T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY SİNİR AĞLARI KULLANARAK GÖRÜNTÜ İŞLEMEYE DAYALI AĞIRLIK TAHMİNİ

ÜNSAL BURAK ŞALVARCI

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRONİK PROGRAMI

DANIŞMAN
YRD. DOÇ. DR. UMUT ENGİN AYTEN

İSTANBUL, 2017

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY SİNİR AĞLARI KULLANARAK GÖRÜNTÜ İŞLEMEYE DAYALI AĞIRLIK TAHMİNİ

Ünsal Burak ŞALVARCI tarafından hazırlanan tez çalışması 21.12.2017 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Yrd. Doç. Dr. Umut Engin AYTEN

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Yrd. Doç. Dr. Umut Engin AYTEN

Yıldız Teknik Üniversitesi _____

Prof. Dr. Lütfiye DURAK ATA

İstanbul Teknik Üniversitesi _____

Yrd. Doç. Dr. Nihan KAHRAMAN

Yıldız Teknik Üniversitesi _____

ÖNSÖZ

Bu yüksek lisans tez çalışması Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı, Elektronik Mühendisliği Yüksek Lisans Programı'nda hazırlanmıştır.

Tezimi hazırlamaya başladığım ilk andan, tezimi tamamlayana kadar geçen sürecin her aşamasında değerli bilgileri ve tecrübesi ile beni yönlendiren, yardım ve katkılarını hiçbir zaman esirgemeyen Sayın Yrd. Doç. Dr. Umut Engin Ayten hocama en içten saygı ve teşekkürlerimi sunarım.

Ayrıca yoğun iş temposunda bu tezi hazırlamaya çalışırken bana her daim destek olan, moral veren ve benimle bu sürecin her anında aynı heyecanı paylaşan başta sevgili eşim olmak üzere değerli aileme de sonsuz teşekkürlerimi sunarım.

Ekim, 2017

Ünsal Burak ŞALVARCI

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ vii

KISALTMA LİSTESİ viii

ŞEKİL LİSTESİ ix

ÇİZELGE LİSTESİ xi

ÖZET xii

ABSTRACT xiv

BÖLÜM 1

GİRİŞ 1

1.1 Literatür Özeti 2
1.2 Tezin Amacı 4
1.3 Hipotez 5

BÖLÜM 2

Genel Bilgiler 6

2.1 Görüntü İşleme 6
2.2 Görüntü İşleme Süreçleri 7
2.2.1 Görüntünün Elde Edilmesi 8
2.2.1.1 Kameranın Kalibre Edilmesi 8
2.2.2 Görüntü Filtreleme ve İyileştirme 8
2.2.3 Görüntü Restorasyonu 9
2.2.3.1 Projektif Dönüşüm 9
2.2.4 Renkli Resim İşleme 10
2.2.5 Dalgacık Dönüşümü ve Çoklu Çözünürlük İşleme 10
2.2.6 Sıkıştırma 10
2.2.7 Morfolojik İşleme 10
2.2.8 Segmentasyon 10
2.2.9 Gösterim ve Tanımlama 11

2.2.9.1 Öz Nitelik Vektörlerinin Çıkartılması 11
2.2.10 Nesne Tanıma 12
2.3 Yapay Zeka ve Makine Öğrenmesi 12
2.4 Yapay Sinir Ağları 13
2.4.1 Yapay Sinir Ağlarının Yapısı 14
2.4.2 Nöronların Çalışma Prensibi 15
2.4.2.1 Yaygın Olarak Kullanılan Aktivasyon Fonksiyonları 17
2.4.3 Yapay Sinir Ağlarında Öğrenme İşlemi 18
2.4.3.1 Eğiticili Öğrenme 18
2.4.3.2 Eğiticisiz Öğrenme 19
2.4.3.3 Takviyeli Öğrenme 20
2.4.4 Yapay Sinir Ağ Modelleri 20
2.4.4.1 Tek Katmanlı Algılayıcılar 21
2.4.4.2 Çok Katmanlı Algılayıcılar 21
2.4.4.3 Radyal Temelli Fonksiyon Ağları 23
2.5 K-Katlı Çapraz Doğrulama Yöntemi 24

BÖLÜM 3

VERİ TABANI 26

3.1 Belirlenen Öz Nitelik Vektörleri 28
3.1.1 Cismin Yüzey Alanı 28
3.1.2 Cismin Çevresi 28
3.1.3 Merkeze En Uzak Kenar Noktası 29
3.1.4 Merkeze En Yakın Kenar Noktası 29
3.1.5 Yuvarlaklık 29
3.2 Cisimlerin Öz Nitelik Vektör Değerleri 29
3.3 Portakal Örnekleri 30
3.4 Yumurta Örnekleri 31

BÖLÜM 4

KULLANILAN MATERYALLER VE METODLAR 33

4.1 Ölçüm Düzeneği 33
4.2 Yapay Sinir Ağının Eğitilmesi ve Test Edilmesi 34
4.3 Ağırlık Tahminleme 35
4.4 Tahminleme Sistemi Süreçleri 36
4.4.1 Cismin Görüntüsünün Okunması 37
4.4.2 Sistemin Kalibre Edilmesi 37
4.4.2.1 Kalibrasyon Parametrelerin Uygulanması 39
4.4.3 Görüntü Dönüşümleri 39
4.4.4 Cismin Görüntüden Ayıklanarak Sınır Noktalarının Belirlenmesi 40
4.4.4.1 Hızlı Ölçüm İçin Cisim Ayıklama Yöntemi 41
4.4.4.2 Hassas Ölçüm İçin Cisim Ayıklama Yöntemi 41
4.4.5 Öz Nitelik Vektör Değerlerinin Belirlenmesi 42
4.4.6 Tahmini Ağırlığın Kullanıcı Arayüzü Kullanılarak Belirlenmesi 43

BÖLÜM 5

SİMÜLASYON VE TEST 45
5.1 Yumurta Simülasyonu 45
5.2 Portakal Simülasyonu 55
BÖLÜM 6
SONUÇ VE ÖNERİLER 64
6.1 Yumurta Örneği 64
6.2 Portakal Örneği 65
6.3 Öneriler 65
KAYNAKLAR 67
EK-A
KAYNAK KODLARI 70
A-1 Kameranın Kalibrasyonu 70
A-2 YSA'nın ÇKA Modeline Göre Eğitilmesi, Testi ve Validasyonu 71
A-3 YSA'nın RTF Modeline Göre Eğitilmesi, Testi ve Validasyonu 72
A-4 K-Katlı Çapraz Doğrulama Yöntemi Datasının Oluşturulması 73
A-5 Kullanıcı Arayüzü 74
A-6 Ağırlık Tahminleme Fonksiyonu 78
ÖZGEÇMİŞ 82

SİMGE LİSTESİ

R	Korelasyon Katsayısı
X,Y,Z	Uzay koordinatları
u,v	Görüntü koordinatları
x'', y''	Projektif dönüşüm için cisim koordinatları
x', y'	Projektif dönüşüm için görüntü koordinatları
w	YSA ağırlıkları
x	YSA girişleri
v	YSA yerel alanı
α	YSA öğrenme oranı
b	YSA ön görü(bias) değeri
Ø	Basamak fonksiyonu eşik değeri
λ	Sigmoid fonksiyonu eğim parametresi
c_i	Radyal tabanlı fonksiyonun merkezi
N	Gizli katmanda bulunan nöron sayısı
σ	Pikselden cm^2 geçmek için kullanılan katsayı
x_i, y_i	Referans görüntü birinci siyah karelerin kesişim noktaları
x_l, y_l	Referans görüntü sağ üst köşedeki siyah karenin kesişim noktaları
k	Referans görüntüdeki karelerin kenar uzunluğu
dx	Referans görüntüdeki yatayda bulunan siyah ve beyaz kare sayısı
dy	Referans görüntüdeki dikeyde bulunan siyah ve beyaz kare sayısı
ρ	Pikselden cm geçmek için kullanılan katsayı
x_m, y_m	Ağırlığı tahmin edilecek cismin merkez noktası
x_n, y_n	Ağırlığı tahmin edilecek cismin sınır noktaları

KISALTMA LİSTESİ

CCD	Yüklenme İliştirilmiş Araç (Charge Coupled Device)
ÇKA	Çok Katmanlı Algılayıcı (Multi Layer Perceptron)
GDA	Adaptif Öğrenme Oranı İle Gradiyent Azalan Geri Yayılım (Gradient Descent with Adaptive Learning Rate Backpropagation)
GDX	Momentum ile Gradiyent Azalan ve Adaptif Öğrenme Oranlı Geri Yayılım (Gradient Descent with Momentum and Adaptive Learning Rate Backpropagation)
KKÇDY	K-Katmanlı Çapraz Doğrulama Yöntemi
MEUKN	Merkeze En Uzak Kenar Noktası
MEYKN	Merkeze En Yakın Kenar Noktası
RP	Esnek Geri Yayılım (Resilient Back Propagation)
RTF	Radyal Taban Fonksiyonu (Radial Basis Function)
SCG	Kademeli Çekim Eğimi (Scaled Conjugate Gradient)
TKA	Tek Katmanlı Algılayıcı (Single Layer Perceptron)
YSA	Yapay Sinir Ağları

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 Görüntü işleme adımları 7
Şekil 2.2 Projektif dönüşüm..... 9
Şekil 2.3 İki boyutlu örnek öz nitelik uzayı 12
Şekil 2.4 Sinir sitemi blok diyagramı..... 13
Şekil 2.5 Biyolojik sinir hücresi görüntüsü..... 14
Şekil 2.6 Çok katmanlı YSA yapısı 15
Şekil 2.7 Lineer olmayan nöron modeli..... 16
Şekil 2.8 Ön görünün (bias) aktivasyon fonksiyonu girişine etkisi 17
Şekil 2.9 Eğiticili öğrenme blok diyagramı..... 19
Şekil 2.10 Eğiticisiz öğrenme blok diyagramı 19
Şekil 2.11 Takviyeli öğrenme blok diyagramı 20
Şekil 2.12 Tek katmanlı algılayıcı 21
Şekil 2.13 Çok katmanlı algılayıcı 22
Şekil 2.14 Radyal temelli fonksiyon ağı 23
Şekil 2.15 k=3 için k-katlı çapraz doğrulama 24
Şekil 3.1 Veri tabanına veri yazma süreci blok diyagramı 27
Şekil 3.2 Görüntü çekim açıları..... 27
Şekil 3.3 Portakal örneği görüntüleri (a) Dik açılı (b) Pozitif açılı (c) Negatif açılı 30
Şekil 3.4 Yumurta örneği görüntüleri (a) Dik açılı (b) Pozitif açılı (c) Negatif açılı..... 32
Şekil 4.1 Ölçüm düzeneği 33
Şekil 4.2 Tahminle sistemi oluşturma süreçleri blok diyagramı..... 35
Şekil 4.3 Tahminleme süreçleri blok diyagramı..... 36
Şekil 4.4 Kalibrasyon akış diyagramı..... 37
Şekil 4.5 Kalibrasyon için kullanılan checkerboard 37
Şekil 4.6 Kamere kalibrasyon arayüzü..... 38
Şekil 4.7 Reprojeksiyon hatası..... 38
Şekil 4.8 a) Sistemden oluşan reprojeksiyon hatası, b) Harici parametrelerin görseli 39
Şekil 4.9 a) Orijinal Görüntü, b) Kalibrasyon sonucunda elde edilen görüntü..... 39
Şekil 4.10 Cisimlerin sınırlarının belirlendiği örnek görüntü 41
Şekil 4.11 Ağırlık tahminle programının kullanıcı arayüzü 43
Şekil 4.12 Örnek bir sistem çıktısı..... 43
Şekil 4.13 Örnek bir ağırlık tahminleme işlemi sonucu 44
Şekil 5.1 ÇKA “Deneme 19” YSA yapısı..... 48

Şekil 5.2 ÇKA “Deneme 19” regresyon grafikleri (a) Bütün veriler (b) Test verileri (c) Doğrulama verileri (d) Eğitim verileri 48
Şekil 5.3 ÇKA “Deneme 19” sistem performansı 49
Şekil 5.4 ÇKA “Deneme 19” hata histogramı 49
Şekil 5.5 RTF “Deneme 3” YSA yapısı 53
Şekil 5.6 RTF “Deneme 3” regresyon grafikleri (a) Bütün veriler (b) Test verileri (c) Doğrulama verileri (d) Eğitim verileri 54
Şekil 5.7 RTF “Deneme 3” eğitim performansı 54
Şekil 5.8 ÇKA “Deneme 15” YSA yapısı 56
Şekil 5.9 ÇKA “Deneme 15” regresyon grafikleri (a) Bütün veriler (b) Test verileri (c) Doğrulama verileri (d) Eğitim verileri 57
Şekil 5.10 ÇKA “Deneme 15” sistem performansı 58
Şekil 5.11 ÇKA “Deneme 15” hata histogramı 58
Şekil 5.12 RTF “Deneme 9” YSA yapısı 61
Şekil 5.13 RTF “Deneme 9” için eğitim performansı 62
Şekil 5.14 RTF “Deneme 9” regresyon grafikleri (a) Bütün veriler (b) Test verileri . (c) Doğrulama verileri (d) Eğitim verileri 62
Şekil 5.15 RTF “Deneme 9” sistem performansı 63

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1 Öz nitelik vektörlerinin en büyük, en küçük ve ortalama değerleri 30
Çizelge 3.2 A sınıfı yumurtalarda ağırlık sınıfları 31
Çizelge 5.1 ÇKA'da başarımı en yüksek denemelerin parametre değerleri 46
Çizelge 5.2 ÇKA'da başarımı en yüksek denemelerin başarımları 46
Çizelge 5.3 Yumurtaların genel sınıflandırma başarımı 47
Çizelge 5.4 Yüksek ağırlık sınıfına, düşük ağırlıklı yumurtaların karışımına göre sınıflandırma başarımı 47
Çizelge 5.5 "Deneme 19"un veri kümesi oranlarındaki değişimin başarıma etkisi ... 50
Çizelge 5.6 "Deneme 19"un KKÇDY başarımı 50
Çizelge 5.7 RTF'de başarımı en yüksek denemelerin parametre değerleri 51
Çizelge 5.8 RTF'de başarımı en yüksek denemelerin başarımları 51
Çizelge 5.9 Yumurtaların genel sınıflandırma başarımı 51
Çizelge 5.10 Yüksek ağırlık sınıfına, düşük ağırlıklı yumurtaların karışımına göre sınıflandırma başarımı 52
Çizelge 5.11 "Deneme 3"ün veri kümesi oranlarındaki değişimin başarıma etkisi 52
Çizelge 5.12 "Deneme 3"ün KKÇDY başarımı 53
Çizelge 5.13 ÇKA'da başarımı en yüksek denemelerin parametre değerleri 55
Çizelge 5.14 ÇKA'da başarımı en yüksek denemelerin başarımları 56
Çizelge 5.15 "Deneme 15"in veri kümesi oranlarındaki değişimin başarıma etkisi 59
Çizelge 5.16 "Deneme 15"in KKÇDY başarımı 59
Çizelge 5.17 RTF'de başarımı en yüksek denemelerin parametre değerleri 59
Çizelge 5.18 RTF'de başarımı en yüksek denemelerin başarımları 60
Çizelge 5.19 "Deneme 9"un veri kümesi oranlarındaki değişimin başarıma etkisi 61
Çizelge 5.20 "Deneme 9"un KKÇDY başarımı 61

ÖZET

YAPAY SİNİR AĞLARI KULLANARAK GÖRÜNTÜ İŞLEMEYE DAYALI AĞIRLIK TAHMİNİ

Ünsal Burak ŞALVARCI

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Umut Engin AYTEN

Nesnelerin ağırlıklarının doğru ve hızlı bir şekilde tespit edilmesi günümüzde akademik ve endüstriyel birçok uygulamada oldukça önemli bir yere sahiptir. Gerçekleştirilen bu çalışmada yapay sinir ağı (YSA) kullanarak görüntü işlemeye dayalı uzaklıktan ve kamera açısından kısmi derecede bağımsız ağırlık tahminlemesi yapılması hedeflenmektedir. Tahminleme işlemi gıda sektöründe farklı kategorilerde olan yumurta ve portakal örnekleri üzerinde gerçekleştirilmiştir. Oluşturulan sistem; bir adet kamera, yapay aydınlatma sistemi, yansıtıcılar ve referans görüntüden oluşmaktadır. Elde edilen görüntü işlenerek cismin ağırlığı tahmin edilmektedir. Yapay sinir ağı olarak ileri beslemeli çok katmanlı algılayıcı (ÇKA) ve radyal taban fonksiyonu (RTF) denenerek sonuçlar karşılaştırılmıştır. Yumurta üzerinde yapılan çalışmalarda sistemin eğitilmesi ve test edilmesi için 4 farklı markadan 4 farklı sınıfta (xlarge - large - medium - small) 250 adet örnek yumurta kullanılmıştır. Portakal üzerinde yapılan çalışmalarda ise farklı büyüklüklerde 150 adet portakal örneği kullanılmıştır. Farklı öznitelik vektörleri, farklı görüntüler ve farklı yapay sinir ağı parametreleri kullanılarak optimum sistem elde edilmeye çalışılmıştır. Bu değişiklikler ile gerçekleştirilen her bir deneme 5'er kez çalıştırılmış ve sonuçların aritmetik ortalaması alınarak yöntemlerin performansı değerlendirilmiştir. Yumurta örneği için 31 farklı deneme arasındaki optimum sonuçta; ÇKA için korelasyon katsayısı (R) 0.9942 ve ortalama mutlak hata (mae) 0.80gr olarak, RTF için korelasyon katsayısı (R) 0.9943 ve ortalama mutlak hata (mae) 0.81gr olarak hesaplanmıştır. Ayrıca Türk Gıda Kodeksi Yumurta Tebliği'ne göre belirlenmiş ve yumurta kutularının üzerinde yazan sınıflandırma değerlerine göre doğruluk oranı %47

iken, gerçekleştirilen bu çalışma sonucunda bu oran ÇKA'da %90.5'e, RTF'de %86.3'e çıkarılmıştır. Portakal örneği için 32 farklı deneme arasındaki optimum sonuçta; ÇKA için korelasyon katsayısı (R) 0.9820 ve ortalama mutlak hata (mae) 2.58gr olarak, RTF için korelasyon katsayısı (R) 0.9825 ve ortalama mutlak hata (mae) 2.42gr olarak hesaplanmıştır. Tüm işlemler Matlab programı ve araç kutuları kullanılarak gerçekleştirilmiştir.

Ayrıca bütün bu çalışmalar sırasında elde edilen görüntüler ileriki çalışmalara yardımcı olması adına paylaşılmaktadır. Yumurta için dik açılı, pozitif açılı ve negatif açılı olmak üzere 750 adet görüntü, portakal için dik açılı, pozitif açılı ve negatif açılı olmak üzere 450 adet görüntü içeren bir veri tabanı oluşturulmuştur.

Anahtar Kelimeler: Yapay sinir ağı, görüntü işleme, ağırlık tahmini, çok katmanlı algılayıcı, radyal taban fonksiyonu, yumurta, portakal, uzaklıktan bağımsız sistem.

ABSTRACT

WEIGHT ESTIMATION BASED ON IMAGE PROCESSING USING ARTIFICIAL NEURAL NETWORK

Ünsal Burak ŞALVARCI

Department of Electronics and Communications Engineering

MSc. Thesis

Adviser: Assist. Prof. Dr. Umut Engin AYTEN

Accurately and fast detection of weight of objects has an important place for lots of academic and industrial application at the present time. In this work, it was aimed to weight estimation in a partially distance and camera angle independent manner based on image processing using artificial neural networks (ANN). Estimation was realized with egg and orange samples which is different categories of food sector. The constituted system consists of a camera, artificial lighting system, reflector and reference image. Object weight was estimated by processing the acquired image. Feed forward multilayer perceptron (MLP) and radial basis function (RBF) was compared as artificial neural network with experiment. 250 samples which consist of 4 different brands and 4 different classes (x-large, large, medium, and small) were used for testing and training the system on egg study. 150 orange samples which were different sizes were used as to orange study. It was tried to be obtained optimum system by using different feature vectors, different images and different artificial neural network parameters. Each tries which was realized with changes was run 5 times and then system performance was evaluated by getting arithmetic mean. Performance of optimum results between 31 methods was calculated for MLP as correlation coefficient (R) was 0.9942 and mean absolute error (mae) was 0.80gr and calculated for RBF as correlation coefficient (R) was 0.9943 and mean absolute error (mae) was 0.81gr for egg samples. As accurate classification rate had been %47 according to defined by Turkish food codex egg communique and classification values written on

the egg cartoon, this rate was increased to %90.5 for MLP and increased to %86.3 for RBF as a result of this work. Performance of optimum results between 32 methods was calculated for MLP as correlation coefficient (R) was 0.9820 and mean absolute error (mae) was 2.58gr and calculated for RBF as correlation coefficient (R) was 0.9825 and mean absolute error (mae) was 2.42gr for orange samples. All processes were realized by using Matlab program and toolboxes.

In addition, all images which were obtained during these works have been shared with the intent of assisting next works. A database which contains right angled, positive angled and negative angled 750 egg images and right angled, positive angled and negative 450 orange images was created.

Keywords: Artificial neural network, image processing, weight estimation, multilayer perceptron, radial basis function, egg, orange, distance independent system.

BÖLÜM 1

GİRİŞ

Cisimlerin ağırlıklarının hesaplanması endüstriyel ve akademik birçok çalışmada oldukça önemli bir yere sahiptir. Yumurta, besleyici değeri yüksek olan ve insan vücudunun gereksinim duyduğu hemen hemen bütün besin maddelerini içeren bir yiyecektir. Ülkemizde ve dünyada sevilerek tüketilen temel bir gıda maddesidir ve biyolojik değeri yüksek proteine sahip olması nedeniyle insan hayatında önemli bir yere sahiptir [1]. Bu nedenle yumurta üretimi ve tüketimi oldukça fazladır.

Yumurtaların ağırlığının hesaplanması iki açıdan önem taşımaktadır. İlk olarak, yapılan araştırmalar yumurtaların ağırlıklarının kuluçka performansı ile doğrudan ilişkili olduğunu ortaya koymuştur [2], [3]. İkinci olarak da, yumurtaların ağırlıklarının doğru hesaplanıp doğru sınıflandırılması tüketicide oluşacak kalite algısını arttıracak ve üretici markaya değer katacaktır. Çünkü yapılan araştırmalar geleneksel tüketicilerin yumurtanın ağırlığına dikkat ettiğini göstermektedir [4]. Yani yumurtanın ağırlığına göre, olması gerekenin altında sınıflandırılması üretici firmaya ekstra bir maliyet oluşturmakta, daha üzerinde bir sınıflandırma gerçekleştirilmesi ise müşteri memnuniyetsizliğine yol açmaktadır.

Portakallar ise farklı ağırlıklarda ve boyutta rahat bir şekilde örneklenebildiği için seçilmiştir. Portakalın ağırlığının hesaplanması sürecinde kullanılan yöntem ile diğer meyvelerin ve sebzelerin ağırlıklarının hesaplanmasında kullanılacak olan yöntem benzerlik göstermektedir. Bu nedenle aynı çalışma, ağırlığı hesaplanacak yeni meyve ve sebzeler için tekrar gerçekleştirilerek diğer meyve ve sebzeler için de ağırlık tahmini yapılabilmektedir.

1.1 Literatür Özeti

Literatürde ağırlık tahminleme üzerine birçok çalışma gerçekleştirilmiştir. Bu çalışmalar genellikle merkez noktasına göre simetrik olan cisimler üzerinde ve sabit kameralar ile gerçekleştirilmiştir. Bunun yanında sabit olmayan kameralar ve hareketli cisimler ile de gerçekleştirilmiş çalışmalar bulunmaktadır. Uzaktan, kamera ile ağırlık tahminleme ile özellikle ağırlık ölçümü sırasında oluşan zaman tüketimi en az düzeye indirilmekte, mekanik sistemler kullanılmadığı için maliyet azaltılmakta ayrıca ağırlık ölçümü sırasında cisimlere zarar verilmesi gibi sorunlar ortadan kaldırılmaktadır.

Lines J. A. ve arkadaşları tarafından 2000 yılında gerçekleştirilen bir çalışmada, yüzen balık kütleleri, görüntü tabanlı sistem ile otomatik olarak tahmin edilmiştir. Bu çalışma 1997 yılında gene aynı ekipte bulunan Hockaday ve arkadaşları tarafından gerçekleştirilen stereo görüntü çiftleri kullanılarak Atlantik somon türlerinin kütlelerinin ölçülmesi üzerine yapılan çalışmayla benzerlik göstermektedir. Bilgisayara direk olarak bağlı 2 kamera ile kameralardan yaklaşık 1-2 metre uzaklıkta olan somon balıklarının görüntüleri elde edilmiş ve bu görüntüler yardımıyla tahminleme işlemi gerçekleştirilmiştir. 17 balık ve 60 görüntü kullanılarak yapılan bu çalışmada, Beddow ve Ross tarafından 1996 yılında geliştirilen, Altlantik somon balıklarının morfolojik boyutlarından kütle tahmini yapılmasını sağlayan lineer regresyon denklemleri kullanılmıştır. Bu çalışma sonucunda ortalama kütle ölçüm hatası %18 olarak hesaplanmıştır [5], [6].

M. Omid ve arkadaşları tarafından 2009 yılında gerçekleştirilen bir çalışmada görüntü işleme tekniğiyle turunçgillerin hacimlerinin ve ağırlıkları tahmin edilmiştir. Bu çalışma portakal, limon, misket limon ve mandalina üzerinde gerçekleştirilmiş ve her bir meyve için 50 adet görüntü örneği kullanılmıştır. Çalışmada öncelikli olarak meyveler su dolu bir kaba daldırılarak, taşan su miktarına göre cismin hacmi hesaplanmaktadır. Ayrıca hassas tartı ile de meyvelerin ağırlıkları ölçülmektedir. Kurulan sistemden elde edilen değerler ile ölçülen değerler karşılaştırılarak sistemin başarımı hesaplanmaktadır. Sistem Visual Basic dili kullanılarak hazırlanmıştır [7].

Determinasyon katsayısı, regresyon analizlerinde tahmin başarımının ne kadar güçlü olduğunu gösteren bir parametredir ve korelasyon katsayısının karesi olarak

tanımlanmaktadır. Korelasyon katsayısı ise iki veya daha fazla değişken arasındaki ilişkiyi yönüyle birlikte gösteren sayısal bir değerdir.

Bu çalışma sonucunda determinasyon katsayısı; limonda %96.2, misket limonda %97, portakalda %98.5 ve mandalinada ise %96 olarak hesaplanmıştır [7].

V. Asadi ve M.H. Raoufat tarafından 2010 yılında, YSA'ya ve görüntü işlemeye dayalı yumurta ağırlıklarının tahmin edildiği bir durum çalışması gerçekleştirilmiştir. Bu çalışmada CCD kamera kullanılmış ayrıca aydınlatma sistemi ve bir ayna yardımıyla yumurtanın tek bir kamera ile farklı açılardan görüntüsünün elde edilmesi sağlanmıştır. Çok katmanlı algılayıcı (ÇKA) kullanılarak gerçekleştirilmiş olan bu çalışmada, ortalama yumurta ağırlığı 60gr'dır. 12 öz nitelik vektörü 3 grup şeklinde kategorize edilerek ÇKA ağının girişine uygulanmıştır. Öğrenme algoritması olarak çok katmanlı algılayıcı – momentum ile gradiyent azalan ve adaptif öğrenme oranlı geri yayılım (MLP-GDX), çok katmanlı algılayıcı – esnek geri yayılım (MLP-RP) ve çok katmanlı algılayıcı – kademeli çekim eğimi (MLP-SCG) denenmiştir. Yapılan çalışma sonucunda en başarılı öğrenme algoritması MLP-SCG bulunmuştur. Bu öğrenme algoritması için korelasyon katsayısı (R) %96 olarak, maksimum hata ise 2.3 gr olarak hesaplanmıştır [8].

P. Javadikia ve arkadaşları tarafından 2011 yılında, ağ tabanlı bulanık çıkarım sistemi (ANFIS) kullanılarak yumurta ağırlığını tahmin eden bir çalışma gerçekleştirilmiştir. Bu çalışmada ticari olarak paketlenmiş 200 adet yumurta kullanılmıştır. CCD kamera ve aydınlatma sistemi içeren ölçüm düzeneği yardımıyla görüntüler elde edilmiştir. Bu görüntüler Matlab programı üzerinde incelenerek yumurta ağırlıkları tahmin edilmiştir. Gerçekleştirilmiş olan bu çalışma sonucunda korelasyon katsayısı %99.42 olarak hesaplanmıştır [9].

