

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**DÂHİLİ WEBCAM KULLANARAK GÖZ
HAREKETLERİNİN TAKİBİ VE İMLEÇ KONTROL
YÖNTEMİ**

DOKTORA TEZİ

Bülent TURAN

**Enstitü Anabilim Dalı : ELEKTRONİK VE BİLGİSAYAR
EĞİTİMİ**
Tez Danışmanı : Doç. Dr. Halil İbrahim ESKİKURT

Haziran 2015

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DÂHİLİ WEBCAM KULLANARAK GÖZ
HAREKETLERİNİN TAKİBİ VE İMLEÇ KONTROL
YÖNTEMİ

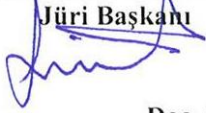
DOKTORA TEZİ

Bülent TURAN

Enstitü Anabilim Dalı : ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ

Bu tez 12 / 06 /2015 tarihinde aşağıdaki jüri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.

Doç. Dr.
Halil İbrahim ESKİKURT
Jüri Başkanı



Doç. Dr.
Yıldırım YALMAN
Üye



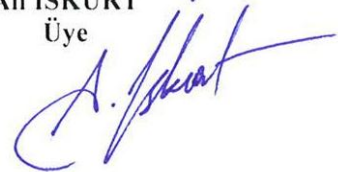
Doç. Dr.
Cüneyt BAYILMIŞ
Üye



Yrd. Doç. Dr.
Gürsel DÜZENLİ
Üye



Yrd. Doç. Dr.
Ali İSKURT
Üye



BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Bülent TURAN

12.06.2015

TEŐEKKÜR

Bu alıőmanın ortaya ıkmasında grüş, fikir ve bilgi birikiminden her zaman istifade ettiđim, alıőmama yön veren ve emeđini hiçbir zaman esirgemeyen danıőman hocam Sayın Do. Dr. Halil İbrahim ESKİKURT'a minnettarım.

alıőmamın ilerlemesinde ve sonuçlandırılmasında bilgi ve tecrübelerinden yararlandığım, takıldığım aőamalarda ıkıő bulmama yardımcı olan, alıőmama yön veren, grüş ve fikirlerini hiçbir zaman esirgemeyen, tez izleme komitesinde yer alan Sayın Do. Dr. Cüneyt BAYILMIŐ ve Sayın Yrd. Do. Dr. Gürsel DÜZENLİ'ye teőekkürü bir bor bilirim.

alıőmam süresince sürekli fikir alışveriőinde bulunduđum alıőmama fikirleri ile katkı sađlayan alıőma arkadaőım Öğr. Gör Mehmet Serhat CAN'a ve uzun süren tez alıőması boyunca yeterince zaman ayıramadığım, sabırları ile bana ok büyük destek olan Eőim Sevda TURAN, ocuklarım Berke Can TURAN ve Beril TURAN'a teőekkür ederim.

İÇİNDEKİLER

TEŞEKKÜR.....	iii
İÇİNDEKİLER.....	iv
SİMGELER VE KISALTMALAR LİSTESİ.....	vii
ŞEKİLLER LİSTESİ... ..	x
TABLOLAR LİSTESİ	xiv
ÖZET.....	xv
SUMMARY.....	xvi
BÖLÜM 1.	
GİRİŞ.....	1
1.1. Çalışmanın Amacı ve Kapsamı	9
1.1.1. Çalışmanın amacı	9
1.1.2. Çalışmanın literatürdeki yeri.....	9
1.1.3. Tez çalışmasının yürütülmesi	10
BÖLÜM 2.	
GÖRÜNTÜ İŞLEME.....	12
2.1. Görüntünün İçerdiği Bilgiler Ve Görüntü İşlemenin Kullanım Alanları.....	12
2.2. Sayısal Görüntü	14
2.3. Sayısal Görüntü İşleme	17
2.3.1. Görüntü çözünürlüğü	18
2.3.2. Görüntü işleme uygulamalarında kullanılan bazı algoritma ve yöntemler.....	19
2.4. İnsan Bilgisayar Etkileşimine ve Görüntü İçinde Yüz/Göz Bölgesine Yönelik Çalışmalar.....	30

BÖLÜM 3.

YAPAY SİNİR AĞLARI	34
3.1. Yapay Sinir Ağlarının Genel Özellikleri	34
3.2. Yapay Sinir Ağlarının Uygulama Alanları	35
3.3. Biyolojik Sinir Hücresi	36
3.4. Yapay Sinir Hücresi.....	36
3.5. Yapay Sinir Ağları	38
3.6. Örnek Yapay Sinir Ağı Modelleri.....	41
3.6.1. Hebb net ağ modeli	41
3.6.2. Perceptron ağ modeli.....	42
3.6.3. SOM (self organizing map) ağ modeli	43
3.6.4. Backpropagation net ağ modeli.....	45
3.6.5. Hopfield yapay sinir ağı modeli	46
3.7. Çalışmada Kullanılan Ağ Modeli	47

BÖLÜM 4.

YÖNTEMİN GELİŞTİRİLMESİ	48
4.1. Materyal	48
4.1.1. Çalışmada kullanılan bilgisayar/bilgisayarlar	48
4.1.2. Göz bölgesinin bulunmasına yönelik eğitim ve test verilerinin hazırlanması.....	49
4.1.3. Göz bölgesinin bulunmasına yönelik kullanılan matlab yazılımı .	50
4.1.4. Koordinat tespitine yönelik eğitim ve test verilerinin hazırlanması	51
4.1.5. Koordinat tespitine yönelik kullanılan matlab yazılımı	53
4.1.6. Çalışmanın denekler ile tekrarlanması amacıyla hazırlanan arayüz	54
4.1.7. Deneklere ait eğitim ve test verilerinin hazırlanması.....	58
4.2. Çalışma Aşamalarının Gerçeklenmesi	58
4.2.1. Göz bölgesinin bulunması.....	61
4.2.2. Göz bölgesine ait özniteliklerin belirlenmesi ve elde edilmesi.....	67
4.2.2.1. İris tespiti	69
4.2.2.2. Göz köşelerinin tespiti	72

4.2.2.3. Göz bölgesine ait piksel değerlerinin alınması	80
4.2.2.4. Göz bölgesine ait özniteliklerin çıkarılması	81
4.2.3. Koordinat tespiti ve imlecin taşınması	83
4.2.4. Farklı kullanıcılar ile çalışmanın tekrarı	88
BÖLÜM 5.	
SONUÇLAR VE ÖNERİLER	99
5.1. Sonuçlar.....	99
5.1.1. Göz bölgesinin bulunmasına yönelik sonuçlar	99
5.1.2. Özniteliklerin çıkarılmasına yönelik sonuçlar	99
5.1.3. Koordinat tespiti ve imlecin taşınmasına yönelik sonuçlar.....	100
5.1.4. Arayüzün kullanılmasına yönelik sonuçlar	103
5.1.5. Farklı kullanıcılar ile koordinat tespitine yönelik sonuçlar.....	104
5.2. Öneriler	104
KAYNAKLAR.....	107
EKLER.....	115
ÖZGEÇMİŞ.....	129

SİMGELER VE KISALTMALAR LİSTESİ

a	: Çember merkezi x koordinatı
b	: Çember merkezi y koordinatı
CHT	: C ircular H ough T ransform (Dairesel Hough Transformu)
CMYK	: C yan- M agenta- Y ellow- B lack (Camgöbeği mavisi-Magenta pembesi-Sarı-Siyah)
EDDP	: E n D üşük D eğerlikli P iksel
$f(net)$: Yapay sinir hücresinde ağırlıklandırılmış girişler toplamına uygulanan aktivasyon fonksiyonu
$H_m(x)$: Görüntüdeki satır piksel değerlerinin ortalaması
$H_m(y)$: Görüntüdeki sütun piksel değerlerinin ortalaması
HSI	: Hue-Saturation-Lightness (Hue-Doygunluk-Yoğunluk)
IDO	: A Discrete Approximation of an Integrodifferential Operator (İntegro-Diferansiyel Operatörünün Ayrık Yaklaşımı)
$I(x, y_i)$: Görüntüdeki satır piksel değerleri
$I(x_i, y)$: Görüntüdeki sütun piksel değerleri
İBE	: İ nsan B ilgisayar E tkileşimi
L	: Max piksel değeri
LVP	: L owest V alued P ixel (En Düşük Değerlikli Piksel)
MAE	: M ean A bsolute E rror (Mutlak Hata Ortalaması)
MAPE	: M ean A bsolute P ercent E rror (Mutlak Hata Oranları Ortalaması)
n	: Tahmin sayısı
nk	: Aynı renk değerine sahip piksel sayısı
PCA	: P rincipal C omponents A nalysis (Temel Bileşenler Analizi)
Pr(r)	: Olasılık yoğunluk fonksiyonu

R	: Ağırlık Faktörü
r	: Çember yarıçapı
RGB	: Red- G reen- B lue (Kırmızı-Yeşil-Mavi)
rk	: Renk değerleri
RMSE	: R oot M ean S quared E rror (Hata Kareleri Ortalamasının Karekökü)
SOM	: S elf O rganizing M ap (özdüzenleyici Haritalar)
SVM (DVM)	: S upport V ector M achines (D estek V ectör M akineleri)
sk	: Renk değerlerinin eşiklenmiş görüntüdeki hesaplanmış piksel Koordinatları
t	: Yapay sinir ağı çıkışında istenen hedef değer matrisi
$V_{n,p}$: Yapay sinir ağı girişleri ile 1. gizli katman arasındaki ağırlık Katsayıları
VPF	: V ariance P rojection F unction (Varyans Bağdaştırma Fonksiyonu)
$w_0, w_1, w_2, \dots, w_n$: Yapay sinir ağı giriş verilerinin ağırlık katsayıları
$W_{p,m}$: Yapay sinir ağı 1. gizli katman ile çıkışlar arasındaki ağırlık Katsayıları
w_i	: Yapay sinir ağı i'inci giriş verisinin ağırlık katsayısı
WVPF	: W eighted V ariance P rojection F unction (Ağırlıklı Varyans Bağdaştırma Fonksiyonu)
$x_0, x_1, x_2, \dots, x_n$: Yapay sinir ağı giriş verileri
x_i	: Yapay sinir ağı i'inci giriş verisi
YSA	: Y apay S inir A ğı
Y_x	: Gerçek x koordinatı
Y_y	: Gerçek y koordinatı
Y_x'	: Tahmin edilen x koordinatı
Y_y'	: Tahmin edilen y koordinatı
$y_0, y_1, y_2, \dots, y_n$: Yapay sinir ağı çıkış verileri
y_i	: Yapay sinir ağı i'inci çıkış verisi

η : Öğrenme Katsayısı
 $\delta_{wy}^2(x)$: Yatay WVP fonksiyonu
 $\delta_{wh}^2(y)$: Dikey WVP fonksiyonu

ŞEKİLLER LİSTESİ

Şekil 1.1. (a) Canesta sanal klavye [3] (b) Dark i-Pen dijital kalem [4]	3
Şekil 1.2. (a) Tekerlekli sandalyeye monte edilmiş portatif Eye Gaze sistem [10] (b) Eye Gaze sistemin kullanılışı [11]	5
Şekil 1.3. (a) Akıllı kaydırma [12] (b) temassız kontrol [13]	7
Şekil 2.1. (a) Orijinal görüntü (b) gri görüntü (c) ikili görüntü (d) negatif görüntü örnekleri	15
Şekil 2.2. Renkli görüntü piksel değerleri	16
Şekil 2.3. Gri seviyeli görüntü piksel değerleri	16
Şekil 2.4. İkili görüntü ve ikili görüntü piksel değerleri	17
Şekil 2.5. (a) Orijinal görüntü (b) parlaklığı artırılmış görüntü ($a = +70$) (c) parlaklığı azaltılmış görüntü ($a = -70$)	19
Şekil 2.6. (a) Orijinal görüntü (b) karşıtlığı artırılmış görüntü ($b = 2$) (c) karşıtlığı azaltılmış görüntü ($b = 0.5$)	20
Şekil 2.7. (a) Orijinal görüntü (b) $a=+50$ $b = 2$ (c) $a=+50$ $b = 0.5$ (d) $a=-10$ $b=2$ (e) $a=-50$ $b = 2$ (f) $a=-50$ $b = 0.5$	21
Şekil 2.8. Orijinal görüntü ve histogramı ile histogramı eşitlenmiş görüntü ve histogramı	23
Şekil 2.9. (a) Orijinal görüntü (b) eşiklenmiş görüntü	24
Şekil 2.10. (a) 3×3 alçak geçiren filtre (b) 5×5 alçak geçiren filtre (c) 9×9 alçak geçiren filtre	25
Şekil 2.11. (a) orijinal görüntü (b) 3×3 alçak geçiren filtre (c) 9×9 alçak geçiren filtre (d) 15×15 alçak geçiren filtre ile filtrelenmiş görüntü [26]	25
Şekil 2.12. Yüksek geçiren filtre örnekleri	26
Şekil 2.13. (a) orijinal görüntü (b), (c) filtrelenmiş görüntü (d) orijinal görüntü (e), (f) filtrelenmiş görüntü [26]	27

Şekil 2.14. (a) Orijinal görüntü (b) tuz biber gürültülü görüntü (c) medyan filtresi ile tuz biber gürültüsü temizlenmiş görüntü [26] ...	28
Şekil 2.15. (a) Orijinal görüntü (b) kenar alınmış görüntü	28
Şekil 2.16. (a) Orijinal görüntü (b) genişletilmiş görüntü (c) daraltılmış görüntü (d) küme kesişimi ile elde edilen kenar bulunmuş görüntü (e) küme farkı ile elde edilen kenar bulunmuş görüntü (f) küme eşleniği ile elde edilen görüntü [26]	29
Şekil 3.1. Biyolojik sinir hücresinin yapısı [53]	36
Şekil 3.2. Yapay sinir hücresinin şematik yapısı [55,56,57]	37
Şekil 3.3. Sıklıkla kullanılan transfer fonksiyonları [55,56]	38
Şekil 3.4. (a) Tek katmanlı tek çıkışlı yapay sinir ağı (b) tek katmanlı çok çıkışlı yapay sinir ağı [59]	39
Şekil 3.5. (a) Çok katmanlı tek çıkışlı yapay sinir ağı (b) çok katmanlı çok çıkışlı yapay sinir ağı [57]	40
Şekil 3.6. Hebb net ağ mimarisi [59]	41
Şekil 3.7. Perceptron ağ mimarisi [63]	43
Şekil 3.8. SOM ağ mimarisi [64]	44
Şekil 3.9. Bir gizli katmanlı backpropagation ağ mimarisi [59]	45
Şekil 3.10. Hopfield yapay sinir ağı mimarisi [51]	47
Şekil 4.1. Mavi renkli bölgeler eğitim, kırmızı renkli bölgeler test bölgeleri olarak kullanılmıştır	52
Şekil 4.2. Denekler için hazırlanan arayüz programı ve hatalı görüntülerin kayıttan önce silinmesi	55
Şekil 4.3. Arayüz yönlendirme ve bölgeleri ekrana basma pencereleri	57
Şekil 4.4. Akış diyagramı	60
Şekil 4.5. (a) Farklı mekan ve ışık şiddetinde elde edilen görüntülerden örnekler, (b) orijinal görüntüden elde edilmiş gri görüntüler (c) bu görüntülerden eğitim seti için elde edilen doğru göz bölgeleri [34, 68], (d) bu görüntülerden eğitim seti için elde edilen yanlış göz bölgeleri	62
Şekil 4.6. Test aşamasında YSA ile tarama usulüne göre göz bölgesinin bulunması	64

Şekil 4.7. Gerçek zamanlı çalışma aşamasında YSA ile tarama usulüne göre göz bölgesinin bulunması.....	65
Şekil 4.8. Yapay sinir ağının uygulandığı test görüntülerinden örnekler[34, 68]...	66
Şekil 4.9. Göz bölgesinin bulunması için geliştirilen YSA'nın gerçek zamanlı çalışma örnekleri	67
Şekil 4.10. (a) iris bölgesinin yapay sinir ağı ile bulunduktan sonra CHT ile iris tespiti, (b) sağ ve sol göz bölgesi içinde CHT ile iris tespiti [68]...	71
Şekil 4.11. (a) Sol göz dış köşe (b) sol göz iç köşe (c) sağ göz iç köşe (d) sağ göz dış köşe (e),(f),(g),(h) göz köşelerinin WVPF kullanılarak bulunması, (i),(j),(k),(l) göz köşelerinin EDDP kullanılarak bulunması [68]	74
Şekil 4.12. Göz köşelerinin bulunmasında kullanılan WVPF ile elde edilen sonuçlar [68].....	74
Şekil 4.13. Göz köşelerinin bulunmasında kullanılan EDDP ile elde edilen sonuçlar [68].....	75
Şekil 4.14. Sol göz dış köşe grafiği [68]	77
Şekil 4.15. Sol göz iç köşe grafiği [68]	77
Şekil 4.16. Sağ göz iç köşe grafiği [68]	78
Şekil 4.17. Sağ göz dış köşe grafiği [68]	78
Şekil 4.18. Göz köşelerinin bulunmasına yönelik örnek görüntü 1 [68]	79
Şekil 4.19. Göz köşelerinin bulunmasına yönelik örnek görüntü 2 [68]	80
Şekil 4.20. Aynı görüntüde göz köşelerinin farklı bulunması	82
Şekil 4.21. (a) Eğitim setinde kullanılmak üzere alınmış bir görüntü (b) göz bölgesi bulunmuş görüntü (c) göz bölgesi (d) matlab ortamında boyutları küçültülmüş göz bölgesi[34].....	84
Şekil 4.22. Koordinat tespiti için eğitilen ağın test verilerine uygulanması ile elde edilen sonuçlar (a) kırmızı renkli kare bölgeler test bölgeleridir (b) test bölgelerinin etrafında daire şeklinde farklı renklerle gösterilen bölgeler ise YSA tarafından tahmin edilen bölgelerdir [34].....	85
Şekil 4.23. 10 numaralı test bölgesine ait imleç kontrol örnekleri.....	88
Şekil 4.24. Göz bölgelerinin bulunması için deneklerden alınan görüntü örnekleri	90

Şekil 4.25. Tüm denekler 75 nolu eğitim bölgesine bakarken alınan birer görüntü	91
Şekil 4.26. Araştırmacı ile en iyi ve en kötü sonuçların elde edildiği 5. ve 7. deneklerin 5 nolu eğitim bölgesine bakarken elde edilmiş görüntüleri	93
Şekil 4.27. Araştırmacı ile en iyi ve en kötü sonuçların elde edildiği 5. ve 7. deneklerin 85 nolu eğitim bölgesine bakarken elde edilmiş görüntüleri	94
Şekil 4.28. Deneklerden elde edilen en iyi sonuçlar	95
Şekil 4.29. Deneklerden elde edilen en kötü sonuçlar	97
Şekil 5.1. (a) 4 numaralı test bölgesi için elde edilmesi beklenen sonuçların ortalaması (b) sonuçların Standart bir simge büyüklüğü ile karşılaştırılması	102

TABLolar LİSTESİ

Tablo 4.1. Denek bilgisayarlarının özellikleri.....	49
Tablo 4.2. Göz bölgesinin bulunması için kullanılan yapay sinir ağının parametreleri [68].....	63
Tablo 4.3. Sol iris bölgesinin bulunması için kullanılan yapay sinir ağının parametreleri [68].....	70
Tablo 4.4. Sağ iris bölgesinin bulunması için kullanılan yapay sinir ağının parametreleri [68].....	71
Tablo 4.5. Göz bölgesi içinde irisin bulunmasına yönelik kullanılan farklı iki yöntem ile elde edilen sonuçlar [68].....	72
Tablo 4.6. Göz köşe bölgeleri içinde göz köşelerinin bulunmasına yönelik kullanılan farklı iki yöntem ile elde edilen sonuçlar [68]	75
Tablo 4.7. Koordinat tespiti için kullanılan yapay sinir ağının parametreleri	84
Tablo 4.8. YSA ile işlenen test görüntüsü verilerinin sonuçları	87
Tablo 4.9. YSA ile işlenen 5. deneğe ait test görüntüsü verilerinin sonuçları	96
Tablo 4.10. YSA ile işlenen 7. deneğe ait test görüntüsü verilerinin sonuçları	98
Tablo 5.1. Deneklerden elde edilen performans ölçümleri	104

ÖZET

Anahtar kelimeler: Fare Kontrol, Dik Bakış, Göz Köşesi Tespiti, İris Tespiti, Göz Takibi, Koordinat Tespiti

Çalışmada dizüstü bilgisayar kullanıcılarının dâhili webcam ile alınan görüntüleri işlenerek imleç kontrolü amaçlanmıştır. Bu amaçla çalışma dâhili webcam ile alınan görüntü içinde göz bölgesinin bulunması, göz bölgesine ait özniteliklerin elde edilmesi, özniteliklerin kullanılarak ekranda bakılan nokta koordinatlarının tespiti ve fare kontrolü şeklinde aşamalandırılmıştır. Çalışmada bölge koordinatlarının tespiti önce araştırmacının kendine ait dâhili webcam görüntüleri kullanılarak gerçekleştirilmiş, daha sonra 11 deneye ait dâhili webcam görüntüleri kullanılarak tekrarlanmıştır.

Göz bölgesinin bulunması aşamasında bir yapay sinir ağı (YSA) geliştirilmiş ve eğitim seti ile birlikte hazırlanan test seti üzerinde %100 başarıya ulaşılmıştır. Sonraki aşamada göz bölgesine ait öznitelikler belirlenmiş ve bu öznitelikler göz bölgesinden elde edilmiştir. Göz bölgesi öznitelikleri olarak iris merkezi, göz köşe noktaları ve göz bölgesine ait piksel değerleri kullanılmıştır. Göz bölgesinin içinde irisin bulunması için Yapay Sinir Ağı (YSA)+Circular Hough Transform (CHT)-Dairesel Hough Transformu, göz köşelerinin bulunması için ise Lowest Valued Pixel (LVP)-En Düşük Değerlikli Piksel (EDDP) kullanılmıştır. Göz bölgesine ait her bir pikselin ifade ettiği renk derinliği değerleri (piksel değerleri) de boyut bazında azaltılarak kullanılmıştır. Ancak otomatik olarak bulunan iris ve göz köşelerinin sonuçlara olumlu katkısının olmadığı gözlemlenmiştir. Manuel olarak iris ve göz köşelerinin bulunmasının ise pratikte her görüntü için uygulanamaması sebebiyle, çalışmanın devamında göz bölgesini ifade etmesi için sadece piksel değerleri kullanılmıştır.

Kullanıcının dâhili webcam ile alınan görüntü içinde bulunan göz bölgesine ait pikseller, veri azaltma yöntemleri ile azaltılmış ve vektör matrisi haline getirilen piksellere ait değerler koordinat tespiti için geliştirilen YSA'da giriş verisi olarak kullanılmıştır. Kullanıcıya ait test görüntüleri, koordinat tespiti için geliştirilen YSA ile test edilmiş ve Mutlak Hata Ortalaması (MAE=1.6648 bölge – 1.6648×24 piksel) bulunmuştur. MAE değerinin küçük olması gerçek bölge koordinatları ile YSA'nın bulunduğu bölge koordinatları arasındaki sapmaların az olduğu anlamına gelmektedir. Kullanıcıya ait test görüntülerinden elde edilen sonuçlara bağlı olarak çalışma farklı kullanıcılar için deneklerin dâhili webcam görüntüleri kullanılarak tekrarlanmıştır. Deneklerden elde edilen sonuçların farklılıklar gösterdiği ve elde edilen en düşük mutlak hata ortalamasının araştırmacının elde ettiği değerlere yaklaştığı görülmüştür.

Çalışmanın sonucunda önceden belirlenmiş kullanıcıların göz hareketlerine bağlı olarak imlecin istenilen noktalara hareketi ortalama (MAE=5,8382 bölge) sapmayla gerçekleşmesi sağlanmıştır. Ortalama MAE değeri hesaplanırken eğitim ve test görüntülerini yanlış aldığı düşünülen deneklerde ortalama dahil edilmiştir.

EYE TRACING AND CURSOR CONTROL METHOD WITH DOMESTIC WEBCAM

SUMMARY

Keywords: Cursor Control, Eye Gaze, Eye Corner Detection, Iris Detection, Eye Tracking, coordinate Detection

In this study, it is aimed to control the cursor on the laptop screen with processing of captured images from the domestic webcam. For this aim, the study was divided into four steps; to detect eye in the image, to obtain the attributes of eye region, to determine the point coordinates looked on the screen with using eye region attributes, and to control the cursor. Firstly, the study has been realized with using the researcher images only. Later on, the study has been repeated again for 11 different test subject images.

In the second step, an Artificial Neural Network (ANN) has been developed and 100% success has been achieved on the test set. In the next step, the attributes, obtained from the eye region have been determined. The iris center, eye corners and pixels values of eye region have been selected as the attributes of eye region. ANN (Artificial Neural Network) & CHT (Circular Hough Transform) methods have been used to detect the iris and LVP (Lowest Valued Pixel) method has been used to detect the eye corners in eye region. The colour depth values (pixel values), define each pixel belong to eye region have been used by reducing the number of pixels. However, it is observed that, the iris and eye corners, detected automatically have not effected the results positively. Meanwhile, the detection of iris and eye corners couldn't be applied manually for each image. So, in the rest of study, only the pixel values has been used.

The number of pixels in eye region within the image of the user has been decreased by data reduction methods. The pixel values, converted into vector matrix have been used as the input data for the ANN, developed for the determination of coordinates. Test images of user have been tested on the ANN and Mean Absolute Error (MAE) has been found as 1,6648 area (1.6648×24 pixels). This small value of the MAE indicates that, there is less deviation between the actual and obtained coordinates. The study has been repeated for different users. It has been observed that, the results obtained from the test subjects were different. However, the minimum Mean Absolute Error obtained from the test subject was very close to the value obtained from the researcher.

As a result of the study, the cursor has been moved to the desired point with the average deviation (MAE=5,8382 area) by depending on eye movements of the predetermined users. The test subjects, which took the training and test images wrong, have been included when the MAE was calculated.

BÖLÜM 1. GİRİŞ

İnsan bilgisayar etkileşimi (İBE), kullanıcının bilgisayarı etkin bir şekilde kullanabilmesi olarak tanımlanabilir. İBE sistemlerinin geliştirilmesine yönelik çalışmalarda temel amaç bilgisayar kullanımının basitleştirilmesi ve kullanımın hızlandırılmasını sağlamaktır. Acartürk ve Çağiltay [1], İBE sisteminin 4 ana bileşenden oluştuğunu ifade etmektedirler. Bunlar; kullanıcı, görev, araç/arayüz ve bağlamdır. Bilgisayarlar, bilgisayar teknolojisinin gelişim süreci diyebileceğimiz uzun bir süre boyunca bilgisayar kullanmayı çok iyi bilen kullanıcılar tarafından kullanıldı. Çünkü bu dönemde bilgisayara veri girmek, bilgisayardan elde edilen çıktıları yorumlamak ve bu çıktıları kullanarak bilgisayara yeni veriler girmek oldukça zordu. İBE sistemlerinde kullanıcının etkisi çok fazla idi. Bu sebeple sayıca bilgisayar kullanıcıları da çok azdı.

Bilgisayar teknolojilerindeki gelişmeler araç/arayüz olarak ifade edilen bilgisayar yazılımlarının ve donanımlarının da gelişmesine olanak sağlamıştır. Arayüz yazılımsal, araç ise donanımsal bir ifade olduğu için birbirlerinden ayrılmaktadırlar. Bu ayrım, arayüzlerin araçlara göre daha hızlı gelişmelerine sebep olmuştur. Geliştirilen her yeni arayüz, kullanıcı üzerine düşen yükün bir kısmını bilgisayarların üzerine almasını sağlamaktadır. Böylece bilgisayar kullanımı giderek kolaylaşmakta, hatta günümüzde insanoğlu konuşmaya ve yürümeye başlamadan önce belli düzeyde bilgisayar kullanmaya başlayabilmektedir. Geliştirilen nesne tabanlı arayüzler sayesinde insanlar, bilgisayarların genel yapıları ve çalışmaları ile ilgili hiç bir bilgiye sahip olmasalar da bilgisayar kullanabilmektedirler. Bu da tüm insanların bilgisayar kullanıcısı olarak ele alınmasını gerektirmektedir. Bu sebeple geliştirilen arayüzlerin tüm insanların özellikleri dikkate alınarak daha kolay kullanıma sahip olması istenmektedir.

