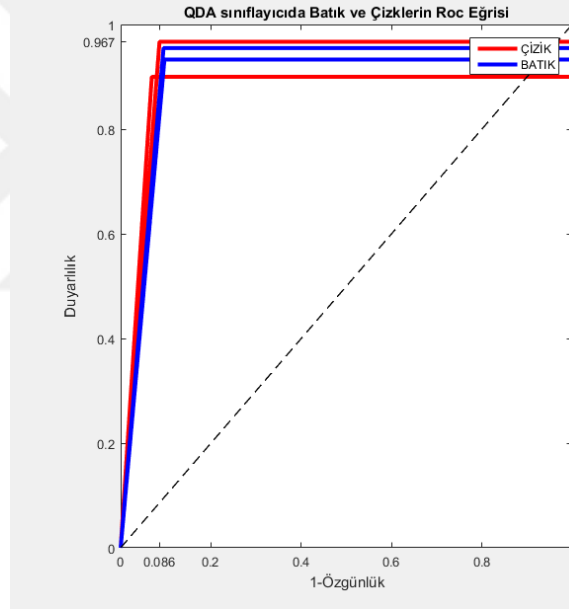
TCC= TP, TN, FP ve FN kullanılarak duyarlılık, özgüllük ölçütleri kullanılarak yapılan sınıflandırmanın toplam doğruluk oranıdır.

$$\text{duyarlılık} = \frac{TP}{TP + FN} \quad (4.1)$$

$$\text{özgünlük} = \frac{TP}{TP + FP} \quad (4.2)$$

$$\text{toplam - doğruluk (TCC)} = \frac{TP + TN}{TN + FN + TP + FP} \quad (4.3)$$

Duyarlılık oranı “Eşitlik 4.1”, verilen veri kümesi içerisinde batık veya çizik hatalarını doğru tespit etme olasılığıdır. Özgünlük “Eşitlik 4.2”, tüm veri kümesi içerisinde belirgin hataları diğerlerinden ayırt etme olasılığıdır. Toplam doğruluk “Eşitlik 4.3”, doğru olarak sınıflandırılmış toplam verinin katsayısıdır.



Şekil 4. 5 ROC eğrisi

Şekil 4.4’de hata matrisine bağlı olarak ROC eğrisi verilmiştir.

Sınıflandırmada en yüksek başarı oranı %93.3 ile QDA sınıflayıcıdır. Bu sınıflayıcının başarı oranının doğruluğunu kontrol etmek için ROC eğrisi çizdirilmiştir. Çizilen ROC eğrisinde, eğrinin altında kalan alan sınıflandırma başarısını göstermektedir. Bu alan incelendiğinde QDA sınıflayıcının doğru sınıflandırma yaptığı ortaya çıkmıştır.

5. SONUÇ

Yürütülen tez çalışması ile ankastre cam ürünlerden oluşan bir veri kümesi üzerinde hata tespit sistemi önerilmiştir. Toplamda test için 90 adet cam üründen oluşan bu veri kümesi sağlam, batık hatalı ve çizik hatalı olmak üzere 3 sınıftan oluşmaktadır.

Cam örnekleri üzerinde yapılan çalışmada, hatalı görüntüler üzerindeki piksel sayılarının yakın olması, çizik hatalarının batık hataları ya da batık hatalarının çizik hataları olarak sınıflandırılmasına sebep olmaktadır. Bu durumda yapılan birçok deneme sonucunda belirlediğimiz sabit eşik değeri ile bu iki hatayı birbirinden ayırılabilme de başarılı sonuç vermiştir.