M. Kashiha ve arkadaşları tarafından 2014 yılında, 2011 yılında gerçekleştirdikleri görüntü işleme ile domuzların üstten görüntüleri kullanılarak ağırlıklarının tahmin edildiği bir çalışmanın bildirisini yayınlamışlardır. Bu çalışma Agrivet araştırma çiftliğinde gerçekleştirilmiştir. Aynı ölçüm düzeneği kullanılarak 2 farklı deney yapılmış bu deneylerin her birinde 156 saat görüntü kaydedilmiştir. Bu görüntüler anlık olarak işlenmemiş, kayıt edilen bu görüntüler daha sonra Matlab programı üzerinde incelenerek, öz nitelik vektörleri elde edilmiştir. Ağırlık tahminleme işlemi transfer

fonksiyonu modeli kullanılarak gerçekleştirilmiş olup, %97.5'lik bir başarım elde edilmiştir [10].

D. Alikhanov ve arkadaşları tarafından 2015 yılında, regresyon analizi kullanarak görüntü işleme tekniğiyle yumurta ağırlıklarının ölçülmesi üzerine bir çalışma yapılmıştır. Bu çalışmada aydınlatma sistemi ve CCD kamera kullanılarak oluşturulan ölçüm düzeneğinden elde edilen görüntüler, Matlab programı kullanılarak işlenmiştir. Öz nitelik vektörleri olarak; yumurtanın çevresel uzunluğu, alanı, hacmi, en uzun ve en kısa ekseni ve şekil katsayısı kullanılmıştır. Bu çalışma sonucunda en önemli öz nitelik vektörlerinin yumurtanın alanı ve hacmi olduğu görülmüş ve korelasyon katsayısı %94.39 olarak hesaplanmıştır [11].

Chaithanya ve Priya tarafından 2015 yılında gerçekleştirilen 2 boyutlu resimlerdeki cisimlerin ağırlıklarının tahmin edilmesi ile ilgili yapmış oldukları çalışmada, ağırlıkların gerçek zamanlı olarak tahmin edilmesi hedeflenmiştir. Bu çalışmada kamera, başparmak ile bir kez kalibre edilmektedir. Daha sonra görüntüdeki öz nitelikler elde edilerek, cismin ağırlığı tahmin edilmektedir. İlgili cismin 2 boyutlu görüntüsünden elde edilen tahmini hacmi ile daha önceden veri tabanına kayıt edilmiş öz kütlesi çarpılarak cismin kütlesine, kütle ile de yer çekimi ivmesi çarpılarak cismin tahmini ağırlığına ulaşılmaktadır. Bu çalışma; elma, ekmek, portakal, yumurta ve salatalık için gerçekleştirilmiş ve yaklaşık olarak %97 oranında bir doğruluk elde edilmiştir [12].

K. Sabancı, S. Ekinci ve arkadaşları tarafından 2016 yılında Matlab programı kullanılarak, görüntü işleme tekniğiyle buğday ağırlıklarının tahmin edilmesi üzerine bir çalışma yapılmıştır. Daha önce elde edilen belirli miktardaki buğday setlerinin ağırlıklarına göre tahminleme işlemi gerçekleştirilmektedir. Makarna ve ekmek buğdayları için gerçekleştirilmiş bu çalışmada 55 parçalı makarna buğdayında yaklaşık %98 oranında, 57 parçalı ekmek buğdayında yaklaşık %97 oranında başarım elde edilmiştir [13].

1.2 Tezin Amacı

Bu tez çalışmasının amacı; görüntü işlemeye dayalı, kamera dışında başka bir sensör kullanmadan ağırlık tahminleme işleminin yüksek doğruluk oranıyla

gerçekleştirilmesidir. Ağırlık tahminleme işlemi için cisim olarak yumurta ve portakal seçilmiştir. YSA kullanarak, kısmi derecede uzaklıktan ve kamera açısından bağımsız ağırlık tahmini yapan, portakal örneği için ağırlık tahminleme başarımının, yumurta örneği için ise hem ağırlık tahminleme hem de bu ağırlık tahminlerine göre ağırlık sınıflarının belirlenme başarımının yüksek olduğu bir sistemin kurulması amaçlanmaktadır.

1.3 Hipotez

Yumurta ve portakal örneklerinin görüntüleri, oluşturulan ölçüm düzeneği yardımıyla elde edilecektir. Bu görüntüler, görüntülerin belirlenmiş olan öz nitelik vektörleri ve hassas tartı ile yapılan ağırlık ölçüm değerleri bir veri tabanına yazılacaktır. Daha sonra farklı YSA parametreleri, YSA modelleri (ÇKA ve RTF) ve öz nitelik vektörleri kullanılarak, kurulacak olan ağırlık tahminleme sisteminin performansı değerlendirilecektir. Ayrıca başarımı en yüksek olan sistem k-katmanlı çapraz doğrulama yöntemi (KKÇDY) ile de değerlendirilecektir.

Yumurta örneği için, sistemin yumurta ağırlıklarını doğru tahmin edebilme ve yumurtaları doğru sınıflandırma performansına bakılacaktır. Portakal örneği için ise sadece ağırlıkların doğru olarak tahmin edilmesi baz alınarak sistem başarımı belirlenecektir. Sonuç olarak, bu tez çalışmasında görüntü işleme ve yapay sinir ağlarını ortak kullanan, başarımı yüksek, uzaklıktan ve kamera açısından kısmi derecede bağımsız, ağırlık tahminleme sistemi kurulması hedeflenmiştir.

Bu çalışmanın özgün değerini; yumurta ve portakal örneklerinin YSA ile ağırlıklarının tahminlenmesine ek olarak dama tahtası görüntüsü de kullanarak, uzaklıktan ve kamera açısından belirli sınırlar dahilinde bağımsız çalışan sistem ile cisimlerin ağırlıklarının ölçülmesi sırasında oluşacak kullanıcı kaynaklı hataların minimize edilmesi, oluşturmaktadır.

BÖLÜM 2

Genel Bilgiler

2.1 Görüntü İşleme

Görüntü işleme iki temel alan için uygulanmaya başlamıştır. Bunlardan ilki resimdeki bilgilerin insanlar tarafından daha gelişmiş bir şekilde algılanabilmesini sağlamak diğeri ise makineler tarafından görüntünün saklanmasını, iletilmesini ve gösterilmesini sağlamaktır.

Bir resim iki boyutlu bir fonksiyon olarak, f(X,Y) şeklinde tanımlanabilir. Burada X ve Y uzay koordinatlarıdır ve bu fonksiyonun X ve Y noktalarındaki genliği görüntünün o noktalardaki yoğunluğu veya gri seviyesi olarak tanımlanır. Bu parametrelerin sonlu ve ayrık olması durumunda, görüntü "dijital görüntü" olarak adlandırılır. Dijital görüntü işleme ise dijital bilgisayarlar yardımıyla dijital görüntülerin amaca uygun olarak değiştirilmesi olarak tanımlanabilir.

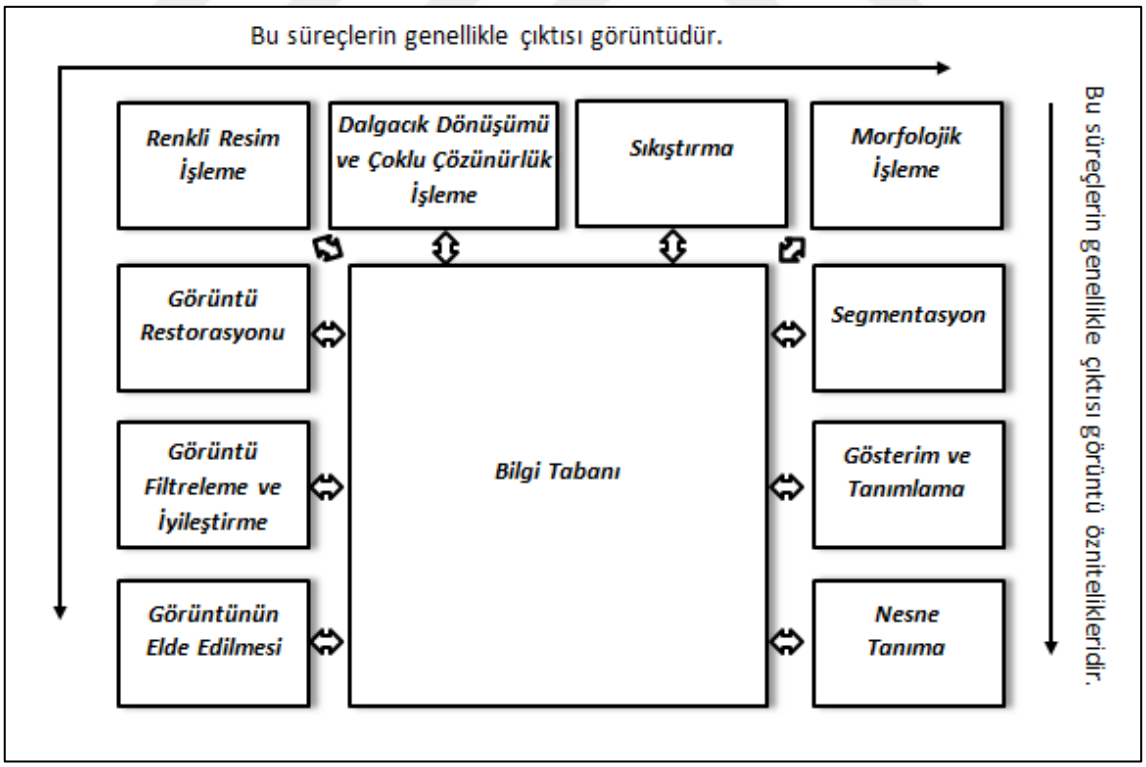
Görüntüler dijital ve analog olmak üzere iki farklı formatta olabilirler. Analog sinyaller zamanın her anı için tanımlı sürekli fonksiyonlardan oluşurken, dijital görüntüler kesikli sinyallerden oluşurlar. Analog görüntülerin analizi yapılırken genellikle süreç manuel olarak işlemektedir. Bir yorumlayıcı tarafından yapılan bu manuel süreç ile hedef belirlenmekte; böylece bu yöntem, yorumlayıcı kaynaklı hataların oluşmasına oldukça açık bir hale gelmektedir. Çünkü bilgisayarlar tarafından sürekli fonksiyonlar işlenememektedir ve sayısal bilgisayarlar tarafından işlenebilmesi için analog görüntülerin sayısal hale dönüştürülmesi gerekmektedir. Dijital görüntüler ise parlaklık derecesine göre sayısal değer alan piksellerden oluşmuş yapılardır. Sayısal bilgisayarlar

ile bu pikseller üzerinde sistematik değişiklikler yapılmakta, böylece dijital görüntülerin işlenmesi ve analiz edilebilmesi sağlanmaktadır.

İnsanlar sadece elektromanyetik spektrumda görünür bölgedeki dalgaları algılayabilme yeteneğine sahiptir fakat görüntüleme cihazları gama ışınlarından radyo dalgalarına kadar hemen hemen bütün elektromanyetik spektrumu kapsayacak şekilde algılama kapasitesine sahiptir. Bundan dolayı görüntüleme cihazları çok geniş ve çeşitli kullanım alanlarına sahiptir [14], [15].

2.2 Görüntü İşleme Süreçleri

Görüntü işleme süreci; girdi, görüntünün işlenmesi ve çıktı aşamalarından oluşmaktadır. Burada görüntü işleme sonucunda elde edilecek çıktı, görüntü olabileceği gibi görüntünün öz nitelikleri de olabilmektedir. Şekil 2.1'de temel olarak kullanılan görüntü işleme adımları gösterilmektedir. İlgili şekilde gösterilen adımların hepsinin uygulanma zorunluluğu yoktur. Farklı amaç ve hedefler için ihtiyaç duyulan adımlar kullanılmaktadır [14].



Şekil 2.1 Görüntü işleme adımları [14]

2.2.1 Görüntünün Elde Edilmesi

Görüntünün elde edilmesi, görüntü işleme adımlarının ilkidir. Adından da anlaşılacağı gibi görüntünün alınarak sisteme uygulanması işlemidir. Genellikle görüntünün elde edilmesi işlemi, ölçeklendirme gibi ön işleme aşamasını içermektedir [14].

2.2.1.1 Kameranın Kalibre Edilmesi

Görüntü işleme uygulamalarında dijital kameralar yaygın olarak kullanılmaktadır. Ölçümlerin hassas olarak yapılabilmesi için dijital kameraların kalibre edilmesi gerekmektedir. Kamera kalibrasyonu, kamera tarafından ölçülen değer ile gerçekte olması gereken değer arasındaki ilişkiyi belirleme ve ölçülen değeri gerçek değere en yakın hale getirme işlemidir. Kamera kalibrasyonu aşamasında ilk olarak kamera parametreleri belirlenmektedir. Burada uzay koordinatları (X,Y,Z) ile görüntü koordinatları (u,v) arasındaki ilişki modellenerek kamera parametreleri hesaplanmaktadır [16].

Fiziksel kamera parametreleri genellikle dahili ve harici parametreler olarak ikiye ayrılmıştır. Harici parametrede, kameranın belirli dünya koordinat sistemindeki 3-boyutlu pozisyonu belirlenmektedir. Kamera merkezli koordinat düzlemine göre nesnelerin koordinatları 3-boyutlu olarak dönüştürülmektedir. Dahili kamera parametreleri ise odak uzaklığı, ölçek çarpanı ve bakış noktasından oluşmaktadır [17].

2.2.2 Görüntü Filtreleme ve İyileştirme

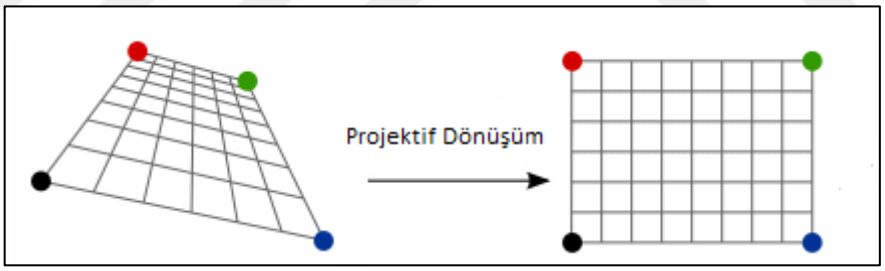
Görüntünün filtrelenmesi ve iyileştirilmesi, görüntünün gerçekleştirilecek olan uygulamaya uygun olarak değiştirilerek görüntünün ilgili uygulamaya daha uygun bir hale getirilmesi işlemidir. Burada görüntünün iyileştirmesi problem odaklı bir çözüm gerektirmektedir. Bir problemin çözümüne uygun olan iyileştirme diğer bir problemin çözümü için uygun olmayabilir. Bu nedenle bütün problemlerin çözümü için kullanılacak genel bir görüntü iyileştirme metodu bulunmamaktadır [14].

2.2.3 Görüntü Restorasyonu

Görüntünün restorasyonu, görüntünün iyileştirilmesinden farklı bir süreçtir. Görüntünün iyileştirilmesi problemden probleme farklılık gösterirken, görüntünün restore edilmesi her problem için aynıdır. Görüntünün restore edilmesi matematiksel ve olasılıksal metotlar ile görüntü üzerindeki bozulmaların düzeltilmesi işlemini ifade etmektedir [14].

2.2.3.1 Projektif Dönüşüm

Bir düzlemden diğer düzeleme yapılan izdüşümler, projektif dönüşüm olarak tanımlanmaktadır. Buradaki düzlemler kesişebileceği gibi aynı zamanda birbirine paralel de olabilmektedir, fakat özellikle düzlemlerin birbirine paralel olmadığı nesnelerin şekil ve büyüklüklerinin görüntü işlemeye dayalı olarak belirlenmesi aşamasında yaygın olarak kullanılmaktadır. En basit ve anlaşılır şekilde, elde edilen görüntüyü tam karşıdan çekilmiş gibi dönüştürme işlemidir. Bu işlem daha önceden bilinen noktalar kullanılarak yapılmaktadır. Örnek bir projektif dönüşüm uygulaması Şekil 2.2'de gösterilmiştir [18].



Şekil 2.2 Projektif dönüşüm [19]

Görüntünün düşeye çevrilme işleminde 8 parametreli projektif dönüşüm uygulanmaktadır. Burada her iki koordinat düzleminden elde edilen minimum 4 noktaya ihtiyaç duyulmakta ve her bir nokta için iki eşitlik yazılmaktadır. Bu eşitlikler (2.1) ve (2.2)'de gösterilmiştir.

$$x'' = \frac{a.x'+by'+c}{g'.x'+h.y'+1} \tag{2.1}$$

$$y'' = \frac{d.x'+ey'+f}{g'.x'+h.y'+1} \tag{2.2}$$

Burada x'' ve y'' cismin koordinatları, x' ve y' ise görüntünün koordinatlarıdır. Diğer parametreler a, b, c, d, e, f, g, h ise bilinmeyen parametrelerdir. 8 tane bilinmeyen parametre olmasından dolayı 8 denklem ile bu parametreler hesaplanabilmektedir [20].

2.2.4 Renkli Resim İşleme

Renkli resimlerin işlenmesi işlemidir. İnternetin yaygın olarak kullanılmaya başlanmasından dolayı renkli resimlerin işlenmesi ve görüntü özelliklerinin çıkarılması önemli bir hal almıştır. Renkli resimler işlenirken iki grupta ele alınmaktadır. Bunlardan ilki tam renkli resimlerdir. Tam renkli resimler, renkli bir algılayıcı tarafından elde edilen görüntülerdir. Diğeri ise sahte renkli görüntülerdir. Sahte renkli görüntüler ise, gri tonlu görüntülerin daha sonra bilgisayar ortamında renklendirilmesiyle oluşmaktadır [14].

2.2.5 Dalgacık Dönüşümü ve Çoklu Çözünürlük İşleme

Dalgacık dönüşümü, görüntünün farklı derecelerdeki çözünürlüklerle gösterilmesi işleminin temelini oluşturmaktadır. Görüntünün sıkıştırılması ve art arda küçük parçalara ayrılması işleminde kullanılmaktadır [14].

2.2.6 Sıkıştırma

Sıkıştırma işlemi, görüntünün saklanması veya iletilmesi esnasında boyutunun düşürülerek daha az yer kaplamasını amaçlayan bir işlemdir [14].

2.2.7 Morfolojik İşleme

Morfolojik işlemler, görüntüdeki elemanların biçimsel yapılarıyla ilgilenmektedir. Uygulanan çeşitli görüntü işleme yöntemleriyle görüntü, şekilsel bazda ele alınmaktadır. Morfolojik işleme ile birlikte görüntüdeki şekilsel elemanların gösterilmesi ve tanımlanması sağlanmaktadır [14].

2.2.8 Segmentasyon

Segmentasyon işlemi, görüntünün anlamlı olacak şekilde parçalara veya nesnelere bölünmesi işlemidir. Örneğin, bir doğa fotoğrafından ağaçlar segmente edilebileceği

gibi başka bir uygulamada nehir segmente edilebilir. Bu nedenle uygulamaya bağlı olarak segmentasyon işlemi farklılık göstermektedir. Dijital görüntü işleme uygulamalarında görüntüdeki segmantasyonun otomatik olarak yapılması oldukça zor bir çalışmadır [14].

2.2.9 Gösterim ve Tanımlama

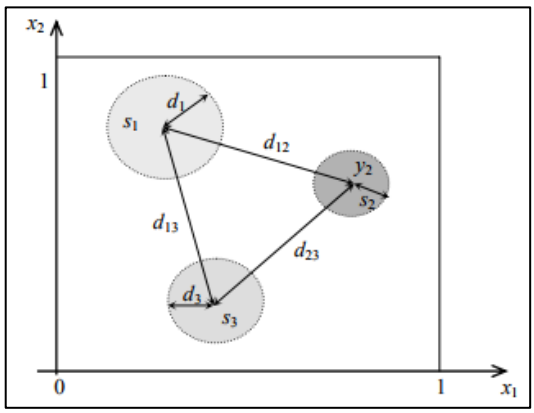
Gösterme ve tanımlama işlemi genellikle segmentasyon işleminden sonra gerçekleştirilmektedir. Segmentasyon işleminden sonra ilgili nesnenin sınırlarını veya tamamını göstermektedir. Nesnenin sınırlarının gösterilmesi genellikle cismin köşe noktalarına veya kıvrılma noktalarına odaklanılan çalışmalarda kullanılmaktadır. Doku veya iskeletsel yapıya odaklanılan çalışmalarda ise genellikle nesnenin tamamı gösterilmektedir. Bazı durumlarda bu iki gösterim de kullanılabilmektedir. Bu iki gösterimde de bilgisayar tarafından görüntünün işlenebilmesi için uygun bir formata dönüştürülmesi gerekmektedir. Tanımlama işleminde ise, nesneyi diğer nesnelerden ayıran öz niteliklerinin çıkartılması sağlanmaktadır [14].

2.2.9.1 Öz Nitelik Vektörlerinin Çıkartılması

Öznitelik vektörleri, belirli bir nesnenin kendine özgü karakteristik özelliklerini içermektedir. Bu nedenle öznitelik vektörlerinin belirlenmesi sistemin performansı açısından çok büyük önem taşımaktadır. Genel olarak öznitelik vektörlerinin tespitinde matematiksel metotlar kullanılamamaktadır. Bu nedenle öznitelik vektörleri, tasarımcının tecrübesi ve içgüdüsel olarak ön gördüğü parametreler üzerinden belirlenmektedir [21].

İki boyutlu, 3 farklı kategoriden (s1,s2 ve s3) oluşmuş öz nitelik uzayı Şekil 2.3' de gösterilmiştir. İlgili şekil incelendiğinde aynı kategoride olan vektörlerin birbirine yakın olduğu, farklı kategorilerde olan vektörlerin ise birbirinden uzakta olduğu görülmektedir. Öz nitelik vektörlerinin çıkartılması sonrasında d1,d2 ve d3 değerlerinin oldukça küçük olması; d_{12}, d_{13} ve d_{23} değerlerinin ise oldukça büyük olması istenmektedir. Bunun nedeni; d1,d2 ve d3 değerlerinin küçülmesi aynı kategorideki nesnelerin birbirlerine olan benzer karakteristiklerinin arttığını; d_{12}, d_{13} ve d_{23}

değerlerinin fazla olması ise farklı kategorideki nesnelerin benzer karakteristiklerinin azaldığını göstermektedir [22].



Şekil 2.3 İki boyutlu örnek öz nitelik uzayı [22]

2.2.10 Nesne Tanıma

Nesne tanıma işlemi, belirlenen öz niteliklere göre nesneyi etiketleme işlemidir. Bu tanımlayıcı öz nitelikler sayesinde cisimler tanımlanmaktadır. Örneğin bir görüntüdeki cisimler araba ve gemi olarak etiketlenerek ayırt edilebilmektedir [14].

2.3 Yapay Zeka ve Makine Öğrenmesi

Yapay zeka uzun yıllar boyunca birçok araştırmacı tarafından üzerinde durulan bir konudur. Yapay zeka ve makine öğrenmesi, insanların doğayı taklit etme çabalarından biridir. Yapay zeka ve makine öğrenmesi ile canlılar tarafından gerçekleştirilmekte olan öğrenme ve karar verme işlemlerinin bilgisayar yardımıyla gerçekleştirilmesi hedeflenmektedir.

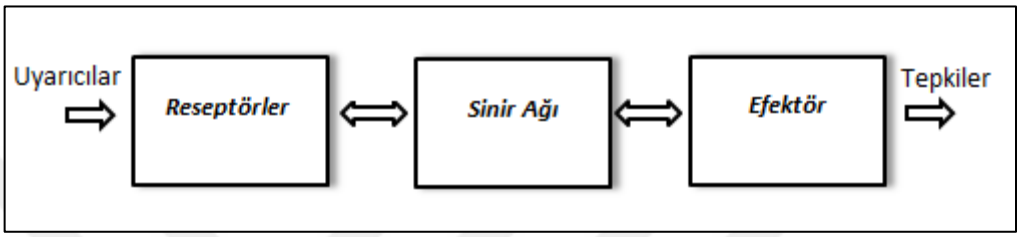
Bu konu üzerinde yapılmış olan çalışmalar sayesinde yapay zeka, ticari sistemler de dahil olmak üzere birçok alanda yaygın olarak kullanılmaktadır. Ayrıca yapay zekanın fonksiyonelliği de gün geçtikçe artmaktadır.

Bilgisayarların, insan beyninin en temel özelliklerinden biri olan öğrenme süreçlerinde YSA'lar yaygın olarak kullanılmaktadır [23].

2.4 Yapay Sinir Ağları

Yapay sinir ağları temel olarak verilen örnekler ile eğitilen, daha sonra önceden tanımlanmamış girdilere eğitim verilerinden öğrendiği bilgiler ile karar vererek bir çıktı üreten sistemlerdir [23].

Yapay sinir ağları biyolojik sinir sistemlerini taklit ederek oluşturulmuştur. Biyolojik sinir sistemleri 3 aşamadan meydana gelmiştir ve Şekil 2.4'de biyolojik sinir sistemlerinin blok diyagramı gösterilmiştir.



Şekil 2.4 Sinir sitemi blok diyagramı

Herhangi bir şekilde reseptörlere gelen uyarılar elektriksel sinyaller olarak sinir ağına iletilmektedir. Sinir ağında verilen karar ise efektöre iletilerek organizmanın gerekli tepkiyi vermesi sağlanmaktadır. Aynı şekilde YSA'lar da benzer şekilde çalışmaktadır. YSA'lar tarafından da bir bilgi alınmakta, yorumlanmakta ve yoruma uygun bir çıktı üretilmektedir [24].

YSA'ların bu kadar talep görmesinin altında hatalara karşı toleransının oldukça yüksek olması, eksik bilgi olması durumunda öğrendiği bilgiler yardımıyla bu eksik bilgileri tamamlayabilmesi, problemlere karşı uyarlanabilirliği gibi önemli özellikleri yer almaktadır.

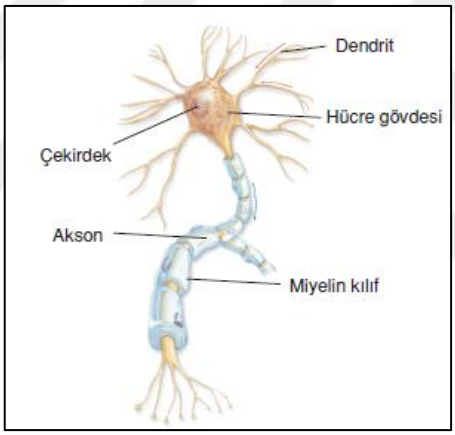
YSA'lar oluşturulurken bütün problemlerin çözümü için tek bir modelin oluşturulması mümkün değildir. Farklı problemler için farklı tasarımların yapılması gerekmektedir [23].

YSA'lar tahminleme, sınıflandırma, verilerin ilişkilendirilmesi, verilerin yorumlanması ve verilerin filtrelenmesi için kullanılabilmektedir. Tahminleme işlemi, YSA'nın eğitim verileri ile eğitilmesinden sonra uygulanan giriş için tahmini bir değer üretmesidir. Cisimlerin ağırlıklarının tahmin edilmesi YSA'ların tahminleme uygulamalarına örnek olarak verilebilir. Sınıflandırma işlemi, girdilerin belirli karakteristik özelliklerine göre

sınıflandırılmasıdır. Verilerin ilişkilendirilmesi, verilerin sınıflandırılması işlemi ile oldukça benzer bir yapıya sahiptir. Verilerin sınıflandırılması işleminden farklı olarak verilerde oluşan hatalar da tespit edilebilmektedir. Verilerin yorumlanması, yaygın olarak veri madenciliğinde kullanılmaktadır. Tam olarak dataların tanımlamalarının yapılmadığı veya data ilişkililerinin bilinmediği durumlarda anlamlı gruplar şeklinde dataların kümelenmesi işlemidir. Dataların filtrelenmesi ise, YSA'lar kullanılarak istenmeyen giriş verilerinin filtrelenmesini sağlamaktadır [25].

2.4.1 Yapay Sinir Ağlarının Yapısı

Daha önce de belirtildiği gibi yapay sinir ağları biyolojik sinir hücrelerinin taklit edilmesiyle oluşturulan sistemlerdir. Örnek bir biyolojik sinir görüntüsü Şekil 2.5'te gösterilmiştir.