Diğer taraftan, bilgisayarlara veri girişi için kullanılan araçlar, arayüzler kadar hızlı olmasa da, geliştirilmeye çalışılmaktadır. Bilinen en klasik ve en yaygın kullanılan veri giriş birimleri klavye ve fare dir. Bununla birlikte bazılarının halen kullanım alanı bulduğu iz topu (trackball), joystick, sayısallaştırıcı (digitizer), tarayıcı (scanner), barkod okuyucu, ışıklı kalem, dokunmatik ekran (touch screen), dokunmatik fare (touchpad), tuş takımı gibi veri giriş birimleri de bulunmaktadır. Bu giriş birimleri incelendiğinde bazılarının kullanım alanlarını yitirdiği, bazılarının da özellikle spesifik alanlarda (dokunmatik fare - dizüstü bilgisayar, dokunmatik ekran – tablet / akıllı tahta / akıllı telefonlar, barkod okuyucu - market / mağaza bilgisayarları, joystick - oyun ekranları / bilgisayarları, tuş takımı - hesap makinesi / bankamatik / endüstriyel cihazlar gibi) yaygın olarak kullanıldığı görülmektedir. Bu noktada giriş birimlerinin sadece bilgisayarlar için değil mikroişlemci tabanlı elektronik cihazların hepsi için önem taşıdığı açıkça görülmektedir.

Bilgisayar kullanımında hedeflenen son noktanın ise “düşünceler ile kontrolün gerçekleştirilmesi” olduğu görülmektedir. Ancak bunun için çok uzun soluklu çalışmalar yapılması gerekliliği de göz ardı edilmemesi gereken bir gerçektir. İnsan bilgisayar etkileşiminde yeni gelişmeler üç bölümde kategorize edilebilir. Bunlar; Giyilebilir cihazlar, kablosuz cihazlar ve sanal cihazlardır [2]. Ancak bu kategoriler bile birbirlerinden tam olarak ayıramamaktadır [2]. Şekil 1a’da endüstriyel bir sanal klavye örneği verilmiştir. Bu donanım düz bir zeminde kullanılabilen, kırmızı bir ışık yayarak bir sanal klavye oluşturmaktadır. Sanal klavyenin tuşlarına basıldığında sesli bir sinyal verebilmekte, böylece kullanımı kolaylaştırmaktadır [3]. Şekil 1b’de bir başka endüstriyel ürün olan dijital kalem verilmiştir. Bu kalem bir aparatla kullanılmaktadır. Bu aparat dosya kıskacı gibi kullanılmakta ve kâğıt üzerine yazılan ya da çizilen her şey bilgisayar ortamına aktarılabilir. Hatta ekrana görsel olarak aktarılan yazılar Microsoft Word formatına çevrilebilmektedir [4].



(a)



(b)

Şekil 1.1. (a) Canesta sanal klavye [3] (b) Dark i-Pen dijital kalem [4]

Günümüzde daha ziyade akıllı ve adaptif insan bilgisayar etkileşim sistemleri geliştirilmektedir. Bu arayüz/araçlar, fiziksel, duyuşsal ve bilişsel kullanıcı etkinliklerini farklı düzeylerde içermektedir. Unimodal insan bilgisayar etkileşim sistemleri, görsel, işitsel ya da algısal tabanlı iletişim kanallarından birine sahiptir. Multimodal insan bilgisayar etkileşim sistemleri ise birden fazla sayıda iletişim kanalına sahiptir [2].

Eye Gaze (dik bakış-bakışlar ile kontrol) sistemler Multimodal insan bilgisayar etkileşim sistemlerine örnek olarak verilebilir [2]. Göz hareketlerinin takibine dayalı olarak çalışan Eye Gaze sistemler kişinin ekranda nereye baktığını bulmak için kullanılırlar [5]. Bu sayede çalışmakta olan bir sistemin kontrolü için kullanılabilirler. Bu sistemler felçli hastalara (başını ve gözlerini oynatabilen) ve okuma zorluğu çeken çocuklara yönelik kullanılmaktadır. Günümüzde halen geliştirme süreci devam eden bu sistemlerde göz hareketlerinin takibi için "Eye Tracker - göz takipçisi" olarak adlandırılan cihazlar kullanılır [5]. Göz takipçilerinin günden güne geliştirilmesi sonucunda, insan bilgisayar etkileşimi sırasında kullanıcıların göz hareketleri daha kolay ve doğru bir şekilde izlenebilmekte [6] buda bakılan nokta tespit başarısını artırmaktadır. Göz hareketlerinin göz takipçileri ile takip edilebilmesi için göz takipçisinin yapısına bağlı olarak, kullanıcıyla fiziksel olarak temas etmesi ya da kullanıcının göz bölgesine kızılötesi ışın yöneltmesi gerekmektedir [5]. Ayrıca Eye Gaze sistemlerde gözlerin sadece o anki pozisyonlarına göre sonuç üretilmez, göz hareketleri kaydedilir ve yeni sonuçların üretilmesinde bir ya da birkaç zaman gecikmeli sonuç da dikkate alınır. Gözler bakış esnasında sürekli hareket halindedir. Bu esnada sıçrama (saccade) ve sabitleme (fixation) hareketi yaparlar [5,7,8,9]. Bu sebeple kişinin ekranda hangi noktaya baktığı tespit edilirken, Eye Gaze sistemlerde gözün sabitleme anları belirlenir ve bu anlarda bakış noktaları tespit edilir [5]. Ancak sabitleme anlarında bile hareketlilik küçük de olsa devam eder [5]. Buda Eye Gaze sistemler gibi bakışlar ile kontrolü amaçlayan çalışmalarda başarı oranını olumsuz etkiler. Şekil 1.2'de Eye Gaze sistemlere örnekler verilmiştir.



(a)



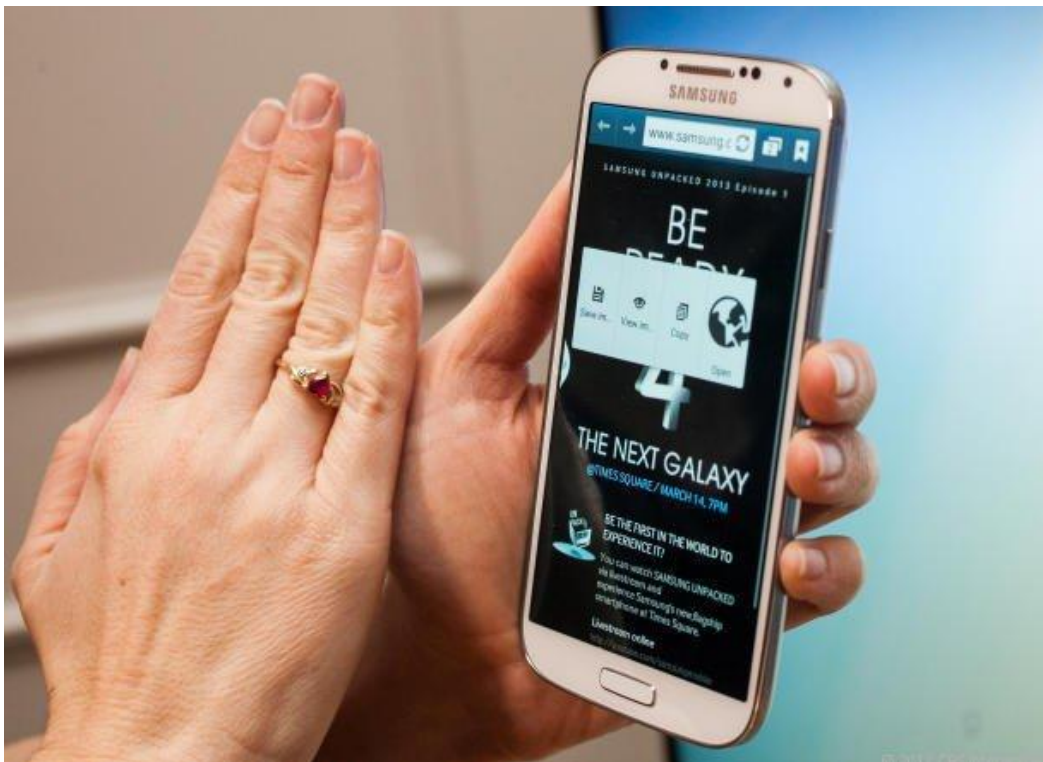
(b)

Şekil 1.2. (a) Tekerlekli sandalyeye monte edilmiş portatif Eye Gaze sistem [10] (b) Eye Gaze sistemin kullanılışı [11]

Günümüzde daha basit kontrol yöntemleri, Eye Gaze sistemler kadar etkin olmasa da farklı amaçlar için kullanılmaktadır. Örneğin, TV kontrolü için sesli komutlar, el-kol ya da vücut-kafa hareketleri, bilgi girişi amaçlı kullanılabilen ve böylece kanal değiştirme, ses arttırma-azaltma gibi kontroller gerçekleştirilebilmektedir. Sesli komutların algılanması ve aktarılması için bilgi giriş donanımı olarak mikrofon, el-kol ya da vücut-kafa hareketlerinin algılanması ve aktarılması için bilgi giriş donanımı olarak ise kamera kullanılmaktadır. Bilgisayar kontrolü için ise bu gibi kontrollerin yanında daha etkin kontrol yöntemleri kullanılmaktadır. Bu gibi cihazlarda kullanıcı ile çok daha fazla bilgi alışverişi yapılmakta ve bilgi alışverişinin hızlı ve kararlı olması gerekmektedir. Bu sebeple, bilgisayara bilgi girişinde yaygın olarak fare ve klavye kullanılmaktadır. Bununla birlikte dokunmatik ekranlarla fare ve klavyeyi sanal olarak ta kullanılabilir. Özellikle telefon gibi her geçen gün yeni bir bilgisayar özelliği daha kazanan cihazlara bilgi girişi, klasik yöntemleri (kumanda, fare, klavye gibi) çok aşmıştır. Çünkü insanlar telefonu sürekli yanlarında taşımakta ve ek donanım elemanlarını da istememektedirler. Hatta uzun bir dönem telefonlarda standart olarak kullanılmış olan tuş takımı bile günümüz akıllı telefonlarında kullanılmamaktadır. Akıllı telefonlarda standart bilgi giriş donanımı dokunmatik ekranlar olmuştur. Uzun bir süre telefon boyutları giderek küçülmeğe iken günümüzde görsel ve dokunsal açıdan kullanımının kolaylaşmasını sağlamak amacıyla boyutları giderek büyümektedir. Bunun yanında özellikle telefonlarda daha birçok bilgi giriş yöntemi kullanılmaktadır. Sesli arama, akıllı kaydırma (Eye Scroll-Herhangi bir görsele bakarken sayfanın en altına ya da en üstüne gelindiğinde ekranın kendi kendine kayması veya yeni sayfaya geçmesi), akıllı duraklatma (Eye Pause-Çalışmakta olan bir videonun bakılmadığında kendiliğinden durması), Temassız kontrol (ekrana temas etmeden el ya da parmak hareketleriyle kontrol) bunlardan bazılarıdır. Şekil 1.3'te akıllı telefon kullanım örnekleri verilmiştir.



(a)



(b)

Şekil 1.3. (a) Akıllı kaydırma [12] (b) temassız kontrol [13]

Görüldüğü gibi, mikrofon ve kameraların görsel elektronik cihazlarda bilgi giriş donanımı olarak kullanılması giderek yaygınlaşmaktadır. Bu çalışmada bakışlar ile kontrol amaçlandığından ve bu amaç için kullanılacak kameranın ekrana göre sabit bir pozisyona sahip olması gerektiğinden dizüstü bilgisayarlarda bulunan dâhili webcam ile çekilen kullanıcı görüntülerinden yararlanılabileceği düşünülmüştür. Bakışlar ile kontrol için kameradan alınan giriş verileri, şu an için amaca yönelik basit verilerdir (son satıra bakılmakta, ilk satıra bakılmakta, ekranın sağına ya da soluna bakılmakta gibi). Bu veriler ile basit kontroller yapılabilmektedir. Daha etkin kontrollerin gerçekleştirilebilmesi ise kameradan alınan giriş verilerinin daha detaylı bilgiler içermesi ile mümkün olacaktır. Böylece hiçbir ek donanım kullanılmadan sadece dâhili webcam görüntüleri kullanılarak bakışlar ile daha etkin kontrol yöntemleri geliştirilebilir.

Görüntü içinde göz bölgesine ait verilerin toplanması birçok alanda kullanılmaktadır. Sağlık [14], Güvenlik sistemleri, kişi tanıma [15,16,17,18] bunların başında gelenlerdir. Günümüzde ise özellikle kontrol sistemlerinde bakışlar ile kontrolün geliştirilebileceği düşüncesiyle, kullanıcı görüntülerinin işlenmesi ve göz bölgesine ait verilerin toplanmasına yönelik birçok çalışma bulunmaktadır [19,20,21,22,23].

İncelenen çalışmalarda temel amacın daha basit donanımlar kullanılarak toplanan verilerin işlenmesiyle (örneğin kızılötesi kamera yerine dâhili webcam) daha etkin özniteliklere (örneğin irisin bulunması, kaşın bulunması, burnun bulunması, göz köşelerinin bulunması) daha yüksek başarı oranları ile ulaşılması olduğu görülmektedir.

Kontrol edilmek istenen sistemin özellikleri (bilgisayar ekranındaki imleç, sanal klavye üzerindeki şifreleme sistemi, metin oluşturucu, bir mekanik robot kol veya robot vb.) ve kontrol amacı her ne kadar kullanılacak donanımı etkileyecek olsa da bu tür sistemlerde görüntü almak amacıyla kamera kullanılacağı ve elde edilen görüntünün işleneceği aşikârdır.

Çalışmada dâhili webcam ile elde edilen görüntüler işlenerek göz bölgesine yönelik özniteliklerin çıkarılması üzerine yoğunlaşmıştır. Bu amaçla webcam görüntüleri

içinde yapay sinir ağıları (YSA) kullanılarak göz ve iris bölgeleri bulunmuştur. Bu bölgelerde Dairesel Hough Transformu (CHT), Ağırlıklı Varyans Bağdaştırma Fonksiyonu (WVPF) ve En Düşük Değerlikli Pikselin bulunması (EDDP) gibi yöntemler kullanılarak otomatik olarak iris merkezi ve göz köşeleri bulunmuştur.

Elde edilen öznitelikler ile bakış koordinatlarının tahmin edilebileceği ve farklı alanlarda kullanılabileceği düşünülmektedir. Örneğin sanal klavyeler üzerinde girilmesi gereken şifrelerin klavyedeki tuşlar kullanılmadan girilebileceği, bakışlar ile kontrol edilmek istenen sistemlerin kontrollerinin gerçekleştirilebileceği, fare kullanmadan imlecin hareket ettirilebileceği, sanal gerçeklik ortamlarında bazı kontrollerin bakışlar ile gerçekleştirilebileceği, üç boyutlu hazırlanmış oyun programlarında joysticklerle yapılan kontrolün bakışlar ile gerçekleştirilebileceği, özel paket programlar ile uyumlu bir şekilde çalıştırılarak program içinde bazı kontrollerin bakışlar ile gerçekleştirilebileceği düşünülmektedir.

1.1. Çalışmanın Amacı ve Kapsamı

1.1.1. Çalışmanın amacı

Bu tez çalışması ile dizüstü bilgisayar kullanıcılarına ait dâhili webcam görüntüleri kullanılarak, kullanıcının ekranda baktığı koordinatların başka bir ek donanım kullanılmadan tespit edilmesi ve imlecin bu koordinatlara taşınması amaçlanmıştır.

1.1.2. Çalışmanın literatürdeki yeri

Bakışlar ile kontrol Eye Tracker ‘göz takipçisi’ olarak bilinen ek donanım ile gerçekleştirilmektedir. Ancak ek donanım kullanılması hem kullanımı zorlaştırmakta hem de maliyeti arttırmaktadır. Bu sebeple yaygın olarak kullanılması da mümkün olmamaktadır. Günümüzde dâhili kameralar(webcam)’dan alınan kullanıcı görüntüleri ile kontrol üzerine çalışmalar devam etmektedir. Böylece ek donanım kullanımından kaynaklanan olumsuz etkilerden kurtulabilmeyebiliriz. Ancak sadece dahili webcam kullanılarak yapılan kontrol, Eye tracker ile yapılan kontrol düzeyine ulaşmamıştır. Şu an için sağa baktı, sola baktı, yukarı baktı, aşağı baktı gibi basit

kontroller yapılabilir. Bunun sebebi şu ana kadarki çalışmalarda görüntü işleme sonucunda bakış ile alakalı olarak yön bilgisine ulaşılabilmesi fakat koordinat bilgisine ulaşamamasıdır. Çalışmada bakılan noktanın koordinat tespiti yapıldığından literatürdeki bu eksikliğin giderilmesi hedeflenmiştir.

1.1.3. Tez çalışmasının yürütülmesi

Bu tez çalışması dört aşamadan oluşmaktadır. Bunlar;



olarak belirlenmiştir.

İlk aşamada, göz bölgesinin bulunması amacıyla dâhili webcam ile elde edilen kullanıcı görüntüleri ve bu görüntüler ile eğitilen bir YSA kullanılmış ve gerçek zamanlı (real time) olarak göz bölgesini bulabilen Matlab tabanlı bir yazılım geliştirilmiştir. Böylece kullanıcının anlık olarak bulunan göz bölgesine ait belirlenmiş özniteliklere gerçek zamanlı (real time) ulaşılabilir. Böylece dâhili webcam'in dizüstü bilgisayarlarda standart olarak bulunan bir donanım olduğu da düşünülerek, özniteliklerin tespit edilmesi amacıyla hiçbir ek donanım kullanılmamış olacaktır.

İkinci aşamada, gözlerin ekranda baktığı nokta koordinatlarının bilgisini içerdiği düşünülen öznitelikler belirlenmiştir. Öngörülen öznitelikler başlangıçta iris merkezi

koordinatları, göz köşe noktaları koordinatları, bu noktalar arasındaki mesafeler ve göz bölgesi piksel değerleridir. İris merkezi koordinatlarının tespit edilmesi amacıyla Dairesel Hough Transformu, göz köşelerinin bulunması için ise en düşük değerlikli pikselin bulunması gerçekleştirilmiştir. Ancak çalışmada öngörülen sonuçlara ulaşılamaması sebebiyle, göz bölgesini içeren görüntünün piksel değerleri kullanılarak çalışma tekrarlanmış ve elde edilen sonuçların istenilen değerlere yakın olması nedeniyle, çalışmaya göz bölgesine ait piksel değerleri kullanılarak devam edilmiştir. Bu aşamada, kullanıcının ekranda baktığı bölgelerin bilindiği eğitim ve test görüntüleri hazırlanmıştır. Eğitim ve test görüntüleri kullanılarak YSA'da kullanılmak üzere eğitim ve test veri setleri oluşturulmuştur.

Üçüncü aşamada, eğitim görüntülerinin göz bölgelerinden elde edilen piksel değerleri kullanılarak hazırlanan eğitim veri seti, koordinat tespiti yapması amacıyla geliştirilen YSA'yı da içeren Matlab tabanlı yazılıma uygulanarak eğitim aşaması tamamlanmıştır. Bu eğitim veri seti ile oluşturulan YSA, test görüntülerinden elde edilen test veri seti ile denenmiş ve imleç elde edilen koordinat noktalarına taşınmıştır.

Son aşamada, hazırlanan Matlab yazılımlarının farklı kullanıcı görüntüleri ile tekrarlanması planlanmıştır. Bu amaçla deneklerden eğitim ve test görüntülerinin alınabilmesi için arayüz tasarlanmış, çalışmaya gönüllü olarak katılmak isteyen denekler belirlenmiş, denekler görüntü alma ve arayüzü kullanma hakkında bilgilendirilmiştir. Deneklerden alınan eğitim ve test görüntüleri kullanılarak çalışma her bir denek için ayrı ayrı tekrarlanmıştır. Farklı kullanıcılar için çalışmanın yaklaşık sonuçlar verdiği görülmüştür.

BÖLÜM 2. GÖRÜNTÜ İŞLEME

Görüntü insan yaşamında önemli bir yere sahiptir. Görme duyusu 5 duyunun en önemlilerindedir. Çünkü insanoğlu öğrenmede dâhil olmak üzere birçok karar aşamasında görüntülerden faydalanmaktadır. Derinlik, boyut, nesne sayısı, rengi, özellikleri ve daha birçok bilgiye görme duyusunu kullanarak ulaşmaktadır.

2.1. Görüntünün İçerdiği Bilgiler Ve Görüntü İşlemenin Kullanım Alanları

Görüntüler birçok bilgi içermekte ve insanlar yaşantılarında bu bilgilerden faydalanmaktadır. Bunun yanı sıra günümüzde fotoğraf makineleri ve kameralar ile elde edilen görüntüler birçok farklı amaç için kullanılmaktadır. Görüntülerin genel olarak derinlik, boyut, nesne sayısı, nesne rengi ve nesne özellikleri gibi bilgileri içerdiği söylenebilir. Bununla birlikte görüntülerin içerdiği bilgileri sıralamak yerine kullanım alanlarını incelemek daha anlamlı ve görüntülerin içerdiği bilgiler açısından daha bilgilendirici olacaktır. Günümüzde görüntü işleme birçok alanda kullanılmakla beraber bu alanlar aşağıdaki gibi gruplandırılabilir;

Arşivleme ve belgelendirme: Bir görüntü mutlu bir anın kayıt altına alınarak arşivlenmesi ya da belgelenmesi amacıyla kullanılabilceği gibi, bir olayın ya da durumun belgelenmesi amacıyla da kullanılabilir. Görüntüler bu amaçlar için kullanıldığında görüntü kalitesi ön plana çıkmakta ve görüntü kalitesini artırmak önem kazanmaktadır. Görüntülerin en eski ve önemli kullanım alanlarından biri olması sebebiyle, görüntü kalitesinin artırılmasına yönelik birçok çalışma yapılmıştır.

İletişim: Günümüzde birey temelli iletişimde sosyal medya ve benzeri platformlar üzerinden görüntüler yaygın olarak kullanılmaktadır. Bu kullanım alanında da görüntü kalitesi ön plana çıkmaktadır. Bununla birlikte görüntüler üzerinde işaretleme vb uygulamalar da kullanılmaktadır.

Medya: Toplum temelli iletişimde dergi, gazete, TV gibi araçlar üzerinden görüntüler yaygın olarak kullanılmaktadır. TV gibi sürekliliği olan araçlarda görüntüler arka arkaya eklenerek videolar oluşturulmaktadır. Bu sebeple görüntü kalitesi yanında, artan görüntü boyutları sebebiyle görüntü sıkıştırma ve uzak mesafelerde görüntü aktarımı önem kazanmaktadır.

Tıp: Röntgen, kanserli hücrelerin tespiti, doku analizi, patolojik araştırma vb amaçlarla alınan görüntülerde belirli nesnelere aranmaktadır. Bu sebeple görüntü kalitesinden ziyade nesnenin belirginliği önemlidir ve gözle kontrol edilmektedir.

Savunma sanayii: Termal, uydu, radar görüntüleri gibi görüntülerin kullanıldığı bu alanda da belirli nesnelere aranmaktadır ve gözle kontrol edilmektedir.

Güvenlik: Parmak izi tarama, avuç içi tarama, retina tespiti gibi uygulamalarda alınan görüntüler işlemciler ile işlenmektedir. Güvenlik kamerası, mobese kamerası gibi kameralar ile alınan görüntüler ise kaydedilerek gerek görüldüğünde ve gözle kontrol edilmektedir. Alarm sistemleri ile entegre edilmiş kameralar ile alınan görüntülerde ise, görüntüler işlemcilerle işlenmekte ve kontrol görüntüleri ile gerçek zamanlı görüntü arasında oluşan farklar sistemin çalışmasını sağlamaktadır.

Trafik: Yolcu yoğunluğu, araç yoğunluğu tespiti gibi uygulamalarda ardışık görüntüler kullanılmakta ve optik akışa (optical flow) göre görüntüler değerlendirilmektedir.

Güvenli sürüş sistemleri: Şerit takibi gibi uygulamalarda yol görüntüleri alınmakta ve bu görüntüler içinde görüntü işleme teknikleri ile şeritler tespit edilerek şerit ihlalleri belirlenmekte ve sürücüler uyarılmaktadır. Araç arka kamerası gibi uygulamalarda ise, sürücünün görmediği bölgeler kamera ile alınıp yol bilgisayarına aktarılmakta ve sürücünün bu bölgeleri kontrol etmesi sağlanmaktadır.

Kalite kontrol: Görüntüler üretim tesislerinde üretilen ürünlerin kalitesinin kontrolü amacıyla kullanıldığı gibi, tarım alanında da ürünlerin olgunlaşmış olgunlaşmadığı ya

da ürün boyutlarının tespiti amacıyla kullanılabilir. Bu kullanım şeklinde istenen özelliklere sahip ürün görüntüleri ile kontrol edilmek istenen ürünün görüntüleri karşılaştırılmakta ve sonuç elde edilmektedir. Ürünün özelliğine göre değişiklik gösterebilen bu yöntemde bir başka kullanım şekli de üründe istenmeyen belirgin özellikler görüntü içinde aranarak karşılaştırma yapmadan da sonuç elde edilebilmektedir.

Nesne tespiti: Görüntü içinde trafik lambası, trafik levhası, araç, plaka, yüz bölgesi, göz bölgesi vb nesnelerin tespiti amacıyla kullanılan bu yöntemde Haar sınıflandırıcılar, yapay sinir ağları, destek vektör makineleri (DVM-Support Vector Machines) gibi sınıflandırıcılar kullanılarak eğitilmekte ve görüntü içinde nesne tespiti gerçekleştirilmektedir. El yazısı tanıma sistemleri gibi çalışmalarda da karakterler ayrı ayrı tespit edilmektedir.

Nesne takibi: Nesne tespiti için geliştirilen sınıflandırıcıların ardışık görüntülere uygulanması ile nesne takibi gerçekleştirilebilmekte ve elde edilen bilgiler kontrol amaçlı kullanılabilir.

Araştırma: Elektromikroskop, gözlem, uzay araştırmaları gibi çalışmalarda da görüntüler kullanılmakta ve bu görüntülerden çıkarımlar yapılabilmektedir.

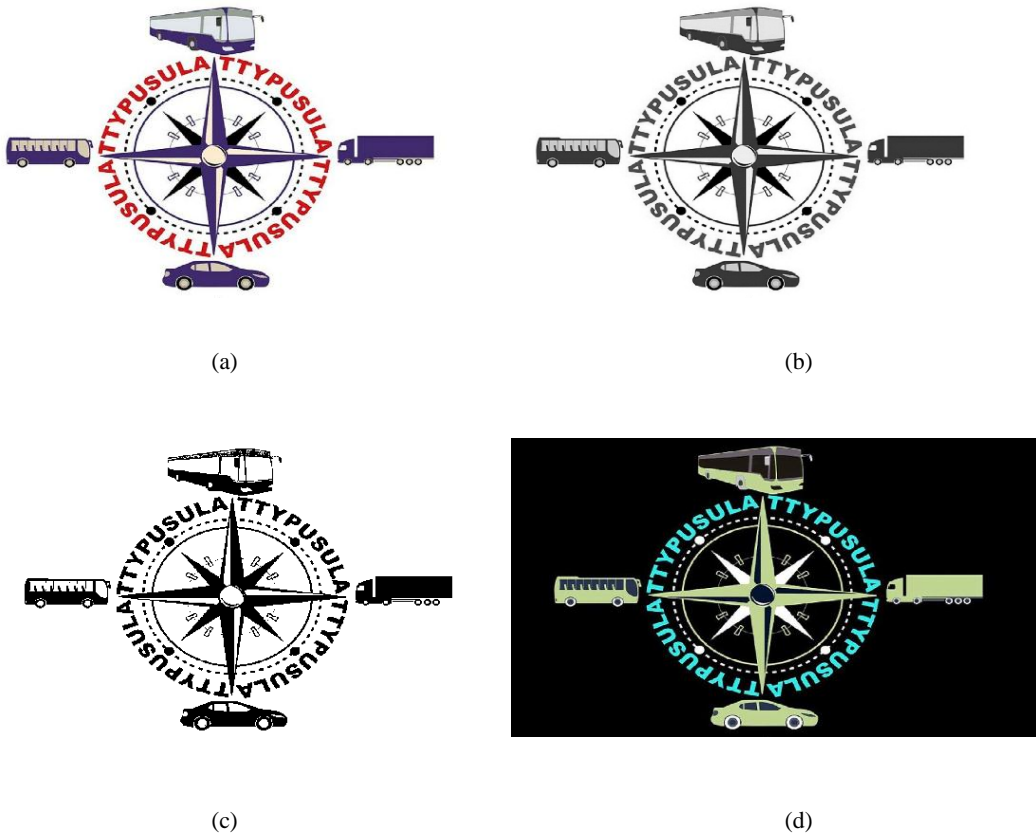
Bilgisayarlı görü: Bilgisayar aracılığı ile görme işleminin gerçekleştirilmesi, yani çevre ile herhangi bir fiziksel etki olmaksızın optik elemanlar kullanılarak çevre ile ilgili veri toplama işlemidir [24, 25]. Elde edilen veriler güvenli sürüş sistemlerinde, bilgisayar kontrolünde, endüstriyel makinelerin kontrolünde ve benzeri kontrol uygulamalarında kullanılabilir.