Cam ürünlerin yüzeyinde bulunan hataların tespitinde, hataların piksel sayılarından çok, bu hataların camın yüzeyinde ne kadar alan kapladığı önemlidir. Cam ürünlerin yüzeyindeki hatalar incelendiğinde, batık hatlarındaki piksel sayısı 0-5.000, çizik hatların 5.000-16.000 pikselden oluştuğu tespit edilmiştir. Farklı cam örneklerine bakıldığında ürünlerin üzerinde hatalı pikseller tespit edilmesine rağmen batık hatası ya da çizik hatası olmadığı görülmüştür. Bu durumda piksel sayısından çok hatanın tüm yüzeydeki yoğunluğunun önemi ortaya çıkmıştır. Cam ürünlerin yüzeyindeki hataların kapladığı alanın küçük olmasından dolayı bu hataların göz ardı edilmesine (hata toleransı) sebep olmaktadır. Örneğin; Likit Kristal Ekran (LCD) ve Thin Film Transistor (TFT) gibi düşük kontrastlı yüzeylerde hata toleransı içerisinde ölü pikseller belli bir oranda göz ardı edilmektedir. Prototip, hataların yüzeyde kapladığı alanı tespit ederek, ilk olarak hatalı ürün / sağlam ürün ayarımı yapmakta, sonrasında hatalı ürünleri batık hatalı ürün ve çizik hatalı ürün diye ikili olarak sınıflandırmaktadır. Bu sınıflandırma işlemini yüzey yoğunluğunu hesaplayarak yapar. Bu hesaplama işleminin sonucunda hatalı / sağlam ürün ve batık / çizik hatalı ürünleri daha doğru biçimde sınıflandırma yapmaktadır.

Cam ürünlerin ya da yansıtıcı yüzeye sahip ürünlerin incelenmesinden sonra hatalar ikiye ayrılmaktadır. İki sınıflı problemlerin çözümünde LSVM ve NLSVM kullanılabilir. Bu tür ürünlerde hatalar yüzeye eşit olarak dağılmadığından dolayı doğrusal sınıflandırmanın başarı oranı düşüktür. NLSVM bu tür problemler için daha doğru sınıflandırma yapmaktadır. Çalışmadan alınan sonuçların sınıflandırılmasında SVM, QDA, DT gibi NLSVM kullanılmaktadır. Başarı oranı ise %93.1'dir.

Cam ürünlerde hata kontrolü geleneksel metot ile yapıldığında ortaya çıkan başarı yüzdesinin, bilgisayar tabanlı görüntü işleme tekniklerine göre çok düşük olduğu görülmüştür. Bilgisayar tabanlı görüntü işleme tekniklerinden, renk uzayı dönüşümü ve morfolojik görüntü işleme teknikleri ile cam ürünler üzerinde hata tespitleri yapılmıştır.

Cam yüzeylerin muayenesi yansıma ve şeffaflıktan dolayı çok zordur. Bu sebeple morfolojik görüntü işleme teknikleriyle açık kaynak tabanında çalışan prototip geliştirilmiştir. Bu prototip bir mikro sürücü üzerindeki kamera yardımıyla konveyör bant üzerinde akan ürünleri sağlam, batık hatalı ve çizik hatalı olarak ayıran bir sistemdir ve % 93.8 gibi yüksek bir başarı oranı ile çalışmaktadır. Yapılan çalışma; yansıtıcı yüzeylere sahip nesnelerdeki hata tespiti için literatüre katkı sağlamıştır.