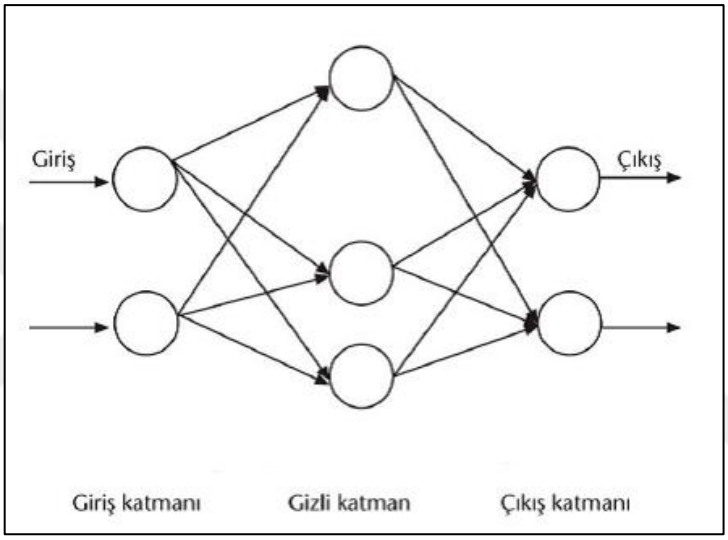


Şekil 2.5 Biyolojik sinir hücresi görüntüsü [26]

Biyolojik sinir hücreleri çekirdek, dendrit ve akson olmak üzere 3 ana kısımdan oluşmaktadır. Dendrit diğer hücrelerden gelen sinyali alarak aksona iletmekle görevlidir. Yapısal olarak bir ağacın dallarını andırmaktadır. Akson ise, her hücrede bir tanedir ve dendrit vasıtasıyla gelen sinyallerin diğer hücrelere iletilmesi görevini üstlenmektedir. Yapay sinir hücreleri de biyolojik sinir hücreleri gibi çevreden gelen bilgiyi bir toplama fonksiyonu ile alıp aktivasyon fonksiyonundan geçirerek bir çıktı üretmektedir. Bu hücrelerde farklı toplama ve aktivasyon fonksiyonu kullanılabilmektedir. Yapay sinir hücreleri yani nöronlar bir araya gelerek YSA'ları oluşturmaktadır. Bu nöronlar birbirlerine paralel işlemler yapabilmekte ve 3 katman halinde YSA'larını oluşturmaktadır. Bu katmanlar;

1. Girdi Katmanı
2. Gizli Katman
3. Çıktı Katmanı olarak sıralanabilir.

Çok katmanlı YSA'nın yapısı Şekil 2.6'da gösterilmiştir. Çevreden gelen verilerin ağa iletimi girdi katmanı ile gerçekleştirilmektedir. YSA'ya uygulanan giriş sayısı kadar girdi katmanında nöron bulunmaktadır. Nöronlar birbirlerine ağırlık değerleri ile bağlıdır ve ağırlık değerlerinin değiştirilmesi ile ağ eğitilmektedir. İstenen çıktı, beklenen çıktıya yaklaşana kadar bu ağırlık değerleri sürekli olarak güncellenmektedir [23], [24].



Şekil 2.6 Çok katmanlı YSA yapısı [27]

Bir YSA'nın 3 temel karakteristik özelliği bulunmaktadır. Bunlar;

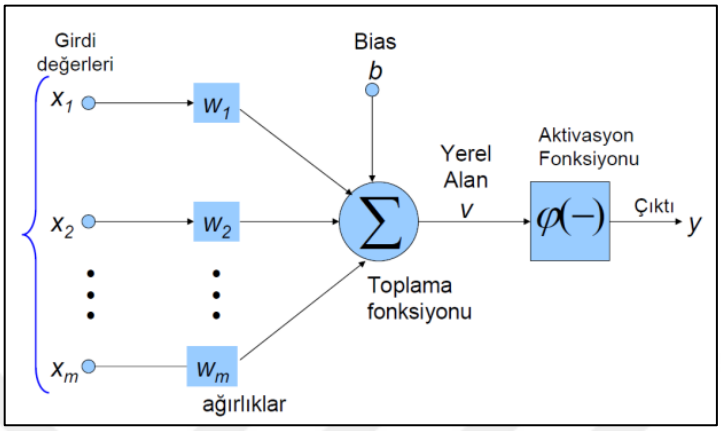
1. Nöronların birbirine bağlantı modeli, yani ağ topolojisi
2. Bağlantılardaki ağırlıklara karar verme metodolojisi, yani öğrenme algoritması
3. Aktivasyon fonksiyonu olarak sıralanabilir [28].

2.4.2 Nöronların Çalışma Prensibi

Daha önceden de belirtildiği gibi YSA'lar nöron adı verilen temel işleme birimlerinden oluşmaktadır. Şekil 2.7'de bir nöronun blok diyagramı gösterilmiştir.

Nöronlara uygulanan girişler üç temel aşamadan geçtikten sonra çıktıya dönüşmektedir. Bunlardan ilki giriş sinyallerinin bağlı oldukları ağırlıklar ile çarpılarak

toplama fonksiyonuna iletilmesidir. Şekil 2.7'de gösterilen xm değeri m'inci veri değerini göstermekte, wm ise m'inci veriye ait olan ağırlığı temsil etmektedir. Burada fizyolojik sinir hücrelerinden farklı olarak YSA'lardaki nöronlar pozitif ve negatif değerler alabilmektedir.



Şekil 2.7 Lineer olmayan nöron modeli [29]

İkinci olarak, toplama fonksiyonuna iletilen ağırlıkla çarpılmış girişler lineer bir toplayıcıda toplanmaktadır. Toplama işlemi (2.1) denklemiyle gösterilmektedir.

$$v = w1,1 *x1 + w1,2 *x2 + ... + w1,m *xm + b = w *x_{toplam} + b \quad (2.1)$$

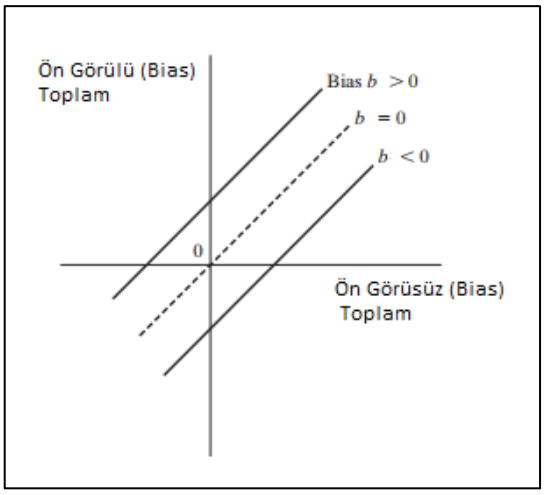
Denklem (2.1)'deki b değeri ön görü (bias) değeri olarak adlandırılmaktadır. Bu değer, toplama fonksiyonuna harici olarak uygulanmakta ve aktivasyon fonksiyonuna uygulanacak olan girişin arttırılmasını veya azaltılmasını sağlamaktadır. Ön görü değerinin pozitif belirlenmesi ile aktivasyon fonksiyonuna uygulanacak giriş arttırılırken, ön görü değerinin negatif seçilmesiyle bu giriş azaltılmaktadır. Ön görü değerinin etkisi Şekil 2.8'de gösterilmiştir.

Son olarak, denklem (2.1)'den elde edilen toplam, aktivasyon fonksiyonuna uygulanarak nöron çıkışı elde edilmektedir.

Nöronun çıktısını veren denklem, (2.2)'de gösterilmiştir.

$$y = f (w* x_{toplam} + b) \quad (2.2)$$

Buradaki aktivasyon fonksiyonu ile nöronun çıktısı, sonlu değerlerde sınırlandırılmaktadır [24].



Şekil 2.8 Ön görünün (bias) aktivasyon fonksiyonu girişine etkisi [24]

2.4.2.1 Yaygın Olarak Kullanılan Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları YSA'ların başarımını etkileyen önemli parametrelerden biridir. Bu nedenle farklı problemler için farklı aktivasyon fonksiyonları kullanılabilmektedir fakat yaygın olarak kullanılan 4 aktivasyon fonksiyonu bulunmaktadır. Birçok durumda ise çok katmanlı YSA'ların avantajlarından yararlanabilmek için lineer olmayan aktivasyon fonksiyonları tercih edilmektedir. Her bir katmanda aynı aktivasyon fonksiyonu kullanılabileceği gibi, farklı katmanlarda farklı aktivasyon fonksiyonlarının kullanılması da mümkündür [28].

Birim fonksiyonu: Tek katmanlı YSA'larda yaygın olarak kullanılmaktadır. Fonksiyona uygulanan giriş değerini aynı şekilde çıkışa aktaran fonksiyonlardır. Birim fonksiyonunun tanımı, (2.3)'te gösterilmiştir [28].

f(x)= x bütün x değerleri için (2.3)

Basamak fonksiyonu: Birim fonksiyonu gibi tek katmanlı YSA'larda yaygın olarak kullanılmaktadır. Çıkışında ürettiği değere göre unipolar ve bipolar olmak üzere ikiye ayrılmaktadır. Girişe uygulanan değişken girişler için bipolar basamak fonksiyonu -1 veya 1 çıkış değerini üretirken, unipolar basamak fonksiyonu 1 veya 0 değerini üretmektedir. Çıkış değeri, belirlenen eşik değerine (Ø) göre belirlenmektedir. Unipolar ve bipolar basamak fonksiyon tanımları sırasıyla, (2.4) ve (2.5)'te gösterilmiştir [28].

$$f(x) = \begin{cases} 1 & Eğer\ x \geq \emptyset \\ 0 & Eğer\ x < \emptyset \end{cases} \quad (2.4)$$

$$f(x) = \begin{cases} 1 & Eğer\ x \geq \emptyset \\ -1 & Eğer\ x < \emptyset \end{cases} \quad (2.5)$$

Sigmoid Fonksiyonu: YSA'lar da en yaygın olarak kullanılan aktivasyon fonksiyonlarından biridir. Sürekli olmasından dolayı türevi alınabilmekte, dolayısıyla doğrusal olmayan problemlerin çözümünde kullanılabilmektedir. Çıkışında ürettiği değere göre unipolar ve bipolar olmak üzere ikiye ayrılmaktadır. Girişe uygulanan değişken girişler için bipolar sigmoid fonksiyonu -1 ile 1 arasında bir çıkış değeri üretirken, unipolar sigmoid fonksiyonu 1 ile 0 arasında bir değerini üretmektedir. Sigmoid fonksiyonunun eğimi, eğim parametresi (λ) kullanılarak değiştirilebilmektedir. Eğim parametresi sonsuza yaklaşırken, sigmoid fonksiyonu basit bir basamak fonksiyonu gibi davranmaktadır. Unipolar ve bipolar sigmoid fonksiyon tanımları sırasıyla, (2.6) ve (2.7)'de gösterilmiştir [24], [28].

$$f(x) = \frac{1}{1+e^{-\lambda x}} \quad (2.6)$$

$$f(x) = \frac{1-e^{-\lambda x}}{1+e^{-\lambda x}} \quad (2.7)$$

2.4.3 Yapay Sinir Ağlarında Öğrenme İşlemi

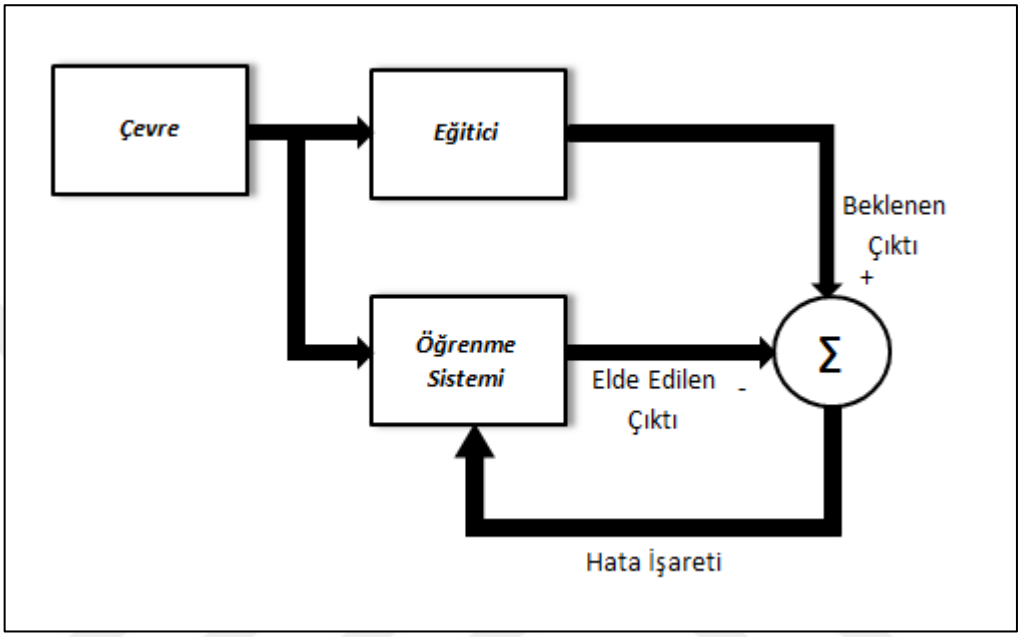
YSA'larda öğrenme işlemi tıpkı insanlardaki öğrenme yöntemi gibi iki farklı şekilde gerçekleştirilmektedir. Bunlar bir eğiticinin bulunduğu eğiticili öğrenme ve bir eğiticinin bulunmadığı eğiticisiz öğrenmedir [24].

2.4.3.1 Eğiticili Öğrenme

Eğiticili öğrenmede, çevreden gelen bilgiler için hangi yanıtın verileceği yapay sinir ağı tarafından bilinmemekte fakat bir eğitici tarafından hangi bilgi için hangi yanıtın oluşması gerektiği sisteme aktarılmaktadır. Yani kısaca, bir eğitmen yardımıyla hangi girdi için hangi çıktının beklendiği sisteme tanımlanmaktadır. Eğiticili öğrenmenin blok diyagramı Şekil 2.9'da gösterilmiştir.

Burada girdi sonucunda eğitmen tarafından en optimum sonuç üretilmektedir. Ayrıca bu girdiye YSA da bir çıkış üretmektedir. Burada istenen durum eğitmen ile YSA çıktıları arasında herhangi bir fark oluşmamasıdır ve beklenen çıktı ile elde edilen çıktı

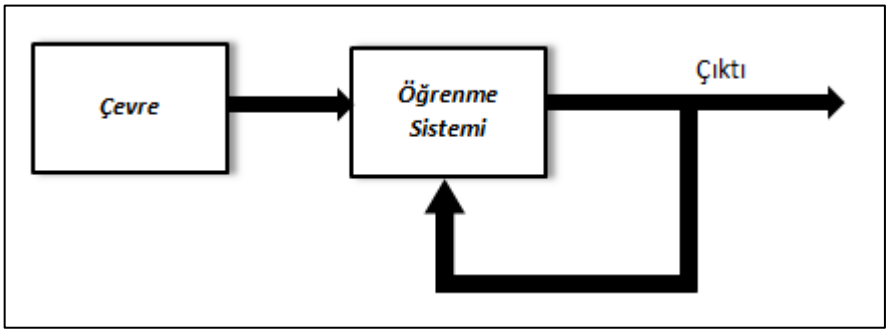
arasındaki fark hata sinyalini oluşturmaktadır. Bu nedenle eğitim vektörü ve hata sinyaline göre YSA'nın parametreleri güncellenmektedir. Bu güncelleme, eğiticinin verdiği değerlere çok yakınsandığı durumda sonlandırılır. Güncelleme işlemi sonlandığında eğitici sistemden ayrılarak çevreden gelen yanıtlara, sistemin öğrendiklerinden hareketle yanıt vermesi beklenmektedir [24].



Şekil 2.9 Eğiticili öğrenme blok diyagramı

2.4.3.2 Eğiticisiz Öğrenme

Eğiticisiz öğrenme, adından da anlaşılacağı gibi sistemde herhangi bir eğiticinin bulunmadığı öğrenme yöntemedir. Eğiticisiz öğrenmenin blok diyagramı Şekil 2.10'da gösterilmiştir.

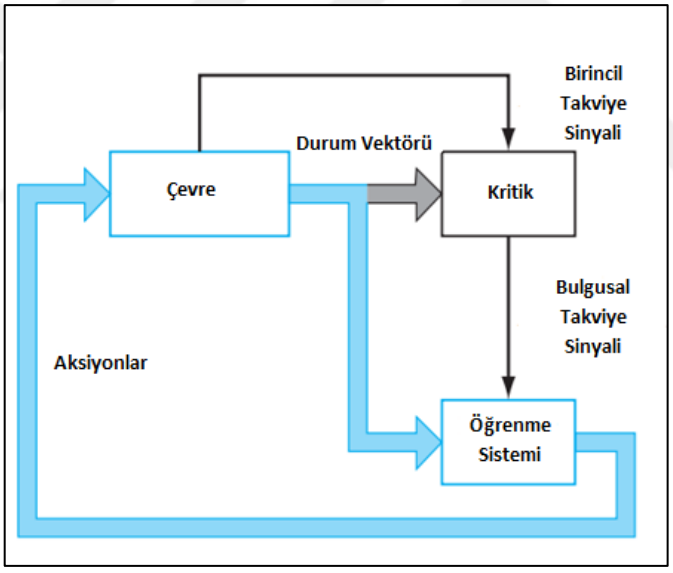


Şekil 2.10 Eğiticisiz öğrenme blok diyagramı

Eğiticisiz öğrenmede beklenen çıktı değeri tanımlanmamıştır. Bu nedenle, giriş verilerinin özelliklerine göre YSA'nın parametreleri güncellenmektedir. Yarışmacı öğrenme ve Hebbian ve Grossberg öğrenme kuralları eğiticisiz öğrenmeye örnek olarak verilebilir [24].

2.4.3.3 Takviyeli Öğrenme

Takviyeli öğrenme eğiticili öğrenmeye benzemekle birlikte aralarındaki temel fark eğiticili öğrenmede her adım sonunda beklenen çıktı değeri ile elde edilen çıktı değerinin karşılaştırılmasıdır. Takviyeli öğrenmede ise her bir adımın iyi ya da kötü olduğu en son çıktının elde edilmesinden sonra anlaşılmaktadır. Sistemin verdiği kararlara göre ödül veya ceza tanımlaması yapılır. Sistemin verdiği doğru kararlar için sistem ödüllendirilirken yanlış verilen kararlar içinse sistem cezalandırılır. Takviyeli öğrenmenin blok diyagramı Şekil 2.11'de gösterilmektedir [24].



Şekil 2.11 Takviyeli öğrenme blok diyagramı [24]

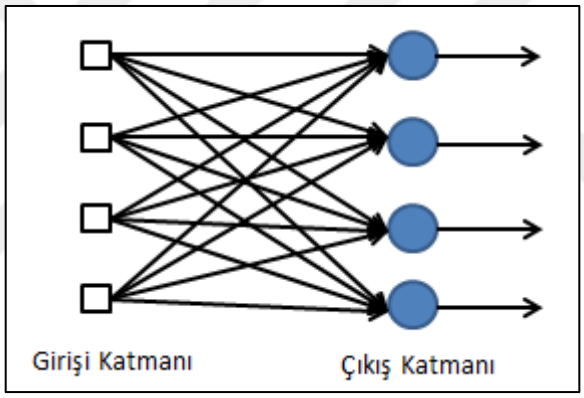
2.4.4 Yapay Sinir Ağ Modelleri

YSA'lar ile ilgili yapılan ilk çalışmalarda tek katmanlı algılayıcı (TKA) modeli kullanılmıştır. SLP Frank Rosenblatt tarafından 1958 yılında tanımlanmıştır. Oluşturulan bu modelde sınıflandırma işlemi doğrusal bir düzlem ile yapılmaktadır ve bu nedenle TKA'lardaki en büyük eksiklik sadece doğrusal olan problemlere çözüm üretebilmesidir. Bundan dolayı doğrusal olmayan problemlerin de çözülebilmesi için

farklı modeller geliştirilmiştir [23], [30]. Bu bölümde, tezin de içeriği olan temel 3 ağ modeli incelenecektir.

2.4.4.1 Tek Katmanlı Algılayıcılar

TKA'lar eğitilebilir ağırlığı bulunan tek bir katmandan oluşan YSA'lardır. TKA yapısı Şekil 2.12'de gösterilmiştir. TKA'larda aktivasyon fonksiyonu olarak unipolar veya bipolar basamak fonksiyonu kullanılmaktadır. TKA'larda giriş değerleri ağırlıklar ile çarpılarak toplanmakta, sonra oluşan değer aktivasyon fonksiyonuna uygulanmakta, son olarak aktivasyon fonksiyonunun yapısına ve eşik değerine göre 1 veya 0 ya da 1 veya -1 ile sınıflandırılmaktadır. Daha sonra beklenen çıktı ile elde edilen çıktının arasındaki farkın öğrenme oranıyla (α) ve girdiler ile çarpılarak ağırlık değerlerine eklenmesiyle öğrenme işlemi gerçekleştirilmektedir [23], [30].

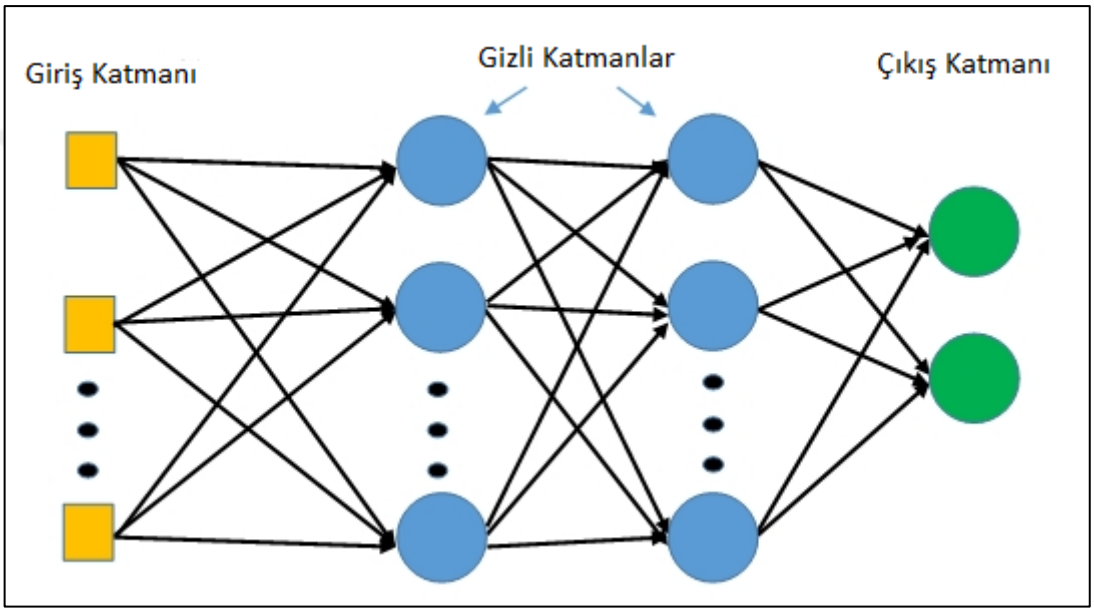


Şekil 2.12 Tek katmanlı algılayıcı

2.4.4.2 Çok Katmanlı Algılayıcılar

Doğrusal olmayan problemlerin çözümünde ÇKA'lar yaygın olarak kullanılmaktadır. TKA'lar VE, VEYA gibi mantıksal problemlere rahatlıkla çözüm üretebilmekte fakat XOR gibi doğrusal olmayan problemlere çözüm bulamamaktadır. ÇKA'lar eğitilebilir ağırlığı bulunan birden fazla katmandan oluşan yapılardır. TKA'ların yetersiz kaldığı XOR probleminin çözümü için gerçekleştirilen çalışmalar sonucunda bulunmuştur. ÇKA yapısı Şekil 2.13'te gösterilmiştir. ÇKA'lar temel olarak 3 katmandan meydana gelmektedir. Bunlar sırasıyla giriş katmanı, gizli katman ve çıkış katmanıdır. Giriş katmanı, verinin alınarak YSA'ya iletildiği katmandır. Bu katmanda herhangi bir veri işleme işlemi gerçekleştirilmemektedir. Gizli katman, giriş katmanından gelen verilerin

işlendiği katmandır. Problemin karmaşıklığına ve çeşidine göre gizli katman ve gizli katmandaki nöron sayısı değişmektedir. Gizli katmanda kaç tane nöron bulunacağı ya da YSA'da kaç tane gizli katman bulunacağını belirlemek üzere kullanılan sabit bir yöntem yoktur. Bu nedenle genellikle bu sayıların belirlenmesinde deneme yanılma yöntemi kullanılmaktadır. Denemeler sonucunda başarımı en yüksek olan yapı, ilgili problemin çözümü için kullanılabilmektedir. Çıktı katmanı ise ara katmandan gelen verileri işleyerek YSA'nın çıktısını üreten katmandır. Burada veri akışı, giriş katmanından çıkış katmanına doğru ileri yönlüdür [23], [30].



Şekil 2.13 Çok katmanlı algılayıcı [31]

ÇKA'ların temel olarak 3 karakteristik özelliği bulunmaktadır:

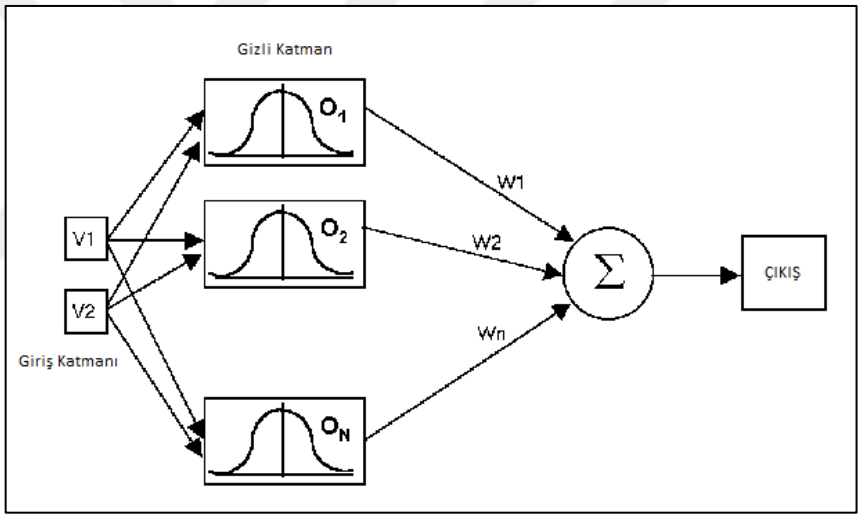
1. Ağdaki her bir nöron modeli çıkış ucunda bir non-lineerlik içermektedir.
2. En az 1 veya birden fazla saklı katmanı bulunmaktadır. Bu gizli katmanların giriş veya çıkışın bir parçası olmaması gerekmektedir.
3. Bütün bağlantılarda ağırlık olması gerekmektedir ve genellikle her bir nörondan (düğümden) diğer katmandaki nöronlara giriş yapılmaktadır.

ÇKA'larda eğiticili öğrenme metodu kullanılmaktadır. Burada ağın eğitilmesi "genelleştirilmiş delta kuralı" uygulanarak gerçekleştirilmektedir. Eğiticili öğrenme metodu kullanılmasından dolayı, ÇKA'ya uygulanan girişler için hangi çıkışların beklendiği bilinmektedir. ÇKA'ya uygulanan giriş doğrultusunda ileri yönlü bir

hesaplama yapılarak çıktılar elde edilmekte ve beklenen çıktılar ile karşılaştırılmaktadır. Elde edilen çıktı ile beklenen çıktı arasındaki fark, yani hata ise geri yönlü geçiş ile geriye doğru dağıtılarak ÇKA'nın ağırlıkları güncellenmektedir [23].

2.4.4.3 Radyal Temelli Fonksiyon Ağları

Radyal temelli fonksiyon (RTF) ağları TKA'lardan çok sonra geliştirilmiş yapılardır. RTF'ler 3 katmana sahiptir ve yalnızca tek bir gizli katmandan oluşmaktadır. RTF'lerde aynı ÇKA'lar gibi ileri beslemeli bir yapıya sahiptir ve bütün katmanları birbirine bağlıdır. Giriş katmanında herhangi bir bilgi işlemesi gerçekleştirilmemektedir. Ayrıca giriş katmanından gizli katmana yapılan bağlantılar ağırlıksızdır. Yani RTF başlangıç koşullarından çok etkilenmemektedir. Radyal temelli fonksiyon ağ yapısı Şekil 2.14'te gösterilmiştir [30].