2.2. Sayısal Görüntü

Sayısal görüntü matematiksel olarak iki boyutlu bir $f(x,y)$ fonksiyonu olarak ifade edilmektedir. Bir (x,y) koordinatının o noktadaki genliği görüntünün parlaklığı veya gri seviyesi olarak isimlendirilir. Görüntü içindeki her bir $x, y, f(x,y)$ değeri ayrık ve sonlu bir tamsayı ise görüntü sayısal görüntü olarak isimlendirilir [26]. Görüntü

içinde her bir (x,y) koordinatı görüntünün ayrı bir elemanı olup piksel olarak isimlendirilir.

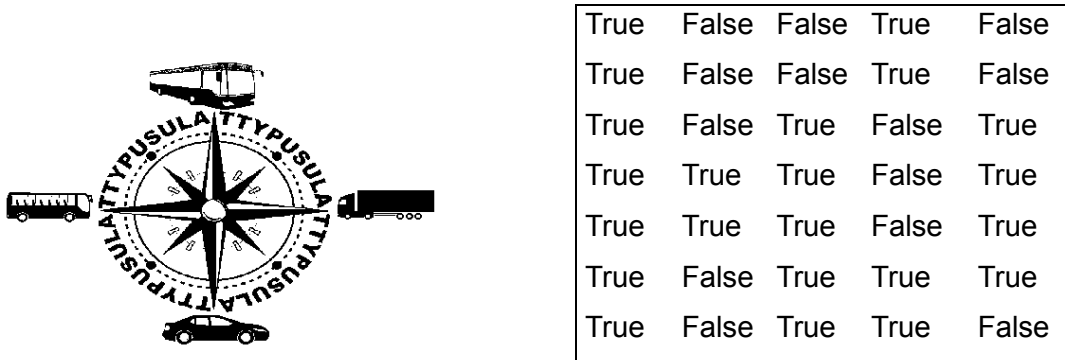
Görüntüler, binary (0-1), gri seviyeli (0-255) ve renkli (0-255×3) görüntüler olarak elde edilerek veya birbirine dönüştürülerek kullanılabilir. Şekil 2.1’de renkli, gri seviyeli, binary ve negatif görüntü örnekleri verilmiştir.



Şekil 2.1. (a) Orijinal görüntü (b) gri görüntü (c) ikili görüntü (d) negatif görüntü örnekleri

Renkli görüntülerin gösteriminde renk modelleri kullanılır. Bu renk modelleri RGB, CMYK ve HSI gibi renk modelleridir. Renkli görüntüler 3 boyut ile temsil edilir. Örneğin RGB (Kırmızı, Yeşil, Mavi) renk modelindeki bir renkli görüntü 3 adet 2 boyutlu matrisin birleşimi ile oluşturulmaktadır. İnsan gözü renkleri kırmızı, yeşil ve mavinin kombinasyonları olarak algılar [27]. RGB renk modelindeki her bir matris kırmızı, yeşil ve mavinin 0-255 aralığındaki parlaklık seviyesi ile ifade edilir ve bu renklerin farklı oranlarda karışımıyla yada dijital görüntüde üst üste çakışmasıyla

seviyesinin altındaki parlaklık seviyeleri 0, üstündeki parlaklık seviyeleri 1 ile ifade edilmektedir. Bu işleme eşikleme (thresholding) denilmektedir. İkili görüntüler görüntü işleme uygulamalarında maskeleme işlemlerinde sıkça kullanılmaktadır [28]. Şekil 2.4’de ikili görüntü ve ikili görüntü piksel değerlerine örnek verilmiştir. İkili görüntülerde piksel değerleri 0-1 şeklinde ifade edilebildiği gibi true-false şeklinde de ifade edilebilmektedir.



Şekil 2.4. İkili görüntü ve ikili görüntü piksel değerleri

Negatif görüntüler renkli görüntüler ile aynı özelliklere sahip olmakla birlikte renkli görüntünün negatifi yani tersidir. Bunu şöyle açıklamak mümkündür: Renk modelindeki her bir renge (kırmızı, yeşil, mavi) ait matrislerdeki piksel değerleri 255’den çıkarılarak yeni piksel değerleri elde edilir. Yeni piksel değerleri ile oluşturulan matrisler tekrar renkli görüntü oluşturacak şekilde birleştirildiğinde görüntünün negatifi elde edilir. Şekil 2.1’de görüntü ve görüntünün negatifi verilmiştir.

2.3. Sayısal Görüntü İşleme

Sayısallaştırılmış görüntülerin bilgisayar ile işlenmesine sayısal görüntü işleme denilmektedir [26]. Görüntü işlemeyi, ölçülmüş veya kaydedilmiş olan sayısal görüntü verilerinin elektronik ortamda amaca uygun şekilde değiştirilmesi olarak tanımlayabiliriz [29]. Bilgisayar teknolojisindeki gelişmeler ve işlemci hızlarındaki artış ve maliyetlerdeki azalma doğrultusunda son yıllarda görüntü işleme, bilimsel,

endüstriyel, uzay ve hükümet uygulamalarında önemli bir yer edinmiştir [30]. Günümüzde görüntü işleme çalışmaları artış göstermektedir. Özellikle endüstriyel alanda görüntü işlemenin kendine yer edinmesi bilim adamlarının ve endüstri kuruluşlarının ilgisini çekmeye devam etmektedir.

Gonzalez ve Woods sayısal görüntü işlemenin temel adımlarını aşağıdaki gibi belirlemiştir;

1. Görüntünün elde edilmesi (image acquisition)
2. Görüntü iyileştirme (image enhancement)
3. Görüntü yenileme (image restoration)
4. Renkli görüntü işleme (colour image processing)
5. Dalgacık ve çoklu çözünürlük işleme (wavelets and multiresolution processing)
6. Görüntü sıkıştırma (image compression)
7. Morfolojik görüntü işleme (morphological image processing)
8. Görüntü bölütleme (image segmentation)
9. Sunum ve özellik çıkarımı (representation ve description)
10. Nesne tanıma (object recognition) [26].

2.3.1. Görüntü çözünürlüğü

Gri seviyeli sayısal görüntülerde her bir (x,y) koordinatı bir pikseli ifade eder. Görüntü çözünürlüğü ise görüntüyü oluşturan piksel sayısı ile doğru orantılıdır. N satıra ve M sütuna sahip bir görüntünün çözünürlüğü $M \times N$ şeklinde ifade edilir. Böylece hem sahip olduğu satır ve sütun sayıları ifade edilirken hem de toplam piksel sayısı verilmiş olmaktadır.

Renkli görüntülerde ise görüntünün 3 boyutlu olması sebebiyle N satır ve M sütuna sahip bir görüntünün çözünürlüğü $M \times N \times 3$ şeklinde ifade edilir. Bu görüntülerde piksel koordinatıda $M \times N \times i$ şeklinde ifade edilmektedir. Burada i, pikselin z eksenindeki yerini ifade etmektedir.

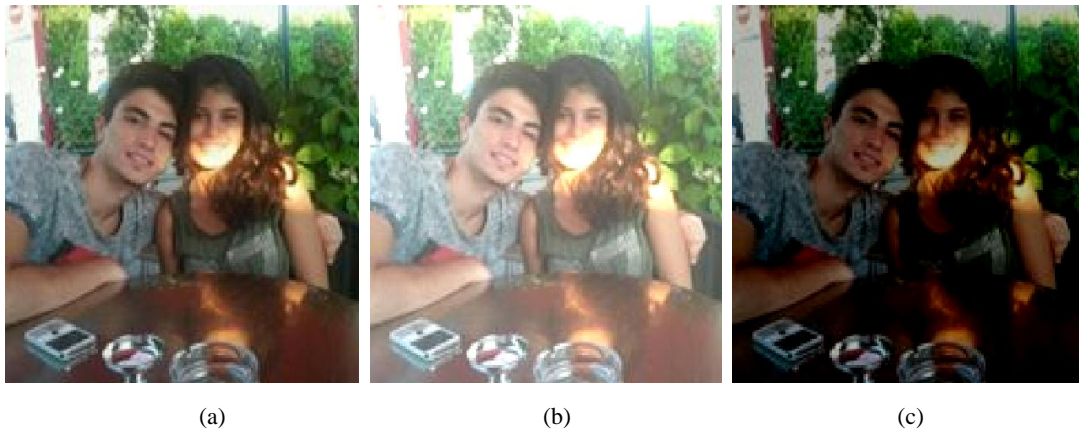
Görüntü aktarma araçlarında (ekran) çözünürlük, desteklediği en üst görüntü çözünürlüğünü ifade etmesi sebebiyle ekran kalitesiyle doğru orantılı olarak anılmaktadır. Benzer şekilde görsel amaçlar için alınan görüntülerde de çözünürlük görüntü kalitesi ile doğru orantılı olarak değerlendirilir. Ancak işlenecek görüntülerde çözünürlüğün yüksek olması işlem sayısının çok artmasına sebep olmaktadır. Bu sebeple işlenecek görüntülerde optimum çözünürlüğün bulunması önem arz etmektedir.

2.3.2. Görüntü işleme uygulamalarında kullanılan bazı algoritma ve yöntemler

Parlaklık (Brightness) Ayarı: Parlaklık ayarı görüntü içindeki piksel değerlerinin bir sabit ile toplanması ile yapılmaktadır.

$$g(x,y) = f(x,y) + a \quad (2.1)$$

Denklem 2.1'de $f(x,y)$ orijinal görüntüyü ifade ederken, $g(x,y)$ parlaklık ayarı yapılmış görüntüyü ifade eder. Eğer a sıfırdan büyükse parlaklık artırılmış, a sıfırdan küçükse parlaklık azaltılmış olur. Parlaklık ayarı yapılmış görüntü örnekleri Şekil 2.5'de verilmiştir.

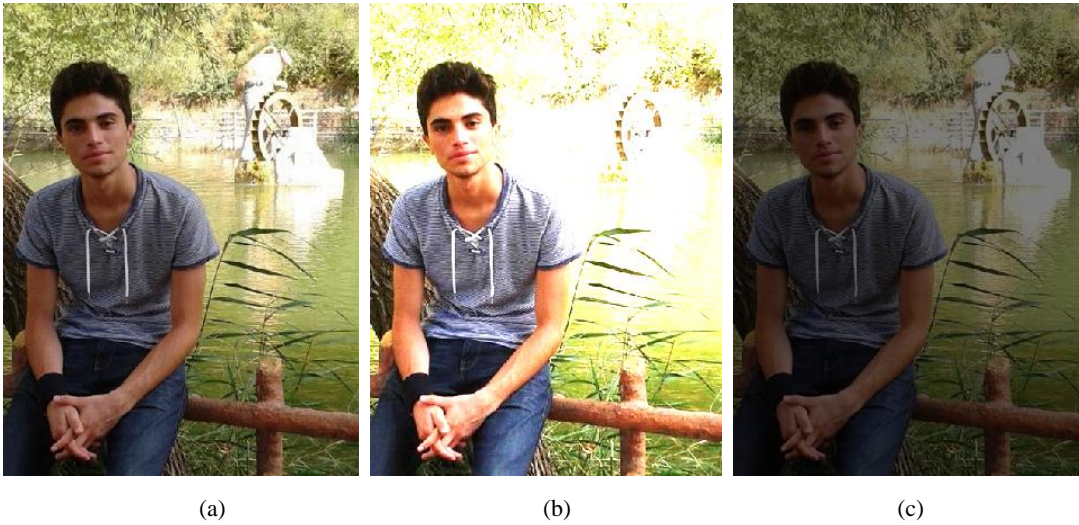


Şekil 2.5. (a) Orijinal görüntü (b) parlaklığı artırılmış görüntü ($a = +70$) (c) parlaklığı azaltılmış görüntü ($a = -70$)

Karşıtlık (Contrast) Ayarı: Karşıtlık ayarı görüntü içindeki piksel değerlerinin bir sabit ile çarpılması ile yapılmaktadır.

$$g(x,y) = f(x,y) \times b \quad (2.2)$$

Denklem 2.2’de $f(x,y)$ orijinal görüntüyü ifade ederken, $g(x,y)$ karşıtlık ayarı yapılmış görüntüyü ifade eder. Eğer a sıfırdan büyükse karşıtlık artırılmış, a sıfırdan küçükse karşıtlık azaltılmış olur. Karşıtlık ayarı yapılmış görüntü örnekleri Şekil 2.6’da verilmiştir.

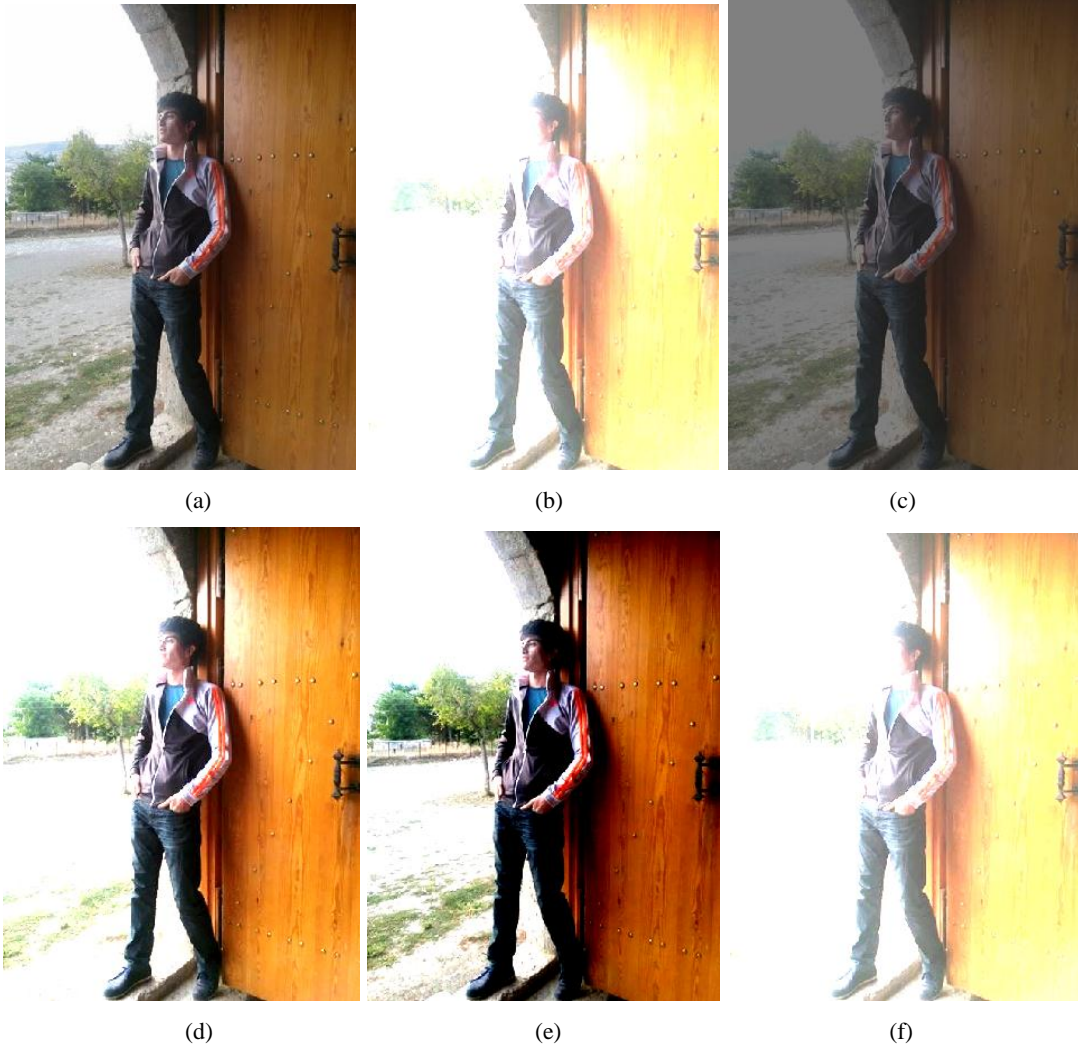


Şekil 2.6. (a) Orijinal görüntü (b) karşıtlığı artırılmış görüntü ($b = 2$) (c) karşıtlığı azaltılmış görüntü ($b = 0.5$)

Karşıtlık ve parlaklık ayarı birlikte de kullanılabilir. Bu durumda denklem 2.3’de verildiği gibi kullanılır.

$$g(x,y) = (f(x,y)+a) \times b \quad (2.3)$$

Karşıtlık ve parlaklık ayarı yapılmış görüntü örnekleri Şekil 2.7’de verilmiştir.



Şekil 2.7. (a) Orijinal görüntü (b) $a=+50$ $b=2$ (c) $a=+50$ $b=0.5$ (d) $a=-10$ $b=2$ (e) $a=-50$ $b=2$ (f) $a=-50$ $b=0.5$

Histogram eşitleme: Sayısal görüntülerde histogram, görüntü içerisinde renk değerinden kaç adet olduğunu gösteren bir grafikdir. Bu grafik, değerlerin sayıca dağılımını göstermektedir. Bu grafiğe bakılarak görüntünün parlaklığı ve karışıklığı hakkında bilgi sahibi olunabilir [29]. Histogramlar, bir görüntüdeki gri seviye değerlerinin ya da farklı renk bileşenlerinin her birinin dağılımını göstermek için elde edilebilirler. Bu özelliğinden dolayı görüntü iyileştirme ve görüntü içinde farklı gri seviyelere sahip bölümlerin ayrıştırılmasında (nesne tespiti vb.) kullanılabilirler.

Histogram eşitlemedeki temel mantık, belli bir bölgede toplanmış olan piksellere ait renk değerlerinin farklı renk değerlerine dağıtılmasıdır. Bunun için önce görüntüde hangi renk değerinden kaç adet olduğu belirlenir (n_k). Daha sonra her bir renk değeri için olasılık yoğunluk fonksiyonu kullanılır ($Pr(r) = n_k/(M \times N)$). Böylece her bir renk

değerinin görüntü içinde kullanım yoğunluğu belirlenmiş olur. Olasılık yoğunluk fonksiyonu sonuçları kullanılarak kümülatif histogram yani renk değerlerinin histogram eşitlenmiş görüntüdeki yeni piksel renk değerleri ($s_k = (L-1) \times \sum \Pr(r)$) belirlenir. Böylece histogram eşitleme gerçekleştirilir ve histogram eşitlenmiş görüntü oluşturulur.

Histogram eşitlemeyi basit bir uygulama örneği ile şöyle açıklanabilir. 3 bitlik ve $[M \times N] = [64 \times 64]$ bir görüntünün rk ve bunlara ait nk değerlerinin Tablo 2.1'deki gibi verildiği varsayalım.

Tablo 2.1. 3 bitlik renk değerleri ve renk değerlerine sahip piksel sayısı

rk	nk
0	790
1	1023
2	850
3	656
4	329
5	245
6	122
7	81

$\Pr(r_k)$ olasılık yoğunluk fonksiyonunu ve s_k hesaplandığında;

$$\Pr(r_0) = 0790/(64 \times 64) = 0,19; \quad s_0 = ,19 \times 7 = 1,33 \equiv 1$$

$$\Pr(r_1) = 1023/(64 \times 64) = 0,25; \quad s_1 = (,19 + ,25) \times 7 = 3,08 \equiv 3$$

$$\Pr(r_2) = 0850/(64 \times 64) = 0,21; \quad s_2 = (,19 + ,25 + ,21) \times 7 = 4,55 \equiv 5$$

$$\Pr(r_3) = 0656/(64 \times 64) = 0,16; \quad s_3 = (,19 + ,25 + ,21 + ,16) \times 7 = 5,67 \equiv 6$$

$$\Pr(r_4) = 0329/(64 \times 64) = 0,08; \quad s_4 = (,19 + ,25 + ,21 + ,16 + ,08) \times 7 = 6,23 \equiv 6$$

$$\Pr(r_5) = 0245/(64 \times 64) = 0,06; \quad s_5 = (,19 + ,25 + ,21 + ,16 + ,08 + ,06) \times 7 = 6,65 \equiv 7$$

$$\Pr(r_6) = 0122/(64 \times 64) = 0,03; \quad s_6 = (,19 + ,25 + ,21 + ,16 + ,08 + ,06 + ,03) \times 7 = 6,86 \equiv 7$$

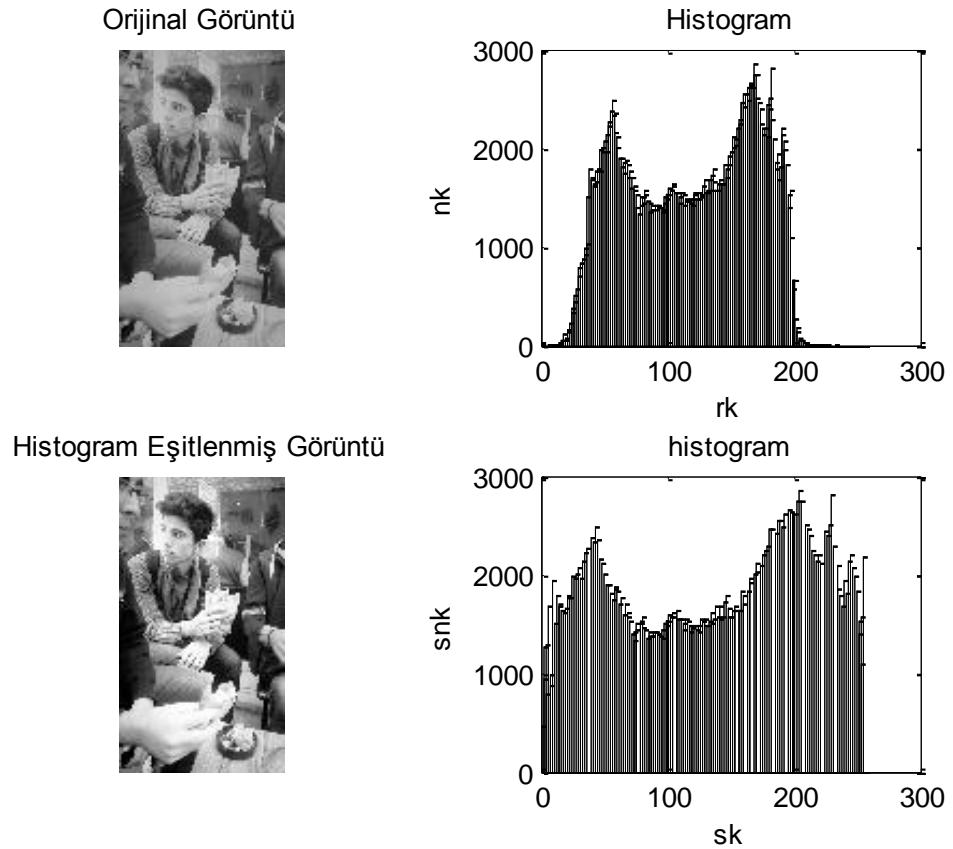
$$\Pr(r_7) = 0081/(64 \times 64) = 0,02; \quad s_7 = (,19 + ,25 + ,21 + ,16 + ,08 + ,06 + ,03 + ,02) \times 7 = 7 \equiv 7$$

Tablo 2.1 yeniden düzenlenerek Tablo 2.2 elde edilir;

Tablo 2.2. 3 bitlik renk deęerleri, bu renk deęerlerine sahip piksel sayısı, renk deęerlerinin grnt iindeki yoęunlukları ve piksellere atanan yeni renk deęerleri

rk	nk	Pr(rk)	sk
0	790	0,19	1
1	1023	0,25	3
2	850	0,21	5
3	656	0,16	6
4	329	0,08	6
5	245	0,06	7
6	122	0,03	7
7	81	0,02	7

Őekil 2.8’de Matlab platformunda histogramı eŐitlenmiŐ grnt rneęi verilmiŐtir.



Őekil 2.8. Orijinal grnt ve histogramı ile histogramı eŐitlenmiŐ grnt ve histogramı

EŐikleme (Thresholding): Genellikle eŐikleme gri seviyedeki grntnn ikilięe dnŐtrlmesi amacıyla kullanılmaktadır. Bylece grnt iinde farklı tonlara sahip nesnelere tespit edilebilir. Bunun iin uygun bir eŐik deęerinin seilmesi gerekir. EŐik deęerinin seiminde grntnn histogramından faydalanılır. EŐik deęerinin

adaptif olarak belirlenmesi amacıyla birçok yöntem önerilmiştir [31]. Eşiklenmiş görüntü örneği Şekil 2.9'da verilmiştir.



(a) (b)

Şekil 2.9. (a) Orijinal görüntü (b) eşiklenmiş görüntü

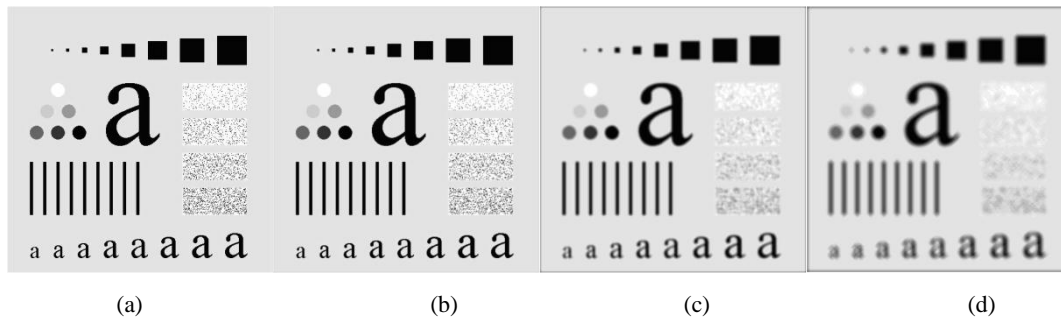
Filtreleme: Görüntü içinde herhangi bir sebeple meydana gelen gürültülerden kurtulmak için görüntülere filtreler uygulanır. Dijital görüntülerde gürültüler kayıt cihazlarından, ışık şiddeti yetersizliğinden, veri aktarım cihazlarından ve benzeri birçok sebepten meydana gelebilmekte ve çok fazla çeşitlilik göstermektedir. Görüntü işlemede kullanılan filtreler de dijital görüntüler gibi iki boyutludur. Farklı boyutlarda filtreler kullanılmakla beraber genellikle 3×3 boyutlarında filtreler kullanılmaktadır. Filtreleme işlemi, filtrenin filtrelenecek görüntü üzerinde kaydırılarak gezdirilmesi ile gerçekleştirilir. Bu gezdirme esnasında filtre elemanı ile ona karşılık gelen piksel değeri çarpılır, tüm çarpımlar toplanır ve filtre elemanlarının toplam değerine bölünerek ağırlıklı ortalaması hesaplanır. Elde edilen değer filtre merkezindeki elemana denk gelen pikselin yeni değeridir [27]. Görüntü işlemede bu işleme konvolüsyon denmektedir. Filtrelemedeki temel mantık böyle olsada bu şekilde gerçekleştirilen filtreler genellikle alçak geçiren, yumuşatma ya da ortalama filtreleri olarak anılır. Bu filtrelerde pozitif değerler kullanılır ve konvolüsyon işleminin sonucu filtre elemanlarının toplamına bölünür. Diğer filtre çeşitlerinde ise, filtre matrisinin değerleri yapılmak istenen filtre işlemine göre belirlenmekte, negatif değerlerde alabilmektedir. Şekil 2.10'da 3×3 , 5×5 , 9×9 boyutlarında alçak geçiren (low pass) filtre örnekleri verilmiştir. Bu filtrelerden 3×3 boyutlarında olan ile görüntü filtrelendiğinde, elde edilen konvolüsyon işleminin sonucu $1/8$ ile, 5×5

boyutlarında filtre kullanıldığında konvolüsyon işleminin sonucu 1/16 ile, 9×9 boyutlarındaki filtre kullanıldığında konvolüsyon işleminin sonucu 1/44 ile çarpılarak (filtre elemanlarının toplamına bölünerek) yeni piksel değeri bulunmaktadır.

$$\begin{array}{l}
 [0 \ 1 \ 0; \\
 1 \ 4 \ 1; \\
 0 \ 1 \ 0]; \\
 \text{(a)}
 \end{array}
 \quad
 \begin{array}{l}
 [0 \ 0 \ 1 \ 0 \ 0; \\
 0 \ 1 \ 1 \ 1 \ 0; \\
 1 \ 1 \ 4 \ 1 \ 1; \\
 0 \ 1 \ 1 \ 1 \ 0; \\
 0 \ 0 \ 1 \ 0 \ 0]; \\
 \text{(b)}
 \end{array}
 \quad
 \begin{array}{l}
 [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0; \\
 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0; \\
 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0; \\
 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0; \\
 1 \ 1 \ 1 \ 1 \ 4 \ 1 \ 1 \ 1 \ 1; \\
 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0; \\
 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0; \\
 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0; \\
 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]; \\
 \text{(c)}
 \end{array}$$

Şekil 2.10. (a) 3×3 alçak geçiren filtre (b) 5×5 alçak geçiren filtre (c) 9×9 alçak geçiren filtre

Şekil 2.11’de alçak geçiren filtre ile filtrelenmiş görüntü örnekleri verilmiştir.