6. KAYNAKLAR

- Acar, U., & Bayram, B., 2009. Morfolojik Görüntü Filtreleri ile İkonos Görüntülerinden Otomatik Bina Çıkarımı, TMMOB Harita ve Kadastro Mühendisleri Odası 12. Türkiye Harita Bilimsel ve Teknik Kurultayı.
- Adelson, E. H., & Wang, J. Y. 1992. Single lens stereo with a plenoptic camera. IEEE transactions on pattern analysis and machine intelligence, 14(2), 99-106.
- Adem, K., & Orhan, U. 2013. Imaging processing based quality control of transversal seams in Tetra Brik Aseptic cartons. 21st Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.
- Agrawal, N. ve Agrawal, S., 2011. Glass Defect Detection Techniques using Digital Image Processing—A Review, Spec. issues IP Multimed. Commun., vol. 1, pp. 65–67,
- Aitkenhead, M. J., 2008. A co-evolving decision tree classification method. Expert Systems with Applications, 34(1), 18-25.
- Anonim, 2017. T.C. Bilim, Sanayi ve Teknoloji Bakanlığı Cam Sektörü Raporu
- Anonim, 2018. T.C. Bilim, Sanayi ve Teknoloji Bakanlığı Cam Sektörü Raporu
- Anonim, 2019. T.C. Bilim, Sanayi ve Teknoloji Bakanlığı Cam Sektörü Raporu
- Ansari, I. A., & Borse, R. Y. 2013. Image Processing and Analysis. International Journal of Engineering Research and Applications, 3(4), 1655-1658.
- Boztoprak, H., Çağlar, M. F., & Merdan, M., 2007. Alternatif morfolojik bir yöntemle plaka yerini saptama, Süleyman Demirel Üniversitesi, Isparta.
- Catanzaro, M., 2013. Encyclopedia of Bioinformatics and Computational Biology. Elsevier, 1104, U.S.A
- Choi, A., Tavabi, N., & Darwiche, A. 2016. Structured features in naive Bayes classification. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 30, No. 1).
- Chao, S. M., Tsai, D. M., Tseng, Y. H., & Jhang, Y. R. 2006. Defect detection in low-contrast glass substrates using anisotropic diffusion. In 18th International Conference on Pattern Recognition (ICPR'06) (Vol. 1, pp. 654-657). IEEE.
- Chen, X., Liu, N., You, B. ve Xiao, B., 2016. A novel method for surface defect inspection of optic cable with short-wave infrared illuminance, Infrared Phys. Technol., vol. 77, pp. 456–463.
- Cabral, J. D. D., & de Araújo, S. A. 2015. An intelligent vision system for detecting defects in glass products for packaging and domestic use. The International Journal of Advanced Manufacturing Technology, 77(1-4), 485-494.

- Deng, Z., Zhu, X., Cheng, D., Zong, M., & Zhang, S. 2016. Efficient kNN classification algorithm for big data. *Neurocomputing*, 195, 143-148.
- Demirbaş, H. Y., & Dursun, İ., 2007. Buğday tanelerinin bazı fiziksel özelliklerinin görüntü işleme tekniğiyle belirlenmesi. *Ankara Üniversitesi Ziraat Fakültesi Tarım Bilimleri Dergisi*, 13(3), 176-185.
- Elbehery, H., Hefnawy, A., & Elewa, M. 2005. Surface defects detection for ceramic tiles using image processing and morphological techniques.
- Fırat M., Güngör M., 2004. Askı Madde Konsantrasyonu ve Miktarının Yapay Sinir Ağları ile Belirlenmesi , *İMO Teknik Dergi* , 3267- 3282
- Foody, G. M., McCulloch, M. B., & Yates, W. B. 1995. Classification of remotely sensed data by an artificial neural network: issues related to training data. *Photogram. Eng. Remote Sens.*, 61(4), 391-401.
- Goldberger, J. J., & Ng, J. (Eds.). 2010. *Practical signal and image processing in clinical cardiology*. Springer Science & Business Media
- Hasan, M. S. 2017. An application of pre-trained CNN for image classification. In 2017 20th International Conference of Computer and Information Technology (ICCIT) (pp. 1-6). IEEE.
- Hocenski, Z., Vasilic, S., & Hocenski, V. 2006. Improved canny edge detector in ceramic tiles defect detection. In *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics* (pp. 3328-3331). IEEE.
- Howse, J., 2013. *OpenCV computer vision with python*. Packt Publishing Ltd.
- Hatipoğlu E. 2018. Decision Tree, Part 5. <https://medium.com/@ekrem.hatipoglu/machine-learning-prediction-algorithms-decision-tree-random-forest-part-5-2970905c021e> (11.10.2020)
- Hastie, T., Tibshirani, R., & Friedman, J. 2009. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.
- Joshi, P. ,2015. *OpenCV with Python by example*. Packt Publishing Ltd.
- Kimura, F., Takashina, K., Tsuruoka, S., & Miyake, Y. 1987. Modified quadratic discriminant functions and the application to Chinese character recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1), 149-153.
- Kahraman, F., Capar, A., Ayvaci, A., Demirel, H., & Gokmen, M. 2004, Comparison of SVM and ANN performance for handwritten character classification. In *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference*, 2004. (pp. 615-618). IEEE.
- Keser, T., Hocenski, Ž., & Hocenski, V. 2010. Intelligent machine vision system for automated quality control in ceramic tiles industry. *Strojarstvo*, 52(2), 105-114.