Şekil 2.14 Radyal temelli fonksiyon ağı [32]

İlgili problemin çözümünde başarım oranının daha yüksek çıkması için RTF modelinin parametrelerinin ÇKA ağında olduğu gibi deneme yanılma yöntemi ile değiştirilmesi sonucu bulunmaktadır. Fakat Deneme yanılma yöntemi ile değiştirilebilecek parametre sayısının ÇKA ağlarına göre daha az oluşu, RTF modelinde optimum sonucu elde etmek için daha kullanışlı olmasını sağlamaktadır. RTF'nin çıkış katmanı, aktivasyon fonksiyonu olarak birim fonksiyonu kullanmaktadır ve çıkış katmanındaki nöron sayısında herhangi bir kısıtlama olmamakla birlikte, gizli katmandaki nöron sayısından daha azdır [24], [30].

RTF ağlarında üretilen çıktıyı veren denklem, (2.8)'te gösterilmiştir.

$$y_l = \sum_{i=1}^{N} W_i \, \theta(\|x - c_i\|) \tag{2.8}$$

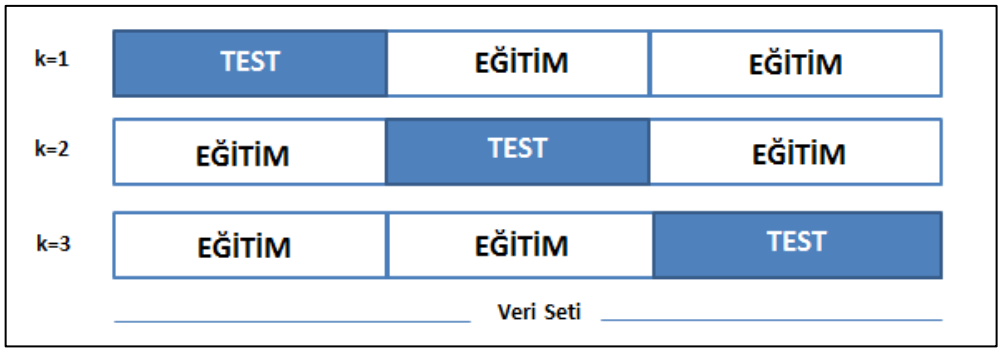
Bu formülde x, girdi değerlerini içeren vektörü temsil etmektedir. $\theta(\|x - c_i\|)$ radyal temelli aktivasyon fonksiyonudur. Gizli katmanda kullanılacak olan fonksiyonlar çeşitlilik göstermektedir fakat yaygın olarak Gauss fonksiyonu kullanılmaktadır. i'inci data noktasındaki c_i ise, radyal tabanlı fonksiyonun merkezini göstermektedir. Saklı katmandaki nöron sayısı kadar merkez noktası oluşmaktadır. Wi çıktı katmanındaki ağırlıklar, N ise gizli katmanda bulunan nöron sayısıdır [24], [30].

2.5 K-Katlı Çapraz Doğrulama Yöntemi

YSA'larda veriler rastgele olarak test, doğrulama ve eğitim amacıyla bölünmekte ve bu rastgele seçim sonucuna göre başarım oranı belirlenmektedir. Bu şekilde başarım oranını belirlemek rassal seçime bağlı olduğundan bütün sistemin başarımını çoğu zaman tam olarak yansıtmamaktadır.

Yapılan çalışmalar, K-Katlı çapraz doğrulama yönteminin rassal seçime dayalı ölçümlemeden dolayı oluşan hataları minimize ettiğini göstermektedir [33].

K-Katlı çapraz doğrulama yönteminde veri seti yaklaşık olarak eşit büyüklüklerde olan k sayısı kadar alt kümelere bölünmektedir. Bu bölünen veri setlerinden bir tanesi sistemi test etmek için geriye kalan k-1 tanesi ise sistemi eğitmek için kullanılmaktadır. Sistemin eğitilmesi ve test edilmesi için veri setinin bölünme şekli k=3 için, Şekil 2.15'te gösterilmiştir.



Şekil 2.15 k=3 için k-katlı çapraz doğrulama

Daha sonra ilgili veri seti için hata hesaplanmaktadır. Bu süreç k kere tekrarlanarak, hataların ortalamaları alınmaktadır. K-Katlı çapraz doğrulama yönteminde başarım oranı (2.9)'da gösterilmiştir [34].

$$Başarım\ Oranı = \frac{1}{k} \sum_{i=1}^{k} Ortalama\ Hata_i \quad (2.9)$$

BÖLÜM 3

VERİ TABANI

Oluşturulan sistemin eğitilmesi, test edilmesi ve doğrulanması için yumurta ve portakal görüntülerinin, bu görüntülere ait öz nitelik vektörlerinin ve yumurta ve portakalların hassas tartı ile ölçülmesi sonucunda elde edilen ağırlıklarının bulunduğu bir veri tabanı oluşturulmuştur. Veri tabanına bilgi aktarım süreci akış diyagramı Şekil 3.1'de gösterilmiştir.

Bu veri tabanı 1.200 tane görüntüyü, her bir görüntü için 5 tane öz nitelik vektörünü ve cisimlerin ağırlıklarını içermektedir.

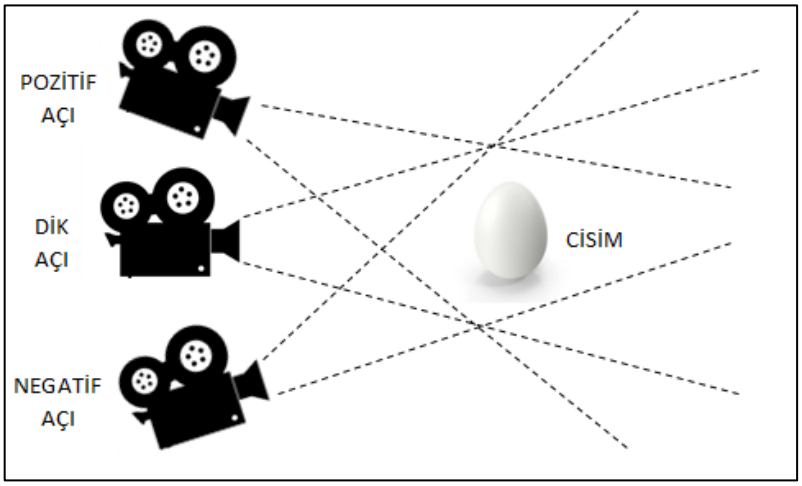
Ağırlıkları ölçülecek yumurta ve portakal görüntülerinin farklı boyutlarda ve farklı ağırlıklarda olmasına özen gösterilmiştir. Cisimlerin görüntüleri kamera yardımıyla 3 farklı açıdan (Dik Açı, Pozitif Açı ve Negatif açı) çekilmiştir. Bu açılar Şekil 3.2'de gösterilmiştir.

Böylelikle, farklı açılardan çekilmiş görüntülerin ayrı ayrı ve aynı anda sisteme uygulanması durumunda sistemin başarımının nasıl değiştiği gözlenmiştir. Bununla birlikte sabit bir tutucu ile kamera sabitlenmediği için elde edilen görüntüler, bütün görüntüler için sabit bir mesafeden ve sabit bir açı ile çekilmemiştir. Bu nedenle yumurta ve portakal örneklerinin görüntüleri, referans görüntü kullanılarak elde edilmiştir. Böylelikle, kameranın konumundan dolayı oluşacak hataların minimize edilmesi hedeflenmiştir. Bunun yanı sıra; bu referans görüntü, uzaklıktan kısmi derecede bağımsız ağırlık ölçümlerinin gerçekleştirilmesine olanak tanımıştır. Veri tabanında bulunan ve sistemin başarımı için kullanılan yumurta ve portakal

örneklerinden elde edilen görüntülerin özellikleri ile öz nitelik vektörlerinin en büyük, en küçük ve ortalama değerleri aşağıda belirtilmiştir.



Şekil 3.1 Veri tabanına veri yazma süreci blok diyagramı



Şekil 3.2 Görüntü çekim açıları

3.1 Belirlenen Öz Nitelik Vektörleri

Elde edilen görüntülerin ağırlık tahminleri bu öznitelik vektörleri yardımıyla yapılmaktadır. YSA'ya her bir öznitelik vektörü girişi olarak uygulanmakta ve çıkış olarak da tahmini ağırlık bilgisi elde edilmektedir.

Sistemin başarım oranları farklı öznitelik vektörleri ile denenmiş ve sonuçları ilerleyen bölümlerde açıklanmıştır. Sistemde kullanılan öznitelik vektörleri;

3.1.1 Cismin Yüzey Alanı

Cisim görüntüden ayrıldıktan sonra, cismin kapladığı alan piksel değerinden hesaplanmaktadır. Elde edilen piksel değeri, $cm^{2,}$'ye geçmek için dönüşüm katsayısı (σ) ile çarpılmaktadır. Alan dönüşüm katsayısının hesaplanma yöntemi (3.1), (3.2) ve (3.3)'te gösterilmektedir. Referans görüntüdeki karelerin kesişim noktalarının piksel değerleri (x, y) olarak tanımlanmıştır. Birinci siyah karenin kesişim noktası (x_i, y_i), sağ üst köşedeki siyah karenin kesişim noktası (x_l, y_l), sol alt köşedeki siyah karenin kesişim noktası (x_k, y_k) ile gösterilmektedir. Denklemde bulunan k, referans görüntüdeki karelerin kenar uzunluğunu; dx, yatayda bulunan siyah ve beyaz kare sayısını; dy ise dikeyde bulunan siyah ve beyaz kare sayısını vermektedir.

$$ra = (dx \times k) \div [(x_i - x_l)^2 + (y_i - y_l)^2]^{-\frac{1}{2}} \quad (3.1)$$

$$rb = (dy \times k) \div [(x_i - x_k)^2 + (y_i - y_k)^2]^{-\frac{1}{2}} \quad (3.2)$$

$$\sigma = ra \times rb \quad (3.3)$$

3.1.2 Cismin Çevresi

Ağırlığı ölçülecek cismin çevresi piksel değerinde hesaplanmaktadır. Elde edilen piksel değerini, cm'ye dönüştürmek için dönüşüm katsayısı *(ρ)* ile çarpılmaktadır. İlgili katsayının hesaplanması (3.4)'te gösterilmiştir.

$$\rho = \frac{ra+rb}{2} \quad (3.4)$$

3.1.3 Merkeze En Uzak Kenar Noktası

En uzak kenarı hesaplama yöntemi (3.5)'te verilmiştir. Ağırlığı tahmin edilecek cismin merkez noktası *(*$\mathrm{x_m}, \mathrm{y_m}$*)* olarak belirtilmiştir. *(*$\mathrm{x_n}, \mathrm{y_n}$*)* ise ağırlığı tahmin edilecek cismin sınır noktalarını göstermektedir. Cismin sınırları t tane veriden oluşmaktadır ve bu Öklid uzaklıklarından en büyüğü merkeze en uzak kenar noktasını vermektedir.

$$\max(d) = \max([(x_m - x_n)^2 + (y_m - y_n)^2]^{-\frac{1}{2}} \quad (3.5)$$

n = 1, 2, 3, 4, ..., t

3.1.4 Merkeze En Yakın Kenar Noktası

En uzak kenar noktası hesaplama yöntemine benzer şekilde en yakın kenarı hesaplama yöntemi (3.6)'da verilmiştir. Öklid uzaklarından en küçük olanı en yakın kenar uzaklığını vermektedir.

$$\min(d) = \min([(x_m - x_n)^2 + (y_m - y_n)^2]^{-\frac{1}{2}} \quad (3.6)$$

n = 1, 2, 3, 4, ..., t

3.1.5 Yuvarlaklık

Yuvarlaklık öznitelik vektörünün hesaplanma yöntemi (3.7)'de gösterilmektedir.

$$R = \frac{\min(d)}{\max(d)} \quad (3.7)$$

3.2 Cisimlerin Öz Nitelik Vektör Değerleri

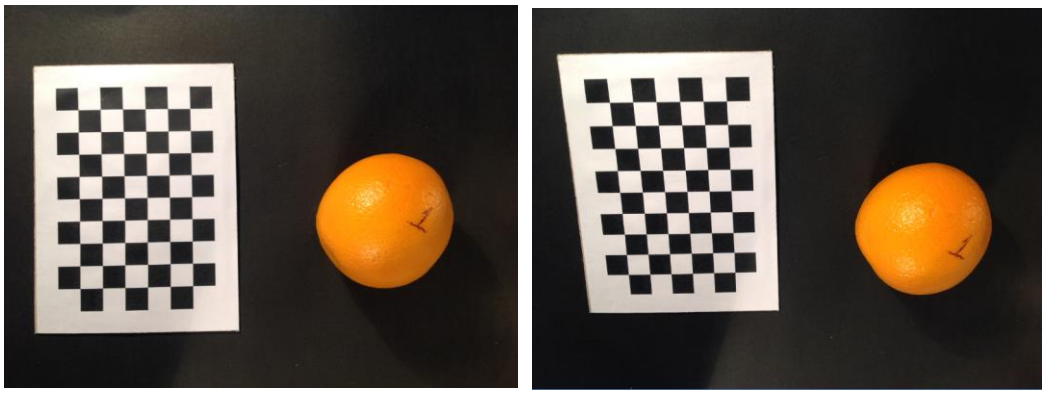
YSA'ların girişine uygulanacak olan yumurta ve portakal görüntülerinden elde edilen öz nitelik vektörleri Matlab programında yazılan kodlar ile hesaplanarak veri tabanına yazılmıştır. Veri tabanında ilgili cisim için alan, yuvarlaklık, çevre, en kısa yarıçap ve en uzun yarıçap değerleri bulunmaktadır. Veri tabanındaki alanın birimi cm^2; çevrenin, en kısa yarıçapın ve en uzun yarıçapın birimi ise cm'dir. Yuvarlaklık öz nitelik vektörünün ise birimi yoktur. Yumurta ve portakal örneklerinin 3 farklı açıdan elde edilmiş görüntüleri üzerinden elde edilen öz nitelik vektörlerinin en büyük, en küçük ve ortalama değerleri Çizelge 3.1'de gösterilmiştir.

Çizelge 3.1 Öz nitelik vektörlerinin en büyük, en küçük ve ortalama değerleri

	YUMURTA			PORTAKAL		
Öz Nitelik Vektörleri	En Büyük	En Küçük	Ortalama	En Büyük	En Küçük	Ortalama
Alan (cm^2)	27,29	14,62	23,15	55,62	31	40,32
Yuvarlaklık	0,85	0,65	0,72	0,97	0,83	0,93
Çevre (cm)	19,07	13,84	17,35	33,79	21,87	26,32
En Kısa Yarıçap (cm)	2,5	1,82	2,32	4,07	2,82	3,42
En Uzun Yarıçap (cm)	3,65	2,55	3,23	4,35	3,31	3,70

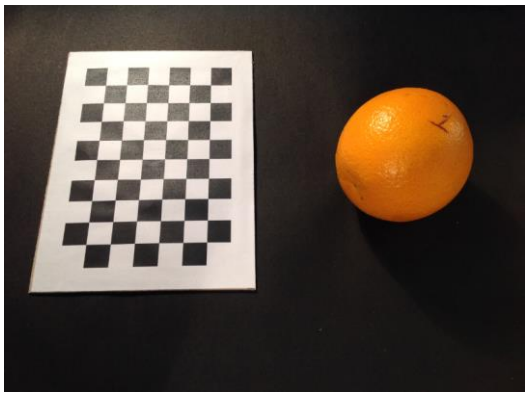
3.3 Portakal Örnekleri

Veri tabanındaki portakal görüntüleri, 150 adet farklı büyüklük ve ağırlıkta olan portakal örneklerinden elde edilmiştir. Bu örnekler 3 farklı açıdan çekilerek toplam 450 adet görüntü elde edilmiş ve bu görüntüler veri tabanına eklenmiştir. Hassas tartı ile yapılan ölçüm sonucunda eğitim ve test kümesindeki en hafif portakalın ağırlığı 89 gr, en ağır yumurtanın ağırlığı 178 gr ve ortalama ağırlık ise 119.9 gr olarak tespit edilmiştir. Veri tabanında bulunan portakal görüntüleri 3 farklı açıdan çekilmiş şekilde, Şekil 3.3'te gösterilmiştir.



(a) (b)

Şekil 3.3 Portakal örneği görüntüleri (a) Dik açılı (b) Pozitif açılı (c) Negatif açılı



(c)

Şekil 3.3 Portakal örneği görüntüleri (a) Dik açılı (b) Pozitif açılı (c) Negatif açılı (devamı)

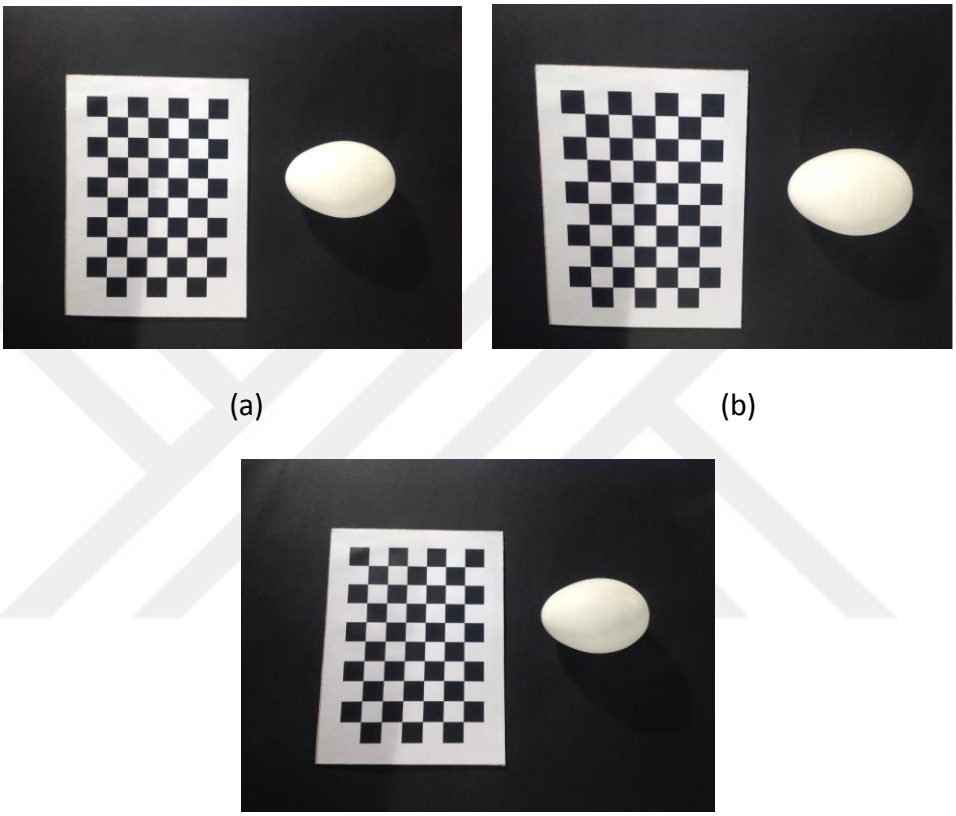
3.4 Yumurta Örnekleri

Veri tabanındaki yumurta görüntüleri, 250 adet farklı büyüklük ve ağırlıkta olan yumurta örneklerinden elde edilmiştir. Ayrıca piyasada satılan yumurtaların sınıflandırma doğruluğunun test edilebilmesi için yumurtalar 4 farklı markadan seçilmiştir. Yumurtalar paket üzerinde yazan sınıflandırma değerlerine göre 60 tane X-Large, 90 tane Large, 40 tane Medium, 60 tane Small yumurta olacak şekilde satın alınmıştır fakat hassas tartı ile yapılan ölçümler sonucunda 9 tane X-Large, 103 tane Large, 89 tane Medium, 49 tane Small yumurta olduğu görülmüştür. Yumurtaların ağırlıklarının sınıflandırılması Türk Gıda Kodeksi Yumurta Tebliği'ne göre yapılmaktadır ve Çizelge 3.2'de A sınıfı yumurtaların Türk Gıda Kodeksine göre ağırlık sınıflandırılması gösterilmektedir.

Çizelge 3.2 A sınıfı yumurtalarda ağırlık sınıfları

Yumurtaların Ağırlık Sınıfları	
Ağırlık Sınıfı	**Ağırlık**
XL-Çok Büyük	≥73 gr
L-Büyük	≥63 – <73 gr
M-Orta	≥ 53 – <63 gr
S-Küçük	<53 gr

Yumurta örneklerinden elde edilen görüntüler 3 farklı açıdan çekilerek toplam 750 adet görüntü elde edilmiş ve bu görüntüler veri tabanına eklenmiştir. Veri tabanında bulunan yumurta görüntüleri 3 farklı açıdan çekilmiş şekilde, Şekil 3.4'te gösterilmiştir. Hassas tartı ile yapılan ölçümler sonucunda eğitim ve test kümesindeki en hafif yumurtanın ağırlığının 32 gr, en ağır yumurtanın ağırlığının 75 gr ve yumurtaların ortalama ağırlığın ise 60.5 gr olduğu görülmüştür.



(a) (b)

(c)

Şekil 3.4 Yumurta örneği görüntüleri (a) Dik açılı (b) Pozitif açılı (c) Negatif açılı

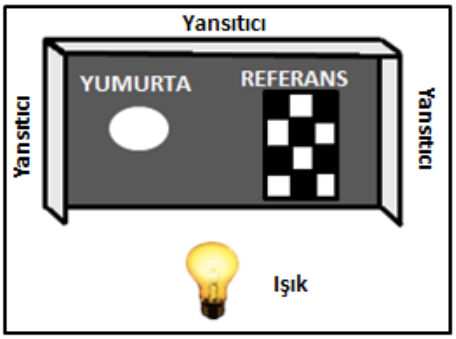
BÖLÜM 4

KULLANILAN MATERYALLER VE METODLAR

Yapılmış olan bu tez çalışmasında Matlab (R2015a) programı ve araç kutuları kullanılarak cisimlerin ağırlıkları tahmin edilmektedir. Kurulan bu sistemde öncelikli olarak YSA eğitilmekte daha sonra test edilerek hata oranları tespit edilmektedir. Farklı YSA modelleri ve parametreleri için eğitim ve test işlemleri tekrarlanmaktadır. Böylece, başarımı en yüksek olan YSA modeli ve parametrelerinin tespit edilmesi hedeflenmektedir. Son olarak başarımı en yüksek olan parametrelerle oluşturulmuş YSA modeline, ağırlığı ölçülecek olan cismin öz nitelik vektörleri uygulanarak ağırlık tahminleme işlemi gerçekleştirilmektedir.

4.1 Ölçüm Düzeneği

Ağırlığı ölçülecek cismin görüntülerinin elde edilmesi için oluşturulan düzenek Şekil 4.1'de gösterilmektedir.



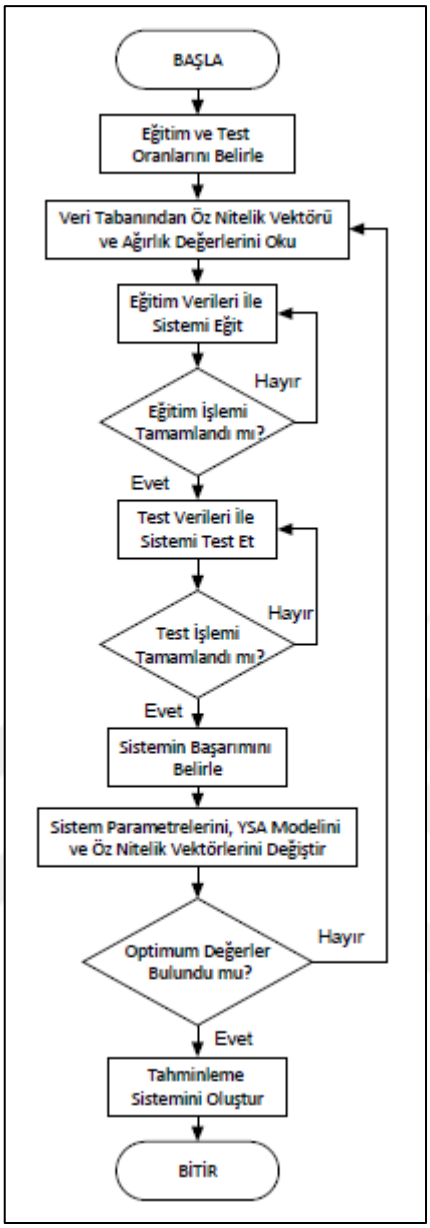
Şekil 4.1 Ölçüm düzeneği

Doğrusal bir kaynaktan alınan ışık, yansıtıcılar yardımıyla cisim ve referans görüntüyü eşit miktarda aydınlatmaktadır. Böylelikle oluşacak karanlıktan dolayı cismin öznitelik vektörlerinin tam olarak elde edilememesi veya referans noktalarının tam olarak belirlenememesi gibi sorunlar ortadan kaldırılmıştır. Uzaklıktan ve çekim açısından belirli sınırlar dahilinde bağımsız olarak kurulan sistem sayesinde, kameranın konumundan bağımsız olarak ağırlık tahmini bu ölçüm düzeneği yardımıyla gerçekleştirilebilmektedir.

4.2 Yapay Sinir Ağının Eğitilmesi ve Test Edilmesi

İlk olarak veri tabanında bulunan ağırlığı ölçülecek cisim görüntülerinden belirli bir oranda sistemi eğitmek için seçilmiştir. Sistemi eğitmek için seçilen görüntülerden çıkarılan öz nitelik vektörleri, YSA'ya uygulanmıştır. Bu çalışmada eğiticili öğrenme kullanıldığı için her bir cisim görüntüsünün yanı sıra aynı zamanda beklenen ağırlık değerleri de sisteme tanıtılmıştır. Daha sonra ilgili cismin veri tabanında kalan görüntüleri YSA'nın test edilmesi için kullanılmıştır. Bu YSA'nın eğitilmesi ve test edilmesi işlemi sonucunda, sistemin hata oranına bakılarak sistemin başarımı belirlenmiştir. Son olarak; YSA modeli, YSA parametreleri ve cismin öz nitelik vektörleri değiştirilerek YSA'nın eğitim ve test edilmesi işlemi tekrarlanmıştır. Her bir deneme sonucunda elde edilen hata oranlarına göre başarımı en yüksek olan YSA parametreleri, YSA modeli ve öz nitelik vektörleri ile ağırlık tahminleme sistemi oluşturulmuştur. YSA'yı eğitme ve test etme süreçlerinin blok diyagramı Şekil 4.2'de verilmiştir.

Belirli bir cisim için oluşturulan ağırlık tahminleme sistemi, farklı cisimler için aynı başarımı veremeyebilir. Bu nedenle farklı cisimler için farklı YSA modeli, YSA parametreleri ve öz nitelik vektörleri içeren sistemlerin tasarlanması gerekebilir.

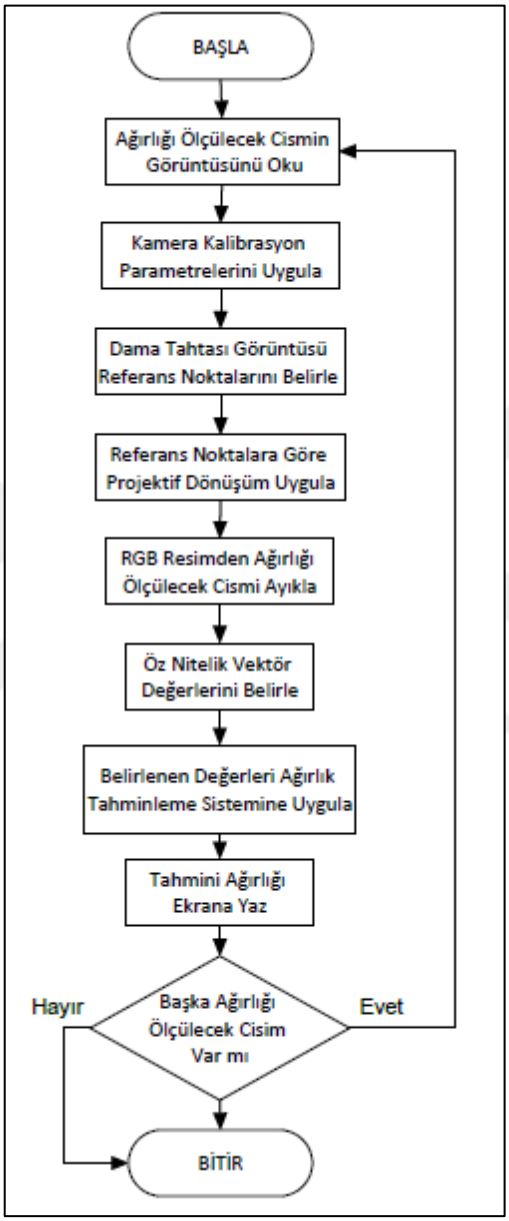


Şekil 4.2 Tahminle sistemi oluşturma süreçleri blok diyagramı

4.3 Ağırlık Tahminleme

Başarımı en yüksek olan YSA modeli, YSA parametreleri ve öz nitelik vektörleri kullanılarak oluşturulan sistem ile ağırlık tahminleme işlemi yapılmaktadır. Öncelikli olarak Matlab programında hazırlanan kodlar yardımıyla ağırlığı ölçülecek cismin öz nitelik vektörleri tespit edilmektedir. Daha sonra başarımı en yüksek sisteme ağırlığı ölçülecek görüntünün öz nitelik vektörleri uygulanarak çıktı elde edilmektedir. Elde edilen bu çıktı, tahmini olarak cismin ağırlığını göstermektedir. Ağırlık tahminleme süreçlerinin blok diyagramı Şekil 4.3'de gösterilmiştir.