Şekil 2.11. (a) orijinal görüntü (b) 3×3 alçak geçiren filtre (c) 9×9 alçak geçiren filtre (d) 15×15 alçak geçiren filtre ile filtrelenmiş görüntü [26]

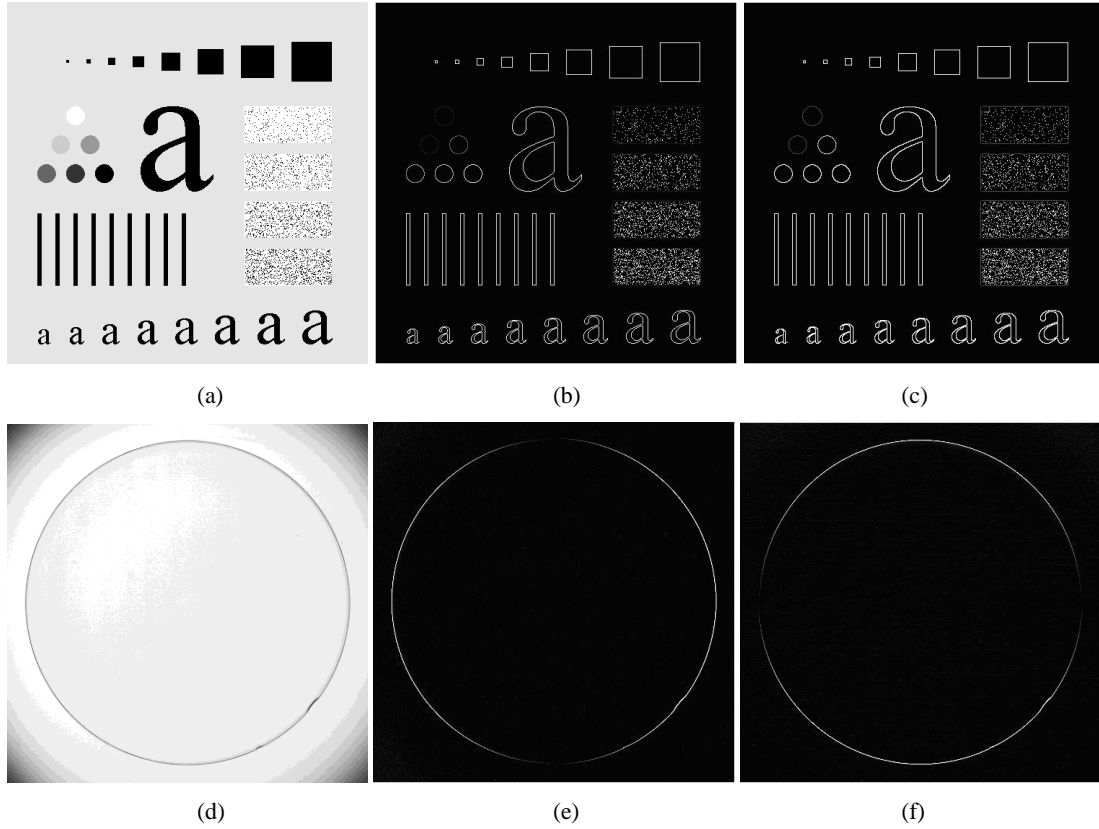
Alçak geçiren filtre, düşük frekansların geçmesine izin vermekte yüksek frekansları geçirmemektedir. Yani görüntü de ani olan renk değişimleri alçak geçiren filtre ile işlendiğinde yumuşamaktadır. Böylece görüntü içindeki küçük boyutlu gürültülerde ortadan kaldırılmış olur. Ancak görüntü içindeki kenarlar, filtrenin boyutlarına ve uygulanma tekrar sayısına bağlı olarak giderek belirginliğini kaybeder. Görüntü işlemedeki amaca göre farklı filtreler kullanılmaktadır. Kenar bulma, görüntüyü keskinleştirme, yumuşatma, gürültü giderme gibi işlemlerde kullanılan filtreler birbirinden farklıdır (Kenarların belirginleşmesi/bulunması için yüksek geçiren filtreler veya küçük gürültülerin görüntüyü yumuşatmadan yok edilmesi için medyan

filtrelerinin kullanılması gibi). Yüksek geçiren (high pass) filtre örnekleri Şekil 2.12’de verilmiştir.

$$\begin{array}{ccccc}
 [0 & 1 & 0; & [1 & 1 & 1; & [1 & 1 & 1; & [1 & 2 & 1; & [1 & 0 & -1; \\
 1 & -4 & 1; & 1 & -8 & 1; & 0 & 0 & 0; & 0 & 0 & 0; & 2 & 0 & -2; \\
 0 & 1 & 1]; & 1 & 1 & 1]; & -1 & -1 & -1]; & -1 & -2 & -1]; & 1 & 0 & -1];
 \end{array}$$

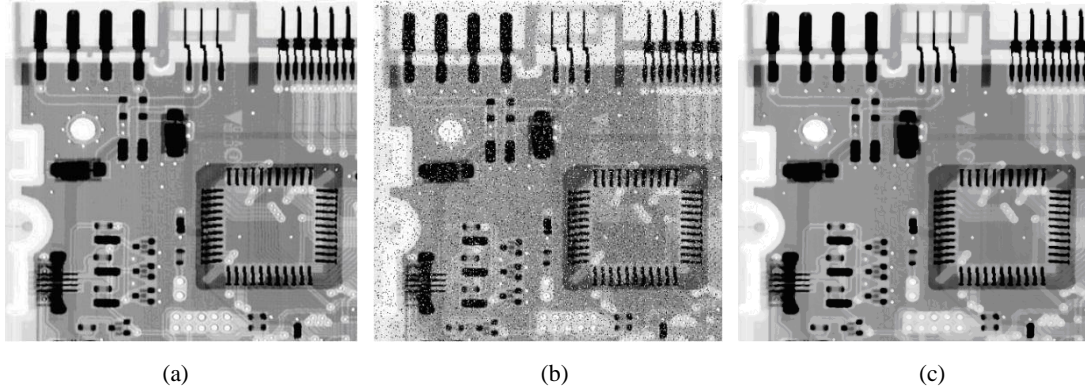
Şekil 2.12. Yüksek geçiren filtre örnekleri

Şekil 2.12’de verilen filtreler görüntü içinde genellikle kenarların belirlenmesi amacıyla kullanılan filtrelerdir. Filtre elemanlarının değerleri tüm kenarların, yatay kenarların ya da dikey kenarların bulunması amacıyla farklı oluşturulmuştur. Filtre elemanlarının toplamı sıfırdır. Toplamları sıfır olduğundan konvolüsyonun ağırlıklı ortalaması değil kendisi yeni piksel değeri olarak kaydedilir. Bu filtreler ile elde edilen görüntü örnekleri Şekil 2.13’te verilmiştir.



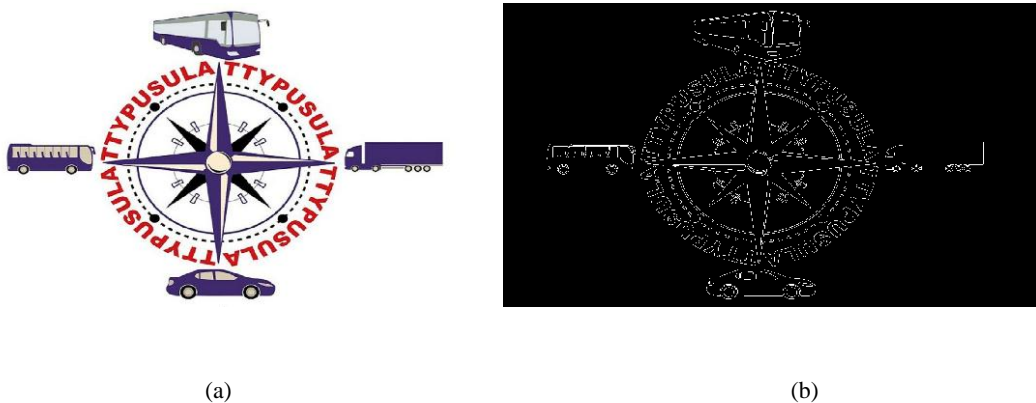
Şekil 2.13. (a) orijinal görüntü (b), (c) filtrelenmiş görüntü (d) orijinal görüntü (e), (f) filtrelenmiş görüntü [26]

Medyan filtrelerde, filtre elemanlarına değer verilmemiştir. Filtre görüntü üzerinde kaydırılırken, filtrenin belirlediği bölgedeki piksel değerleri küçükten büyüğe doğru sıralanır. Sıralama işleminden sonra ortanca büyüklüğe sahip değer pikselin yeni değeri olarak atanır. Şekil 2.14’da orijinal, siyah ya da beyaz görüntü verecek şekilde piksel değerleri değiştirilerek oluşturulmuş, tuz biber gürültülü ve tuz biber gürültüsü medyan filtre ile temizlenmiş görüntü örnekleri verilmiştir. Medyan filtre, filtrelenmek istenen görüntüde tuz biber gürültüsü benzeri gürültüler bulunan görüntülerin filtrelenmesi amacıyla kullanılır.



Şekil 2.14. (a) Orijinal görüntü (b) tuz biber gürültülü görüntü (c) medyan filtresi ile tuz biber gürültüsü temizlenmiş görüntü [26]

Kenar bulma: Görüntü içinde nesnelere ait kenarların bulunduğu bölgelerde renk değerlerinde büyük değişimler meydana gelmektedir. Nesne kenarlarının bu özelliğinden faydalanılarak türev tabanlı kenar bulma algoritmaları geliştirilmiştir. Görüntülerin satırlardaki türevleri alınarak dikey kenarlar, sütun türevleri alınarak yatay kenarlar bulunabilmektedir. Görüntünün satır ve sütun türevleri birlikte ele alındığında ise hem yatay hem dikey kenarlar bulunur. En yaygın kullanılan kenar bulma algoritmaları ise Sobel (dikey sobel $[-1, 0, 1; -2, 0, 2; -1, 0, 1]$, yatay sobel $[1, 2, 1; 0, 0, 0; -1, -2, -1]$), Prewit (dikey prewit $[-1, 0, 1; -1, 0, 1; -1, 0, 1]$, yatay prewit $[1, 1, 1; 0, 0, 0; -1, -1, -1]$) ve Canny kenar bulma algoritmalarıdır. Şekil 2.15’de Canny kenar bulma algoritması kullanılarak elde edilmiş görüntü örneği verilmiştir.



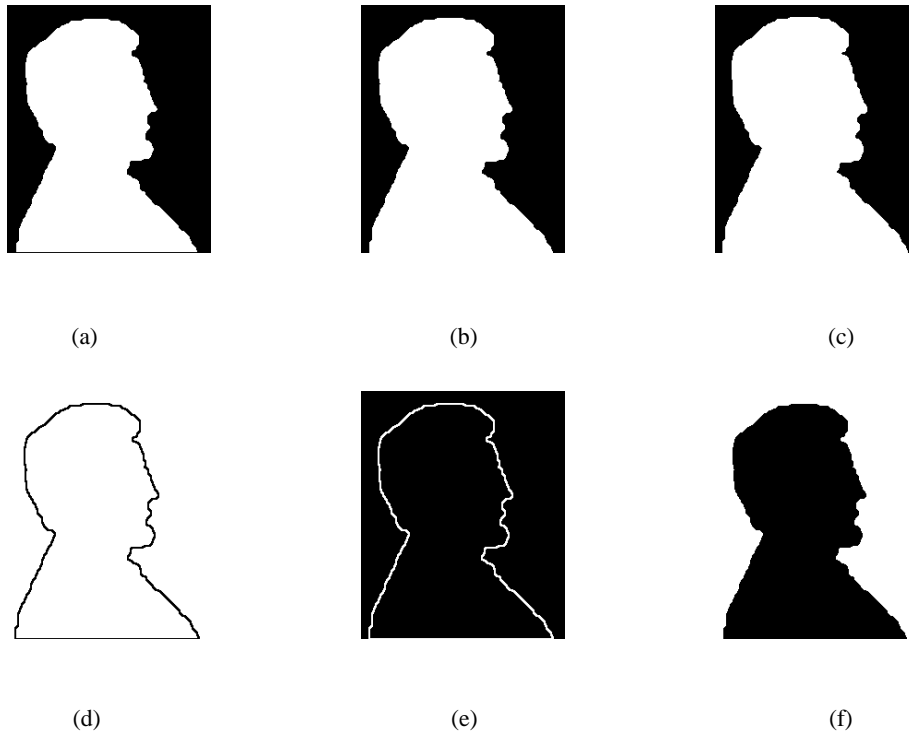
Şekil 2.15. (a) Orijinal görüntü (b) kenar alınmış görüntü

Morfolojik işlemler: İkili morfolojik operatörler binary görüntülere uygulanır ve gürültülerin temizlenmesi ya da küçük nesnelere yok edilmesi, eksik noktaların

tamamlanması gibi işlemlerde kullanılırlar [27]. Morfolojik işlemlerde ikili yapısal elemanlar kullanılır. Bu yapısal elemanlar görüntü üzerinde yapılmak istenen uygulamaya uygun olarak seçilir veya geliştirilir. Bununla birlikte çok sıklıkla kullanılan temel yapısal elemanlar da bulunmaktadır. Bu yapısal elemanlar aşağıda verilmiştir.

1. Genişleme
2. Aşınma

Yapısal elemanlar tek tek veya amaca göre birlikte kullanılarak görüntüye uygulanabilmektedir. Örneğin genişleme ve aşınma yapısal elemanlarının farklı kombinasyonları ile kapama ve açma morfolojik işlemleri gerçekleştirilebilmektedir. Morfolojik işlemler ile elde edilmiş görüntü örnekleri Şekil 2.16'da verilmiştir.



Şekil 2.16. (a) Orijinal görüntü (b) genişletilmiş görüntü (c) daraltılmış görüntü (d) küme kesişimi ile elde edilen kenar bulunmuş görüntü (e) küme farkı ile elde edilen kenar bulunmuş görüntü (f) küme eşleniği ile elde edilen görüntü [26]

2.4. İnsan Bilgisayar Etkileşimine ve Görüntü İçinde Yüz/Göz Bölgesine Yönelik Çalışmalar

İnsan bilgisayar etkileşimine yönelik birçok çalışma bulunmaktadır. Bu çalışmalar görsel [32,33,34], akustik [35], zihinsel ve dokunsal [36,37] veri aktarımı temelli olabilmektedir. Bilgisayarlar ve elektronik cihazlara görsel veri aktarımı kameralar kullanılarak, görüntü içindeki değişikliklerin takip edilmesi ile gerçekleştirilmektedir. Örneğin bir otoyol görüntüsünü alan bir kameranın aldığı görüntüler kullanılarak otoyoldan geçen araç bilgileri (sayısı, plakası) belirlenebilir [38], böylece araçların hangi tarihte, saatte otoyolu kullandığı ya da otoyolun hangi yoğunlukta kullanıldığı belirlenebilir. Ancak bu örnekte kullanılan görüntülerde bulunan nesnelere herhangi bir elektronik cihaza kontrol amacıyla hareket etmemektedirler. Oysa insan bilgisayar etkileşimine yönelik çalışmalarda görüntü alınan bölgedeki değişiklikler kontrol amacı taşımaktadır [39,40]. Örneğin TV karşısında kolunu hareket ettirmekte olan bir insan TV’de bulunan kameraya kol hareketini algılatıp kanal değiştirmek ya da sesi azaltmak-arttırmak istemektedir. Kamera ile alınan bu görüntü işlenir ve kullanıcının isteği doğrultusunda kontrol işlemi hatasız olarak gerçekleştirilmeye çalışılır.

Bakışlar ile kontrol ise el kol ve kafa hareketlerine göre daha dar bir alanda gerçekleşmesi sebebi ile tespiti zor, ancak etkin bir kontrol yöntemidir. Bu sebeple göz hareketlerinin takip edilmesi ve elde edilen verilerin kullanılmasına yönelik çalışmalar bulunmaktadır [5,6,7,8,9]. Bu çalışmalar Eye Gaze sistemler olarak isimlendirilmektedir. Eye Gaze sistemlerde göz hareketlerinin takibi amacıyla göz takipçileri (Eye-tracker) kullanılır. Göz takipçilerinin, göz hareketlerini izleyebilmesi için kullanıcı ile dokunsal olarak irtibat halinde olması ya da göz bölgesine kızılötesi ışık göndermesi gerekmektedir. Eye Gaze sistemlerin ek donanıma ihtiyaç duyuyor olmaları aslında bir dezavantajdır. Çünkü kullanımı zorlaştırır ve maliyeti artırır. Bu sebeple de yaygın olarak kullanılamamakta, genelde bu alanda araştırmalarda bulunan bazı üniversite laboratuvarlarında bulunmakta ve deneysel amaçlı kullanılmaktadır. Oysa Eye-tracker kullanmadan da bakış noktaları, görüntü işleme yöntemi kullanılarak bulunabilir [34,41]. Bu da Eye Gaze sistemlerin kullanımını kolaylaştırır ve maliyetlerinin azalmasına yardımcı olur. Böylece Eye Gaze

sistemlerin kullanımı da yaygınlaşır. Bunun için uygun görüntü işleme ve göz bölgesine ait öznelik çıkarma yöntemlerinin belirlenmesi gerekmektedir.

Görüntü içinde insan yüzü ile ilgili çalışmalar; çalışma şekli, yöntemi ve amacı gibi noktalarda birbirinden farklılıklar göstermektedir. Örneğin bazı çalışmalar göz hastalıklarına yönelik iken, bazıları biyometrik tanıma sistemlerine yönelik veya göz hareketlerinin takip edilmesine yönelik olabilmektedir. Bunlardan bazıları aşağıda verilmiştir.

İşler [42] çalışmasında; kişilerin ön cepheden webcam ile çekilen resimleri içinde göz bölgelerinin bulunmasını amaçlamıştır. Bu işlem için yapay sinir ağı kullanmış ve bu yöntemi farklı yöntemler ile karşılaştırmıştır. Yapay sinir ağları ile geliştirilen yöntemin farklı günlerde ve farklı aydınlanma koşullarında çekilmiş resimler üzerinde daha başarılı sonuçlar ürettiğini tespit etmiştir [42]. Türkan ise “Edge projections for eye localization” isimli çalışmasında göz bölgesini dalgalık dönüşümü ve bir destek vektör makinesi (Support Vector Machine-SVM) tabanlı sınıflandırıcı kullanarak tespit etmiştir [43].

Göz bölgesi içinde irisin bulunmasına yönelik incelenen çalışmalardan ilki ise Kuo'nun çalışmasıdır. Kuo [21] çalışmasında, göz bölgesi içinde iris bölgesini belirlemek amacıyla Temel Bileşenler Analizini (Principal Components Analysis - PCA) kullanmıştır. Bununla birlikte irisin bulunmasına yönelik çalışmalarda en çok kullanılan yöntemlerden biri CHT'dir. CHT özellikle gürültüye karşı dayanıklılığı sebebi ile tercih edilmektedir [14]. Greco [17] çalışmasında; kişiye özel iris özelliklerini belirlemek amacı ile görüntü içinde irisi İntegro-Diferansiyel Operatörünün Ayrık Yaklaşımını (A discrete approximation of an integrodifferential Operator-IDO) kullanarak bulmuştur [17]. Benzer şekilde Husam [16] irisi IDO'yu kullanarak bulmuştur. Husam, IDO'nun Hough Transformunun gelişmiş bir varyantı olarak kabul edilebileceğini belirtmiştir [16].

Göz bölgesi içinde yürütülen diğer çalışmalarda ise, göz köşe bölgelerinin bulunması amaçlanmıştır. Göz köşelerinin bulunması için kullanılan yöntemlerden biri Xia tarafından önerilen [44] WVPF (Weighted Variance Projection Function-Ağırlıklı

varyans bağdaştırma fonksiyonu)'dir. Bu yöntemde, önce görüntünün içinde göz köşesinin bulunduğu bölge belirlenir ve bu bölge içindeki piksel değerlerinin satır ve sütun olarak varyansları tespit edilir, daha sonra bu değerler Weight Factor (ağırlık faktörü) ile çarpılarak ağırlıklı varyans değerleri tespit edilir. Ağırlıklı varyans değerlerinin satır ve sütunlarda maksimum değerleri bulunur. Bu değerler bize görüntü içindeki göz köşesinin koordinatlarını verir [44].

Göz köşelerinin bulunmasına yönelik bir başka çalışma ise alt ve üst göz kapaklarının eğrilerinin tespit edilmesi esasına dayanmaktadır. Bu yöntem ile göz kapaklarının eğrilerinin kesiştiği noktaların göz köşelerini vereceği düşünülmektedir [21, 45].

Bazı çalışmalarda, göz köşeleri, göz bölgelerinin tespiti amacı ile de kullanılabilir. Bu tür çalışmalarda, gri seviyeli görüntüler üzerinde ten süzgeci ile yüz bölgesi bulunduktan sonra lineer filtreler yardımıyla göz köşeleri belirlenir ve göz bölgesi bulunur. Sirohey [46] çalışmasında, bu yöntemler ile yapılan göz tespitinde karşılaşılan problemleri aşabilmek için non-lineer filtreler ile daha iyi sonuçlara ulaşılabileceğini ön görmektedir. Çalışmada renkli görüntüler üzerinde göz köşelerinin ve buna bağlı olarak göz tespitinin yapılabilmesi için lineer olmayan bir filtre tasarımı yapılmıştır.

Batista'nın [20] çalışmasında; göz köşelerinin bulunması amacıyla aşağıda verilen algoritma önerilmiştir.

Gözler kapalıken;

- Renkli görüntüden gri seviyeli görüntü elde edilir ve kontrastı gerilerek artırılır;
- Düşey gradyan maskesi uygulanarak yatay kenarlar keskinleştirilir ve eşiklenerek siyah beyaz resme çevrilir;
- Pruning operatörü kullanılarak göz bölgesinin iskeleti elde edilir;
- İskeletin son noktaları seçilerek göz köşeleri belirlenir [20].

Gözler açıkken;

- Renkli görüntüden gri seviyeli görüntü elde edilir ve kontrastı gerilerek artırılır;

- Göz bölgesinde yatay VPF değerlerinin sınırladığı bölgeler belirlenir. Bu bölgeler açık renkli sclera bölgesini ve koyu renkli göz kapağı bölgesini içerir.
- VPF değerleri eşiklenerek göz köşe bölgelerinin yeri tahmin edilir.
- Pruning operatörü kullanılarak göz bölgesinin iskeleti elde edilir;
- İskelet polinom fonksiyonunun son noktaları göz köşelerini bulmak için kullanılır.

Her iki yaklaşım da elde edilen bilgiler birleştirilir [20].

Ayrıca literatürde dâhili webcam ile alınan görüntünün işlenmesi ve imleç kontrolü yapılmasına yönelik çalışmalarla da karşılaşmaktadır. Bunlar genellikle Head Control-Head Tracking gibi isimlerle anılırlar. Çünkü bu çalışmalarda imleç kontrolü baş hareketleri takip edilerek yapılır. Bununla birlikte literatürde webcam görüntüleri kullanılarak, bakışlar ile imleç kontrolü yapan bir çalışma da bulunmaktadır. Avrupa komisyonu, Gatsby vakfı ve Samsung firmasının desteklediği Opengazer isimli projede bakışlarla imleç kontrolü gerçekleştirilmektedir [41]. Bu çalışmada koordinat tespiti üç aşamada yapılır. İlki olan Feature Point Selection (Özellik noktası seçimi); aşamasında kullanıcı fareyi kullanarak yüz bölgesinde özellik noktalarını seçmektedir. İkinci aşama olan Calibrating the System (Sistemin kalibre edilmesi); aşamasında kullanıcı ekranda beliren kalibrasyon noktalarına bakarak sistemin kalibrasyonunu yapar. Son aşama olan Tracking (Takip); aşamasında ise özellik noktaları takip edilir ve bakılan nokta koordinatı belirlenir [41].

BÖLÜM 3. YAPAY SINİR AĞLARI

Yapay sinir ağı (YSA) insan beynindeki sinir hücreleri temel alınarak geliştirilmektedir. İnsan beynindeki biyolojik sinir hücreleri çok karmaşık bir yapıya sahiptir. Yapay sinir ağı oluşturulurken hem hücrelerin yapısı hem de çalışması modellenmeye çalışılmaktadır. Günümüzde araştırmacıların tek bir biyolojik sinir hücresinin yapısı noktasında hem fikir olmasına rağmen bu hücrelerin birbirleri ile olan yapısal bağları ve çalışma prensipleri araştırmacılar için henüz karmaşıklığını devam ettirmektedir. Bu sebeple araştırmacılar birçok farklı ağ modeli üzerinde çalışmaya devam etmektedir.

3.1. Yapay Sinir Ağlarının Genel Özellikleri

Doğrusal olmama (Nonlinearity): Doğrusal ya da doğrusal olmayan yapay sinir ağı oluşturulabilmektedir [47]. Böylece yapay sinir ağı ile karmaşık problemlere de çözüm bulunabilmektedir.

Uyarlanabilirlik (Adaptivity): Yapay sinir ağı eğitim esnasında ağırlıkların güncellenmesi ve en uygun ağırlıkların bulunması ile probleme uyarlanabilme yeteneğine sahiptir. Ayrıca eğitilmiş bir ağ, çözümü için geliştirildiği problemdeki küçük değişiklikler için tekrar eğitilebilir hatta gerçek zamanlı kullanımda öğrenmeye devam ederek ağırlıklar güncellenecek şekilde tasarlanabilir [47].

Kanıtı dayalı cevap (Evidential response): Yapay sinir ağı, girişin hangi sınıflandırmaya dâhil edileceği hakkında bilgi verdiği gibi sınıflandırmanın doğruluğunu belirleyecek şekilde de tasarlanabilir [47,48].

Genelleme (Generalization): Yapay sinir ađları eđitimi esnasında ađlırlıklar g¼ncellenir. Eđitim tamamlandıđında problem ađ tarafından ¼đrenilir ve ađlırlıklar son halini alır. B¼ylece ađ eđitim esnasında karřılařmadıđı ¼rnekler i¼in de dođru sonu¼ ¼retebilir [48].

Hata Toleransı (Fault Tolerance): Yapay sinir ađları, giriř verilerinde meydana gelebilecek bozulma, g¼r¼lt¼, eksilme gibi durumlarda sonu¼ ¼retebilmektedir. ¼¼nk¼ bilgi t¼m yapay sinir ađına dađıtılmıřtır [48].