- Kumar, N. Kaur, R. 2013. Detection of Defects in Glass Using Edge Detection with Adaptive Histogram Equalization, *Int. J. Innov. Res. Comput. Commun. Eng.*, pp. 1321–1327, 1970.
- Liu, Y. H., & Chen, Y. J. 2011. Automatic defect detection for TFT-LCD array process using quasiconformal kernel support vector data description. *International Journal of Molecular Sciences*, 12(9), 5762-5781.
- Liu, S., Xiao, J., Liu, J., Wang, X., Wu, J., & Zhu, J. 2017. Visual diagnosis of tree boosting methods. *IEEE transactions on visualization and computer graphics*, 24(1), 163-173.
- Mathur, A., & Foody, G. M., 2008. Multiclass and binary SVM classification: Implications for training and classification users. *IEEE Geoscience and remote sensing letters*, 5(2), 241-245.
- Miller, Z., Deitrick, W., Hu, W., 2011. Anomalous network packet detection using data stream mining. *Journal of Information Security*, 2(04), 158
- Niu, Y., Wu, X., & Shi, G. 2016. Image enhancement by entropy maximization and quantization resolution upconversion. *IEEE Transactions on Image Processing*, 25(10), 4815-4828.
- Osuna, E. E. 1998. Support vector machines: Training and applications (Doctoral dissertation, Massachusetts Institute of Technology).
- Öztürk, Ş., & Akdemir, B. 2018. Fuzzy logic-based segmentation of manufacturing defects on reflective surfaces. *Neural Computing and Applications* 29(8), 107-116.
- Öztürk, Ş., & Akdemir, B. 2015. Comparison of edge detection algorithms for texture analysis on glass production. *Procedia-Social and Behavioral Sciences*, 195, 2675-2682.
- Qu, T., Zou, L., Zhang, Chen, Q., X., ve Fan, C., 2016. Defect detection on the fabric with complex texture via dual-scale over-complete dictionary, *J. Text. Inst.*, vol. 107, no. 6, pp. 743–756.
- Rosli, N., Fauadi, M., Awang, N., & Noor, A. , 2018. Vision-Based Defects Detection for Glass Production based on Improved Image Processing Method. *Journal of Advanced Manufacturing Technology (JAMT)*, 12(1(1), 203-212. Retrieved from <https://jamt.utem.edu.my/jamt/article/view/3935>
- Schöning, J., & Heidemann, G. 2016. Taxonomy of 3D Sensors-A Survey of State-of-the-Art Consumer 3D-Reconstruction Sensors and their Field of Applications. In *VISIGRAPP (3: VISAPP)* (pp. 194-199).
- Spinola, C.G., Canero, J., Moreno-Aranda, G., Bonelo, J. M. ve Martin-Vazquez, M., 2011. Real-time image processing for edge inspection and defect detection in stainless steel production lines, *IEEE Int. Conf. Imaging Syst. Tech. IST 2011 - Proc.*, no. May, pp. 170–175, 2011.