Farklı cisimler için farklı parametrelerin başarımı daha yüksek olabilmektedir. Bu nedenle farklı cisimler için, YSA'nın eğitilmesi ve test edilmesi aşamasına geri dönülerek ilgili cisim için optimum parametrelerin revize edilmesi başarımı artırabilmektedir.



Şekil 4.3 Tahminleme süreçleri blok diyagramı

4.4 Tahminleme Sistemi Süreçleri

Şekil 4.3'te blok diyagramı verilen tahminleme süreçleri, Matlab programı araçları ve fonksiyonları kullanılarak gerçekleştirilmiştir. Tahminleme sistemi süreçlerinde uygulanan işlemler, ayrıntılı olarak bu bölümde ele alınmıştır.

4.4.1 Cismin Görüntüsünün Okunması

Cismin ağırlığının ölçülebilmesi için, öncelikli olarak cisim görüntüsünün çalışma ortamına aktarılması gerekmektedir. Bu nedenle görüntü işlenmek üzere, Matlab çalışma alanına aktarılmaktadır. Ağırlığı ölçülecek olan cismin çalışma ortamına aktarılmasıyla birlikte, görüntünün çözünürlüğü, boyutu gibi parametreler de elde edilebilmektedir.

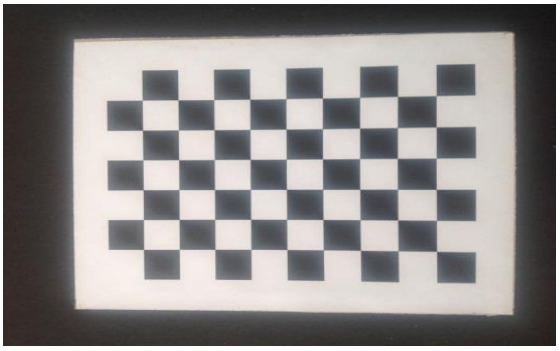
4.4.2 Sistemin Kalibre Edilmesi

Görüntülerin elde edilmesi için 4. Nesil Ipad kullanılmıştır ve elde edilen bu görüntülerdeki piksel çözünürlüğü 2592x1936'dır. Tek bir kamera ile görüntülerin elde edilmesi nedeniyle tek kamera kalibrasyonu gerçekleştirilmiştir. Kalibrasyon için Matlab üzerinde bulunan "MATLAB SINGLE CALIBRATION APP" kullanılmıştır. Bu yöntem ile kameranın dahili, harici ve lens bozulma parametreleri hesaplanarak, kameranın kalibre edilmesi hedeflenmiştir. Kamera kalibrasyonu için uygulanan algoritma Şekil 4.4'te gösterilmiştir.



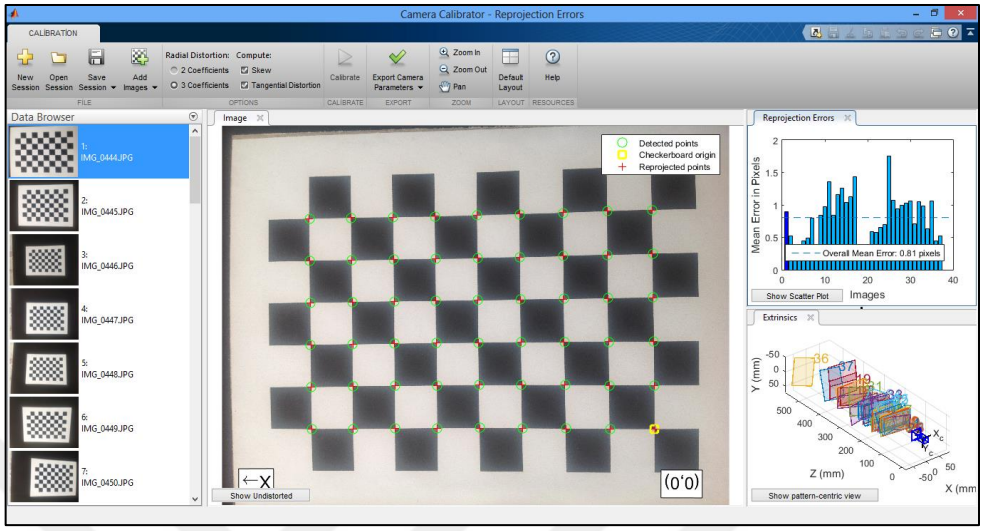
Şekil 4.4 Kalibrasyon akış diyagramı

Kalibrasyon paterni olarak 10x7'lik dama tahtası (checkerboard) görüntüsü kullanılmıştır. Kamerayı kalibre etmek için kullanılan görüntü Şekil 4.5'te gösterilmiştir.



Şekil 4.5 Kalibrasyon için kullanılan checkerboard

Farklı açılardan çekilmiş 38 tane dama tahtası görüntüsü, sisteme yüklenerek, sistemin kalibre edilmesi sağlanmıştır. Sistemin kalibrasyonu sırasında kullanılan arayüz Şekil 4.6'da gösterilmiştir.



Şekil 4.6 Kamere kalibrasyon arayüzü

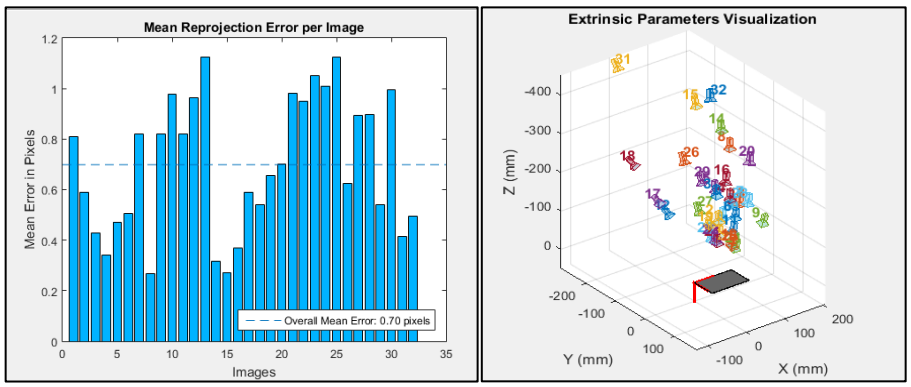
Sistemin kalibre edilmesi sonucunda oluşan reprojeksiyon hatası ve kamera harici parametre görüntüsü kontrol edilmiştir. Reprojeksiyon hatası, elde edilen görüntü ile elde edilmesi gereken görüntü arasındaki piksel farkının meydana getirdiği ortalama hatadır. Reprojeksiyon hatasının görüntüsü Şekil 4.7'de gösterilmiştir [35].



Şekil 4.7 Reprojeksiyon hatası [36]

Kullanılan kalibrasyon işleminde reprojeksiyon hatası 1.1 pikselden büyük olan görüntüler sistemden çıkarılmıştır. Ayrıca kamera harici parametrelerinin görüntüsüne göre kamera paterninin arkasında olması durumu gibi paterne göre kameranın konumunun doğru olmaması durumunda görüntünün hatalı olduğu varsayılarak sistemden çıkarılmıştır. Bu kontroller doğrultusunda, sistemin kalibrasyonunu bozan 6 görüntü çıkarılarak 32 tane görüntü ile sistem kalibre edilmiştir. Sistemin kalibre

edilmesi sonucunda oluşan reprojeksiyon hatası ve kamera harici parametrelerinin görüntüsü Şekil 4.8'de verilmiştir.

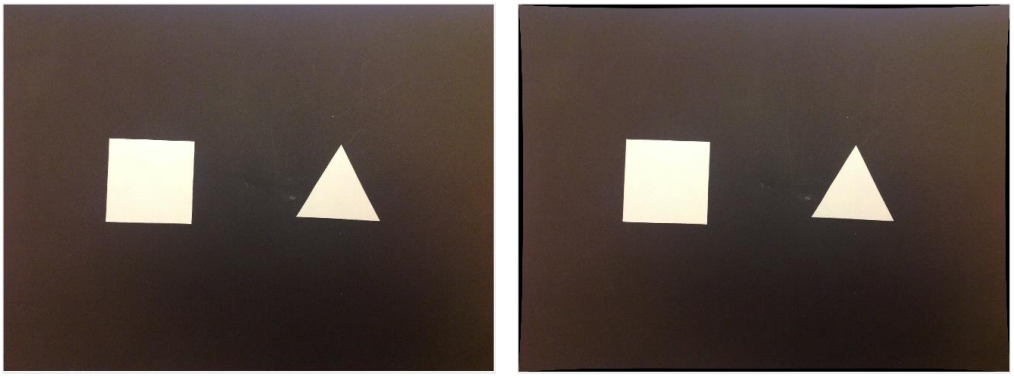


(a) (b)

Şekil 4.8 a) Sistemden oluşan reprojeksiyon hatası, b) Harici parametrelerin görseli

4.4.2.1 Kalibrasyon Parametrelerin Uygulanması

Matlab programının "MATLAB SINGLE CALIBRATION APP" aracı kullanılarak elde edilen kalibrasyon parametreleri orijinal görüntüye uygulanarak, orijinal görüntüde oluşan lens bozulmalarının giderilmesi sağlanmıştır. Orijinal görüntü ve kalibrasyon parametreleri uygulandıktan sonra lens bozulmaları giderilmiş görüntü Şekil 4.9'da gösterilmiştir.



(a) (b)

Şekil 4.9 a) Orijinal Görüntü, b) Kalibrasyon sonucunda elde edilen görüntü

4.4.3 Görüntü Dönüşümleri

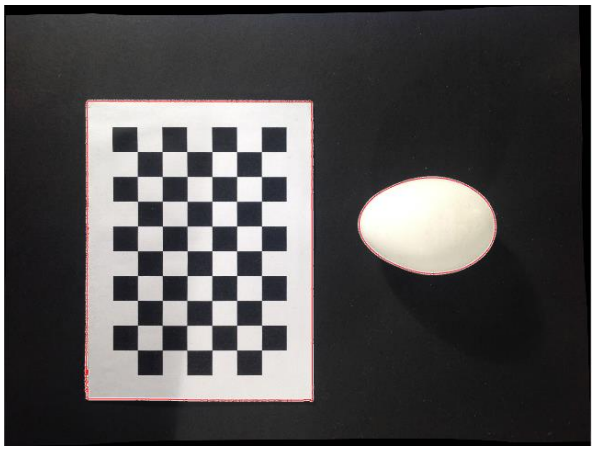
Öncelikle görüntü, merkez noktasından belirli bir açıda saat yönünün tersine döndürülerek, sisteme uygun hale getirilmektedir. Daha sonra kamera açısından ve

uzaklığından kısmi derecede bağımsız olarak ölçüm yapılabilmesi için gerekli görüntü dönüşümlerinin yapılması gerekmektedir çünkü ağırlığı ölçülecek cisimlerin görüntüleri sabit bir mesafeden ve sabit bir açıyla elde edilmemiştir. Burada görüntüler, kamera yardımıyla hem farklı açılardan çekilen görüntülerin sisteme etkisinin ölçülebilmesi için 3 farklı açıdan çekilmiştir hem de sabit bir tutucu ile kamera sabitlenmediği için görüntülerin elde edilmesi sırasında açı ve uzaklık farklılıkları oluşmuştur. Bu nedenle kameranın (x, y, z) koordinatlarından kaynaklanan perspektif görüntüde oluşacak farklılıkların giderilmesi gerekmektedir. Yani kamerayı (0, 0, 0) noktasına yerleştirerek tam dik bir iz düşüm elde edilebilmesi amaçlanmaktadır. Bu nedenle görüntünün kalibre edilebilmesi için projektif dönüşümü uygulanmıştır. Projektif dönüşümün uygulanabilmesi için referans görüntü kullanılmaktadır. Bu nedenle öncelikle dama tahtası referans görüntüsünün referans noktaları ve boyutu belirlenmektedir. Daha sonra beklenen değer ve elde edilen değer arasındaki farklar kullanılarak uygulanacak olan geometrik dönüşüm tahmin edilmektedir. Son olarak, tahmin edilen bu projektif geometrik dönüşümü, ilgili görüntüye uygulanarak görüntü dönüşümü işlemi tamamlanmaktadır.

4.4.4 Cismin Görüntüden Ayıklanarak Sınır Noktalarının Belirlenmesi

Cismin görüntüden ayıklanması işlemi hızlı ölçüm için ve hassas ölçüm için olmak üzere iki şekilde yapılabilmektedir. Hızlı ölçüm sırasında gerçekleştirilen görüntü ayıklama işleminin başarımı hassas ölçüm sırasında görüntü ayıklama işleminin başarımına göre genellikle daha düşük olmakla birlikte cismin ayıklanma süresi hızlanmaktadır. Bununla birlikte birçok görüntüde hızlı ölçüm sırasında kullanılan ayıklama işlemi gereksinimleri karşılamaktadır.

Hassas ölçüm için cisim ayıklama işleminde, görüntüyü piksel piksel inceleyecek kod dizileri yazılarak cisim belirlenirken, hassas ölçüm için görüntü ayıklama işleminde Matlab fonksiyonları kullanılarak cisim belirlenmektedir. Görüntüdeki cismin ayıklanarak sınır noktalarının belirlendiği örnek görüntü Şekil 4.10'da gösterilmiştir.



Şekil 4.10 Cisimlerin sınırlarının belirlendiği örnek görüntü

4.4.4.1 Hızlı Ölçüm İçin Cisim Ayıklama Yöntemi

Öncelikli olarak RGB (Red – Green – Blue) renk formatında olan görüntünün parlaklığı korunurken, ton ve saturasyon bilgileri ortadan kaldırılarak gri düzeyli görüntüye dönüştürülmektedir. Gri düzeyli görüntüdeki cismin, arka plan görüntüsünden ayrılabilmesi amacıyla bir eşik değeri belirlenmektedir. Eşik değerinin belirlenmesi aşamasında Otsu metodunu kullanmaktadır. Bu metotta gri düzeyli renklerin görüntüde kaçar defa kullanıldığına bakılarak eşik değeri belirlenmektedir. Daha sonra belirlenen bu eşik değere göre görüntü ikili sisteme, dönüştürülmektedir. İkili sisteme dönüştürme işlemi ile eşik değerinin üzerinde kalan piksellerde cisim var (1), altında kalan değerler ise cisim yok (0) olarak değerlendirilmektedir. Sonraki aşamada bağlantılı objelerin 10.000 pikselden az olması durumunda, bu objeler de cisim yok (0) olarak değerlendirilmektedir. Buna ek olarak cisim var (1) olarak belirlenen bölgelerdeki küçük objeler temizlenmekte ve cisim görüntüden ayıklanmaktadır. Son olarak da ayıklanan cisimlerin sınır noktaları çizdirilerek, cisim ayıklama işlemi tamamlanmaktadır.

4.4.4.2 Hassas Ölçüm İçin Cisim Ayıklama Yöntemi

Başlangıç aşamasında her bir renk piksel değeri double değişken tipine dönüştürülmektedir. Bu piksel değeri 0.56'dan büyük ise veya 3 rengin piksel ortalaması 0.56'dan büyük ise veya 0.30'dan büyük bir rengin piksel değeri diğer renklerin piksel değerlerinin 2 katından fazla ise piksel değeri 1 (nesne var) olarak

belirlenmektedir. Eğer 3 rengin piksel ortalaması 0.25'ten küçük ise piksel değeri 0 (nesne yok) olarak belirlenmektedir. Daha sonra, RGB renk formatında olan görüntünün parlaklığı korunurken, ton ve saturasyon bilgileri ortadan kaldırılarak gri düzeyli görüntüye dönüştürülmektedir. Sonraki aşamada görüntüye medyan filtre uygulanmaktadır. Meydan filtre ile belirlenen matris boyutundaki gürültüler görüntüden temizlenmektedir. Sonraki aşamada değeri 0.30'dan fazla olan ve ilk nesne tanıma işleminde durumu tespit edilememiş piksellerin bütün komşu pikseller ile karşılaştırması yapılmaktadır. Eğer komşu piksel ilk nesne tanımlama işleminde "nesne var" olarak tanımlanmış ve komşu piksel ile pikselin kendi değeri arasındaki fark %10'dan küçük 2 adet piksel varsa, bu piksel de 'nesne var' olarak işaretlenmektedir. Bu işlem nesnenin sınırları daha hassas belirleyebilmek için kullanılmaktadır. Daha sonra durumu hakkında karar verilememiş her bir pikselin kırmızı, yeşil ve mavi bileşenleri için ilgili pikselin ortalama renk değeri bulunmaktadır. Bu ortalama piksel değerleri bütün pikseller için belirlendikten sonra tekrar ortalaması alınarak, eşik değeri belirlenmektedir. Bu eşik değerinin altında kalan pikseller nesne yok (0) olarak işaretlenmekte, üzerinde kalan pikseller ise nesne var (1) olarak işaretlenmektedir. Sonraki aşamada, bağlantılı objelerin 10.000 pikselden az olması durumunda, bu objeler de cisim yok (0) olarak değerlendirilmektedir. Buna ek olarak cisim var (1) olarak belirlenen bölgelerdeki küçük objelerin temizlenmesi işlemleri de gerçekleştirilerek, cisim görüntüden ayıklanmaktadır. Son olarak da ayıklanan cisimlerin sınır noktaları çizdirilerek, cisim ayıklama işlemi tamamlanmaktadır.

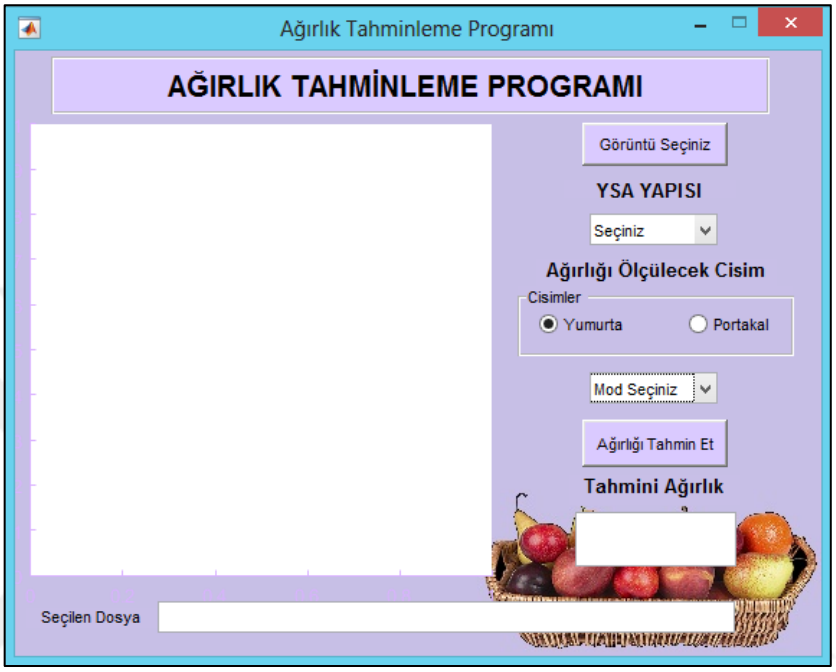
Buradaki sayısal değerler problem odaklı olarak deneme yanılma yöntemi ile bulunmuştur ve bu yüksek lisans tezi kapsamında olan yumurta ve portakal örnekleri için çalışmaktadır. Başka problemler için bu sayısal değerler değiştirilerek cismin ayıklanma başarımı en uygun değerlerin bulunması gerekmektedir.

4.4.5 Öz Nitelik Vektör Değerlerinin Belirlenmesi

İlgili cisim için başarımı en yüksek çıkan sistemde kullanılan öz nitelik vektörleri, ağırlığı tahmin edilecek cisim için hesaplanmaktadır. Burada veri tabanının oluşturulması aşamasında kullanılan öz nitelik vektörleri ve hesaplama yöntemlerinin aynısı ağırlığı tahmin edilecek cismin görüntüsü için de kullanılmaktadır.

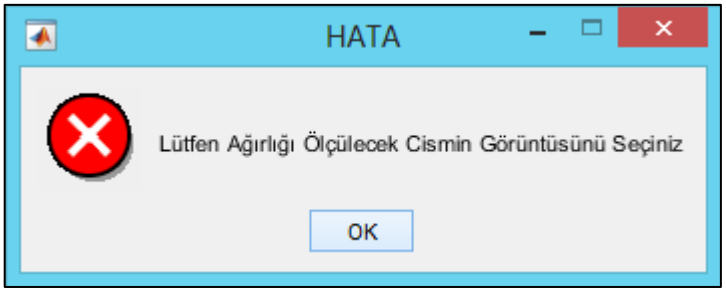
4.4.6 Tahmini Ağırlığın Kullanıcı Arayüzü Kullanılarak Belirlenmesi

Ağırlığı tahmin edilecek olan cismin öz nitelik vektörleri, ilgili cisim için kurulan başarımı en yüksek sisteme uygulanmaktadır. Bu nedenle kullanıcı arayüzü, bu parametreler baz alınarak hazırlanmıştır. Hazırlanmış olan kullanıcı arayüz görüntüsü Şekil 4.11'de gösterilmektedir.



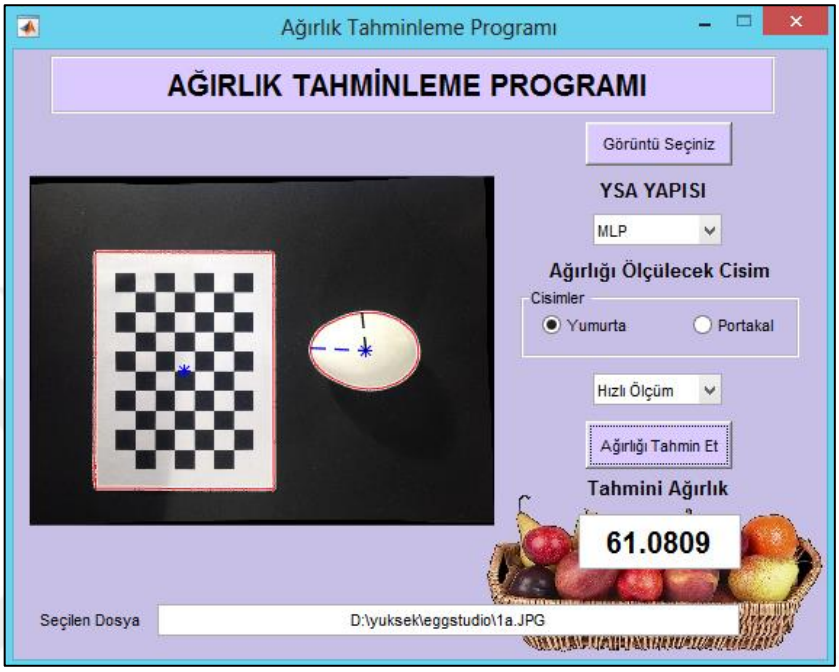
Şekil 4.11 Ağırlık tahminle programının kullanıcı arayüzü

Bu kullanıcı arayüzü ile ağırlığı ölçülecek cismin görüntüsü, YSA yapısı, ağırlığı ölçülecek cismin ne olduğu ve hangi modda ölçüm yapılacağı gibi bilgiler seçilmekte ve "Ağırlığı Tahmin Et" butonuna basılarak, ağırlığı tahmin etme işlemi gerçekleştirilmektedir. Bu seçimlerden birinin yapılmaması durumunda kullanıcıyı bilgilendirmek amacıyla uyarı mesajı gösterilmektedir. Program tarafından üretilen örnek bir uyarı mesajı görüntüsü Şekil 4.12'de gösterilmiştir.



Şekil 4.12 Örnek bir sistem çıktısı

Kullanıcı arayüzü çıktısında; ağırlığı tahmin edilecek cismin ve referans görüntünün sınırları ve merkez noktaları, MEUKN ve MEYKN değerleri, seçilen görüntünün dosya yolu ve tahmin edilen cismin ağırlığı gösterilmektedir. Örnek bir ağırlık tahminleme işlemi sonucu Şekil 4.13'de gösterilmektedir.



Şekil 4.13 Örnek bir ağırlık tahminleme işlemi sonucu

BÖLÜM 5

SİMÜLASYON VE TEST

Kurulan sistemin parametreleri değiştirilerek, denenen yöntemler arasındaki en iyi ağırlık tahminini yapan sistem tespit edilmiştir. Performans parametresi olarak ortalama mutlak hata (mae) ve korelasyon katsayısı kullanılmıştır. YSA'nın; modeli, girişine uygulanacak olan öznitelik vektörleri, öğrenme oranları, gizli katman ve gizli katmandaki nöron sayıları, iterasyon sayısı, transfer fonksiyonu, öğrenme fonksiyonu, öğrenme, test ve validasyon seti oranları gibi parametreleri değiştirilerek başarımı arttıracak farklı denemeler gerçekleştirilmiştir. Her bir deneme 5 kez çalıştırılarak elde edilen sonuçların aritmetik ortalaması, ilgili parametreleri içeren denemelerin başarımını oluşturmaktadır.

5.1 Yumurta Simülasyonu

Yumurta örnekleri üzerinde yapılan çalışmalar, ÇKA ve RTF YSA modelleri ile denenmiştir. ÇKA ile 20 farklı deneme, RTF ile 11 farklı deneme gerçekleştirilmiştir. Daha sonra başarımı en yüksek olan sonuçlar için K-katmanlı çapraz doğrulama yöntemi uygulanmıştır.

ÇKA için en iyi ağırlık tahmini yapan 5 yöntemin farklı parametreleri Çizelge 5.1'de ve bu yöntemlerin sonuçları da Çizelge 5.2'de gösterilmiştir. Başarımı en yüksek bu yöntemler için; tek gizli katman, öğrenme oranı artım oranı 1.05, öğrenme oranı azaltma oranı 0.7, momentum katsayısı 0.9 (Öğrenme fonksiyonu için GDX kullanılan

denemelerde geçerlidir.), iterasyon sayısı 30,000 ve öz nitelik vektörlerinden dik açı ile elde edilmiş; cismin yüzey alanı, çevresi ve yuvarlaklığı ortak olarak kullanılmıştır.

Çizelge 5.1 ÇKA'da başarımı en yüksek denemelerin parametre değerleri

	Deneme 19	**Deneme 9**	**Deneme 5**	**Deneme 8**	**Deneme 14**
Gizli Katman Nöron Sayısı	40	40	20	20	40
Gizli Katman Aktivasyon Fonk.	Tanjant Sigmoid	Tanjant Sigmoid	Tanjant Sigmoid	Lineer	Tanjant Sigmoid
Çıkış Katmanı Aktivasyon Fonk.	Tanjant Sigmoid	Tanjant Sigmoid	Tanjant Sigmoid	Lineer	Tanjant Sigmoid
Öğrenme Fonksiyonu	GDX	GDX	GDX	GDX	GDA
Öğrenme Oranı	0.5	0.1	0.5	0.1	0.5

Çizelge 5.2 ÇKA'da başarımı en yüksek denemelerin başarımları

	Deneme 19	**Deneme 9**	**Deneme 5**	**Deneme 8**	**Deneme 14**
Test (MAE)	0.90 gr	0.83 gr	0.84 gr	0.79 gr	0.91 gr
Test (R)	0.9916	0.9940	0.9933	0.9945	0.9930
Doğrulama (MAE)	0.83 gr	0.83 gr	0.87 gr	0.86 gr	0.84 gr
Doğrulama (R)	0.9938	0.9943	0.9946	0.9931	0.9935
Genel (MAE)	0.80 gr	0.81 gr	0.81 gr	0.83 gr	0.83 gr
Genel (R)	0.9942	0.9942	0.9941	0.9939	0.9938

Başarımı en yüksek olan "Deneme 19" için yumurtaların sınıflandırma başarımı Çizelge 5.3'te gösterilmiştir. "Deneme 19" için, daha yüksek ağırlık sınıfında olmasına rağmen hatalı olarak daha düşük sınıfta ölçülen yumurtalara göre elde edilmiş sınıflandırma başarımı ise Çizelge 5.4'te gösterilmiştir.