Paralel İřlem Yeteneđi (parallel distributed processing): Yapay sinir ađları yapıları itibariyle bir¼ok iřlemi aynı anda yapabilmektedir [49].

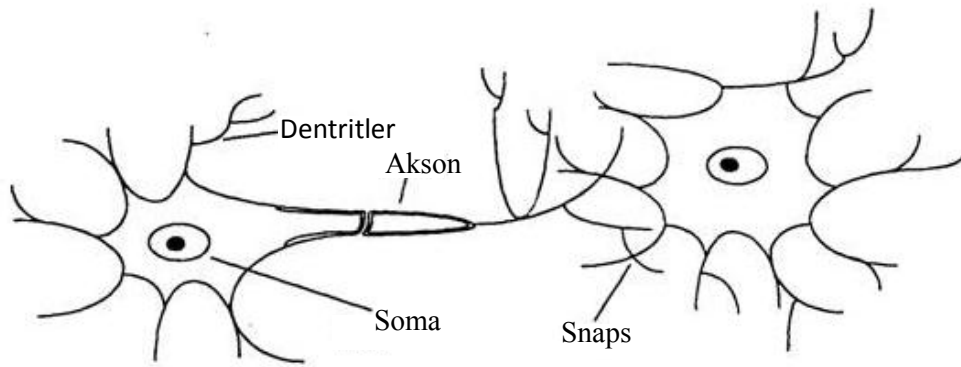
3.2. Yapay Sinir Ađlarının Uygulama Alanları

Yapay sinir ađları g¼n¼m¼zde bir¼ok uygulama alanı bulmakla birlikte bazıları ařađıdaki gibi sıralanabilir;

1. Sinyal iřleme (Signal Processing)
2. ¼r¼nt¼ Tanıma (Pattern Recognition)
3. Konuřma Tanıma (Speech Recognition)
4. Kontrol (Control) [50]
5. Sınıflandırma (Classification)
6. En İyileme (Optimization)
7. Yaklařık Fonksiyon Belirleme (Function Approximation)
8. Veri Sıkıřtırma (Data Compression)
9. Zaman serisi analizleri (Time Series Analysis) [51]
10. ¼bekleme (Clustering)
11. Vekt¼r Nicelendirme (Vector Quantization)
12. Tahmin (Forecasting) [52]

3.3. Biyolojik Sinir Hücresi

Biyolojik sinir hücreleri soma, dentrit, akson ve snaps olarak isimlendirilen yapısal elemanlardan oluşmaktadır. Bu yapısal elemanlardan soma hücre gövdesine verilen isimdir. Soma, dentritler üzerinden diğer sinir hücreleri ya da dış dünyadan bilgi almaktadır. Alınan bilgiler hücre gövdesinde birleştirilerek çıkış sinyali üretilip üretilmeyeceğine karar verilmekte ve çıkış sinyali aksonlar yardımıyla diğer sinir hücrelerine veya terminal organlara aktarılmaktadır [53]. Snapslar, sinir hücreleri arasında bilgi alış verişini gerçekleştiren boşluklardır [54]. Aksonlar ve dentritler arasında yer alırlar. Snapsa gelen elektriksel darbeler işlenerek dentrite aktarılır. Biyolojik sinir hücresinin yapısı Şekil 3.1’de verilmiştir.

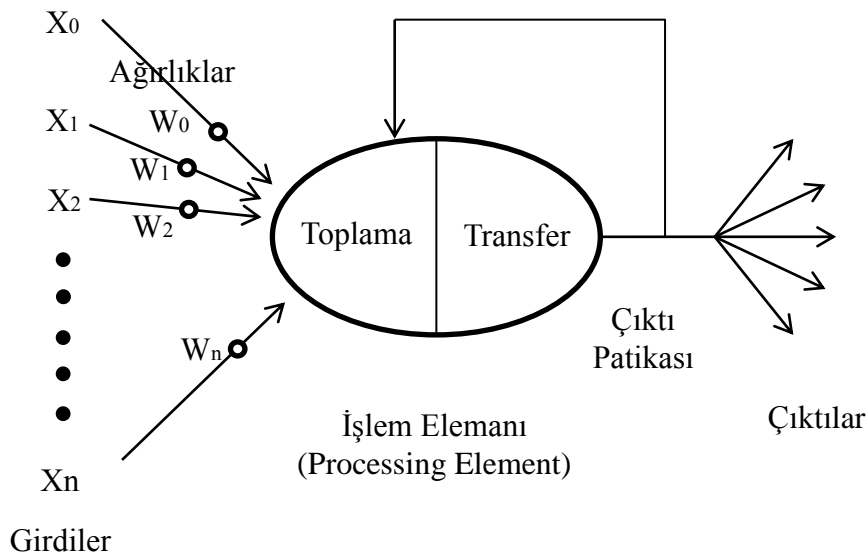


Şekil 3.1. Biyolojik sinir hücresinin yapısı [53]

3.4. Yapay Sinir Hücresi

McCulloch ve Pitts (1943) biyolojik sinir hücresinin matematiksel modelini çıkarmışlardır [55]. Yapay sinir ağları ile ilgili ilk önemli çalışmalar ise Hebb (1949) tarafından yapılmıştır. Hebb, biyolojik sinir hücrelerinin birbirleri ile iletişimlerini inceleyerek Hebb yapay sinir ağı teorisini geliştirmiştir.

Şekil 3.2’de bir yapay sinir ağının şematik yapısı görülmektedir. Bu şemada $x_0, x_1, x_2, \dots, x_n$ hücreye gelen giriş verilerini ifade etmektedir. $w_0, w_1, w_2, \dots, w_n$ giriş verilerinin ağırlıklarını ifade etmektedir. Her bir giriş verisi hücreye ulaşmadan önce farklı ağırlıklar ile işlenmektedir. Ağırlıklar ile işlenen giriş verileri hücreye alındığında önce toplanır daha sonra ise transfer fonksiyonundan geçirilerek çıkışa aktarılır. Bu çıkışlar ağ modeline bağlı olarak çıkış olarak kullanılabilir gibi bir sonraki katmanda bulunan hücrelerde giriş verisi olarak kullanılabilir.



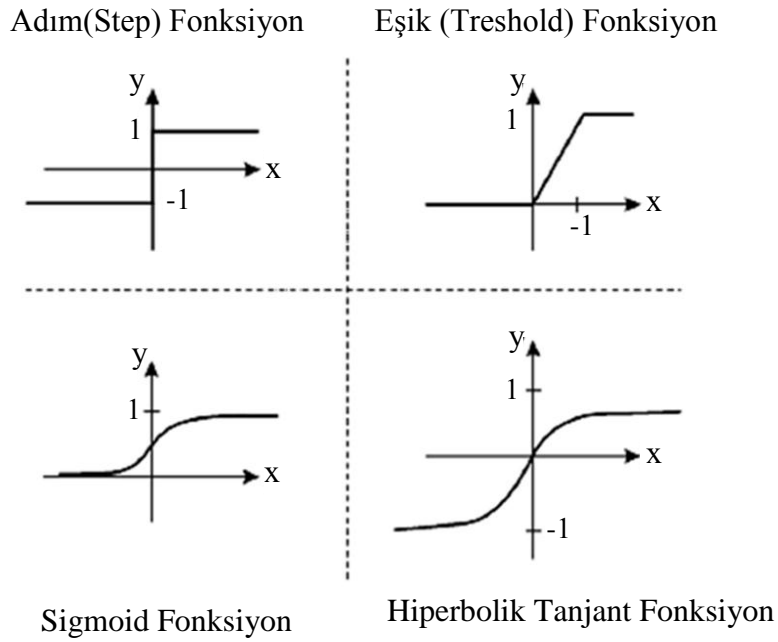
Şekil 3.2. Yapay sinir hücresinin şematik yapısı [55,56,57]

Basit bir yapay sinir hücresi için çıkış ifadesi aşağıdaki gibi yazılabilir.

$$y = f\left(\sum_{i=1}^n w_i \times x_i - \theta\right) \quad (3.1)$$

Burada w_i her bir girişe ait ağırlık katsayısını, x_i sinir hücresine gelen giriş verilerini, θ sinir hücresine ait eşik seviyesini, f sinir hücresine ait transfer fonksiyonunu, y ise çıkış verisini ifade etmektedir. Sinir hücrelerinde çok farklı transfer fonksiyonları kullanılabilir. Transfer fonksiyonu olarak genellikle

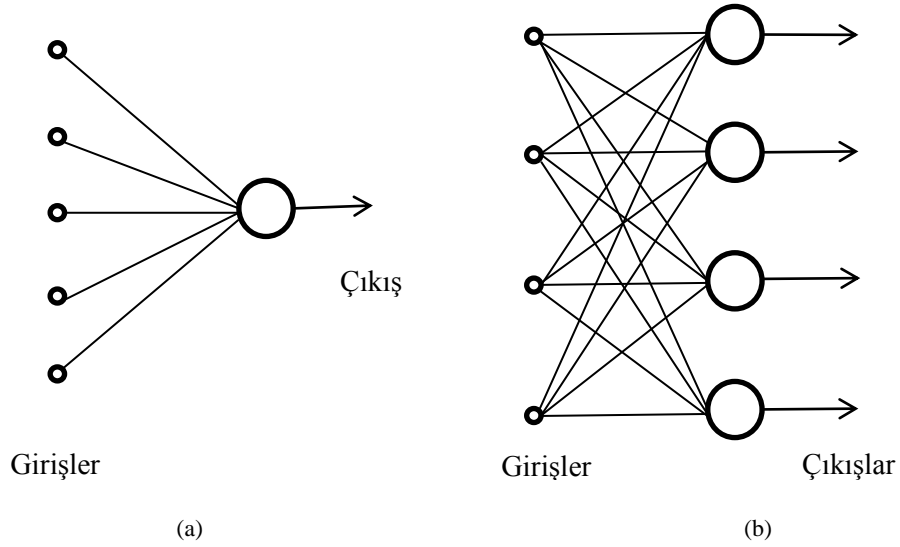
doğrusal olmayan fonksiyonlar kullanılır. Çünkü doğrusal fonksiyonlarda çıkış verisi giriş verisi ile orantılı olacaktır [55]. Bu da yapay sinir ağlarının başarısız olmasına sebep olmaktadır [58]. Şekil 3.3'te en çok kullanılan transfer fonksiyonları verilmiştir.



Şekil 3.3. Sıklıkla kullanılan transfer fonksiyonları [55,56]

3.5. Yapay Sinir Ağları

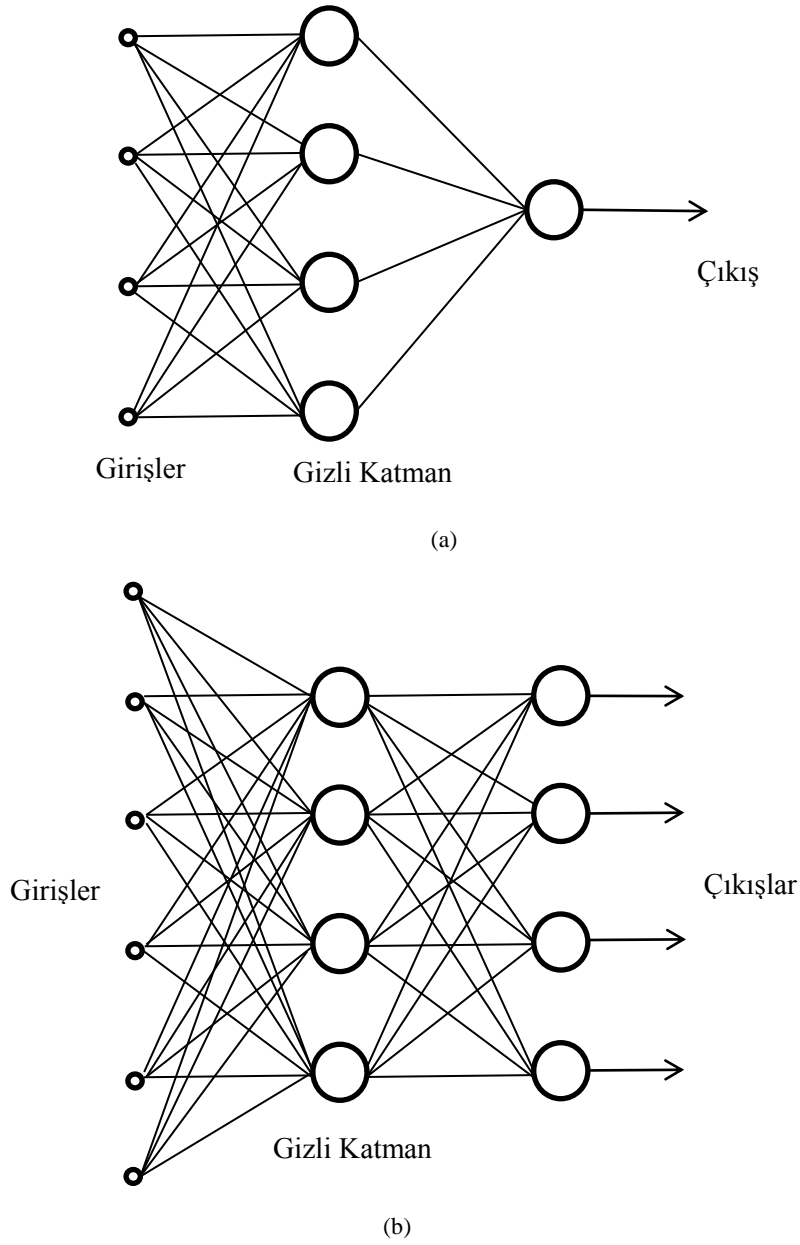
Yapay sinir ağları, genellikle çok sayıda yapay sinir hücresinin bir arada kullanılması ile oluşturulmaktadır. En basit yapay sinir ağında 1 tane yapay sinir hücresi kullanılır. Girişler ağırlıklandırılarak bu hücrede toplanır ve transfer fonksiyonu ile işlendikten sonra hücrenin çıkışı yani ağın çıkışı elde edilir. Bu tip ağ mimarilerinde sadece girişler ve çıkış katmanı vardır. Bunlara tek katmanlı yapay sinir ağları denir. Tek katmanlı yapay sinir ağı örnekleri Şekil 3.4'te verilmiştir. Bu ağlarda çıkış nöron sayısı elde edilecek çıkış verisi sayısına göre belirlenir.



Şekil 3.4. (a) Tek katmanlı tek çıkışlı yapay sinir ağı (b) tek katmanlı çok çıkışlı yapay sinir ağı [59]

Ağlar çok katmanlı olarak ta gerçekleştirilebilmektedir. Bu durumda girişler ile çıkış katmanı arasında ara katmanlar ilave edilebilmektedir. Bu katmanlara gizli katman denilmektedir. Herhangi bir sınırlama olmamasına rağmen literatürde genellikle 1, en çok 2 gizli katman kullanılarak yapay sinir ağları oluşturulduğu görülmektedir. Toplamda 4 katmandan fazlası ağın performansını ters etkilemektedir [60]. Şekil 3.5'te çok katmanlı tek çıkışlı, çok katmanlı çok çıkışlı ağ modelleri verilmiştir. Yapay sinir ağlarının katman sayılarına göre sınıflandırması;

1. Tek katmanlı yapay sinir ağları,
 2. Çok katmanlı yapay sinir ağları,
- şeklinde yapılabilir.



Şekil 3.5. (a) Çok katmanlı tek çıkışlı yapay sinir ağı (b) çok katmanlı çok çıkışlı yapay sinir ağı [57]

Gizli katmandaki nöron sayısı çözüm aranan probleme, veri miktarına ve tasarımda kullanılan ağ modeline bağlı olmakla birlikte giriş veri sayısının %75'i olarak önerilmektedir [61]. Bir başka çalışmada ise gizli katmandaki nöron sayılarının, ağdan en iyi performansı alacak şekilde kullanıcı tarafından seçilmesi gerektiği belirtilmektedir [62].

Yapay sinir ağları, eğitimi esnasında her bir giriş için bulunan hatanın dönüt olarak kullanılıp kullanılmamasına göre;

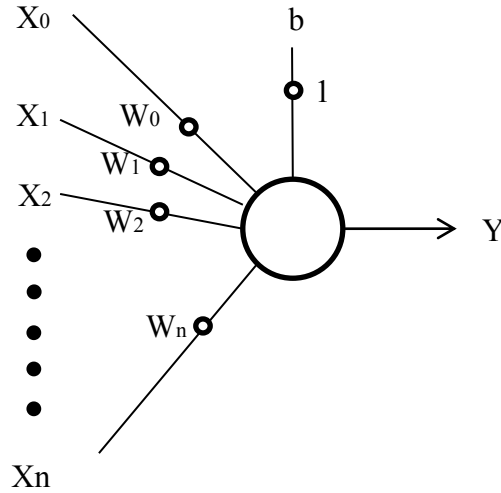
1. İleri beslemeli ağlar,
 2. İleri beslemeli geri yayımlı ağlar,
- şeklinde de sınıflandırılmaktadır.

Bu sınıflandırmalar genel olarak yapay sinir ağlarını basitçe sınıflandırma için kullanılsa da günümüzde yapay sinir ağlarının birçok modeli vardır ve bu modellerin kendi içinde de sınıflandırılmaları mümkündür.

3.6. Örnek Yapay Sinir Ağı Modelleri

3.6.1. Hebb net ağ modeli

En basit ve ilk yapay sinir ağı modelidir. Öğrenme yöntemi olarak Hebb kuralını kullanmaktadır. İsmi buradan alır. Hebb kuralı oldukça basittir. Eğer iki hücre aynı aktivasyonu gösteriyorsa bu hücreler arasındaki bağ kuvvetlendirilmeli, eğer farklı aktivasyon gösteriyorsa bağ zayıflatılmalı ya da olduğu gibi bırakılmalıdır [50,51,56,59]. Şekil 3.6'da Hebb net mimarisi görülmektedir.



Şekil 3.6. Hebb net ağ mimarisi [59]

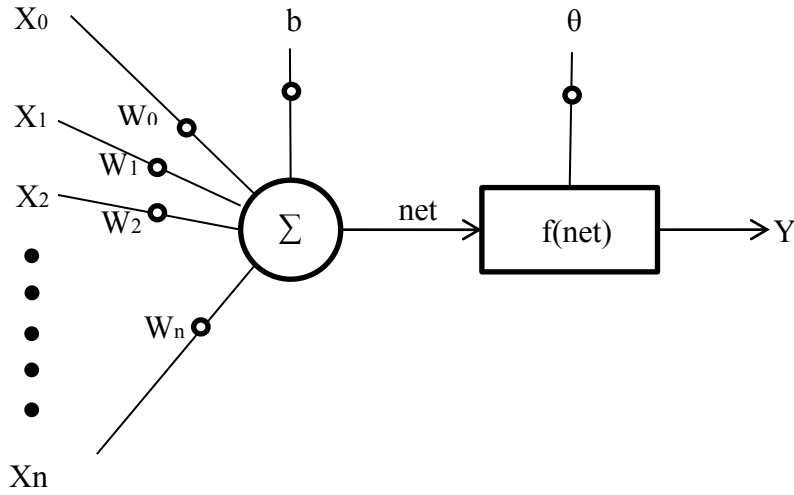
Hebb net ađ modelinde öğrenme süreci ađrılıkların (w_i) çıkıştaki hataya göre güncellenmesi ile gerçekleştirilir. Ađrılıkların güncellenmesi için kullanılan denklem 3.2’de verilmiştir.

$$w_i(\text{yeni}) = w_i(\text{eski}) + x_i \times y \quad (3.2)$$

Denklem 3.1’de ađrılıkların güncellenmesi $x_i \times y$ ifadesine bađlıdır. Bu ifade hata olarak kabul edilmiş ve hataya göre güncelleme yapılmıştır. x_i Sinir hücresinin i ’inci girişini ifade etmekle birlikte aynı zamanda sinir hücresine bađlı bir önceki sinir hücresinin çıkışını ifade etmektedir. y ’de işlem yapılan hücrenin çıkışı olduğundan her iki hücrenin aktifliđi/pasifliđi yani çıkışlarının pozitifliđi/negatifliđi durumunda aradaki bađ kuvvetlendirilmiş (artırılmış) olacaktır. Hata durumunda ise yani hücrelerden biri aktif diđeri pasif iken aradaki bađ zayıflatılmış (azaltılmış) olur.

3.6.2. Perceptron ađ modeli

Hebb net mimarisinin kullanıldığı Perceptron’da Hebb net’deki öğrenme kavramı yerine eşikleme fonksiyonlu öğrenme kullanılmaktadır [59]. Eşiklenmiş aktivasyon fonksiyonu [53] ile birlikte tekrarlı öğrenme prosedürü (iterasyonlu), Perceptron öğrenme kuralının Hebb öğrenme kuralına göre daha güçlü bir öğrenme kuralı olmasını sağlamıştır [50,59]. Şekil 3.7’de Perceptron mimarisi görülmektedir.



Şekil 3.7. Perceptron ağ mimarisi [63]

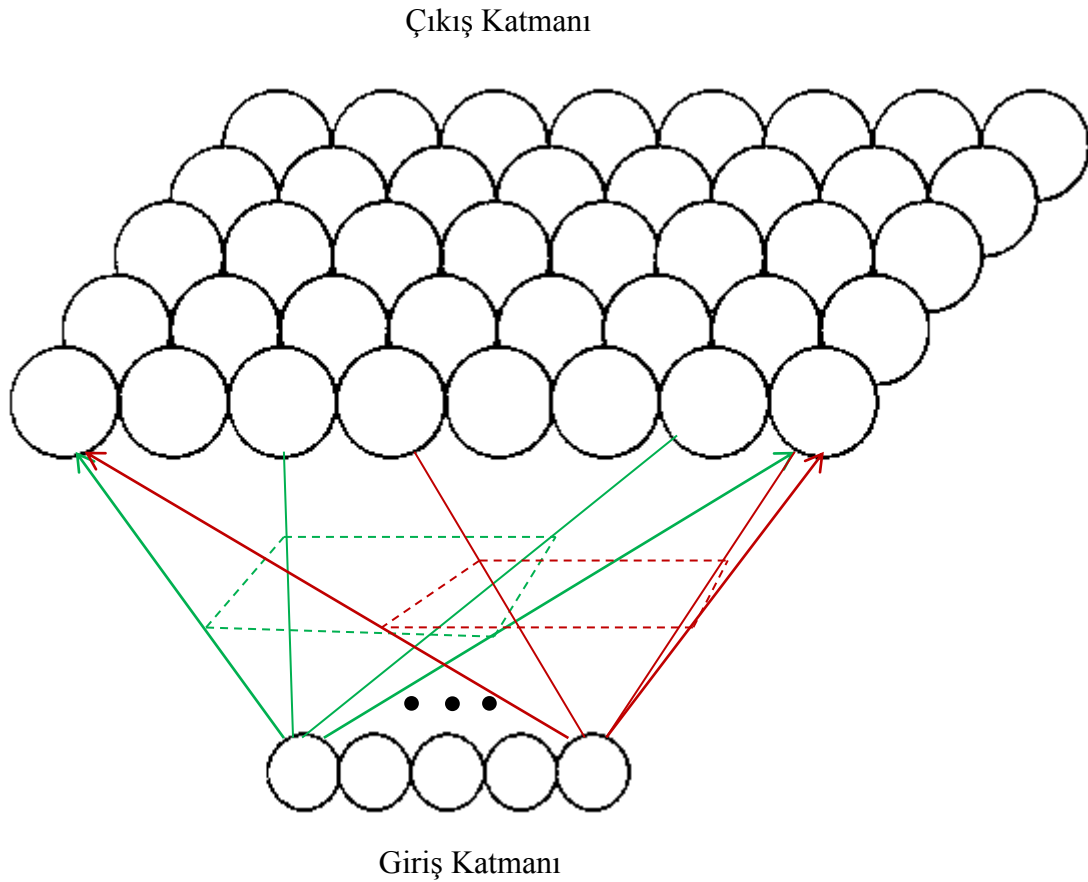
Perceptron ağ modelinde de öğrenme süreci, ağırlıkların (w_i) çıkıştaki hataya göre güncellenmesi ile gerçekleştirilir. Ağırlıkların güncellenmesi için kullanılan denklem 3.3'de verilmiştir.

$$w_i(\text{yeni}) = w_i(\text{eski}) + \eta \times t \times x_i \quad (3.3)$$

Denklem 3.2'ye öğrenme katsayısı η ilave edilmiştir. Ayrıca perceptron modelinde her giriş için çıkış hesaplanır ve olması gereken çıkış değeri ile karşılaştırılır eğer çıkış olması gereken değerden farklı ise ağırlıklar güncellenir değilse ağırlıklar güncellenmez. Yine bu modelde Hebb net'ten farklı olarak ağırlıklar değişmez oluncaya kadar işlem devam ettirilir [50].

3.6.3. SOM (self organizing map) ağ modeli

Genellikle sınıflandırma yapmak için kullanılan bu ağ modeli Kohonen [51] tarafından geliştirilmiştir. Kohonen yapay sinir ağları için farklı bir ağ mimarisi önermiş ve bu ağ mimarisi araştırmacılar tarafından büyük ilgi görmüştür [59]. SOM ağlarında çıktı katmanı 2 boyutludur ve ağ yapısında tek katman bulunur. Şekil 3.8'de SOM mimarisi görülmektedir.



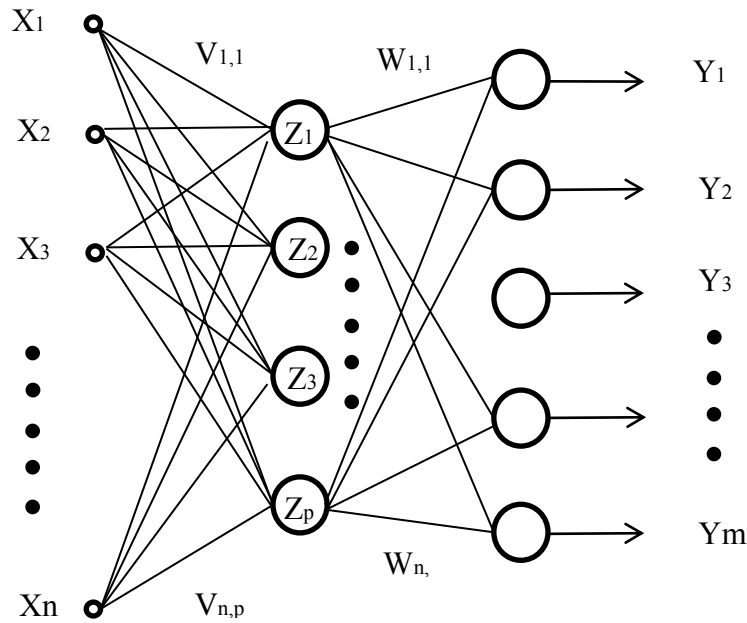
Şekil 3.8. SOM ağ mimarisi [64]

SOM ağlarında eğitim öğretmensiz öğrenme (unsupervised learning) şeklinde yapılmaktadır. Bu ağlarda eğitim esnasında ağa çıkış bilgileri verilmez. SOM ağı, m sayıdaki giriş verilerini çıkışta n sayıda grupta sınıflandırır. SOM ağı tarafından 2 boyutlu çıkış katmanında önce küme merkezleri oluşturulur ve giriş verileri aralarındaki ilişkiler ağ tarafından değerlendirilerek girişin hangi kümeye dâhil edileceği belirlenir. Ağın eğitimi esnasında küme merkezlerinin yeri ve küme sayısı değişebilir. Ayrıca kümeler tek bir hücreden oluşmak zorunda değildir. Küme merkezinin etrafındaki hücrelerin komşuluk dereceleri ve sınırı ağ tarafından belirlenir. Küme merkezlerinin ve komşulukların belirlenmesinde, ağın eğitiminde kullanılan girişler için hangi çıkışın aktif olacağını belirlemek etkilidir. Her bir giriş için çıkışta küme belirlenir ve bu küme içinde bir sinir hücresi aktif olur. Bu çıkış hücresinin SOM ağlarındaki kazanan hepsini alır [47,50,59] ilkesi ile ağırlıkları güncellenir ve küme merkezini kendine doğru çeker [59]. Bu sebeple SOM ağlarındaki öğrenme, rekabetçi öğrenme (competitive learning) olarak [47]

adlandırılır. Ağ eğitildiğinde küme içinde en çok kazanan hücre yeni küme merkezi olur.