- Singh, T., Lal Dua, R., Agrawal, S. ve Acharya, A., 2013. Detection of Defects in Glass Sheet using C. S. C based Segmentation Method, Int. J. Comput. Appl., vol. 68, no. 14, pp. 29–32.
- Şentürk, E. 2011. Mutluluk düzeyinin sosyo-demografik özelliklerle lojistik regresyon analizi aracılığıyla incelenmesi ve Türkiye için bir uygulama.
- Toraman, S. 2006. Histopatolojik imgelerin değerlendirilmesinde örüntü tanıma temelli karar destek sistemleri/Decision support systems based on pattern recognition for evaluating of histopathologic images.(Y. Lisans Tezi),Fırat Üniv. Elektronik ve Bilgisayar Eğitim A.B.D. , ELAZIĞ.
- Yang S.2019. A Tour To Machine Learning And Deep Learning, An Introduction to Naïve Bayes Classifier From theory to practice, learn underlying principles of Naïve Bayes. <https://towardsdatascience.com/introduction-to-na%C3%AFve-bayes-classifier-fa59e3e24aaf> (09.10.2020)
- Zhao, J., Kong, Q. J., Zhao, X., Liu, J., & Liu, Y. 2011. A method for detection and classification of glass defects in low resolution images. Sixth International Conference on Image and Graphics (pp. 642-647). IEEE.

7. EKLER

7.1 EK-1 Matlab kodu

```
a=imread('*.jpg');
a=imread('*.jpg');
b=rgb2gray(a);
yiq=rgb2ntsc(a);
imshow(yiq);
y=yiq(:,:,1);
figure, imshow(y);
i=yiq(:,:,2);
figure, imshow(i);
q=yiq(:,:,3);
figure, imshow(q);
hsv=rgb2hsv(a);
imshow(hsv);
h=hsv(:,:,1);
figure, imshow(h);
s=hsv(:,:,2);
figure, imshow(s);
v=hsv(:,:,3);
figure, imshow(v);
adapt_v=adapthisteq(v);
figure, imshow(adapt_v);
batik=adapt_v(:,:,1)>0.63;
figure, imshow(batik);
batik2=bwareaopen(batik,80);
figure, imshow(batik2);
batiksay=bwarea(batik2);
batik_rgb=bw2rgb(batik2);
figure, imshow(batik_rgb);
sonuc=imsubtract(a,batik_rgb);
figure, imshow(sonuc);
```

7.2. EK-2 Görüntülerde Çizik Batık Hatalarını bulan Python Kodu

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from os.path import join, basename, exists
from os import mkdir
from glob import glob
from sys import argv
import shutil

def get_file_list(path, ext="JPG"):
    """
    """
    print (join(path, "*" + ext ))
    return glob(join(path, "*" + ext ))

file_list = get_file_list('./resimler','jpg')

verbose = False
if "verbose" in argv:
    verbose = True
screen_tags = True
if "no_screen_tags" in argv:
    screen_tags = False

n = "sonuclar/"
if exists(n):
    shutil.rmtree(n)
mkdir(n[:-1])

print ("Imge listesi: ", file_list)
for f in file_list:
    print ("Imge yukleniyor...")
    imrgb = cv2.imread(f)
    imrgb = cv2.resize(imrgb,(400,400))
    ##### A gaussian blur may work here

    im = cv2.imread(f, cv2.IMREAD_GRAYSCALE)
    if verbose:
        plt.imshow(im, cmap="gray")
        plt.show()

    imlab = cv2.cvtColor(imrgb,cv2.COLOR_BGR2LAB)
    l,a,b = cv2.split(imlab)
```

```

clahe = cv2.createCLAHE(clipLimit=40, tileGridSize=(12,12))
#clahe = cv2.createCLAHE(clipLimit=20, tileGridSize=(512,512)) # Effects are
interesting
l = clahe.apply(l)
im_lab = cv2.merge((l,a,b))
iml = cv2.cvtColor(im_lab, cv2.COLOR_LAB2RGB)
iml = cv2.cvtColor(iml,cv2.COLOR_RGB2GRAY)
#### Contrast ayarinin sonu