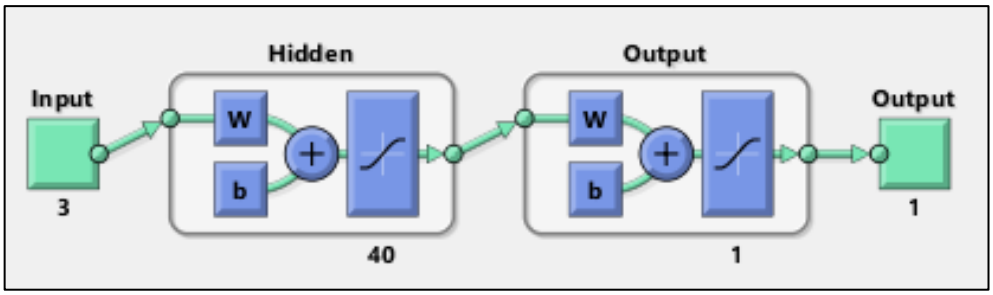
Çizelge 5.3 Yumurtaların genel sınıflandırma başarımı

	Toplam Yumurta Örneği	**Doğru Sınıflandırılan**	**Yanlış Sınıflandırılan**	**Doğruluk**
1. İterasyon	38	37	1	97.4 %
2. İterasyon	38	35	3	92.1 %
3. İterasyon	38	36	2	94.7 %
4. İterasyon	38	33	5	86.8 %
5. İterasyon	38	31	7	81.6 %
Aritmetik Ortalama	**190**	**172**	**18**	**90.5 %**

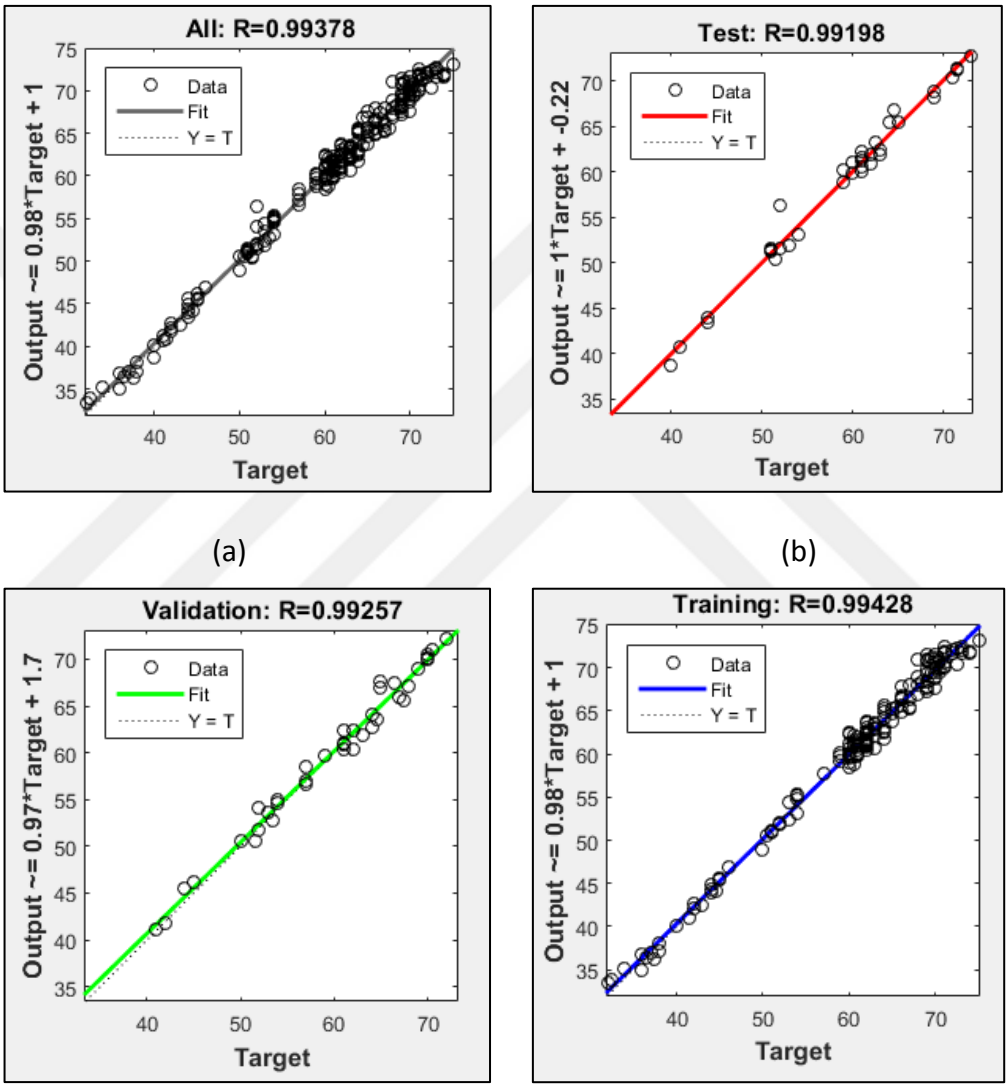
Çizelge 5.4 Yüksek ağırlık sınıfına, düşük ağırlıklı yumurtaların karışımına göre sınıflandırma başarımı

	Toplam Yumurta Örneği	**Doğru Sınıflandırılan**	**Yanlış Sınıflandırılan**	**Doğruluk**
1. İterasyon	38	37	1	97.4 %
2. İterasyon	38	35	3	92.1 %
3. İterasyon	38	38	0	100 %
4. İterasyon	38	37	1	97.4 %
5. İterasyon	38	34	4	89.5 %
Aritmetik Ortalama	**190**	**172**	**18**	**95.3 %**

“Deneme 19” için Matlab programı üzerinden elde edilen YSA yapısı Şekil 5.1’de gösterilmiştir. Aynı deneme için Matlab programı üzerinden elde edilen; bütün veriler, eğitim, test ve doğrulama verileri için regresyon grafikleri Şekil 5.2’de gösterilmiştir.

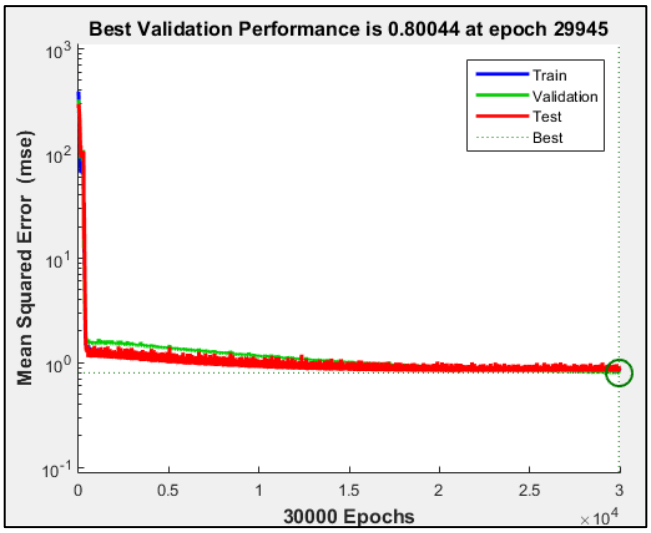


Şekil 5.1 ÇKA "Deneme 19" YSA yapısı

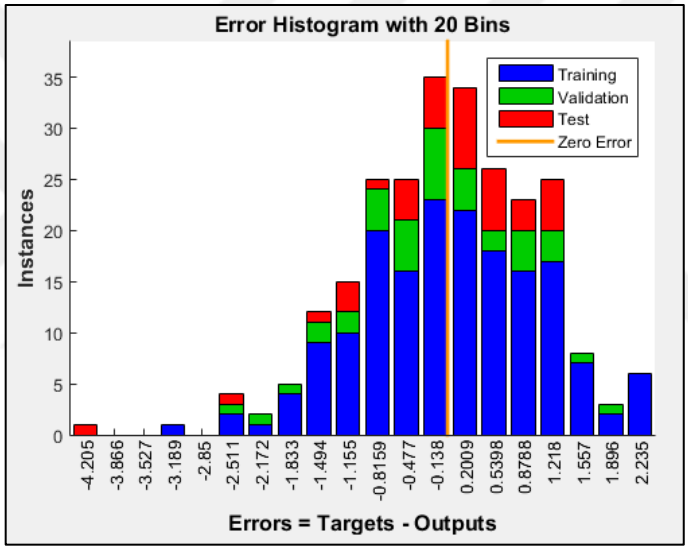


Şekil 5.2 ÇKA "Deneme 19" regresyon grafikleri (a) Bütün veriler (b) Test verileri (c) Doğrulama verileri (d) Eğitim verileri

"Deneme 19" için elde edilen örnek bir sistem performansı ve hata histogramı sırasıyla Şekil 5.3'te ve Şekil 5.4'te gösterilmiştir.



Şekil 5.3 ÇKA "Deneme 19" sistem performansı



Şekil 5.4 ÇKA "Deneme 19" hata histogramı

YSA modeli ÇKA ile elde edilen, başarımı en yüksek olan "Deneme 19" için 250 yumurta örneğinden elde edilen görüntülerden seçilen eğitim, test ve doğrulama kümesi oranları değiştirilerek başarım oranları tekrar hesaplanmıştır. Veri kümesi 1, veri tabanından ilgili cisim görüntülerinin %50'sinin eğitim, %25'inin test ve %25'inin doğrulama verisi olacak şekilde seçilmesiyle oluşturulmuştur. Veri kümesi 2 ise, veri tabanından ilgili cisim görüntülerinin %34'ünün eğitim, %33'ünün test ve %33'nün doğrulama verisi olacak şekilde seçilmesiyle oluşturulmuştur. İlgili başarım sonuçlar Çizelge 5.5'te gösterilmiştir.

Çizelge 5.5 “Deneme 19”un veri kümesi oranlarındaki değişimin başarıma etkisi

	Test (MAE)	**Test (R)**	**Doğrulama (MAE)**	**Doğrulama (R)**	**Genel (MAE)**	**Genel (R)**
Veri Kümesi 1	0.88 gr	0.9930	0.92 gr	0.9929	0.84 gr	0.9936
Veri Kümesi 2	1.01 gr	0.9899	0.77 gr	0.9910	0.91 gr	0.9922

Başarımı en yüksek “Deneme 19” için KKÇDY, K=5 için uygulanmış sonuçları Çizelge 5.6’da gösterilmiştir.

Çizelge 5.6 “Deneme 19”un KKÇDY başarımı

	Test (MAE)	**Test (R)**	**Genel (MAE)**	**Genel (R)**
K=1	0.87 gr	0.9926	0.83 gr	0.9939
K=2	0.88 gr	0.9929	0.76 gr	0.9947
K=3	0.80 gr	0.9942	0.82 gr	0.9942
K=4	0.93 gr	0.9937	0.83 gr	0.9940
K=5	0.86 gr	0.9932	0.80 gr	0.9943
Aritmetik Ortalama	**0.87 gr**	**0.9933**	**0.81 gr**	**0.9942**

RTF için en iyi ağırlık tahmini yapan 5 yöntemin farklı parametreleri Çizelge 5.7’de ve bu yöntemlerin sonuçları da Çizelge 5.8’de gösterilmiştir. Başarımı en yüksek bu yöntemler için; karesel ortalama hata hedefi 1, ekranlar arasına eklenecek nöron sayısı 25 ve maksimum nöron sayısı giriş vektör sayısına eşit olacak şekilde tanımlanmıştır. “Deneme 3” ve “Deneme 4”te dik açı ile elde edilmiş bütün öz nitelik vektörleri kullanılmıştır. “Deneme 2”de dik açı ile elde edilmiş bütün öz nitelik vektörleri kullanılmış fakat yuvarlaklık öz nitelik vektörü yerine elipsin odak noktaları ve elipsin büyük çap uzunluğu arasındaki oranı veren, dış bükeylik (eccentricity) değeri kullanılmıştır. “Deneme 7”de ise dik açı ile elde edilmiş öz nitelik vektörleri; cismin yüzey alanı, cismin çevresi, MEUKN ve MEYKN kullanılmıştır. Bunların yanı sıra, “Deneme 9”da ise 3 farklı açıdan elde edilmiş öz nitelik vektörleri; cismin yüzey alanı, cismin çevresi, MEUKN ve MEYKN aynı anda YSA’ya uygulanmıştır.

Çizelge 5.7 RTF'de başarımı en yüksek denemelerin parametre değerleri

	Deneme 3	**Deneme 7**	**Deneme 2**	**Deneme 9**	**Deneme 4**
Öz Nitelik Vektörleri	5	4	5	12	5
Yayılma Değeri	100	100	1	100	1000

Çizelge 5.8 RTF'de başarımı en yüksek denemelerin başarımları

	Deneme 3	**Deneme 7**	**Deneme 2**	**Deneme 9**	**Deneme 4**
Test (MAE)	0.78 gr	0.87 gr	0.84 gr	0.90 gr	0.80 gr
Test (R)	0.9947	0.9933	0.9936	0.9921	0.9948
Doğrulama (MAE)	0.88 gr	0.85 gr	0.85 gr	0.77 gr	0.90 gr
Doğrulama (R)	0.9921	0.9940	0.9937	0.9953	0.9937
Genel (MAE)	0.81 gr	0.81 gr	0.81 gr	0.80 gr	0.82 gr
Genel (R)	0.9943	0.9943	0.9942	0.9942	0.9941

YSA modeli RTF ile elde edilen, başarımı en yüksek olan "Deneme 3" için yumurtaların sınıflandırma başarımı Çizelge 5.9'da gösterilmiştir.

Çizelge 5.9 Yumurtaların genel sınıflandırma başarımı

	Toplam Yumurta Örneği	**Doğru Sınıflandırılan**	**Yanlış Sınıflandırılan**	**Doğruluk**
1. İterasyon	38	32	6	84.2 %
2. İterasyon	38	32	6	84.2 %
3. İterasyon	38	32	6	84.2 %
4. İterasyon	38	34	4	89.5 %
5. İterasyon	38	34	4	89.5 %
Aritmetik Ortalama	**190**	**164**	**26**	**86.3 %**

“Deneme 3” için, daha yüksek ağırlık sınıfında olmasına rağmen hatalı olarak daha düşük sınıfta ölçülen yumurtalara göre elde edilmiş sınıflandırma başarımı ise Çizelge 5.10’da gösterilmiştir.

Çizelge 5.10 Yüksek ağırlık sınıfına, düşük ağırlıklı yumurtaların karışımına göre sınıflandırma başarımı

	Toplam Yumurta Örneği	**Doğru Sınıflandırılan**	**Yanlış Sınıflandırılan**	**Doğruluk**
1. İterasyon	38	35	3	92.1 %
2. İterasyon	38	36	2	94.7 %
3. İterasyon	38	35	3	92.1 %
4. İterasyon	38	36	2	94.7 %
5. İterasyon	38	34	4	89.5 %
Aritmetik Ortalama	**190**	**176**	**14**	**92.6 %**

Başarımı en yüksek olan “Deneme 3” için 250 yumurta örneğinden elde edilen görüntülerden seçilen eğitim, test ve doğrulama kümesi oranları değiştirilerek başarım oranları tekrar hesaplanmıştır. Veri kümesi 1, veri tabanından ilgili cisim görüntülerinin %50’sinin eğitim, %25’inin test ve %25’inin doğrulama verisi olacak şekilde seçilmesiyle oluşturulmuştur. Veri kümesi 2 ise, veri tabanından ilgili cisim görüntülerinin %34’ünün eğitim, %33’ünün test ve %33’nün doğrulama verisi olacak şekilde seçilmesiyle oluşturulmuştur. İlgili başarım sonuçlar Çizelge 5.11’de gösterilmiştir.

Çizelge 5.11 “Deneme 3”ün veri kümesi oranlarındaki değişimin başarıma etkisi

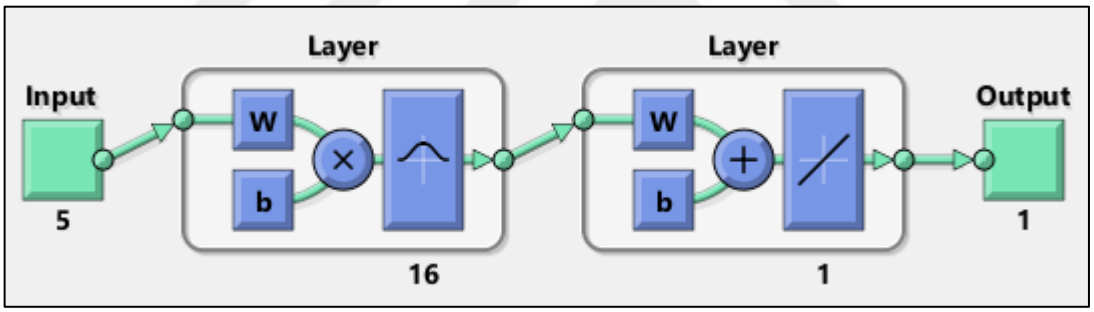
	Test (MAE)	**Test (R)**	**Doğrulama (MAE)**	**Doğrulama (R)**	**Genel (MAE)**	**Genel (R)**
Veri Kümesi 1	0.84 gr	0.9937	0.88 gr	0.9938	0.83 gr	0.9940
Veri Kümesi 2	0.86 gr	0.9937	0.87 gr	0.9933	0.84 gr	0.9938

Başarımı en yüksek “Deneme 3” için KKÇDY, K=5 için uygulanmış sonuçları Çizelge 5.12’de gösterilmiştir.

Çizelge 5.12 “Deneme 3”ün KKÇDY başarımı

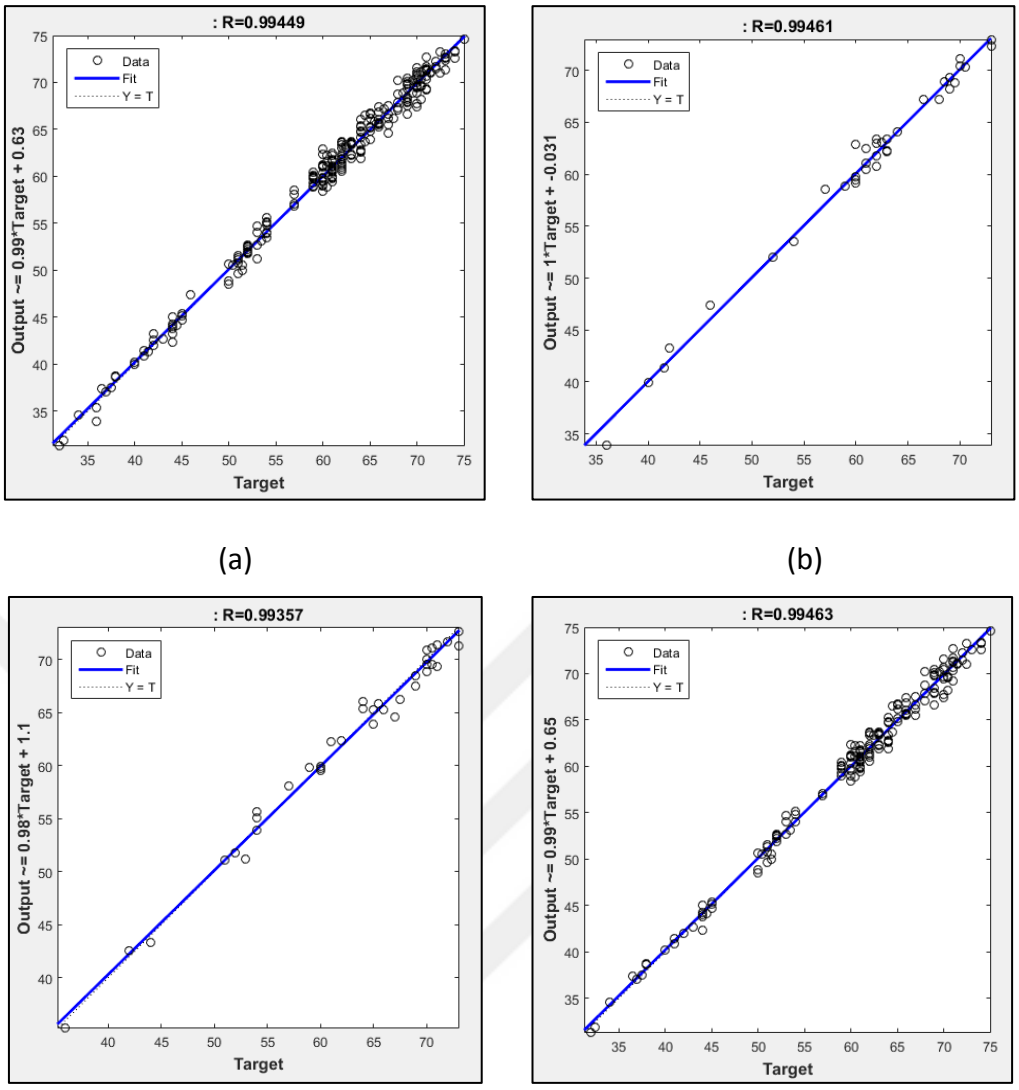
	Test (MAE)	**Test (R)**	**Genel (MAE)**	**Genel (R)**
K=1	0.88 gr	0.9947	0.81 gr	0.9942
K=2	0.81 gr	0.9939	0.80 gr	0.9944
K=3	0.80 gr	0.9943	0.80 gr	0.9943
K=4	0.78 gr	0.9946	0.80 gr	0.9942
K=5	0.80 gr	0.9946	0.79 gr	0.9945
Aritmetik Ortalama	**0.81 gr**	**0.9944**	**0.80 gr**	**0.9943**

“Deneme 3” için Matlab programı üzerinden elde edilen YSA yapısı Şekil 5.5’te gösterilmiştir.

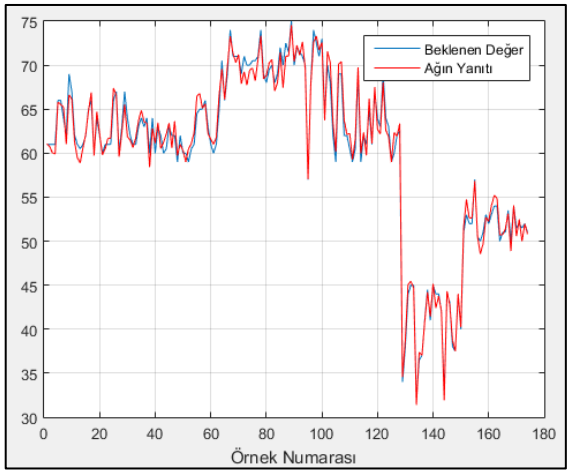


Şekil 5.5 RTF “Deneme 3” YSA yapısı

“Deneme 3” için Matlab programı üzerinden elde edilen; bütün veriler, eğitim, test ve doğrulama verileri için regresyon grafikleri Şekil 5.6’da gösterilmiştir. Aynı deneme için sistemin eğitilmesi aşamasında beklenen değer ile ağ tarafından üretilen çıktı Şekil 5.7’de gösterilmiştir.



Şekil 5.6 RTF "Deneme 3" regresyon grafikleri (a) Bütün veriler (b) Test verileri (c) Doğrulama verileri (d) Eğitim verileri



Şekil 5.7 RTF "Deneme 3" eğitim performansı

5.2 Portakal Simülasyonu

Portakal örnekleri üzerinde yapılan çalışmalar, ÇKA ve RTF YSA modelleri ile denenmiştir. ÇKA ile 21 farklı deneme, RTF ile 11 farklı deneme gerçekleştirilmiştir. Daha sonra başarımı en yüksek olan sonuçlar için K-katmanlı çapraz doğrulama yöntemi uygulanmıştır.

ÇKA için en iyi ağırlık tahmini yapan 5 yöntemin farklı parametreleri Çizelge 5.13'te ve bu yöntemlerin sonuçları da Çizelge 5.14'te gösterilmiştir. Başarımı en yüksek bu yöntemler için; tek gizli katman, öğrenme oranı 0.5, öğrenme oranı artım oranı 1.05, öğrenme oranı azaltma oranı 0.7, momentum katsayısı 0.9 (Öğrenme fonksiyonu için GDX kullanılan denemelerde geçerlidir.), iterasyon sayısı 30,000 ve öz nitelik vektörlerinden üç farklı açı ile elde edilmiş; cismin yüzey alanı, çevresi, MEUKN ve MEYKN ortak olarak kullanılmıştır. "Deneme 15" ve "Deneme 19" için 3 farklı açıdan elde edilmiş bu öz nitelik vektörleri direk olarak YSA'ya uygulanmıştır. "Deneme 21" ve "Deneme 17" için cismin yüzey alanının 3 farklı açıdan çekilmiş görüntüye göre elde edilen değerlerin geometrik ortalaması, diğer öz nitelik vektörlerinin ise aritmetik ortalaması alınarak elde edilmiş 4 giriş YSA'ya uygulanmıştır. "Deneme 16"da ise 3 farklı açıdan çekilmiş ortak öz nitelik vektörlerinin geometrik ortalaması alınarak elde edilmiş 4 giriş, YSA'ya uygulanmıştır.

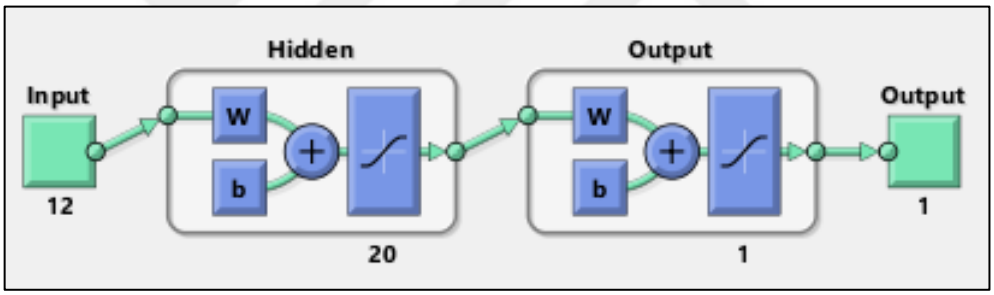
Çizelge 5.13 ÇKA'da başarımı en yüksek denemelerin parametre değerleri

	Deneme 15	**Deneme 21**	**Deneme 16**	**Deneme 17**	**Deneme 19**
Giriş Sayısı	12	4	4	4	12
Gizli Katman Nöron Sayısı	20	20	20	20	20
Gizli Katman Aktivasyon Fonk.	Tanjant Sigmoid	Tanjant Sigmoid	Tanjant Sigmoid	Tanjant Sigmoid	Tanjant Sigmoid
Çıkış Katmanı Aktivasyon Fonk.	Tanjant Sigmoid	Tanjant Sigmoid	Tanjant Sigmoid	Tanjant Sigmoid	Tanjant Sigmoid
Öğrenme Fonksiyonu	GDA	GDX	GDA	GDX	GDX

Çizelge 5.14 ÇKA'da başarımı en yüksek denemelerin başarımları

	Deneme 15	**Deneme 21**	**Deneme 16**	**Deneme 17**	**Deneme 19**
Test (MAE)	3.30 gr	3.20 gr	3.05 gr	3.62 gr	3.97 gr
Test (R)	0.9695	0.9581	0.9717	0.9732	0.9581
Doğrulama (MAE)	2.69 gr	3.07 gr	3.66 gr	2.66 gr	3.07 gr
Doğrulama (R)	0.9777	0.9770	0.9590	0.9803	0.9702
Genel (MAE)	2.58 gr	2.64 gr	2.84 gr	2.89 gr	2.70 gr
Genel (R)	0.9820	0.9800	0.9783	0.9775	0.9772

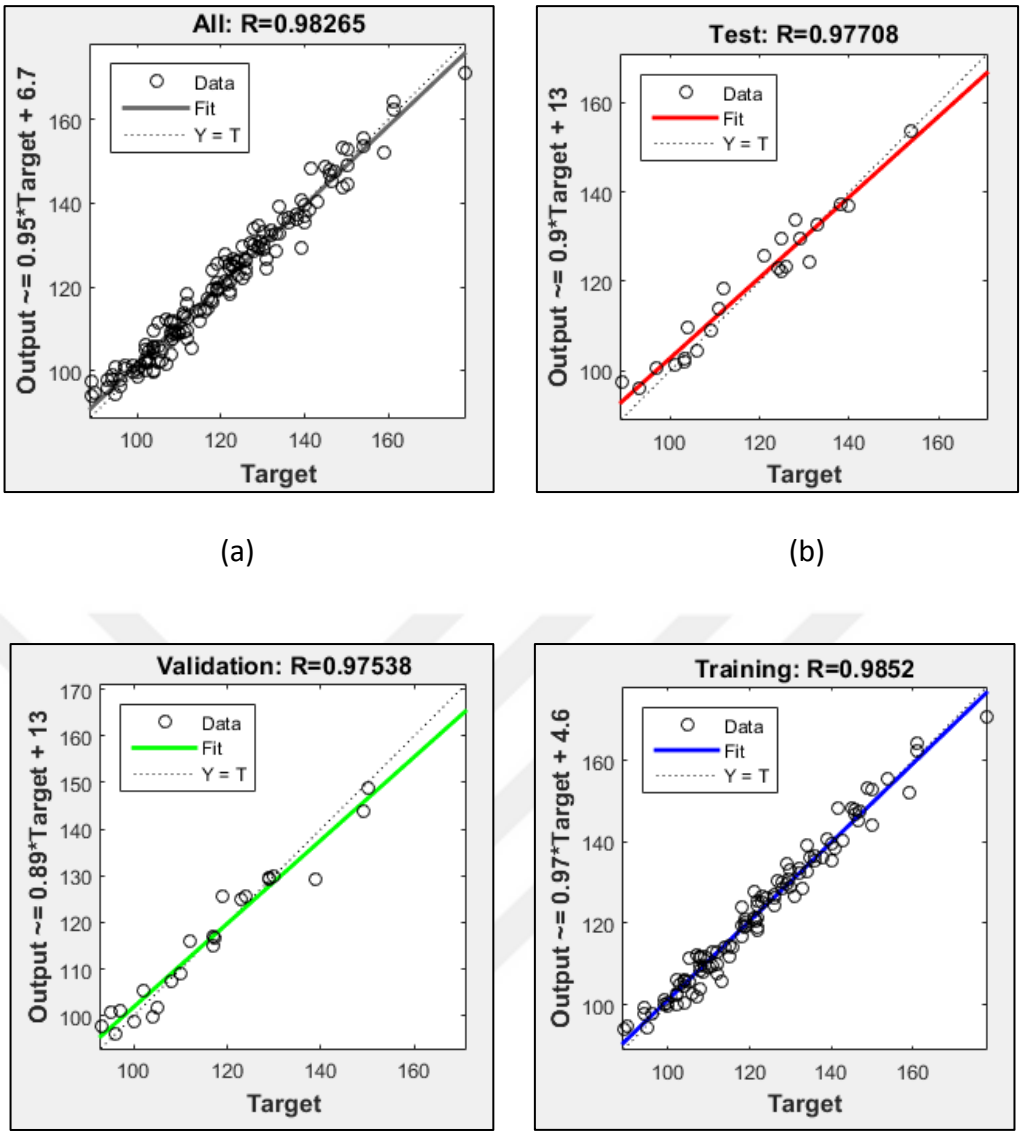
"Deneme 15" için Matlab programı üzerinden elde edilen YSA yapısı Şekil 5.8'de gösterilmiştir.