3.6.4. Backpropagation net ağ modeli

Çok katmanlı ileri beslemeli ağlarda hatanın geri yayılımı amacıyla kullanılan özel bir ağ modelidir [47]. Birçok alanda problem çözümünde yaygın olarak kullanılmaktadır [50]. Backpropagation net ağ modeli, öğretmenli öğrenme (supervised learning) stratejisini kullanmaktadır [65]. Şekil 3.9'da backpropagation ağ mimari örneği verilmiştir.



Şekil 3.9. Bir gizli katmanlı backpropagation ağ mimarisi [59]

Backpropagation ağ yapısında, çıkış katmanında elde edilen y değerlerinde meydana gelen hata tespit edilir. Bu hata, giriş-gizli katman arasındaki ağırlıklar ($v_{n,p}$) ile gizli-çıkış katmanı arasındaki ağırlıklar ($w_{p,m}$) üzerine yansıtılarak tüm ağırlıklar aynı anda güncellenmektedir [59]. Backpropagation ağlarda aktivasyon fonksiyonunun sürekli ve türevi alınabilen bir fonksiyon olması gerekmektedir [60]. Genellikle aktivasyon fonksiyonu olarak sigmoid fonksiyonu, öğrenme fonksiyonu

olarakta delta öğrenme kuralı kullanılmaktadır [59]. Sigmoid fonksiyonu, türevi ve ağırlıkların güncellenmesi için kullanılan denkleler 3.4, 3.5 ve 3.6'da verilmiştir.

$$f_{sig}(x) = 2 / (1 + \exp(x)) \quad (3.4)$$

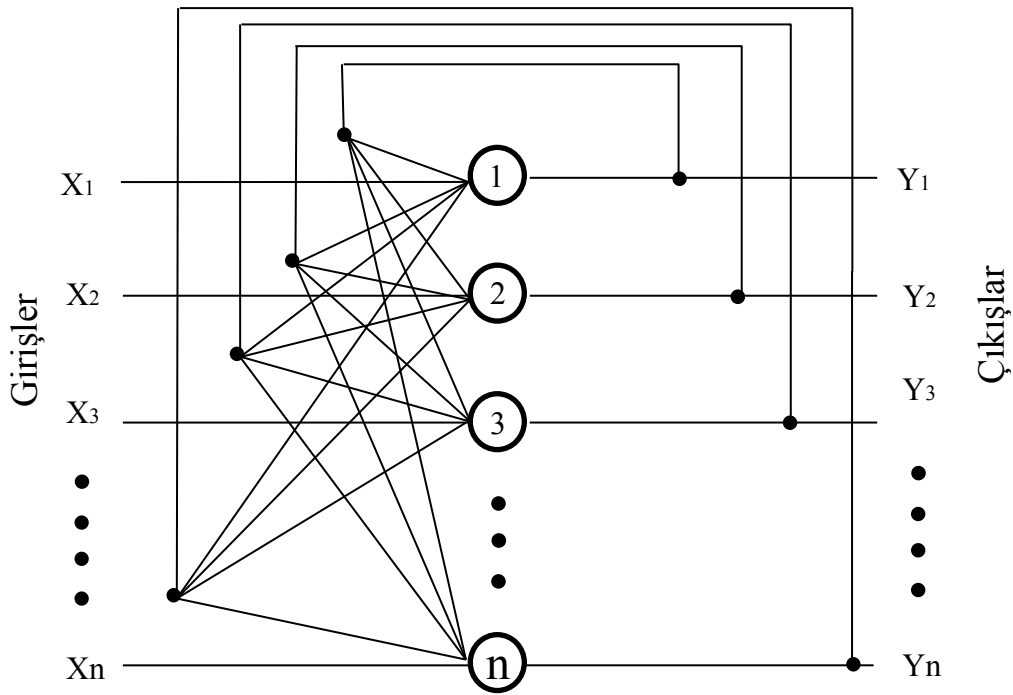
$$f'_{sig}(x) = f_{sig}(x) \times [1 - f_{sig}(x)] \quad (3.5)$$

$$w_i(\text{yeni}) = w_i(\text{eski}) + (\eta \times [t - f(y_{in})] \times f'(y_{in})) \quad [50] \quad (3.6)$$

Denklemleri verilen sigmoid fonksiyonu ikili (0,1) sigmoid fonksiyonudur. Bununla birlikte iki kutuplu (-1,1) sigmoid fonksiyonu da backpropagation net ağ modelinde [50] sıklıkla kullanılmaktadır.

3.6.5. Hopfield yapay sinir ağı modeli

Hopfield ağı tek katmanlı ve geri dönüşümlü (recurrent) bir ağ modelidir [51,66]. Hopfield ağ modeli tekrar beslemeli bir yapıya sahiptir. Bu sebeple diğer yapay sinir ağı modellerinden ayrılmaktadır. [59]. Hopfield katmanında bulunan hücrelerin tamamı hem giriş hem çıkış elemanıdır ve bu hücrelerin her biri birbirleri ile bağlıdır. Başlangıçta verilen giriş değerleri bir enerji fonksiyonu olarak saklanır [51]. Bu giriş değeri örüntüsü önceden öğrenilmiş bir başka giriş değeri örüntüsüne yavaş yavaş değiştirilerek benzetilmeye çalışılır [59]. Şekil 3.10'da Hopfield ağ mimarisi örneği verilmiştir.



Şekil 3.10. Hopfield yapay sinir ağı mimarisi [51]

3.7. Çalışmada Kullanılan Ağ Modeli

Çalışmada göz bölgesinin bulunması, iris bölgelerinin bulunması ve bakılan bölge koordinatlarının tespiti amacıyla 4 farklı YSA geliştirilmiştir. Çalışmada çok katmanlı ileri beslemeli geri yayımlı (feedforward Backpropagation) ağ modeli kullanılmıştır. Çok katmanlı ağlar mühendislik uygulamalarında ve özellikle doğrusal olmayan karmaşık problemlerin çözümünde yaygın olarak kullanılmaktadır. Çalışmada koordinat tespiti için bir düzleme yönelik sonuçlar elde edilmek istendiğinden SOM ağ modelide incelenmiş ancak 57×32 nörona sahip SOM yapay sinir ağına işlem yükünün çok artması sebebiyle sonuç alınamamış ve bu problemin çözümünde de çok katmanlı ileri beslemeli geri yayımlı ağ modeli tercih edilmiştir.

BÖLÜM 4. YÖNTEMİN GELİŞTİRİLMESİ

4.1. Kullanılan Materyaller

4.1.1. Çalışmada kullanılan bilgisayar/bilgisayarlar

Çalışmada kullanılan dizüstü bilgisayar HP pavillion, 15.6" ekran, 1366×768 ekran çözünürlüğü, 2.13 GHz çift çekirdekli intel P 7450 işlemci, 2 GB ram, 1.3 Mega Piksel dâhili web kamerasına sahiptir. Araştırmacının göz bölgesinin bulunması ve tek kullanıcı için ekranda bakılan bölgenin koordinatlarının tespiti amacıyla kullanılan görüntüler bu bilgisayarın dâhili web kamerası ile 640×480×3 boyutlarında yakalanarak çalışmada kullanılmıştır. Çalışmada kullanılan kullanıcı görüntüleri EK1'de verilmiştir.

Ayrıca çalışmada deneklerin göz bölgesinin bulunması ve deneklerin ekranda baktığı bölge koordinatlarının tespiti amacıyla kullanılan görüntülerin alınması amacıyla deneklere ait dizüstü bilgisayarlar kullanılmıştır. Bu bilgisayarların özellikleri Tablo 4.1'de verilmiştir;

Tablo 4.1. Denek bilgisayarlarının özellikleri

	Marka	Ram	İşlemci	İşlemci Hızı	Webcam	İşletim Sistemi	Ekran Boyutu	Ekran Çözünürlüğü
1.Denek Bilgisayarı	Lenova G 580	4 GB	Intel i5-3210	2,5 GHz	0,3 MP	Windows 8 x64	15,6"	1366×768
2.Denek Bilgisayarı	Casper Nirvana	8 GB	Intel i7-3630QM	2,4 GHz	1,3 MP	Windows 8 x64	15,6"	1366×768
3.Denek Bilgisayarı	Toshiba Satellite	8 GB	Intel i5-3210	2,5 GHz	1 MP	Windows 8,1 x64	15,6"	1366×768
4.Denek Bilgisayarı	Asus	8 GB	Intel i7-4700HQ	2,4/3,4 GHz	1,3 MP	Windows 8	15,6"	1366×768
5.Denek Bilgisayarı	Asus	8 GB	Intel i7-3610QM	2,3 GHz	1,3 MP	Windows 8 x64	15,6"	1366×768
6.Denek Bilgisayarı	Acer Aspira	4 GB	İntel i5-M430	2,27 Ghz	1,3 MP	Windows 7 x64	15,6"	1366×768
7.Denek Bilgisayarı	HP Compaq	4 GB	Intel B-950	2,10 GHz	0,3 MP	Windows 7 x64	15,6"	1366×768
8.Denek Bilgisayarı	Asus	4 GB	Intel Celeron	1,80 GHz	0,3 MP	Windows 8 x64	15,6"	1366×768
9.Denek Bilgisayarı	Acer Aspira	6 GB	Intel i5-2430M	2,4/3,0 GHz	1,3 MP	Windows 7 x64	15,6"	1366×768
10.Denek Bilgisayarı	Casper Nirvana	8 GB	Intel i5-3210M	2,50 GHz	1,3 MP	Windows 8 x64	15,6"	1366×768
11.Denek Bilgisayarı	Toshiba Satellite	8 GB	Intel i7-2670	2,2/3,1 GHz	0,3 MP	Windows 8 x64	15,6"	1366×768
12.Denek Bilgisayarı	Grundig	8 GB	Intel i7-3670QM	2,2 GHz	1,3 MP	Windows 7 x64	15,6"	1366×768

4.1.2. Göz bölgesinin bulunması amacıyla kullanılan eğitim ve test verileri

Çalışmanın ilk aşamasında geliştirilen YSA'nın eğitim ve test setinin oluşturulması için, bilgisayar kullanıcının bilgisayar ekranına bakarken, farklı ortam, ışık şiddeti ve bakış açılarındaki görüntüleri alınmıştır. Farklı ortam, ışık şiddeti ve bakış açılarındaki görüntülerin alınması; göz bölgesinin bulunması aşamasında eğitilecek YSA'nın daha

kararlı çalışmasını ve gerçek zamanlı çalışmaya elverişli olmasını sağlamak amacıyla öngörülmüştür. Böylece YSA'nın dış ortamdaki fiziki değişkenlerden en az seviyede etkilenmesi amaçlanmıştır. Bilgisayarın dâhili webcam'i kullanılarak elde edilen görüntülerden 107'si eğitim setinde, 20'si test setinde kullanılmıştır (EK1). Eğitim setindeki 107 görüntünün her birinden 48×160 boyutlarında 1 doğru (göz bölgesini temsil edebilecek), 5 yanlış (göz bölgesini temsil edemeyecek) olmak üzere toplam 642 adet görüntü alınmıştır. Elde edilen doğru/yanlış göz bölgeleri önce Matlab "imresize" komutu kullanılarak 6×20 boyutlarına daha sonra da "cat" komutu kullanılarak 1×120 boyutuna dönüştürülmüş daha sonra her bir görüntü için elde edilen vektörler birleştirilerek 642×120 boyutlarında eğitim seti oluşturulmuştur. YSA eğitim ve test aşamalarında kullanılan veriler EK2'de verilmiştir. Test seti için görüntüler alınmış ancak sabit bir test seti oluşturulmamıştır. Çünkü test aşamasında eğitimde olduğu gibi 1×120 boyutuna sahip veriler kullanılması gerekmektedir. Oysa test amacıyla kullanılacak görüntüden, birbirinden farklı birçok 1×120 boyutuna sahip veri elde edilebilir. Bu sebeple test, tarama usulü ile yapılmış, test veri seti de bu aşamada sadece araştırmacının göz bölgesinin bulunması amaçlandığından, araştırmacıya ait 20 adet görüntü olarak belirlenmiştir (EK2). YSA'nın test aşamasında kullanılan test görüntüleri ayrıca EK3'de verilmiştir.

Bu aşamada alınan görüntülerin ekrandan uzaklığı birbirinden farklı olmakla birlikte ortalama uzaklıkları 50 cm civarındadır.

4.1.3. Göz bölgesinin bulunması amacıyla kullanılan matlab yazılımı

Bilgisayarın dâhili web kamerası ile alınan kullanıcı görüntülerinin içinde göz bölgesinin bulunması amacıyla MATLAB R2010a® platformu kullanılmıştır. Araştırmacı tarafından Matlab neural network, image processing ve wavelet toolboxları ve fonksiyonları kullanılarak görüntü tarama ve YSA eğitimi Matlab ortamında gerçekleştirilmiştir.

YSA'nın giriş neuron sayısı her bir görüntünün 120 veri ile ifade edilmesi sebebiyle 120, çıkış neuron sayısı test edilen görüntünün göz bölgesi olup olmadığını ifade edebilmesi için 1 olarak belirlenmiştir.

Çalışmanın bu aşamasında kullanılan Matlab yazılımları EK4, EK5, EK6'da verilmiştir.

4.1.4. Koordinat tespiti için kullanılan eğitim ve test verileri

Bu aşamada öncelikle bilgisayar ekranında bakılan noktanın nasıl temsil edileceği kararlaştırılmıştır. Seçenekler piksel bazında çalışma ve belli boyutlara sahip bölge bazında çalışma olarak belirlenmiştir. Piksel bazında çalışıldığında bilgisayar ekranında çözünürlüğe bağlı olarak 1366×768 ayrı nokta olacak ve bu noktaları iyi temsil edebilmesi için örneklemede çok fazla sayıda nokta seçilmesi gerekecektir. Bölge bazında çalışıldığında ise bölge sayısının toplam piksel sayısından çok daha az olacağı ve bunları temsil için örnekleme grubunda daha az eleman bulunacağı aşikârdır. Ayrıca bilgisayar ekranındaki nesnelere de belli boyutlara sahip olması ve birkaç pikselden çok daha büyük olmaları sebebiyle bölge bazlı çalışmanın sonuçlar açısından yeterli olacağı öngörülmüştür. Böylece bilgisayar ekranı 24×24 piksel boyutlarına sahip 57×32 adet bölgeye ayrılmıştır. Bölge boyutları belirlenirken 15.6" bilgisayar ekranındaki Windows penceresinin simge durumuna küçültme (30×18), ekranı kaplama (26×18), kapatma (48×18) butonlarının boyutları dikkate alınmıştır. Oluşturulan bölgelerden random olarak seçilen 100 tanesi eğitim setini, benzer şekilde random olarak seçilen 20 tanesi de test setini hazırlamak için kullanılmıştır.

Kullanıcı belirlenen bölgelere bakarken 20'şer adet görüntü ($480 \times 640 \times 3$) almış, bu görüntüler gri resme çevrilerek ve ilk YSA ile işlenerek 48×160 boyutlarında göz bölgeleri belirlenmiştir. Elde edilen göz bölgeleri önce Matlab "imresize" komutu kullanılarak 6×20 boyutlarına daha sonra da "cat" komutu kullanılarak 1×120 boyutlarına dönüştürülmüştür. Görüntü içinde göz bölgesinin yerinin ekrana bakış açısı bilgisini içereceği ve bunun da elde edilecek sonuçları etkileyeceği düşünülerek göz bölgesinin başlangıç koordinatları da bu veriye ilave edilmiş ve her bir görüntü için 1×122 boyutunda veri elde edilmiştir. Daha sonra her bir görüntü verisi birleştirilerek 2000×122 boyutlarında eğitim, benzer şekilde 400×122 boyutlarında test veri seti oluşturulmuştur (EK2). Koordinat tespitinin yapılması için geliştirilen YSA'nın test aşamasında kullanılan test görüntüleri ayrıca EK8'de verilmiştir.

Bu aşamada alınan görüntüler tek bir kullanıcıya aittir ve ekrandan uzaklıkları birbirinden farklı olmakla birlikte ortalama 50 cm civarındadır. Görüntüler, benzer ışık şiddetine sahip ortamlarda ve ekrana dik bakış açısı ile bakarken alınmıştır.

4.1.5. Koordinat tespiti için kullanılan matlab yazılımı

Bilgisayarın dâhili web kamerası ile alınan kullanıcı görüntüleri kullanılarak ekranda bakılan bölge koordinatlarının bulunması amacıyla MATLAB R2010a® platformu kullanılmıştır. Araştırmacı tarafından Matlab neural network, image processing, wavelet toolboxları ve Young tarafından geliştirilen CHT fonksiyonları (circle_hough, circle_houghpeaks, circlepoints) [67] kullanılarak görüntü tarama, öznitelik çıkarımı ve YSA eğitimi Matlab ortamında gerçekleştirilmiştir.

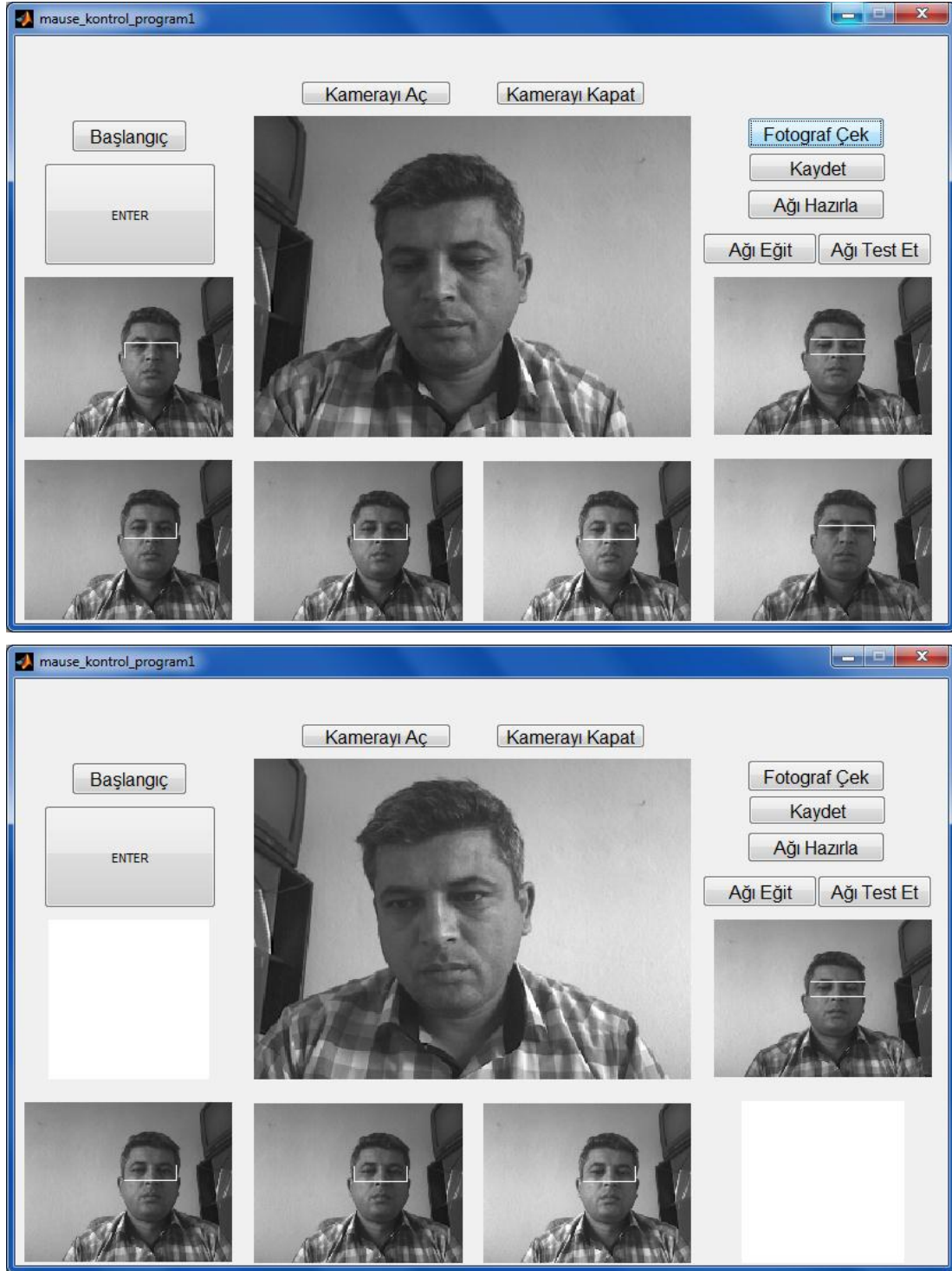
YSA'nın giriş neuron sayısı, her bir görüntünün 122 veri ile ifade edilmesi sebebiyle 122, çıkış neuron sayısı test edilen görüntüden ekranda bakılan bölge koordinatlarının (xkoordinatı, y koordinatı) tespit edilebilmesi amacıyla 2 olarak belirlenmiştir.

Çalışmanın bu aşamasında kullanılan Matlab yazılımları EK9, EK10'de verilmiştir.

4.1.6. Çalışmanın denekler ile tekrarlanması amacıyla hazırlanan arayüz

Bu aşamada, önceki aşamalarda geliştirilen YSA'ların farklı kullanıcılar için benzer sonuçlar vereceği öngörüsü doğrultusunda, Matlab Gui kullanılarak bir arayüz geliştirilmiş ve çalışma farklı denekler ile tekrarlanmıştır. Geliştirilen arayüz programı EK11'de verilmiştir.

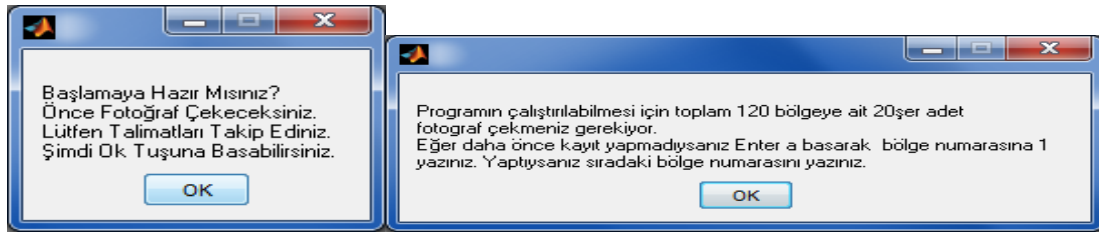
Öncelikle çalışmanın ilk aşamasında araştırmacı tarafından araştırmacının göz bölgesini bulmak amacıyla geliştirilen YSA'nın, bu aşamada görüntü alacak deneklerin göz bölgesini bulabilmesi gerekmektedir. Bu sebeple deneklerden, farklı mekân, farklı ışık şiddeti ve farklı bakış açılarında bilgisayar ekranına bakarken görüntüler alınmış ve YSA eğitim seti genişletilmiştir. Bunun için 2 denekten 50, 6 denekten 20 adet olmak üzere 8 denekten alınan 220 adet görüntü YSA eğitim setine ilave edilmiştir. Böylelikle göz bölgesini bulması amacıyla kullanılan YSA eğitim seti 327 kullanıcı görüntüsünden elde edilen 327'si doğru toplam 1962 eğitim verisi ile tekrar eğitilmiştir. Deneklerden birisi çalışmaya devam edememiş, ancak bu aşamada 4 yeni denek çalışmaya dâhil olmuş ve çalışma süreci 8 erkek, 3 kız denekten oluşan 11 farklı kullanıcı ile tamamlanmıştır. YSA çalışmaya sonradan katılan 4 kullanıcının göz bölgesi ile eğitilmemesine rağmen bu kullanıcıların göz bölgelerini de bulmuştur. Kullanılan arayüz ile görüntü gerçek zamanlı olarak alınmakta ve kullanıcı göz bölgeleri bulunarak ekrana basılmaktadır.



Şekil 4.2. Denekler için hazırlanan arayüz programı ve hatalı görüntülerin kayıttan önce silinmesi

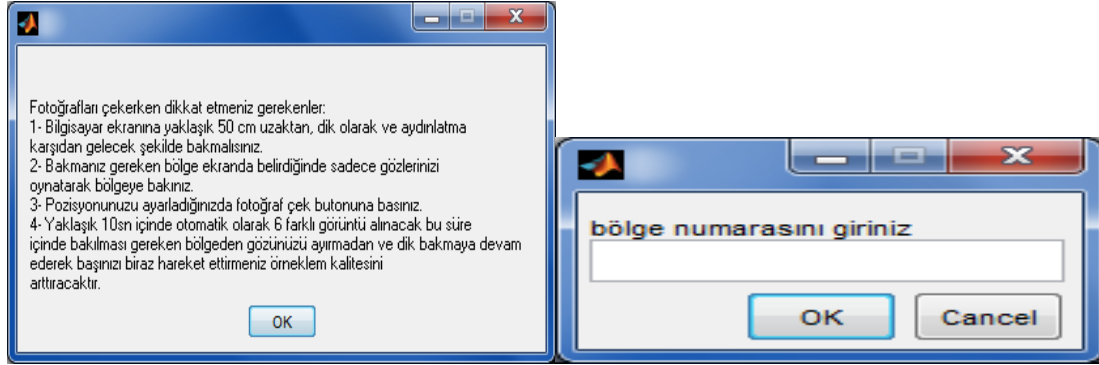
Arayüz, çalışma süreci hakkında bilgisi olmayan deneklerin görüntü alma sürecini en az hata ile gerçekleştirmelerini sağlamak amacıyla, kullanım esnasında denekleri yönlendirecek ve bakılması gereken bölgeleri sırayla ekrana tek tek basabilecek şekilde tasarlanmış ve denekler kullanım hakkında bilgilendirilmişlerdir. Böylece

bilgi eksikliği ve programsal nedenler ile oluşabilecek hataların önüne geçilmeye çalışılmıştır. Arayüz kullanım esnasında, her defasında gerçek zamanlı 6 adet görüntü almakta ve göz bölgesinin bulunamaması ya da herhangi başka bir hata sonucu oluşabilecek yanlış çekimlerde, deneklere hatalı görüntüyü silme imkânı tanımaktadır. Kaydet butonu ile alınan görüntüler ve göz bölgesine ait piksel değerleri bölge numarasına göre kaydedilmektedir. Her bölge için kayıt sayısı 20'ye ulaştığında program kayıt işlemini tamamlayacak ve kayıtlı görüntülerin ve verilerin üzerine kayıt gerçekleştirmeyecek şekilde tasarlanmıştır. Deneklerin, 100 eğitim bölgesi ve 20 test bölgesi için olmak üzere göz bölgesi doğru bulunmuş 2400 görüntü alması gerekmektedir. Denekler kaydet butonuna bastığında arayüz programı bu görüntüleri ve görüntülere ait göz bölgesi piksel değerlerini kaydetmektedir. Bölge tespiti için geliştirilen YSA dik bakışları algıladığından, görüntülerin farklı bakış açılarıyla alınması test sonuçlarını olumsuz etkilemektedir.