print ("Binarize ediliyor")
"""
_,iml = cv2.threshold(iml,175,255,cv2.THRESH_BINARY)
plt.imshow(iml, cmap="gray")
plt.show()
"""
iml =
cv2.adaptiveThreshold(iml,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRE
SH_BINARY,75,-45)
if verbose:
plt.imshow(iml, cmap="gray")
plt.show()
print ("Preprocessing...")
iml = cv2.erode(iml,(1,5))
if verbose:
plt.imshow(iml, cmap="gray")
plt.show()
iml = cv2.dilate(iml,(25,33))
if verbose:
plt.imshow(iml, cmap="gray")
plt.show()
# Kernel boyutlariyla oynanabilir.
closed = cv2.morphologyEx(iml, cv2.MORPH_OPEN, np.ones([3, 3]) )
if verbose:
plt.imshow(iml, cmap="gray")
plt.show()
iml = cv2.erode(iml,(3,3))
if verbose:
plt.imshow(iml, cmap="gray")
plt.show()
#closed = cv2.morphologyEx(closed, cv2.MORPH_CLOSE, np.ones([3, 3]) )
#plt.imshow(closed, cmap="gray")
#plt.show()
#### Preprocessing sonu

b2, contours, hierarchy = cv2.findContours(closed, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
cizik_count = 0

```

```

batik_count = 0
cizik_area = 0
batik_area = 0
toplam_alan = closed.shape[0] * closed.shape[1]

for cnt in contours:
    rect = cv2.minAreaRect(cnt)
    box = cv2.boxPoints(rect)
    box = np.int0(box)
    print ("Cevreleyen koordinatlar:", box)
    p0 = box[0]
    p1 = box[1]
    p2 = box[2]
    cnt_len = np.sqrt(np.square(p1[0] - p2[0]) + np.square(p1[1] - p2[1]))
    a_len = np.sqrt(np.square(p1[0] - p2[0]) + np.square(p1[1] - p2[1]))
    b_len = np.sqrt(np.square(p0[0] - p1[0]) + np.square(p0[1] - p1[1]))
    print ("A uzunlugu, ", a_len, " B uzunlugu, ", b_len)
    ratio = np.float(a_len) / np.float(b_len)

    THRESH = 1.5
    if ratio > THRESH or ratio < 1./THRESH:
        text = "Cizik"
        cizik_count += 1
        cizik_area += cv2.contourArea(cnt)
        cv2.drawContours(imrgb,[box],0,(0,0,255),2)
    else:
        text = "Batik"
        batik_count += 1
        batik_area += cv2.contourArea(cnt)
        cv2.circle(imrgb,tuple(np.mean(box,
axis=0).astype(np.int)),int(cnt_len/np.pi*2),(0,255,0),2)
        print (ratio, text)
        font = cv2.FONT_HERSHEY_SIMPLEX
        if screen_tags:
            cv2.putText(imrgb,text,tuple(p1), font, 1,(255,0,0),1,cv2.LINE_AA)

plt.imshow(imrgb)
title_msg = ""
title_msg += "Cizik Sayisi: " + str(cizik_count) + " "
title_msg += "Cizik Alani(%): " + "{:.2f}".format(cizik_area*100/toplam_alan) + " "

title_msg += "Batik Sayisi: " + str(batik_count) + " "
title_msg += "Batik Alani(%): " + "{:.2f}".format(batik_area*100/toplam_alan)