Şekil 5.8 ÇKA "Deneme 15" YSA yapısı

Aynı deneme için Matlab programı üzerinden elde edilen; bütün veriler, eğitim, test ve doğrulama verileri için regresyon grafikleri Şekil 5.9'da gösterilmiştir.

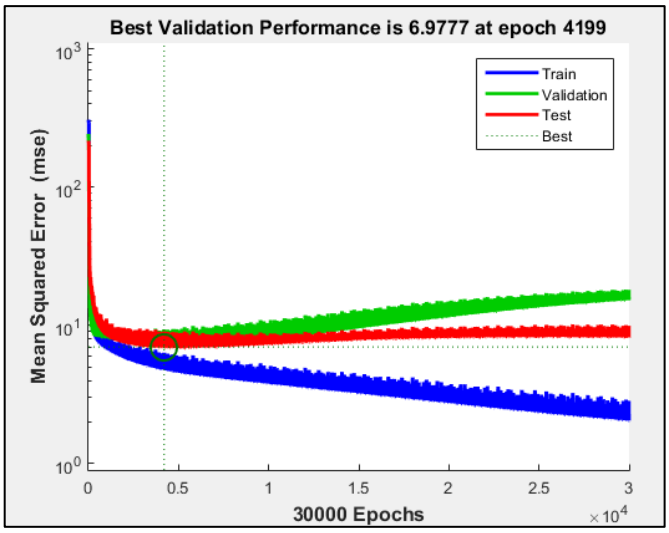
"Deneme 15" için elde edilen, örnek bir sistem performansı ve hata histogramı ise sırasıyla Şekil 5.10'da ve Şekil 5.11'de gösterilmiştir.



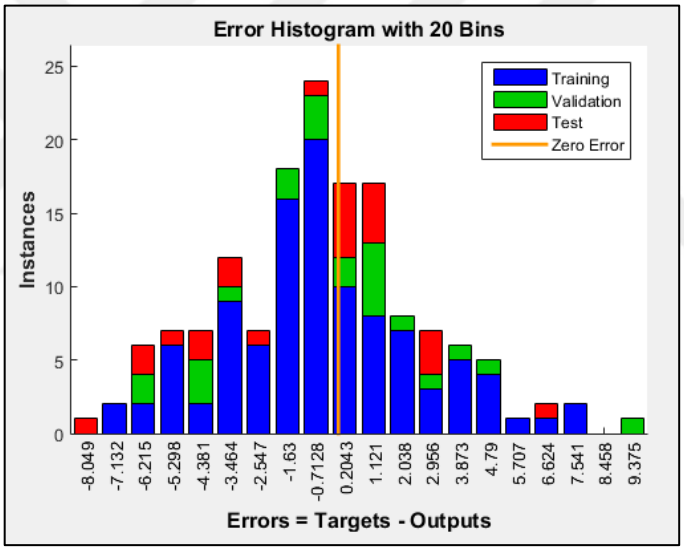
(a) (b)

(c) (d)

Şekil 5.9 ÇKA "Deneme 15" regresyon grafikleri (a) Bütün veriler (b) Test verileri (c) Doğrulama verileri (d) Eğitim verileri



Şekil 5.10 ÇKA "Deneme 15" sistem performansı



Şekil 5.11 ÇKA "Deneme 15" hata histogramı

YSA modeli ÇKA ile elde edilen, başarımı en yüksek olan "Deneme 15" için 150 portakal örneğinden elde edilen görüntülerden seçilen eğitim, test ve doğrulama kümesi oranları değiştirilerek başarım oranları tekrar hesaplanmıştır. Veri kümesi 1, veri tabanından ilgili cisim görüntülerinin %50'sinin eğitim, %25'inin test ve %25'inin doğrulama verisi olacak şekilde seçilmesiyle oluşturulmuştur. Veri kümesi 2 ise, veri tabanından ilgili cisim görüntülerinin %34'ünün eğitim, %33'ünün test ve %33'nün doğrulama verisi olacak şekilde seçilmesiyle oluşturulmuştur. İlgili başarım sonuçlar Çizelge 5.15'te gösterilmiştir.

Çizelge 5.15 “Deneme 15”in veri kümesi oranlarındaki değişimin başarıma etkisi

	Test (MAE)	**Test (R)**	**Doğrulama (MAE)**	**Doğrulama (R)**	**Genel (MAE)**	**Genel (R)**
Veri Kümesi 1	3.63 gr	0.9621	3.33 gr	0.9687	3.03 gr	0.9744
Veri Kümesi 2	3.74 gr	0.9638	3.27 gr	0.9710	3.27 gr	0.9717

Başarımı en yüksek “Deneme 15” için KKÇDY, K=5 için uygulanmış sonuçları Çizelge 5.16’da gösterilmiştir.

Çizelge 5.16 “Deneme 15”in KKÇDY başarımı

	Test (MAE)	**Test (R)**	**Genel (MAE)**	**Genel (R)**
K=1	3.72 gr	0.9828	2.08 gr	0.9869
K=2	2.92 gr	0.9680	1.67 gr	0.9905
K=3	3.85 gr	0.9069	2.08 gr	0.9809
K=4	3.92 gr	0.9592	1.94 gr	0.9857
K=5	2.96 gr	0.9846	2.77 gr	0.9825
Aritmetik Ortalama	**3.47 gr**	**0.9603**	**2.11 gr**	**0.9853**

RTF için en iyi ağırlık tahmini yapan 5 yöntemin farklı parametreleri Çizelge 5.17’de gösterilmiştir.

Çizelge 5.17 RTF’de başarımı en yüksek denemelerin parametre değerleri

	Deneme 9	**Deneme 10**	**Deneme 11**	**Deneme 5**	**Deneme 7**
Giriş Sayısı	12	4	4	5	4
Öz Nitelik Vektörleri	4	4	4	5	4
Yayılma Değeri	100	100	1	100	1000

Bu yöntemlerin sonuçları da Çizelge 5.18’de gösterilmiştir. Başarımı en yüksek bu yöntemler için; karesel ortalama hata hedefi 1, ekranlar arasına eklenecek nöron sayısı

25 ve maksimum nöron sayısı giriş vektör sayısına eşit olacak şekilde tanımlanmıştır. “Deneme 9”, “Deneme 10”, “Deneme 11” ve “Deneme 7” için öz nitelik vektörlerinden; cismin yüzey alanı, çevresi, MEUKN ve MEYKN ortak olarak kullanılmıştır. “Deneme 9” da 3 farklı açıdan elde edilmiş bu öz nitelik vektörleri YSA’ya uygulanmıştır. “Deneme 10”da 3 farklı açıdan çekilmiş ortak öz nitelik vektörlerinin geometrik ortalaması alınarak elde edilmiş 4 giriş, ağırlık tahmini için kullanılmıştır. “Deneme 11”de cismin yüzey alanının 3 farklı açıdan çekilmiş görüntüye göre elde edilen değerlerin geometrik ortalaması, diğer öz nitelik vektörlerinin ise aritmetik ortalaması alınarak elde edilmiş 4 giriş YSA’ya uygulanmıştır. “Deneme 7”de bu öz nitelik vektörlerinin sadece dik açıyla çekilmiş görüntülerinden elde edilen değerler YSA’nın girişi olarak uygulanmıştır. Son olarak “Deneme 5”de ise dik açıyla elde çekilmiş görüntülerden elde edilen bütün öz nitelik vektörleri kullanılmıştır.

Çizelge 5.18 RTF’de başarımı en yüksek denemelerin başarımları

	Deneme 9	**Deneme 10**	**Deneme 11**	**Deneme 5**	**Deneme 7**
Test (MAE)	2.74 gr	2.95 gr	2.63 gr	3.65 gr	3.43 gr
Test (R)	0.9760	0.9786	0.9808	0.9621	0.9612
Doğrulama (MAE)	2.73 gr	2.47 gr	2.95 gr	3.05 gr	3.17 gr
Doğrulama (R)	0.9767	0.9835	0.9782	0.9755	0.9710
Genel (MAE)	2.42 gr	2.62 gr	2.58 gr	3.12 gr	3.10 gr
Genel (R)	0.9825	0.9808	0.9808	0.9727	0.9726

Başarımı en yüksek olan “Deneme 9” için 150 portakal örneğinden elde edilen görüntülerden seçilen eğitim, test ve doğrulama kümesi oranları değiştirilerek başarım oranları tekrar hesaplanmıştır. Veri kümesi 1, veri tabanından ilgili cisim görüntülerinin %50’sinin eğitim, %25’inin test ve %25’inin doğrulama verisi olacak şekilde seçilmesiyle oluşturulmuştur. Veri kümesi 2 ise, veri tabanından ilgili cisim görüntülerinin %34’ünün eğitim, %33’ünün test ve %33’nün doğrulama verisi olacak şekilde seçilmesiyle oluşturulmuştur. İlgili başarım sonuçlar Çizelge 5.19’da gösterilmiştir.

Çizelge 5.19 “Deneme 9”un veri kümesi oranlarındaki değişimin başarıma etkisi

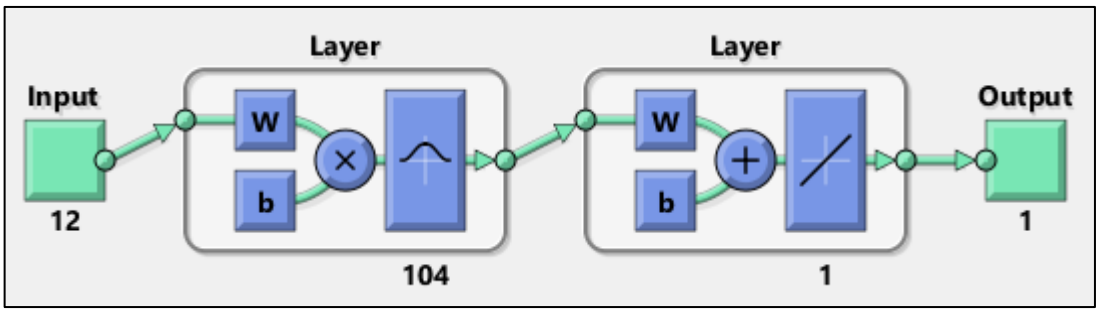
	Test (MAE)	Test (R)	Doğrulama (MAE)	Doğrulama (R)	Genel (MAE)	Genel (R)
Veri Kümesi 1	1.13 gr	0.9550	1.27 gr	0.9569	1.17 gr	0.9566
Veri Kümesi 2	1.30 gr	0.9310	1.21 gr	0.9537	1.17 gr	0.9536

Başarımı en yüksek “Deneme 9” için KKÇDY, K=5 için uygulanmış sonuçları Çizelge 5.20’de gösterilmiştir.

Çizelge 5.20 “Deneme 9”un KKÇDY başarımı

	Test (MAE)	Test (R)	Genel (MAE)	Genel (R)
K=1	2.81 gr	0.9835	2.40 gr	0,9836
K=2	2.82 gr	0.9659	2.44 gr	0,9823
K=3	2.66 gr	0.9770	2.49 gr	0,9828
K=4	3.42 gr	0.9685	2.38 gr	0,9830
K=5	2.18 gr	0.9901	2.34 gr	0,9839
Aritmetik Ortalama	**2.78 gr**	**0.9770**	**2.41 gr**	**0,9831**

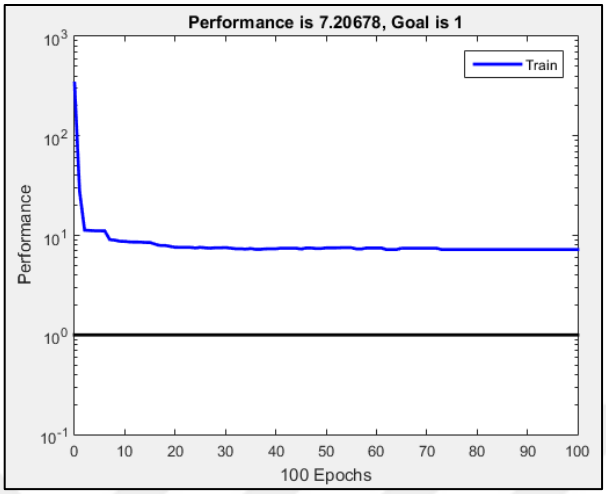
“Deneme 9” için Matlab programı üzerinden elde edilen YSA yapısı Şekil 5.12’de gösterilmiştir.



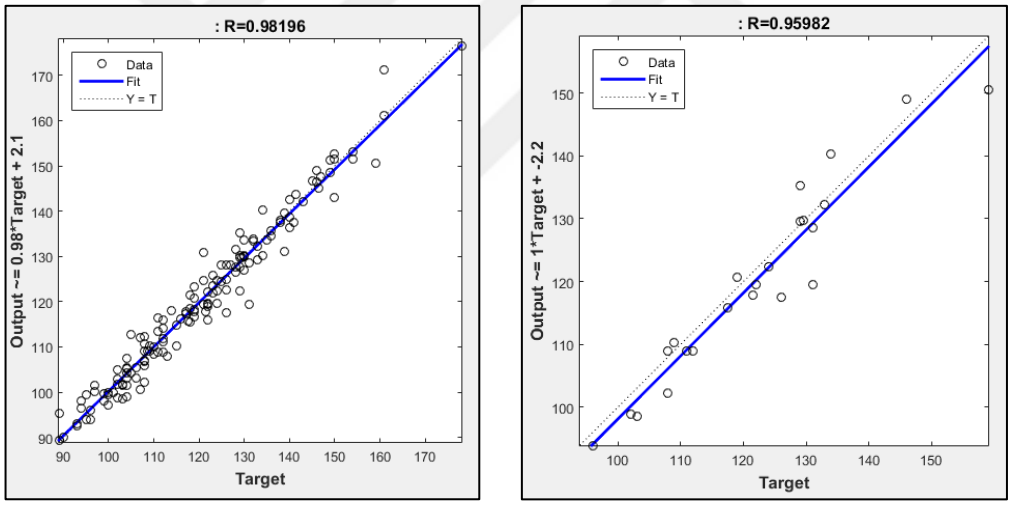
Şekil 5.12 RTF “Deneme 9” YSA yapısı

“Deneme 9” için Matlab programı üzerinden elde edilen sistem performansı ve bütün veriler, eğitim, test ve doğrulama verileri için regresyon grafikleri sırasıyla Şekil 5.13 ve

Şekil 5.14'te gösterilmiştir. Aynı deneme için sistemin eğitilmesi aşamasında beklenen değer ile ağ tarafından üretilen çıktı ise Şekil 5.15'te gösterilmiştir.

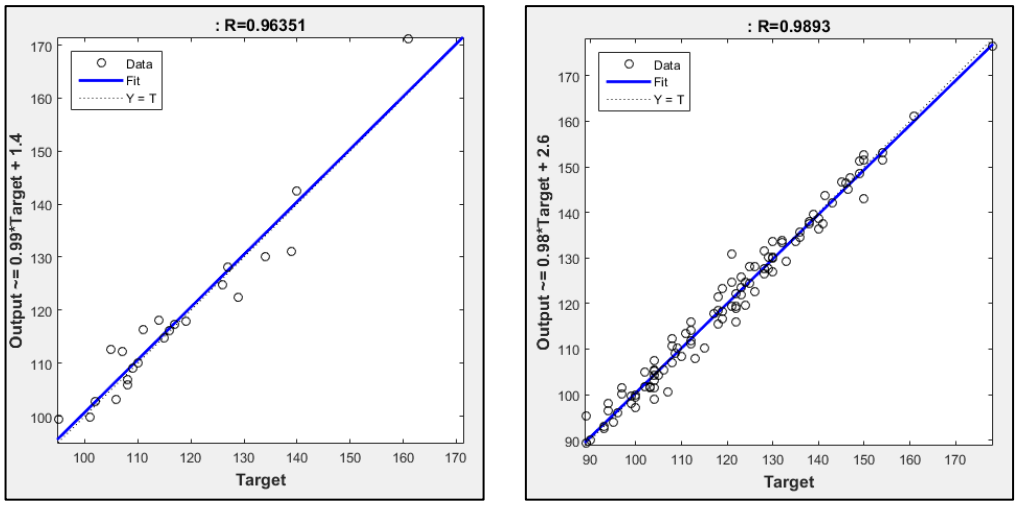


Şekil 5.13 RTF "Deneme 9" için eğitim performansı



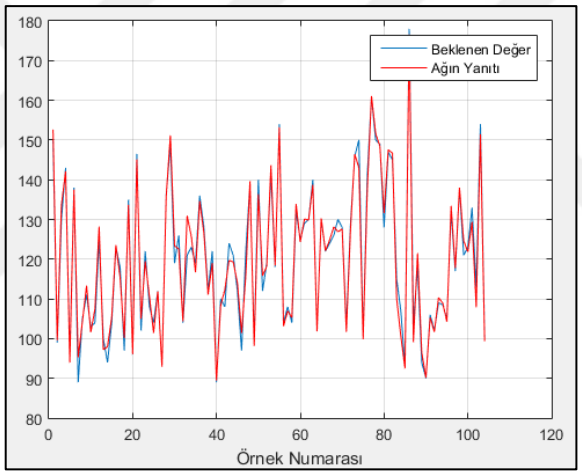
(a) (b)

Şekil 5.14 RTF "Deneme 9" regresyon grafikleri (a) Bütün veriler (b) Test verileri (c) Doğrulama verileri (d) Eğitim verileri



(c) (d)

Şekil 5.14 RTF "Deneme 9" regresyon grafikleri (a) Bütün veriler (b) Test verileri (c) Doğrulama verileri (d) Eğitim verileri (devamı)



Şekil 5.15 RTF "Deneme 9" sistem performansı

BÖLÜM 6

SONUÇ VE ÖNERİLER

6.1 Yumurta Örneği

Yumurta örneği üzerinde yapılmış olan çalışmada 250 adet yumurta örneği ile yapay sinir ağı kullanılarak uzaklıktan bağımsız ağırlık tahmini gerçekleştirilmiştir. Yapılan deneyler sonucunda farklı yöntemler için farklı başarımlar elde edilmiştir. Farklı YSA modelleri için en iyi ağırlık tahmini yapan 5 denemede ise başarımlarının oldukça birbirine yakın olduğu ve oldukça yüksek başarım sağlandığı gözlenmiştir. Çok küçük farklar olmasına rağmen ÇKA modeli için gerçekleştirilen 5 deneme arasından genel sistem başarımı daha yüksek olan "Deneme 19" seçilmiştir. Bu yöntemde korelasyon katsayısı (R) 0.9942 ve ortalama mutlak hata (mae) 0.80 gr olarak hesaplanmıştır. Sistemde tespit edilen en büyük mutlak hatanın ise 3.19 gr olduğu gözlenmiştir. Aynı şekilde çok küçük farklar olan RTF modeli için gerçekleştirilen 5 deneme arasında ise genel başarımı daha yüksek olan "Deneme 3" seçilmiştir. Bu yöntemde korelasyon katsayısı (R) 0.9943 ve ortalama mutlak hata (mae) 0.81 gr olarak hesaplanmıştır. Sistemde tespit edilen en büyük mutlak hatanın ise 2.93 gr olduğu gözlenmiştir.

Adana'da satışa sunulan yumurtalar üzerinde yapılan bir çalışmada, yumurtaların paket üzerinde yazan ağırlıklara uyma oranının markalara göre %95 ile %60 arasında değiştiği belirtilmiştir [37]. Türk Gıda Kodeksi Yumurta Tebliğine göre 180 örnek ve fazlasının kontrol edilmesi durumunda bir alt ağırlık sınıfından karışma oranı %5'i, 180 örnekten daha az yumurtanın kontrol edilmesi durumunda ise %10'u geçmemelidir. Üst ağırlık sınıflarında bulunan yumurtalar için ise herhangi bir sınırlama yoktur.

Hassas tartı ile yapılan deneylerde 4 farklı markadan alınan yumurtaların, yumurta kutularının üzerinde yazan sınıflandırma değerlerine göre alt ağırlık sınıfından karışma oranının %11, genel doğruluk oranının ise %47 olduğu görülmüştür. Gerçekleştirilen bu çalışma sonucunda ise ÇKA için alt ağırlık sınıfından karışma oranı %5'e düşürülmüş ve genel doğruluk oranı %90.5'e çıkarılmıştır. RTF için ise alt ağırlık sınıfından karışma oranı %7.5'e düşürülmüş ve genel doğruluk oranı %86.3'e çıkarılmıştır.

6.2 Portakal Örneği

Yumurta örneği üzerinde yapılmış olan çalışmada 150 adet portakal örneği ile yapay sinir ağı kullanılarak uzaklıktan bağımsız ağırlık tahmini gerçekleştirilmiştir. Yapılan deneyler sonucunda farklı yöntemler için farklı başarımlar elde edilmiştir. Farklı YSA modelleri için yumurta örneğinde de olduğu gibi en iyi ağırlık tahmini yapan 5 denemede başarımlarının oldukça birbirine yakın olduğu ve oldukça yüksek başarım sağlandığı gözlenmiştir. Çok küçük farklar olmasına rağmen ÇKA modeli için gerçekleştirilen 5 deneme arasından genel sistem başarımı daha yüksek olan "Deneme 15" seçilmiştir. Bu yöntemde korelasyon katsayısı (R) 0.9820 ve ortalama mutlak hata (mae) 2.58 gr olarak hesaplanmıştır. Sistemde tespit edilen en büyük mutlak hatanın ise 10.6 gr olduğu gözlenmiştir. Aynı şekilde çok küçük farklar olan RTF modeli için gerçekleştirilen 5 deneme arasında ise genel başarımı daha yüksek olan "Deneme 9" seçilmiştir. Bu yöntemde korelasyon katsayısı (R) 0.9825 ve ortalama mutlak hata (mae) 2.42 gr olarak hesaplanmıştır. Sistemde tespit edilen en büyük mutlak hatanın ise 11 gr olduğu gözlenmiştir.

6.3 Öneriler

Yapılan bu tez çalışmasında cisimlerin görüntüleri 3 farklı açıdan elde edilmiş ve sonuçlar bu açılar baz alınarak değerlendirilip, optimum ağırlık tahmini yapan sistem bulunmaya çalışılmıştır. Buna ek olarak 2 veya 3 kamera ile eş zamanlı görüntülerin alınarak sistemin verdiği tepki ve başarımlar ileriki çalışmalarda ölçülebilir.

Yumurtaların ağırlıkları ölçülürken üretim tarihi göz önünde bulundurulmamıştır. Üretim tarihinin eskiliğinden dolayı oluşacak bayatlamanın yumurta ağırlığı üzerindeki

etkisi de göz önünde bulundurularak sistem tekrar eğitilerek, sistemin başarımı daha da arttırılabilir.

Bu çalışmada yumurta ve portakal örnekleri kullanılmıştır. İleride gerçekleştirilecek olan çalışmalarda farklı cisimler kullanılabilir. Ayrıca hayvan ve insanların da ağırlıklarının hesaplanması konusunda benzer çalışmalar gerçekleştirilebilir.

Sebze ve meyveler üzerinde yapılan çalışmalarda, sabit kameralar yardımıyla günlük olarak sebze ya da meyvelerin ağırlıkları tespit edilerek, gelişim aşamasındaki ağırlıkları periyodik olarak bir veri tabanında saklanabilir. Böylelikle hastalıklı veya gerekli gelişim performansını gösteremeyen sebze veya meyveler tespit edilerek, erken müdahale sağlanabilir. Bu yöntem ile de üretim verimliliği arttırılabilir.

Derin öğrenme algoritmaları kullanılarak, kurulmuş olan sistemin performansı tekrar değerlendirilebilir.

Yazılmış olan Matlab program kodu üzerinde iyileştirmeler yapılarak veya daha hızlı bilgisayarlar kullanarak veya aynı anda birden fazla cismin ağırlığının tahmin edilmesi sağlanarak daha hızlı ağırlık tahmin etme işleminin gerçekleştirilmesi sağlanabilir.

KAYNAKLAR

[1] Çelebi Ş., Karaca H., (2006). "Yumurtanın besin değeri, kolesterol içeriği ve yumurtayı n-3 yağ asitleri bakımından zenginleştirmeye yönelik çalışmalar", Atatürk Üniversitesi Ziraat Fakültesi Dergisi, 37 (2):257-265.

[2] Gonzalez A., Sattelee DG., Moharer F., CADD GG., (1999). Factors Affecting Ostrich Egg Hatchability, Poultry Science, 78:1257-1262.

[3] Wilson HR, (1991). "Interrelationship of egg size, chick size, posthatching growth and hatchability", World's Poultry Science Journal, 47:5-20.

[4] Çelik Y., Şengül T., (2001). "Şanlıurfa İli Kentsel Alanında Tüketicilerin Yumurta Tüketim Düzeyleri ve Tüketim Alışkanlıklarının Belirlenmesi", Hayvansal Üretim, 42 (2):53-62.

[5] Lines J. A. Ve Ark., (2001). "An automatic image based system for estimating the mass of free swimming fish", Computer and Electronics in Agriculture, 31 (2):151-168.

[6] Hockaday s. ve Ark., (1997). "Using stereo image pairs to measure mass in strains of Atlantic salmon (Salmo salar L).", Sensors and their applications VIII, Institute of Physics Publishing, 7-10 September 1997, Bristol, 21–26.

[7] Omid M., Khojastehnazhand M., Tabatabaeefar A., (2010). "Estimating Volume And Mass of Citrus Fruits by Image Processing Technique", Journal of Food Engineering, 100 (2):315-321.

[8] Asadi V., Raoufat H. M., (2010). "Egg Weight Estimation by Machine Vision and Neural Network Techniques (A Case Study Fresh Egg)", Internatıonal Journal of Natural and Engineering Sciences, 4 (2):1-4.

[9] Javadikia P. ve Ark., (2011). "Measuring the Weight of Egg with Image Processing and ANFIS Model", "Swarm, Evolutionary and Memetic Computing",Part I, 19-21 December 2011, Visakhapatnam, 407-416.