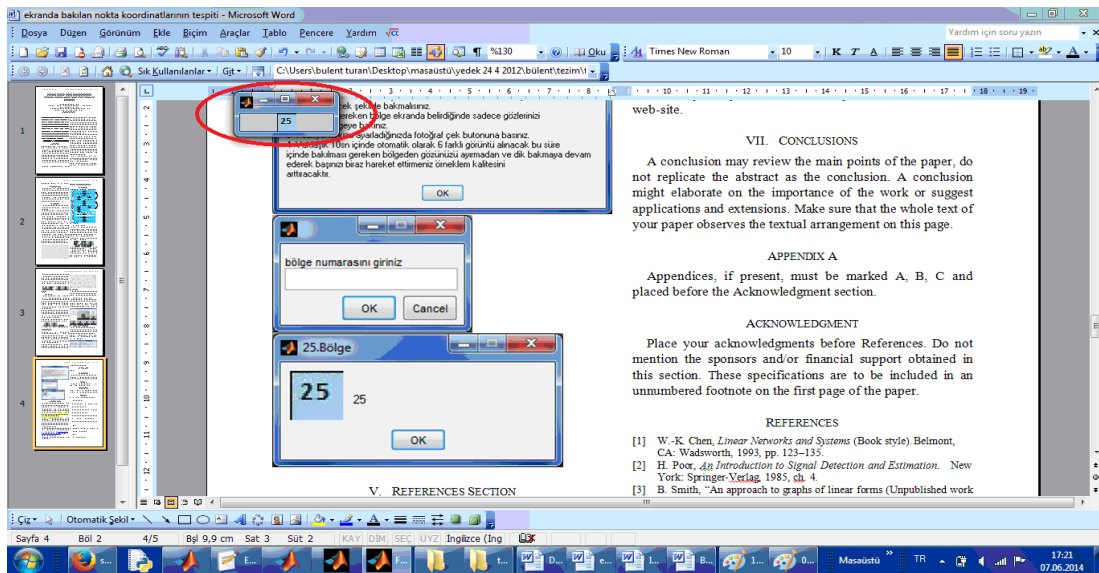
(a)

(b)

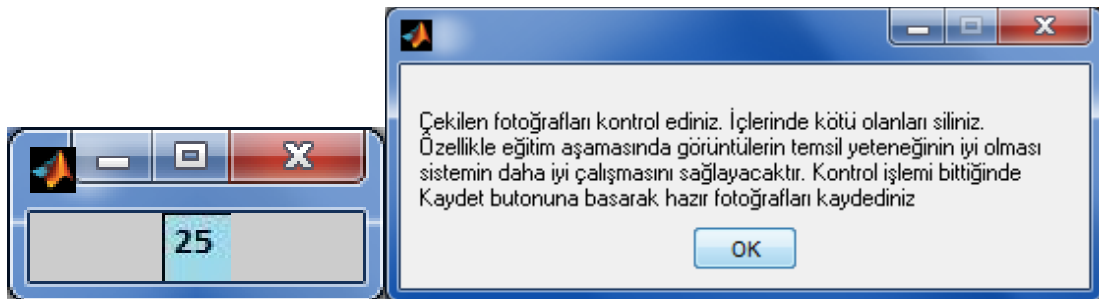


(c)

(d)



(e)



(f)

(g)

Şekil 4.3. Arayüz yönlendirme ve bölgeleri ekrana basma pencereleri

Şekil 4.3. deki pencereler şu bilgileri içermektedir;

- a- “START” butonuna basıldığında ekrana gelen açıklama penceresi
- b- İlk açıklama penceresinde “OK” butonuna basıldığında ekrana gelen açıklama penceresi
- c- Fotoğraf çekimi esnasında dikkat edilecek hususları açıklayan pencere
- d- Görüntüsü alınmak istenen bölge numarasının girilme ekranı
- e- Numarası girilen bölgenin ekrana basılması f- bölge numarası girildiğinde ekrana bölgeye ait koordinat bilgisi dahilinde basılan 24×24 boyutlarında bölgeyi içeren pencere
- g- Çekilen fotoğrafların kayıttan önce kontrol edilmesini sağlayan açıklama penceresi.

4.1.7. Deneklere ait eğitim ve test verilerinin kullanılması

Deneklerden alınan görüntüler araştırmacı tarafından kontrol edilmiş ve daha sonra kullanılmıştır. Kontrol aşaması, genel olarak deneklerin dik bakış açısı ve uygun aydınlatma ortamlarında görüntü alıp almadıkları şeklinde yapılmakla birlikte, aynı bölgeye bakılarak kaydedilmiş görüntüler arasına farklı bölgelere bakılan görüntülerin karışıp karışmadığı ve farklı bölgelere bakarken çekilmiş görüntüler arasında kafa pozisyonlarının değişip değişmediği de ayrıca kontrol edilmiştir. Deneklerin hatalı görüntüleri yeniden alması ayrıca görüntü alırken kafa pozisyonları değişen deneklerin tüm görüntüleri yeniden alması sağlanmıştır.

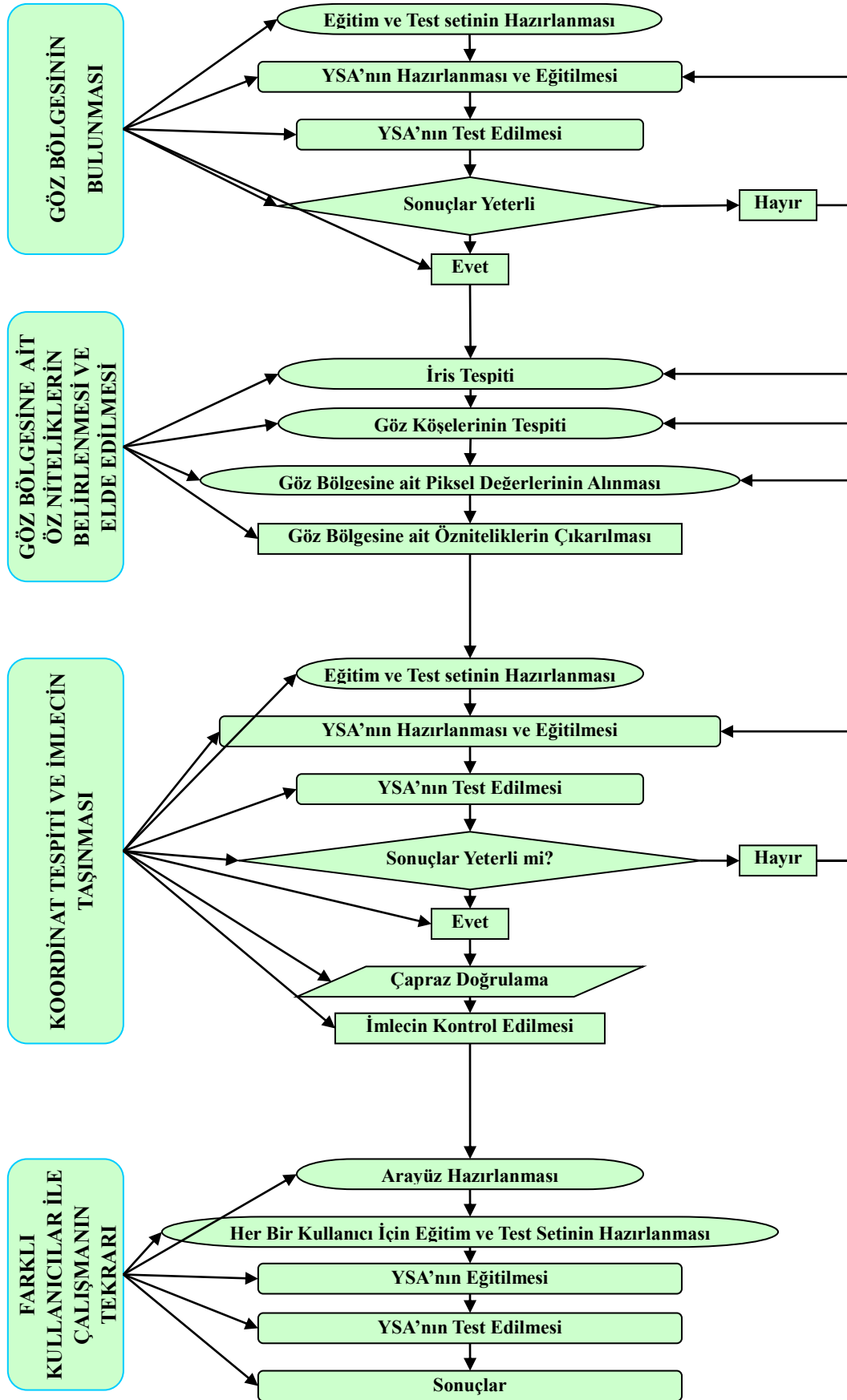
Deneklerden alınan görüntülerin uygunluğuna karar verildikten sonra araştırmacı görüntüleri için koordinat tespitine yönelik eğitim ve test verilerinin hazırlanmasına benzer şekilde her bir denek için eğitim ve test verileri hazırlanmıştır.

4.2. Çalışma Aşamalarının Gerçeklenmesi

Bakışlar ile bilgisayar kontrolü günümüzün yaygın araştırma konularındandır. Bu amaçla yapılan çalışmalarda görüntü işleme teknikleri kullanılmaktadır. Bakışlar ile bilgisayar kontrolünde en yaygın kullanılan yöntem Eye Gaze sistemlerdir. Eye Gaze sistemlerde göz hareketlerinin takibi eyetracker kullanılarak gerçekleştirilmektedir.

Görüntü işleme tekniklerindeki gelişmeler sonucu dizüstü bilgisayar kullanıcılarına ait görüntüler işlenerek görüntü içinde göz bölgesinin bulunmasına ve bu bölgeye ait piksel değerlerinden veri toplamaya yönelik çalışmalar yapılmaktadır.

Çalışmada bakışlar ile bilgisayar kontrolüne temel olacağı öngörülerek herhangi bir ek donanım kullanmadan bakışlar ile imleç kontrolü amaçlanmıştır. Çalışma dizüstü bilgisayar kullanıcısının webcam ile alınan görüntülerinin Matlab neural network ve image processing toolbox'ları kullanılarak işlenmesi doğrultusunda planlanmıştır. Çalışmanın aşamaları ise aşağıdaki akış diyagramına göre gerçekleşmiştir.



Şekil 4.4. Akış diyagramı

4.2.1. Göz bölgesinin bulunması

Çalışmanın ilk aşamasında, kullanıcı görüntüleri ile imleç kontrolü için kullanıcının bilgisayar kullanırken alınan görüntüleri içinden göz bölgelerinin bulunması hedeflenmiştir. Çünkü görüntü işleyerek imleç kontrolünün gerçekleştirilebilmesi için göz bölgesine ait özniteliklerin kullanılabilmesi öngörülmüştür. Böylelikle dizüstü bilgisayarlarda ek donanım kullanmadan imleç kontrolü gerçekleştirilebilir. Bu amaçla kullanıcı görüntüsü içinde göz bölgesinin bulunması için sınıflandırıcı problemlerinde sıklıkla kullanılan yapay sinir ağının kullanılmasına karar verilmiştir. Bu aşamada dizüstü bilgisayar kullanıcılarından farklı ışık şiddeti, farklı mekan ve farklı bakış açılarındaki bilgisayar ekranına bakarken dâhili web kamerası ile 640×480×3 boyutlarında görüntüleri alınmıştır. Alınan görüntüler gri görüntüye dönüştürülerek çalışmada kullanılmıştır. Bu görüntüler içinde kullanıcının göz bölgesini (48×160) doğru ifade edecek bölgeler ve ayrıca (48×160) yanlış olarak ifade edecek bölgeler manüel olarak belirlenmiştir. Bu görüntülerden elde edilen veriler azaltılarak Matlab ortamında 1×120 vektör matrisi haline getirilerek, YSA eğitiminde kullanılmıştır.

Eğitim setinin hazırlanmasında ve YSA eğitiminde kullanılan doğru ve yanlış göz bölge örnekleri ve bu göz bölgelerinin elde edildiği kullanıcı görüntüleri Şekil 4.5’de verilmiştir.



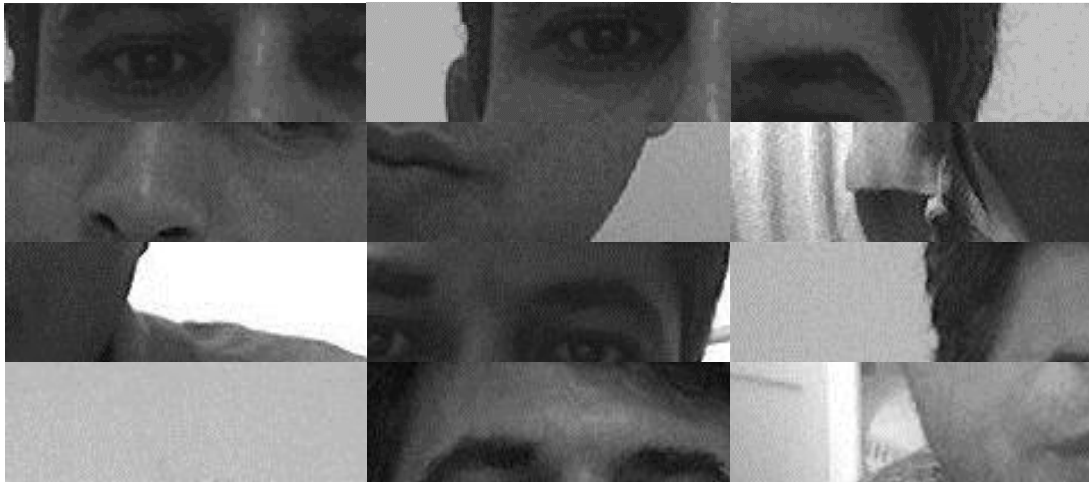
(a)



(b)



(c)



(d)

Şekil 4.5. (a) Farklı mekan ve ışık şiddetinde elde edilen görüntülerden örnekler, (b) orijinal görüntüden elde edilmiş gri görüntüler (c) bu görüntülerden eğitim seti için elde edilen doğru göz bölgeleri [34, 68], (d) bu görüntülerden eğitim seti için elde edilen yanlış göz bölgeleri.

Bu aşamada YSA oluşturulurken ileri beslemeli geri yayımlı ağ modeli kullanılmıştır. Oluşturulan YSA 1 ara, 1 çıkış katmanından oluşmaktadır. Ara katmanda 150 adet, Çıkış katmanında ise 1 adet nöron kullanılmıştır. Ağ parametreleri ağın tekrar tekrar eğitilip test edilmesi sonucu elde edilen test sonuçlarının başarıım yüzdesine göre belirlenmiştir. Çalışmada kullanılan YSA'lar

EK12’de verilmiştir. Eğitimde 107 kullanıcı görüntüsü kullanılmıştır. Bu görüntülerin her birinden 1’i doğru 5’i yanlış göz bölgesi olmak üzere toplam 642 görüntü kullanılmıştır. Her bir giriş için ağa giriş görüntüsünden elde edilen 120 değerden oluşan öznitelik vektörü uygulanmış, doğru girişler için çıkış 1, yanlış girişler için ise çıkış -1 olarak ağ eğitilmiştir. Eğitim ve test aşamaları sonucunda geliştirilen YSA parametreleri Tablo 4.2’de verilmiştir.

Tablo 4.2. Göz bölgesinin bulunması için kullanılan yapay sinir ağının parametreleri [68]

Giriş seti	107’i doğru, 535’i yanlış görüntü verisi giriş vektörleri oluşturulurken kullanılmıştır.
Ağ modeli	İleri beslemeli geri yayımlı
Katman sayısı	1 ara katman 1 çıkış katmanı
Nöron sayısı	Ara katman 150 adet, Çıkış katmanı 1 adet
Aktivasyon fonksiyonu	Ara katman logaritmik sigmoid; Çıkış katmanı tanjant sigmoid
Öğrenme algoritması	trainsecg
Öğrenme katsayısı	0.6
Momentum katsayısı	0.8

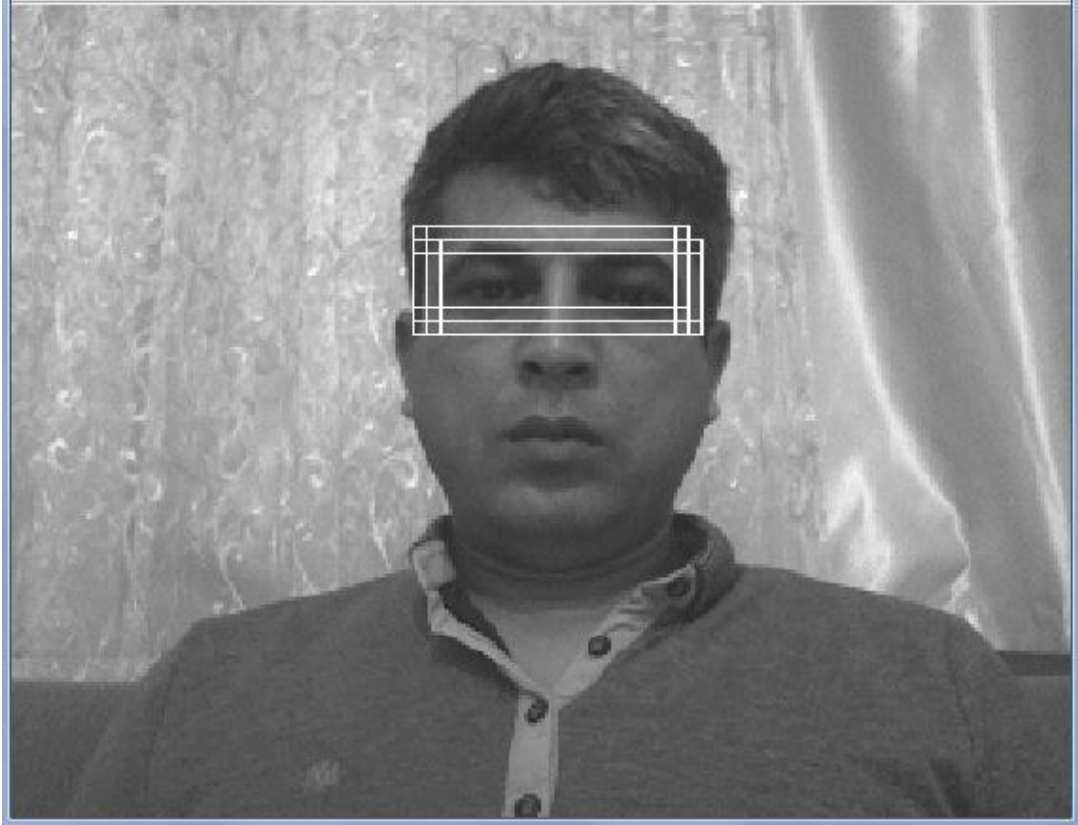
YSA’nın test aşamasında kullanılan test görüntüleri EK3’de verilmiştir. Eğitim aşamasında göz bölgesini ifade eden ve etmeyen manüel olarak belirlenmiş görüntüler kullanılarak eğitim veri seti hazır olarak YSA’ya verilmiştir. Ancak test aşamasında bu şekilde hazır bir test veri seti kullanılmamış, test edilecek görüntülerde tarama usulü ile göz bölgesi olması ihtimali olan bölgelerden elde edilen veriler, YSA ile test edilmiştir. Her bir görüntüde göz bölgesi bulunana kadar tarama usulü ile veri alma işlemi tekrarlanmıştır. Test, aşamasında tarama usulü ile veri alma işleminin gerçekleştirilmesine yönelik hazırlanmış örnek görüntü Şekil 4.6’da verilmiştir. Bu aşamada elde edilen ardışık görüntüler birleştirilerek gif animasyon oluşturulmuştur. Oluşturulan animasyonlar EK13’de verilmiştir.



Şekil 4.6. Test aşamasında YSA ile tarama usulüne göre göz bölgesinin bulunması

Test esnasında farklı zaman, farklı mekân ve farklı ışık şiddetlerinde örnekler kullanılması gerekliliğinden dolayı, test görüntüleri ayrık zamanlı oluşturulmuştur. Bu sebeple her bir test görüntüsünde tarama baştan başlamakta ve bir önceki sonuçtan olumlu ya da olumsuz herhangi bir şekilde etkilenmemektedir. Ancak bu aşamada geliştirilen YSA'nın, çalışmanın ilerleyen aşamalarında gerçek zamanlı olarak kullanılması planlanmaktadır. Gerçek zamanlı çalışmalarda bir önceki test süreci sonucunda elde edilen değerlerin biliniyor olması, bir sonraki test sürecinde taramaya başlanacak bölge hakkında kullanılabilir bir veri içermektedir. Kullanıcı her ne kadar bilgisayar kullanırken görüntü içinde farklı koordinat bölgelerinde yer alma ihtimaline sahip olsa da, ardışık yakalanan görüntülerde birbirinden çok farklı bölgelerde olma ihtimali yoktur. Bu sebeple yazılım gerçek zamanlı çalışma için tekrar düzenlenmiş, ilk alınan görüntüde, elde herhangi bir veri olmaması sebebiyle taramanın başlangıç koordinatlarından başlaması ancak sonraki görüntülerde bir önceki görüntüde elde edilen göz bölgesi başlangıç koordinatlarından başlayarak göz bölgesinin bu koordinatlar etrafında aranması sağlanmıştır.

Gerçek zamanlı çalışma aşamasında tarama usulü ile veri alma işleminin gerçekleştirilmesine yönelik hazırlanmış örnek görüntü Şekil 4.7’de verilmiştir. Gerçek zamanlı elde edilen ardışık görüntüler birleştirilerek gif animasyon oluşturulmuştur (EK13).



Şekil 4.7. Gerçek zamanlı çalışma aşamasında YSA ile tarama usulüne göre göz bölgesinin bulunması

Tarama usulünün gerçek zamanlı çalışma temelli olarak geliştirilmesi sonucu taramanın her seferinde başlangıç koordinatlarından başlamak zorunda olmaması yazılımın gerçek zamanlı çalışması esnasında işlem hızını da arttırmaktadır. Örneğin dâhili web kamerası ile arka arkaya yakalanan iki görüntüde kullanıcının göz bölgesinin aynı koordinatlardan başladığı düşünüldüğünde, ikinci görüntünün işlenmesi esnasında taramanın ilk basamağında göz bölgesi bulunabilecektir.

Geliştirilen YSA yazılımının hem gerçek zamanlı hem ayrık zamanlı olarak çalıştırılması esnasında 640×480 görüntü içinde 48×160 boyutlarında göz bölgesi aranmaktadır. YSA tarafından test edilen çerçevenin göz bölgesi olmadığı belirlendiğinde, çerçeve yatay ya da dikey ekseninde 4 piksel kaydırılarak yeni çerçeve

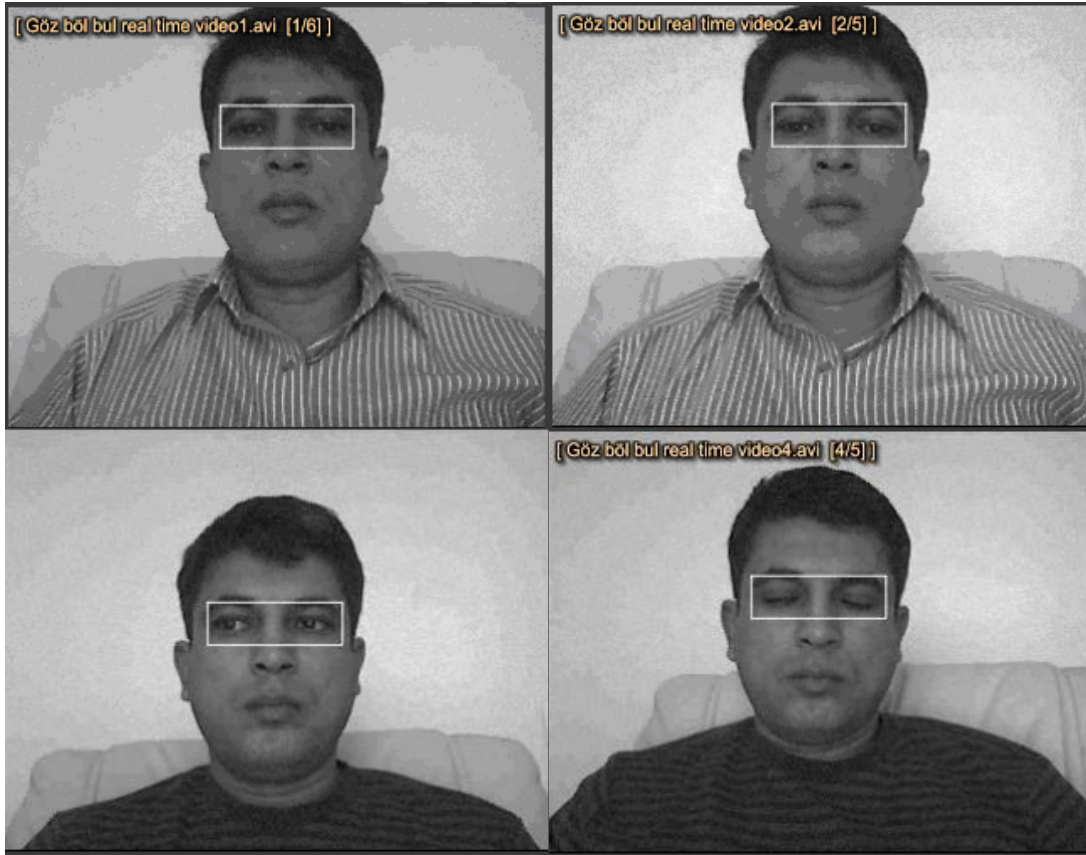
belirlenmektedir. Bu çerçeve, tekrar YSA ile test edilmekte ve böylece göz bölgesi bulunana kadar tarama devam etmektedir. Çerçevenin tarama esnasında piksel piksel değilde 4 piksel kaydırılmasının sebebi, çalışmada başarı oranının azalmasına neden olmadan işlem yükünü 16 kat azaltmasıdır.

Çalışma esnasında göz bölgesinin bulunması amacıyla bir çok YSA eğitilip test edilmiştir. Tekrarlanan test aşamalarında %90-%95-%100 başarı oranlarına sahip YSA'lar belirlenmiştir. Çalışmanın devamında, kullanılan test seti üzerinde %100 oranında başarıya ulaşılan YSA kullanılmıştır. Yapay sinir ağının test setine uygulanması sonucu elde edilen görüntülerden bazıları Şekil 4.8'da verilmiştir.



Şekil 4.8. Yapay sinir ağının uygulandığı test görüntülerinden örnekler[34, 68].

Bu aşamada geliştirilen YSA tabanlı Matlab Yazılımı çalışmanın öznitelik çıkarımı, koordinat tespiti ve imlecin taşınması aşamalarında gerçek zamanlı olarak başarılı bir şekilde kullanılmıştır. Gerçek zamanlı çalışma ile ilgili videolar (EK13) hazırlanmıştır. Hazırlanan videolardan örnek görüntüler Şekil 4.9'da verilmiştir.



Şekil 4.9. Göz bölgesinin bulunması için geliştirilen YSA'nın gerçek zamanlı çalışma örnekleri

Çalışmada YSA'nın eğitimi için hazırlanan Matlab yazılımı EK4'te, ayırık zamanlı test görüntüleri ile test edilmesi için geliştirilen Matlab yazılımı EK5'te, gerçek zamanlı kullanılması için geliştirilen Matlab yazılımı EK6'da verilmiştir.

4.2.2. Göz bölgesine ait özniteliklerin belirlenmesi ve elde edilmesi

Göz hareketleri ile ek donanım kullanmadan imleç kontrolünün gerçekleştirilebilmesi amacıyla, bu aşamada görüntü içinde belirlenmiş olan göz bölgesinden ekranda bakılan noktanın koordinat bilgilerini içeren verilerin toplanması amaçlanmıştır. Bu amaçla göz bölgesine ait piksel değerlerinin ayrıca iris ve göz köşelerinin bulunması ile ekranda bakılan nokta koordinatlarına ait özniteliklere ulaşılabileceği düşünülmüştür. Böylece piksel değerleriyle iris merkezi, iris genişliği, göz köşe noktalarının koordinatları ile bu noktaların birbirlerinden yatay ve

düŖey eksenindeki uzaklıkları öznitelik olarak kullanılabilir. Öznitelik olarak kullanılması düŖünülen verilerden;

1. Göz bölgesine ait piksel deęerlerinin ekranda bakılan nokta koordinat bilgisinin bileŖeni olduęu,
2. İris geniŖlięinin ve aynı göze ait göz köŖe noktalarının birbirinden olan uzaklıklarının ekrandan olan uzaklık bilgisinin bileŖeni olduęu,
3. İris ve göz köŖe nokta koordinatlarının kullanıcının ekrana göre pozisyon bilgisinin bileŖeni olduęu,
4. Göz köŖe noktaları ve iris merkezi koordinatlarının birbirinden dikey ve yatay yönde olan uzaklıklarının kullanıcının kafa pozisyon bilgisinin ve ekrana bakıŖ açısı bilgisinin bileŖeni olduęu,
5. İris merkez koordinatlarının aynı göze ait göz köŖelerinin birleŖimi ile oluŖan hattın üstünde ya da altında kalmasının kullanıcının ekranda baktığı noktanın dikey eksenindeki koordinat bilgisinin bileŖeni olduęu,
6. İris merkez koordinatlarının aynı göze ait göz köŖelerine yatay eksenindeki uzaklık bilgisinin kullanıcının ekranda baktığı noktanın yatay eksenindeki koordinat bilgisinin bileŖeni olduęu,

ön görölmüŖtür.