plt.title(title_msg)
# plt.show()
name = n+basename(f)

```

```

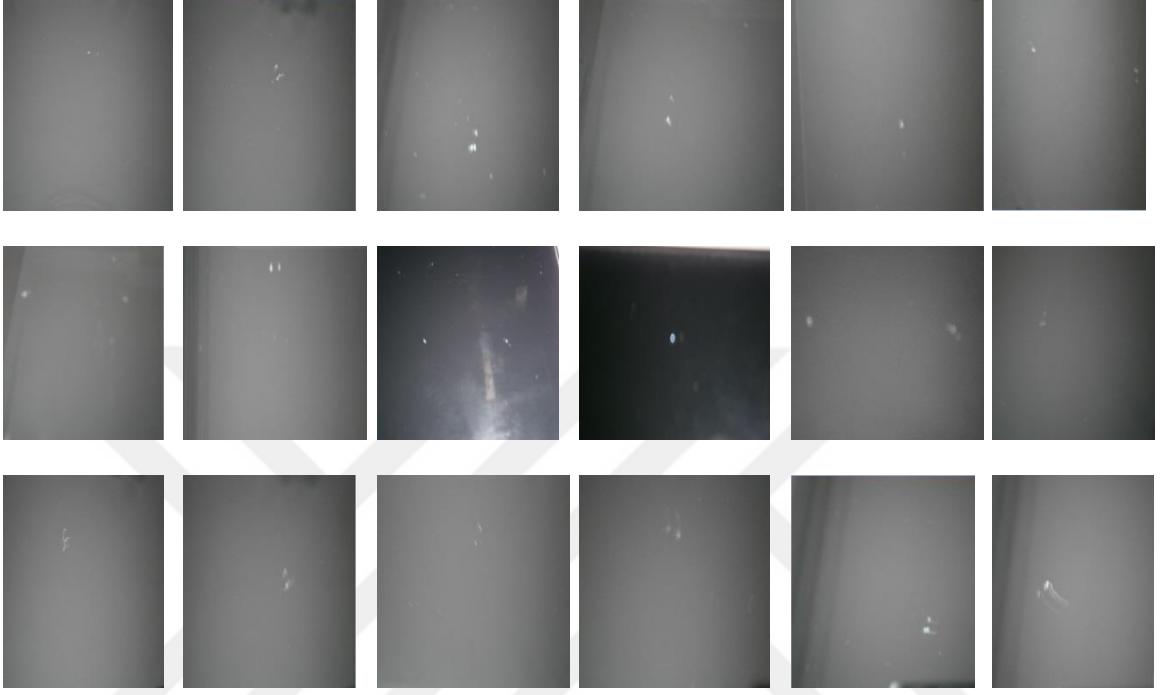
name = name.split('.')[:-1]
name = "".join(name)
name += ".png"
plt.savefig(name)

# this may be used for illumination correction in the future
def adjust_gamma(image, gamma=1.0):
    # build a lookup table mapping the pixel values [0, 255] to
    # their adjusted gamma values
    invGamma = 1.0 / gamma
    table = np.array([(i / 255.0) ** invGamma * 255
                      for i in np.arange(0, 256)]).astype("uint8")

    # apply gamma correction using the lookup table
    return cv2.LUT(image, table)

```

7.2 EK-3 Çalışmada kullanılan “BATIK” cam örnekleri



7.3 EK-4 Çalışmada kullanılan “ÇİZİK” cam örnekleri



7.4 Ek 18 Özgeçmiş

ÖZGEÇMİŞ

Adı Soyadı	Çetin Cem BÜKÜCÜ
Kişisel Bilgiler	Uyruğu: T.C. Doğum Tarihi 01.04.1977 Yer: Merzifon
İletişim Bilgileri	E-posta: cem.bukucu@amasya.edu.tr
Öğrenim Bilgileri	Lise: 1991-1994 Nazilli Lisesi Lisans: 1995-2000 Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik- Bilgisayar Öğretmenliği Bölümü Yüksek Lisans:2004-2006 Bahçeşehir Üniversitesi Fen Bilimleri Enstitüsü Bilgi Teknolojileri Programı
İş Deneyimi	2000-2004 Milli Eğitim Bakanlığı Bilgisayar Öğretmenliği 2004-Halen Amasya Üniversitesi Akademik Personel