[10] Kashiha M. ve Ark., (2014). "Weight Estimation of Pigs Using Top-View Image Processing", ICIAR, Part I, 22-24 October 2014, Vilamoure, 496-503.

[11] Alikhanov D. ve Ark., (2015). "Indirect Method for Egg Weight Measurement Using Image Processing", IJETAE, 5 (11):30-34.

[12] Chaithanya C., Priya S., (2015). “Object Weight Estimation From 2D Images”, ARPN Journal of Engineering and Applied Sciences, 10 (17):1-5.

[13] Sabancı K. ve Ark., (2016). “Weight Estimation of Wheat by Using Image Processing Techniques”, Journal of Image and Graphics, 4 (1):51-54.

[14] Gonzalez R. C., Woods R. E., (2007). “Digital Image Processing”, Third Edition, Prentice Hall, New Jersey.

[15] Lillesand T.M., R.W. Kiefer, (1994). “Remote Sensing and Photo Interpretation”, 3rd. Edition, John Wiley & Sons, New York.

[16] Taşdemir Ş. ve Ark., (2009). “Sayısal Görüntü Analiz İşleminde Kamera Kalibrasyon Parametrelerinin Belirlenmesi”, IATS, 13-15 Mayıs 2009, Karabük.

[17] Heikkila J., Silven O., (1997). “A Four-step Camera Calibration Procedure with Implicit Image Correction”, IEEE International Conference on Computer Vision and Pattern, 17-19 June 1997, San Juan.

[18] Ünsal B.F., (2009). “İki Boyutlu Doğrusal Koordinat Dönüşümleri”, 12. Türkiye Harita Bilimsel ve Teknik Kurultayı, 11-15 Mayıs 2009, Ankara.

[19] Graphics Mill, Affine and Projective Transformations, https://www.graphicsmill.com/docs/gm/affine-and-projective-transformations.htm, 5 Kasım 2017

[20] Aydar U., (2007). “Cephe Değerlendirmelerinde Fotogrametrik ve Görselleştirme Yöntemlerinin Kıyaslanması”, Yüksek Lisans Tezi, İTÜ Fen Bilimleri Enstitüsü, İstanbul.

[21] Çölkesen İ., (2009). “Uzaktan Algılamada İleri Sınıflandırma Tekniklerinin Karşılaştırılması ve Analizi”, Yüksek Lisans Tezi, Gebze Yüksek Teknoloji Enstitüsü Mühendislik ve Fen Bilimleri Enstitüsü, Gebze.

[22] Ölmez T., Dokur Z., (2009). “Uzman Sistemlerde Örüntü Tanıma: Yapay Sinir Ağları Genetik Algoritmalar Bulanık Mantık Makine Öğrenmesi”, İTÜ, 1. Baskı, İstanbul.

[23] Öztemel E., (2006). “Yapay Sinir Ağları”, Papatya Yayıncılık, 2. Baskı, İstanbul.

[24] Haykin S., (1994). “Neural Networks A Comprehensive Foundation”, Prentice Hall Publishing, Vol.1, New Jersey.

[25] Sağıroğlu Ş., Erler M., Beşdok E., (2003). “Mühendislikte Yapay Zeka Uygulamaları 1 / Yapay Sinir Ağları”, Ufuk Kitap, 1.Baskı, Kayseri.

[26] Bilgicik, Sinir Hücresi ve Çeşitleri, http://www.bilgicik.com/yazi/sinir-hucresi-ve-cesitleri/, 6 Kasım 2017

[27] Demirhan A., Kılıç Y. A., Güler İ., (2010). “Tıpta Yapay Zeka Uygulamaları”, Yoğun Bakım Dergisi, 9 (1): 31-41

[28] Fausett L., (1994). “Fundamentals Of Neural Networks: Architectures, Algorithms, and Application”, Prentice-Hall Inc., International Editions, New Jersey.

[29] Psikolojik, Yapay Sinir Ağları, http://www.psikolojik.gen.tr/yapay-sinir-aglari.html, 7 Kasım 2017

[30] Kriesel D., (2005). "A Brief Introduction to Neural Networks", http://www.dkriesel.com/en/science/neural_networks, 4 Kasım 2017.

[31] Safari Book Online, Multi Layer Perceptron, https://www.safaribooksonline.com/library/view/getting-started-with/9781786468574/ch04s04.html, 10 Kasım 2017

[32] DTREG, RBF Neural Networks, https://www.dtreg.com/solution/view/25, 8 Kasım 2017

[33] Kohavi, R., (1995). "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection", Proc. IJCAI'95 Conf., 20-25 August 1995, Montreal, 1137-1143.

[34] James G. ve Ark., (2013). "An Introduction to Statistical Learning: with Applications in R", Springer Science & Business Media, First Edition, London.

[35] Hartley R., Zisserman A., (2003). " Multiple View Geometry in Computer Vision", Cambridge University Press, Second Edition, New York.

[36] Mathworks, Single Camera Calibration App, https://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html, 9 Kasım 2017

[37] Doğan H., (2008). "Adana'da Satışa Sunulan Yumurtalarda Sunuş Çeşitliliği ve Kalite Değişimi Üzerine Bir Çalışma", Yüksek Lisans Tezi, Çukurova Üniversitesi Fen Bilimleri Enstitüsü, Adana.

EK-A

KAYNAK KODLARI

Sistemin kurulması sırasında yazılmış olan kaynak kodları aşağıda verilmiştir.

A-1 Kameranın Kalibrasyonu

```
% Farklı Açılardan Çekilmiş Dama Tahtası Görüntülerinin Sisteme Tanıtılması
imageFileNames = {'C:\Users\burak\Desktop\checkerboard3\IMG_0444.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0445.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0446.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0447.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0448.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0449.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0450.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0451.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0452.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0453.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0456.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0459.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0460.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0462.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0463.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0464.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0465.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0466.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0467.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0468.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0470.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0471.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0472.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0473.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0474.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0475.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0476.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0477.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0479.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0480.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0481.JPG',...
    'C:\Users\burak\Desktop\checkerboard3\IMG_0482.JPG',...
    };
```

```
% Dama Tahtası Görüntüsünün Tespit Edilmesi
[imagePoints, boardSize, imagesUsed] = detectCheckerboardPoints(imageFileNames);
imageFileNames = imageFileNames(imagesUsed);

% Karelerin Köşe Noktalarının Beklenen Değerlerinin Belirlenmesi
squareSize = 12;  % in units of 'mm'
worldPoints = generateCheckerboardPoints(boardSize, squareSize);

% Kameranın Kalibre Edilmesi
[cameraParams, imagesUsed, estimationErrors] = estimateCameraParameters(imagePoints,
worldPoints, ...
   'EstimateSkew', true, 'EstimateTangentialDistortion', true, ...
   'NumRadialDistortionCoefficients', 3, 'WorldUnits', 'mm');

% Reprojeksiyon Hatasının Gösterilmesi
h1=figure; showReprojectionErrors(cameraParams, 'BarGraph');

% Harici Parametrelerin Gösterilmesi
h2=figure; showExtrinsics(cameraParams, 'PatternCentric');

% Tahmini Hataların Gösterilmesi
displayErrors(estimationErrors, cameraParams);
```

A-2 YSA'nın ÇKA Modeline Göre Eğitilmesi, Testi ve Validasyonu

```
%ÇKA Parametreleri
clear train
net = feedforwardnet([40]);
net.performFcn = 'mse';
net.performParam.regularization = 0.5;
net.layers{1}.transferFcn = 'tansig';
net.layers{2}.transferFcn = 'tansig';
net.trainfcn = 'traingdx';

%Verilerin Rast Gele Seçilmesi Durumu
net.dividefcn = 'dividerand';
net.divideParam.trainRatio = 70;
net.divideParam.valRatio   = 15;
net.divideParam.testRatio  = 15;

%K Katmanlı Çapraz Doğrulama Yöntemi Veri Seti
set=5;% K. veri seti numarası
net.divideFcn = 'divideind';
net.divideParam.trainInd = 1:120;
net.divideParam.testInd  = 121:150;
net.divideParam.valInd   = 121:150;
target= kfoldwholedata(13,:,set); % Beklenen Değer
train1= kfoldwholedata(1:12,:,set); % Öz Nitelik Vektörleri

%ÇKA Parametreleri
net.trainparam.lr = 0.5;
net.trainparam.mc = 0.9;
```

```
net.trainParam.min_grad = 0;
net.trainParam.max_fail = 60000;
net.trainparam.epochs = 30000;
net.trainparam.goal = 0;

%ÇKA'nın Eğitilmesi
net = configure (net,train1,target);
net(train1);
[net,tr] = train (net,train1,target);
assignin('base', 'net', net);
y=net(train1);

%ÇKA Performansının Belirlenmesi
trainTargets = target .* tr.trainMask{1};
valTargets = target .* tr.valMask{1};
testTargets = target .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)
perf  = perform(net,target,y)
maetest = mae(testTargets-y)
maevalidation=mae(valTargets-y)
maetrain=mae(trainTargets-y)
maegeneral=mae(target-y)
```

A-3 YSA'nın RTF Modeline Göre Eğitilmesi, Testi ve Validasyonu

```
%Verilerin Rast Gele Seçilmesi Durumu
[trainInd,valInd,testInd] = dividerand(wholedata,0.70,0.15,0.15);

%Verilerin Rast Gele Seçilmesi Durumu
set=4; ;% K. veri seti numarası
[trainInd,testInd,valInd] = divideind(kfoldwholedata(:,:,set),1:120,121:150,121:150);
wholedata = kfoldwholedata(:,:,set);

%Veri Kümelerinin Belirlenmesi
trainD = trainInd(1:12,:);
trainT = trainInd(13,:);
valD = valInd(1:12,:);
valT = valInd(13,:);
testD = testInd(1:12,:);
testT = testInd(13,:);
wholeD = wholedata(1:12,:);
wholeT = wholedata(13,:);

%RTF'in Eğitilmesi ve Parametreleri
net = newrb(trainD,trainT,1,10000);

%RTF'in Performansı
ytrain = sim(net,trainD);
figure;
plotregression(trainT,ytrain);
maetrain = mae(trainT-ytrain)
yval = sim(net,valD);
figure;
```

```
plotregression(valT,yval);
maeval = mae(valT-yval)
ytest = sim(net,testD);
figure;
plotregression(testT,ytest);
maetest = mae(testT-ytest)
ywhole = sim(net,wholeD);
figure;
plotregression(wholeT,ywhole);
maewhole = mae(wholeT-ywhole)
max(abs( wholeT-ywhole))
view(net);

% Ağ Performansının Gösterilmesi
Y= net(trainD);
figure;
plot(trainT')
hold on
grid on
plot(Y','r')
set(gca,'ytick')
legend('Beklenen Değer','Ağın Yanıtı')
xlabel('Örnek Numarası')
```

A-4 K-Katlı Çapraz Doğrulama Yöntemi Datasının Oluşturulması

```
c = cvpartition(150,'kfold',5);

for fold=1:5

testind = transpose(test(c,fold));
trainind = transpose(training(c,fold));

for v = 2:13
testind(v,:) = testind(1,:);
end

for v = 2:13
trainind(v,:) = trainind(1,:);
end

testdata = wholedata .* testind;
traindata = wholedata .* trainind;

newsutun=1;
for sutun = 1:150;
   if traindata(1,sutun) ~= 0;
      for satir = 1:13;
      kfoldwholedata(satir,newsutun,fold)= traindata(satir,sutun);
      end
      newsutun = newsutun+1;
   end
end
```

```
for sutun = 1:150;
    if testdata(1,sutun) ~= 0;
       for satir = 1:13;
       kfoldwholedata(satir,newsutun,fold)= testdata(satir,sutun);
       end
       newsutun = newsutun+1;
    end
end

end
```

A-5 Kullanıcı Arayüzü

```
function varargout = AgirlikTahmin(varargin)

% Başlatma Kodları
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @AgirlikTahmin_OpeningFcn, ...
                   'gui_OutputFcn',  @AgirlikTahmin_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end


 %Değişkenlerin Başlangıç Değerlerinin Tanımlanması
function AgirlikTahmin_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
global image hiz calistir filename cisim ysa tahmin ghazir ucresim ucoznitelik
image = 0;
hiz = -1;
calistir = 0;
filename = 0;
cisim = 1;
ysa = -1;
tahmin = 0;
ghazir = 1;
ucresim = 0;
ucoznitelik(5,3) = 0;

% Bütün Kullanıcı Arayüzünü Kapsayan Eksenin Oluşturulması
ah = axes('unit', 'normalized', 'position', [0 0 1 1]);
```

```
% Arka Plan Görüntüsünün Okunması
bg = imread('arkaplan.jpg'); imagesc(bg);

% Arka Plan Parametrelerinin Belirlenmesi
set(ah,'handlevisibility','off','visible','off')
uistack(ah, 'bottom');
guidata(hObject, handles);

function varargout = AgirlikTahmin_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

% Cismin Görüntüsünün Seçilmesi
function goruntu_secim_Callback(hObject, eventdata, handles)
global image filename
[filename,pathname] = uigetfile();

if filename ~=0
axes(handles.goruntu_cikti)
image = imread(strcat(pathname,filename));
imshow(image);
set(handles.path_edit,'String',strcat(pathname,filename));
set(handles.agirlik_cikti,'String','');
end

% Cismin Ağırlığının Tahmin Edilmesi
function agirlik_tahmin_Callback(hObject, eventdata, handles)
global image hiz calistir filename ysa cisim tahmin ghazir ucoznitelik ucresim

calistir =1;
ghazir =1;

if filename==0
   msgbox(sprintf('Lütfen Ağırlığı Ölçülecek Cismin Görüntüsünü Seçiniz'),'HATA','Error');
   calistir = 0;

elseif (ysa~=0) && (ysa~=1)
   msgbox(sprintf('Lütfen YSA Yapısını Seçiniz'),'HATA','Error');
   calistir = 0;

elseif (hiz~=0) && (hiz~=1)
   msgbox(sprintf('Lütfen Mod Seçimi Yapınız'),'HATA','Error');
   calistir = 0;

end

if calistir == 1
   cameraParams = evalin('base', 'cameraParams1');
   kilotahmin(image,cameraParams,0,0,hiz);
      if(cisim==1 && ysa==0)
         oznitelik = evalin('base', 'oznitelik');
         net = evalin('base', 'mlpyumurta');
         hesapla = oznitelik(1:3,:);
      end
      if(cisim==1 && ysa==1)
         oznitelik = evalin('base', 'oznitelik');
         net = evalin('base', 'rbfyumurta');
```

```
        hesapla=oznitelik;
      end
      if(cisim==2 && ysa==0)
        oznitelik = evalin('base', 'oznitelik');
        net = evalin('base', 'mlpportakal');
        ghazir = 0;
        if(ucresim == 0 )
        ucoznitelik(:,1) = oznitelik;
        ucresim = 1;
        elseif (ucresim == 1)
        ucoznitelik(:,2) = oznitelik;
        ucresim = 2;
        elseif (ucresim == 2)
        ucoznitelik(:,3) = oznitelik;
        ucresim = 3;
        end
       if(ucresim == 3)
       ghazir = 1;
       ucoznitelik (2,:) = [];
       hesapla = [ucoznitelik(:,1);ucoznitelik(:,2);ucoznitelik(:,3)];
       end
      end
      if(cisim==2 && ysa==1)
        oznitelik = evalin('base', 'oznitelik');
        net = evalin('base', 'rbfportakal');
        ghazir = 0;
        if(ucresim == 0 )
        ucoznitelik(:,1) = oznitelik;
        ucresim = 1;
        elseif (ucresim == 1)
        ucoznitelik(:,2) = oznitelik;
        ucresim = 2;
        elseif (ucresim == 2)
        ucoznitelik(:,3) = oznitelik;
        ucresim = 3;
        end
       if(ucresim == 3)
       ghazir = 1;
       ucoznitelik (2,:) = [];
       hesapla = [ucoznitelik(:,1);ucoznitelik(:,2);ucoznitelik(:,3)];
       end

      end


      if (ghazir ==1)
      tahmin = net(hesapla);
      else
      msgbox(sprintf('Bu Cisim İçin %1g. Görüntüyü Seçiniz',ucresim+1),'Uyarı','Warn');
      end
    set(handles.agirlik_cikti,'String',tahmin);
end


function path_edit_Callback(hObject, eventdata, handles)
```

```
% Dosya Yolunu Gösteren Metin Kutusunun Parametreleri
function path_edit_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end


% Belirlenen Tahmini Ağırlığı Gösteren Metin Kutusunun Parametreleri
function agirlik_cikti_Callback(hObject, eventdata, handles)
function agirlik_cikti_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end


% Ölçüm Modunun Belirlenmesi
function hiz_popup_Callback(hObject, eventdata, handles)
global hiz;
contents = get(hObject,'Value');

switch contents

   case 1
      hiz = -1;

   case 2
      hiz = 1;

   case 3
      hiz = 0;
end

% Ölçüm Modunun Belirlendiği Açılır Menünün Parametreleri
function hiz_popup_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end


% Ağırlığı Ölçülecek Cisimin Seçilmesi
function Yumurta_Callback(hObject, eventdata, handles)
global cisim;
cisim = 1;

function portakal_Callback(hObject, eventdata, handles)
global cisim;
cisim = 2;

%  YSA Yapısının Belirlenmesi
function ysa_yapisi_popup_Callback(hObject, eventdata, handles)

global ysa;
contents = get(hObject,'Value');

switch contents
```

```
    case 1
        ysa = -1;

    case 2
        ysa = 0;

    case 3
        ysa = 1;
end

% YSA  Yapısını Belirleyen Açılır Menünün Parametreleri
function ysa_yapisi_popup_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

A-6 Ağırlık Tahminleme Fonksiyonu

```
function[output]=kilo(input,kk,net,aci,secim)

%Görüntülerin ve Kamera Parametrelerinin Okunması
giris = input;
cameraParams=kk;
originalImage = giris;
undistortedImage = undistortImage(originalImage, cameraParams);
ilk=undistortedImage;
ilk=imrotate(ilk,aci);

%Projektif Dönüşümün Uygulanması
[imagePoints, boardSize] = detectCheckerboardPoints(ilk);
squareSize = 12;
worldPoints = generateCheckerboardPoints(boardSize, squareSize);
worldPoints=[60 0;48 0;36 0;24 0;12 0;0 0;60 12;48 12;36 12;24 12;12 12;0 12;60 24;48 24;36 24;24
24;12 24;0 24;60 36;48 36;36 36;24 36;12 36;0 36;60 48;48 48;36 48;24 48;12 48;0 48;60 60;48 60;36
60;24 60;12 60;0 60;60 72;48 72;36 72;24 72;12 72;0 72;60 84;48 84;36 84;24 84;12 84;0 84;60 96;48
96;36 96;24 96;12 96;0 96 ];
movingPoints = [imagePoints];
fixedPoints  = [worldPoints*9];
tform = fitgeotrans(movingPoints,fixedPoints,'projective')
ilk=imwarp(ilk,tform);
[imagePoints, boardSize] = detectCheckerboardPoints(ilk);

%Hassas Ölçüm İçin Cisim Ayıklama Yöntemi – 4. Bölümde Ayrıntılı Olarak Anlatılmıştır.
hiz = secim;
if (hiz == 0 )

[satir sutun renk] = size(ilk);
ilk1=im2double(ilk);
yeni=ilk1;
toplam=0;
toplam1=0;
toplam2=0;
toplam3=0;
say=0;
say1=0;
```

```
 %Hassa Ölçüm İle Cisim Ayıklama 1. Aşama
for i=1:satir
for j=1:sutun
   if(yeni(i,j,1) > 0.56) yeni(i,j,:)=1; end
   if(yeni(i,j,2) > 0.56) yeni(i,j,:)=1; end
   if(yeni(i,j,3) > 0.56) yeni(i,j,:)=1; end
   if((yeni(i,j,1)/2 > yeni(i,j,2) && yeni(i,j,1)/2 > yeni(i,j,3)) && yeni(i,j,1) > 0.30 ) yeni(i,j,:)=1; end
   if((yeni(i,j,1)+yeni(i,j,2)+yeni(i,j,3))/3) > 0.56 yeni(i,j,:)=1; end
   if((yeni(i,j,1)+yeni(i,j,2)+yeni(i,j,3))/3) < 0.25 yeni(i,j,:)=0; end
end
end

%Görüntü Dönüşümleri
yeni1=rgb2gray(yeni);
yeni1=medfilt2(yeni1,[5 5]);
yeni1=im2double(yeni1);

%Hassas Ölçüm İçin Cisim Ayıklama 2. Aşama
for i=2:satir-3
for j=2:sutun-3

   if ( j>1000 && ilk1(i,j,1) > 0.30 )
     var=0;
    if( ilk1(i-1,j,1)- ilk1(i,j,1) < ilk1(i-1,j,1)*0.1 && yeni1(i-1,j,1)==1 ) var=var+1; end
    if( ilk1(i+1,j,1)- ilk1(i,j,1) < ilk1(i+1,j,1)*0.1 && yeni1(i+1,j,1)==1 ) var=var+1; end
    if( ilk1(i,j-1,1)- ilk1(i,j,1) < ilk1(i,j-1,1)*0.1 && yeni1(i,j-1,1)==1 ) var=var+1; end
    if( ilk1(i,j+1,1)- ilk1(i,j,1) < ilk1(i,j+1,1)*0.1 && yeni1(i,j+1,1)==1 ) var=var+1; end
    if( ilk1(i-1,j-1,1)- ilk1(i,j,1) < ilk1(i-1,j-1,1)*0.1 && yeni1(i-1,j-1,1)==1 ) var=var+1; end
    if( ilk1(i+1,j+1,1)- ilk1(i,j,1) < ilk1(i+1,j+1,1)*0.1 && yeni1(i+1,j+1,1)==1 ) var=var+1; end
    if( ilk1(i-1,j+1,1)- ilk1(i,j,1) < ilk1(i-1,j+1,1)*0.1 && yeni1(i-1,j+1,1)==1 ) var=var+1; end
    if( ilk1(i+1,j-1,1)- ilk1(i,j,1) < ilk1(i+1,j-1,1)*0.1 && yeni1(i+1,j-1,1)==1 ) var=var+1; end

    if(var>2) yeni(i,j,:)=1; yeni1(i,j,:)=1; end

   end

%Hassas Ölçüm İçin Cisim Ayıklama 3. Aşama
   if (yeni(i,j,1) ~= 1  )
      toplam = toplam + (ilk1(i,j,1)+ ilk1(i,j,2)+ ilk1(i,j,3))/3;
   say=say+1;

   end

end
end
esik= toplam / say;
esik=esik+0.2;

for i=1:satir
for j=1000:sutun

if ((yeni(i,j,1)+ yeni(i,j,2)+ yeni(i,j,3))/3 < esik )

   yeni(i,j,:)=0; else yeni(i,j,:)=1; end

end
```

```
end

%Hassas Ölçüm İçin Cisim Ayıklama 4. Aşama

b=rgb2gray(yeni);
ss=graythresh(b);
d=im2bw(b,ss+0.2);
e=bwareaopen(d,10000);
e=imfill(e,'holes');
imshow(e);
end

%Hızlı Cisim Ayıklama  - 4. Bölümde Ayrıntılı Olarak Açıklanmıştır.
if(hiz == 1)

b=rgb2gray(ilk);
ss=graythresh(b);
d=im2bw(b,ss+0);
e=bwareaopen(d,10000);
e=imfill(e,'holes');
imshow(e);
end

%Öz Nitelik Vektörlerinin Belirlenmesi
s=regionprops(e,'Area','Centroid','BoundingBox','Perimeter','ConvexArea','Eccentricity');
centoids = cat(1,s.Centroid);
area=cat(1,s.Area);
carea=cat(1,s.ConvexArea);
peri=cat(1,s.Perimeter);
rround=cat(1,s.Eccentricity);
bound = bwboundaries(e);
cisimsinir=bound{2};
cisimsinir1=bound{1};

%Çevre Hesaplaması Alternatif Yöntem
[satir sutun] = size(cisimsinir);
cevre=0;
for i=1:satir-1
cevre = cevre+sqrt( (cisimsinir(i,1)-cisimsinir(i+1,1))^2 +  (cisimsinir(i,2)-cisimsinir(i+1,2))^2);
end

%Cismin Merkez Noktasına En Uzak ve En Yakın Noktaların Belirlenmesi
[satir sutun] = size(cisimsinir1);
maxru = 0;
minru = 99999;
for i=1:satir
uzaklik = sqrt( (cisimsinir1 (i,1)-centoids(1,2))^2 +  (cisimsinir1 (i,2)-centoids(1,1))^2);
if maxru < uzaklik maxru=uzaklik;  end
if minru > uzaklik minru=uzaklik;  end
end

% σ  ve ρ Katsayılarının Belirlenmesi
refx=60/pdist([imagePoints(1,:);imagePoints(6,:)],'euclidean')
refy=96/pdist([imagePoints(1,:);imagePoints(49,:)],'euclidean')
kar= refx*refy;
ort=(refx+refy)/2;
```

```
%Cismin Merkez Noktasına En Uzak ve En Yakın Noktaların Belirlenmesi
[satir sutun] = size(cisimsinir);
maxu = 0;
minu = 99999;
for i=1:satir
uzaklik = sqrt( ((cisimsinir1(i,1)-centoids(2,2))*refx)^2 +  ((cisimsinir (i,2)-centoids(2,1))*refy)^2);
if maxu < uzaklik maxu=uzaklik; maxx= cisimsinir (i,1);maxy= cisimsinir (i,2); end
if minu > uzaklik minu=uzaklik; minx= cisimsinir (i,1);miny= cisimsinir (i,2);end
end

%Öz Nitelik Vektörleri ve Katsayılar İle Çarpılarak Gerekli Dönüşümlerinin Sağlanması
area
centoids;
kare=area(1)* kar
cisim=area(2)* kar
minu
maxu
perimeter=peri(2)*ort
roundess=minu/maxu

%Öz Nitelik Vektörlerine Göre Tahminleme İşlemi
oznitelik = [cisim/100;roundess;perimeter/10;minu/10;maxu/10];
assignin('base', 'oznitelik', oznitelik);

imshow(ilk);
hold on
visboundaries(bound,'Color','r','LineWidth',1);
plot(centoids(:,1),centoids(:,2),'b*');
plot([maxy centoids(2,1)],[maxx centoids(2,2)],'Color','Blue','LineWidth',1,'LineStyle','--')
plot([miny centoids(2,1)],[minx centoids(2,2)],'Color','Black','LineWidth',1,'LineStyle','--')

end
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Ünsal Burak ŞALVARCI

Doğum Tarihi ve Yeri : 22.04.1988 - İstanbul

Yabancı Dili : İngilizce

E-posta : buraksalvarci88@gmail.com

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Lisans	Endüstri Müh.	Süleyman Demirel Üni.	2012
Lisans	Elektronik ve Haberleşme Müh.	Süleyman Demirel Üni.	2011
Lise	Yabancı Dil Ağırlıklı Lise	Ankara Ayrancı Süper Lisesi	2006

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2015	Türk Hava Yolları	Data ve Strateji Uzmanı
2014	Ultra Teknoloji	Ar-Ge Mühendisi
2013	Pervacio Inc.	Saha Mühendisi

YAYINLARI

Bildiri

1. “Yapay Sinir Ağları Kullanılarak İmgelerden Yumurta Ağırlığının Uzaklıktan Bağımsız Sezimi”, Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Antalya

2. “2.4 Ghz ISM Bandında Çalışan, Data Logger Özelliği Olan, Düşük Güçlü, Verimli, Kablosuz RF Prob Tasarımı” Çankaya Üniversitesi 4. Mühendislik ve Teknoloji Sempozyumu, Ankara, Sayfa 59.

3. “Müşteri Memnuniyetine Yönelik Sistem Tasarımı: I-Q Randevu Bankacılığı Modeli” Üretim Araştırmaları Sempozyumu, İzmir

4. “Kalorifer Kazanı İmalat Atölyesinde Simülasyon Destekli Süreç İyileştirme ve Personel Planlama” Üretim Araştırmaları Sempozyumu, İzmir

Proje

1. Rfid Teknolojisi Kullanılarak Mobil El Cihazı İle Kitapların Uzaktan (Telemetri) Takibi Ve Personelsiz İşlem Yapabilen Rfid İstasyon Sistem Tasarımı: Süleyman Demirel Üniversitesi Kütüphanesi Örneği Projesi

2. Zigbee Teknolojisi Kullanarak, Data Logger Özelliği Taşıyan Gezgin (Kablosuz) Problar Yardımıyla Bilgisayar Destekli Uzaktan Ölçüm (Telemetri) Sistem Tasarımı Projesi