Bu sebeple görüntü içinde kullanıcının irisi ve göz köŖe nokta koordinatları bulunabilirse, göz bölgesine ait piksel deęerleri ile birlikte ekranda bakılan nokta koordinatlarının tespiti amacıyla geliŖtirilecek olan YSA eęitiminde kullanılabilir düŖünülmüŖtür.

4.2.2.1. İris tespiti

Günümüzde görüntü içinde geometrik şekillerin tespit edilmesi, görüntü işlemede birçok uygulama alanı bulmaktadır. Bu alanlardan bazıları, nesne tanımlama, iris tespiti, plaka tespiti olarak sıralanabilir. Örneğin Hough Transformu siyah-beyaz görüntülerin içindeki düzgün doğruları tespit etmek için kullanılmaktadır. Bunun yanı sıra, doğrusal olmayan eğrilerin tespitinde de kullanılmaktadır.

Özellikle dairesel şekillerin bulunması amacıyla CHT yaygın bir şekilde kullanılmaktadır. CHT oy verme yöntemine göre çalışmaktadır [69]. Bu teknikte eşiklenmiş görüntüdeki her pikselin merkez olabilme potansiyeli değerlendirilir ve en yüksek puanı alan piksel merkez olarak belirlenir [14]. Denklem 4.1, 4.2 ve 4.3'de a,b çember merkezinin koordinatları, r çemberin yarıçapı olarak alındığında belirlenen aralıktaki yarıçap değerleri için çember merkezi verilen denklem kullanılarak bulunabilir.

$$r^2 = (x - a)^2 + (y - b)^2 \quad (4.1)$$

$$x = a + r \times \sin(\theta) \quad (4.2)$$

$$y = b + r \times \cos(\theta) \quad [68] \quad (4.3)$$

Çalışmada, göz bölgesi bulunan görüntü içinde irisin bulunması amacıyla iki farklı yöntem kullanılmıştır. İlk kullanılan yöntemde sağ ve sol göz bölgeleri ayrıldıktan sonra, ön filtreleme, Canny kenar bulma algoritması ve CHT uygulanarak görüntü içinde iris tespit edilmeye çalışılmıştır. İrisin arandığı görüntüye CHT uygulanırken, önce irisin olması muhtemel büyüklüğe göre minimum ve maximum yarıçaplar belirlenmiştir. Sonra kenarları bulunmuş, siyah beyaz görüntü içindeki her noktanın etrafında belirlenen yarıçap uzaklığındaki beyaz noktalar saydırılmış ve en çok beyaz noktaya sahip olan piksel koordinatı iris merkezi olarak belirlenmiştir. Bu uygulamada elde edilen başarı oranı, farklı ön filtremeler yapılarak tekrarlanmasına rağmen yüksek değildir.

CHT'nin iris bulmada yaygın olarak kullanılmasına ve yüksek başarı oranlarına ulaşılmasına rağmen çalışmada düşük bir başarı oranına ulaşılmasının iki nedene

bağlı olduğu düşünülmektedir. İlki yüksek başarı oranı elde edilen çalışmalarda iris bölgesinin daha yakından görüntüleniyor olması, ikincisi de alınan görüntülerin göz hastalıkları veya biyometrik tanıma gibi hassas alanlarda kullanılması sebebiyle farklı ışık şiddeti etkilerinden kurtulmak amacıyla kızıl ötesi kamera kullanılmasıdır. Bunlardan ilkinin iris bölgesinin doğru bulunmasına, ikincisinin ise iris için çizilen çemberin hassasiyetine etki ettiği görülmüştür.

Bu sebeple ikinci yöntem geliştirilmiştir. Bu yöntemde göz bölgesinin içinde sağ ve sol iris bölgelerinin bulunması için farklı iki yapay sinir ağı geliştirilerek CHT'nin irisi arayacağı bölgenin daraltılması hedeflenmiştir. Farklı ışık şiddetlerinin oluşabileceği düşük hassasiyet oranı ihmal edilerek, irisin doğru bulunma oranının arttırılacağı öngörülmüştür. Bu yöntemde, sağ ve sol göz bölgelerinin içinde iris bölgesini içeren 24×24 boyutlarında bölgeler belirlenmiş ve dalgacık dönüşümü uygulanarak 6×6 boyutlarında matrisler elde edilmiştir. Bu matrisler 1×36 vektör haline getirilerek giriş verisi olarak kullanılmıştır. Hedef matrisi ise doğru girişler için 1 yanlış girişler için -1 olarak ağa uygulanmıştır. Ağ parametreleri test verilerinden elde edilen sonuçlara göre güncellenmiştir. Sol iris bölgesinin bulunması için geliştirilen yapay sinir ağının özellikleri Tablo 4.3'de verilmiştir.

Tablo 4.3. Sol iris bölgesinin bulunması için kullanılan yapay sinir ağının parametreleri [68]

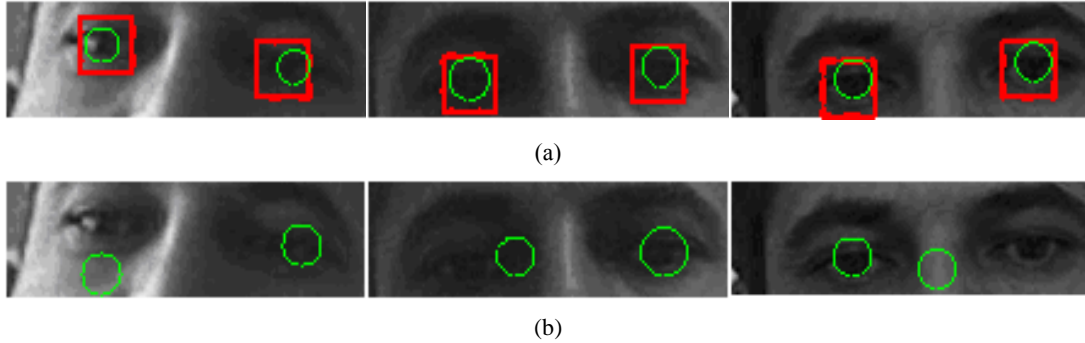
Giriş seti	107'i doğru, 535'i yanlış görüntü verisi giriş vektörleri oluşturulurken kullanılmıştır.
Ağ modeli	İleri beslemeli geri yayımlı
Katman sayısı	1 ara katman 1 çıkış katmanı
Nöron sayısı	Ara katman 150 adet, Çıkış katmanı 1 adet
Aktivasyon fonksiyonu	Ara katman radyal tabanlı; Çıkış katmanı tanjant sigmoid
Öğrenme algoritması	trainscg
Öğrenme katsayısı	0.7
Momentum katsayısı	0.8

Sağ iris bölgesinin bulunması için geliştirilen yapay sinir ağının özellikleri Tablo 4.4'te verilmiştir.

Tablo 4.4. Sağ iris bölgesinin bulunması için kullanılan yapay sinir ağının parametreleri [68]

Giriş seti	107'i doğru, 535'i yanlış görüntü verisi giriş vektörleri oluşturulurken kullanılmıştır.
Ağ modeli	İleri beslemeli geri yayımlı
Katman sayısı	1 ara katman 1 çıkış katmanı
Nöron sayısı	Ara katman 100 adet, Çıkış katmanı 1 adet
Aktivasyon fonksiyonu	Ara katman radyal tabanlı; Çıkış katmanı tanjant sigmoid
Öğrenme algoritması	trainsecg
Öğrenme katsayısı	0.7
Momentum katsayısı	0.8

Bu ağlar kullanılarak bulunan sağ ve sol iris bölgesine CHT uygulanarak, irisin bulunmasındaki başarı oranının arttığı görülmüştür. Şekil 4.10'da iris bölgesinin yapay sinir ağı uygulandıktan sonra CHT uygulanarak ve sadece CHT uygulanarak bulunmasına ilişkin örnekler verilmiştir.



Şekil 4.10. (a) iris bölgesinin yapay sinir ağı ile bulunduktan sonra CHT ile iris tespiti, (b) sağ ve sol göz bölgesi içinde CHT ile iris tespiti [68]

İrisin bulunması için kullanılan her iki yöntem sonucunda elde edilen başarı oranları Tablo 4.5'de verilmiştir.

Tablo 4.5. Göz bölgesi içinde irisin bulunmasına yönelik kullanılan farklı iki yöntem ile elde edilen sonuçlar [68]

Sol ve sağ göz bölgeleri ayrıldıktan sonra			
DAİRESEL HOUGH TRANSFORMU		YSA + DAİRESEL HOUGH TRANSFORMU	
Sol irisin sol göz bölgesi içinde CHT ile bulunma oranı	Sağ irisin sağ göz bölgesi içinde CHT ile bulunma oranı	Sol iris bölgesinin YSA ile bulunma oranı	Sağ iris bölgesinin YSA ile bulunma oranı
		% 95	% 90
% 75	% 65	Sol irisin sol iris bölgesi içinde CHT ile bulunma oranı	Sağ irisin sağ iris bölgesi içinde CHT ile bulunma oranı
		% 100	% 100

Tablo 4.5’de verilen değerler iris bölgesinin doğru olarak bulunmasına yönelik olarak elde edilen değerlerdir. Çizilen çemberin hassasiyeti dikkate alınmamıştır. Tablo 4.5 incelendiğinde görülmektedir ki, yapay sinir ağı kullanılarak irisin arandığı bölge daraltıldığında, YSA’nın yanlış sonuç ürettiği görüntüler hariç irisin bulunma oranı %100’dür. Bu görüntüleri de dikkate aldığımızda, sol irisin bulunma oranı %95, sağ irisin bulunma oranı %90’dır. Ayrıca ikinci yöntem kullanıldığında başarı, birinci yönteme göre sol irisin bulunmasında % 26,66, sağ irisin bulunmasında %38,46 oranında artmıştır [68].

4.2.2.2. Göz köşelerinin tespiti

Göz köşelerinin bulunması için iki farklı yöntem kullanılmıştır. Bunlardan ilki Ağırlıklı Varyans Bağdaştırma Fonksiyonu (WVPF) diğeri çalışma için geliştirilen köşe bölgesi içindeki En Düşük Değerlikli Pikselin (EDDP) bulunması yöntemidir [68]. Her iki yöntem kullanılarak göz köşeleri bulunmaya çalışılmış ve elde edilen sonuçlar karşılaştırılmıştır.

Ağırlıklı varyans bağdaştırma fonksiyonu (WVPF) göz köşelerinin bulunması amacıyla kullanılan yöntemlerden biridir. Varyans bağdaştırma fonksiyonu (VPF) göz bölgesi içinde iris bölgesinin bulunması için kullanılan bir yöntemdir [70,71]. Bu yöntem geliştirilerek göz köşelerinin bulunması amacıyla da kullanılmıştır. Yöntem kullanılırken önce göz köşesi olması muhtemel bölgeler belirlenmekte, daha sonra Haris Köşe Dedektörü ile görüntü içindeki pikseller köşe olma olasılıklarına göre ağırlıklandırılmakta ve varyans bağdaştırma fonksiyonu bu ağırlık değerleri ile

birlikte işlenerek göz köşeleri bulunmaktadır [17]. Yatay ve dikey WVPF denklemleri 4.4 ve 4.5'te verilmiştir.

Denklem 4.4, ve 4.5'te varyans hesaplanırken her bir piksel için bulunan ağırlık faktörü (R) kullanılmıştır. Satır ve sütun varyanslarının maksimum değerleri köşe noktasının koordinatlarını vermektedir.

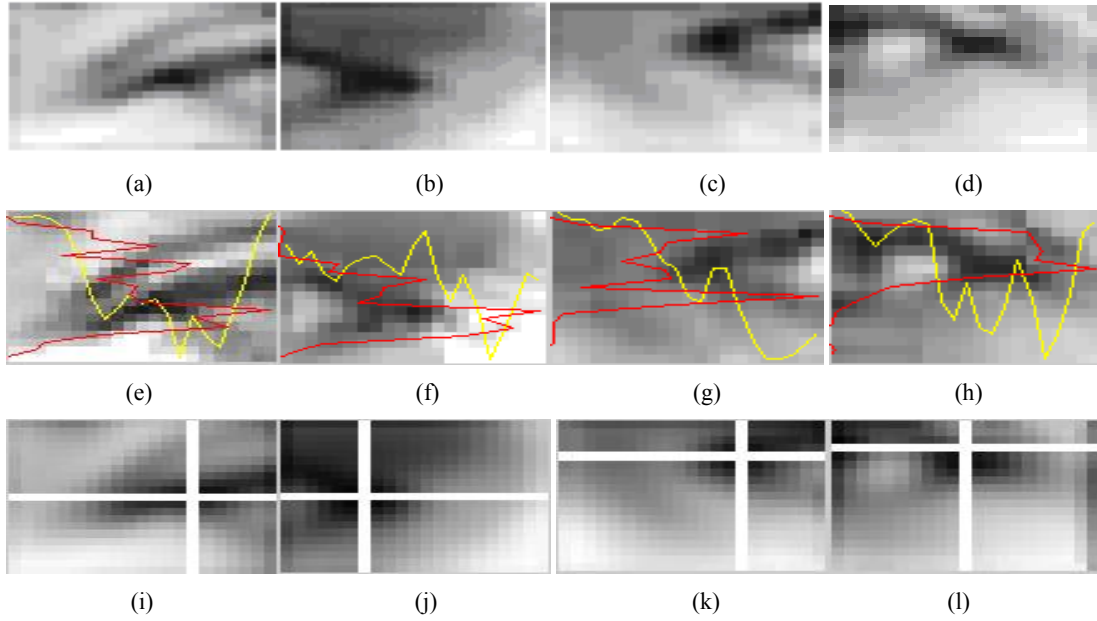
$$\delta_{wv}^2(x) = \frac{1}{x_2 - x_1} \sum_{x_i=x_1}^{x_2} R(x, y_i) * [I(x, y_i) - H_m(x)]^2 \quad (4.4)$$

yatay WVPF

$$\delta_{wh}^2(y) = \frac{1}{y_2 - y_1} \sum_{y_i=y_1}^{y_2} R(x_i, y) * [I(x_i, y) - H_m(y)]^2 \quad (4.5)$$

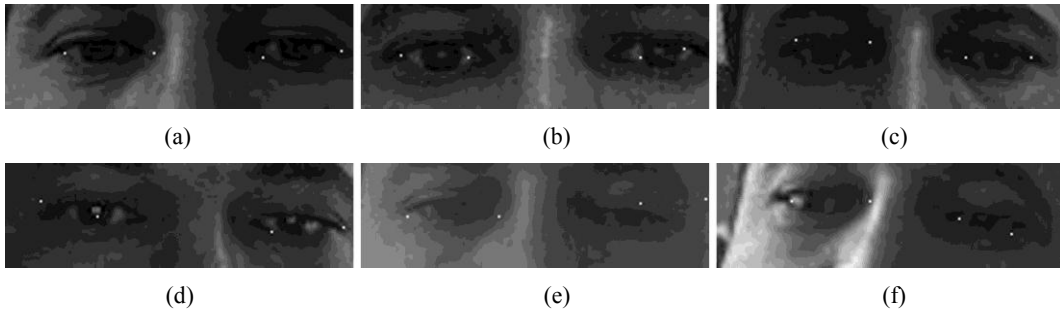
dikey WVPF [68]

Birçok çalışmada, en düşük değerin bulunması basit bir strateji [72] olarak kullanılmaktadır [73,74,75]. Bu çalışmada en düşük değerlikli pikselin (EDDP) bulunması, göz köşelerinin tespiti için kullanılmıştır. İrisin bulunmasından sonra göz köşelerinin bulunması amacıyla göz köşe bölgeleri belirlenmiştir. Göz köşe bölgelerine ve göz köşelerinin bulunmasına örnekler şekil 4.11'de verilmiştir.



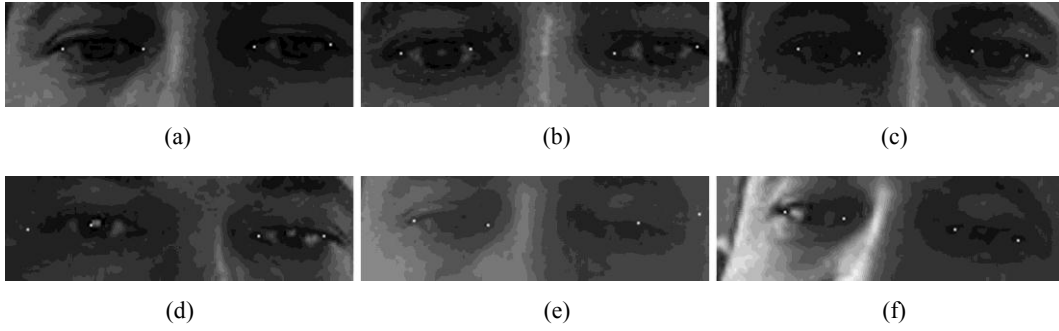
Şekil 4.11. (a) Sol göz dış köşe (b) sol göz iç köşe (c) sağ göz iç köşe (d) sağ göz dış köşe (e),(f),(g),(h) göz köşelerinin WVPF kullanılarak bulunması, (i),(j),(k),(l) göz köşelerinin EDDP kullanılarak bulunması [68]

Göz bölgesi içinde WVPF kullanılarak göz köşelerinin bulunmasına yönelik doğru ve yanlış sonuç üretmiş örnek görüntüler Şekil 4.12’de verilmiştir.



Şekil 4.12. Göz köşelerinin bulunmasında kullanılan WVPF ile elde edilen sonuçlar [68]

Köşe Bölgesi İçindeki En Düşük Değerlikli Pikselin bulunması yöntemi kullanılarak göz köşelerinin bulunmasına yönelik doğru ve yanlış sonuç üretmiş örnek görüntüler Şekil 4.13’te verilmiştir.



Şekil 4.13. Göz köşelerinin bulunmasında kullanılan EDDP ile elde edilen sonuçlar [68]

Şekil 4.12. ve 4.13. d’de sol iris bölgesi, Şekil 4.12. ve 4.13. e’de ise sağ iris bölgesi, geliştirilen YSA’lar tarafından yanlış olarak bulunduğu için iris ve göz köşeleri de yanlış olarak bulunmuştur. Bu sebeple göz köşelerinin bulunması için kullanılan yöntemler karşılaştırılırken dikkate alınmamıştır.

Elde edilen başarı oranları Tablo 4.6’da verilmiştir.

Tablo 4.6. Göz köşe bölgeleri içinde göz köşelerinin bulunmasına yönelik kullanılan farklı iki yöntem ile elde edilen sonuçlar [68]

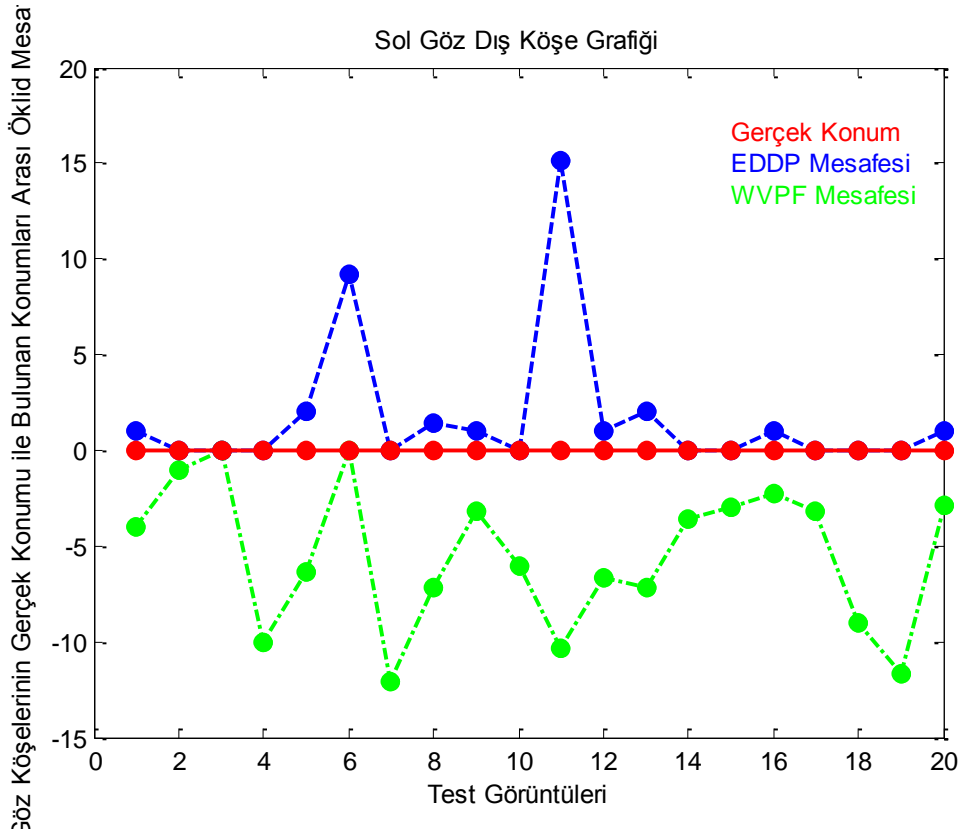
Sol ve sağ göz iç ve dış köşe bölgeleri belirlendikten sonra							
WVPF				EN DÜŞÜK DEĞERLİKLİ PİKSELİN BULUNMASI			
Sol Göz		Sağ Göz		Sol Göz		Sağ Göz	
Dış Köşe	İç Köşe	İç Köşe	Dış Köşe	Dış Köşe	İç Köşe	İç Köşe	Dış Köşe
% 60	% 40	% 70	% 40	% 90	% 80	% 85	% 80

Tablo 4.6’da da görüldüğü gibi Ağırlıklı Varyans Bağdaştırma Fonksiyonu ile elde edilen başarı oranı farklı ön filtreler kullanılmasına rağmen arttırılamamıştır. Kullanılan ikinci yöntemde ise, daha yüksek bir başarı oranına ulaşılmıştır. Bu sayede başarı sol göz dış köşe için %50, sol göz iç köşe için %100 sağ göz iç köşe için %21,43, sağ göz dış köşe için %100 oranlarında artmıştır. Elde edilen oranlara irisin yanlış bulunduğu görüntüler dâhil edilerek ulaşılmıştır [68].

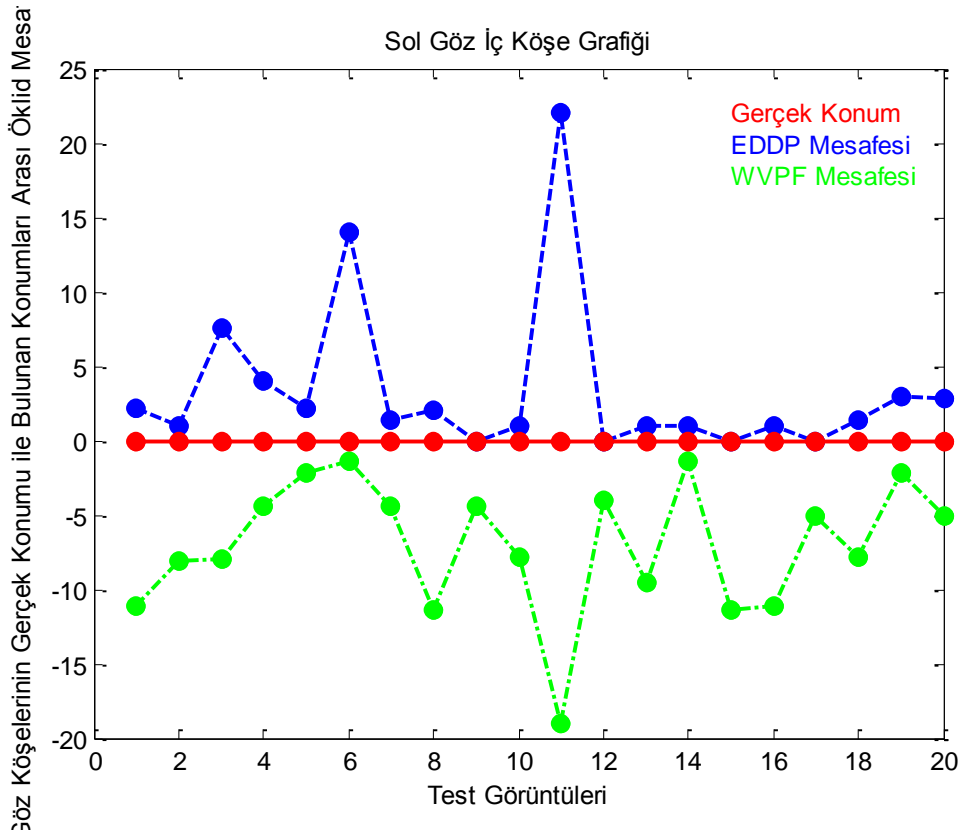
Sonuçların daha iyi anlaşılması için Şekil 4.14, Şekil 4.15, Şekil 4.16 ve Şekil 4.17’de verilen grafikler incelenebilir. Bu şekillerde yatay eksen, her bir test

görüntüsünü ifade ederken, dikey eksen, manüel olarak yerleri belirlenen gerçek göz köşeleri ile WVPF & EDDP yöntemleri kullanılarak bulunan göz köşeleri arası öklid uzaklıklarını temsil edecek şekilde tasarlanmıştır [68].

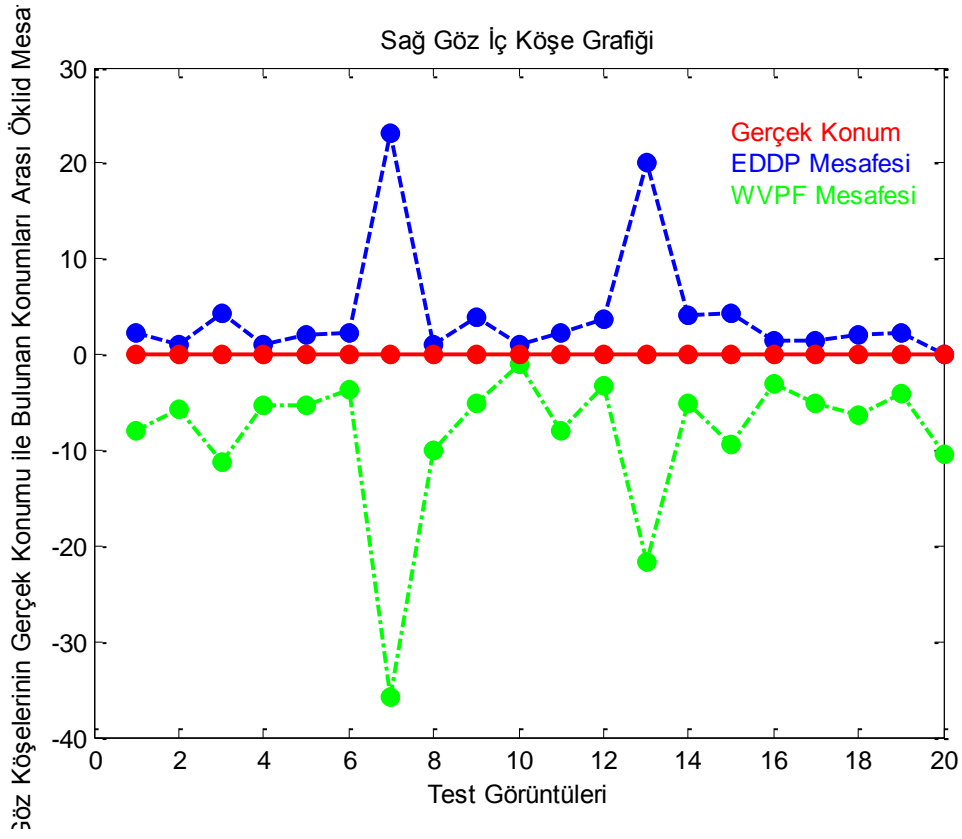
Grafiklerde göz köşelerinin gerçek konumu (manuel bulunan konum) kırmızı çizgiyle ifade edilmiştir. WVPF yöntemi ile bulunan göz köşesi ile gerçek konumu arası öklit mesafesi, negatif yönde ve yeşil kesik çizgiyle, EDDP yöntemi ile bulunan göz köşesi ile gerçek konumu arası öklit mesafesi ise, pozitif yönde ve mavi kesik çizgiyle ifade edilmiştir [68].



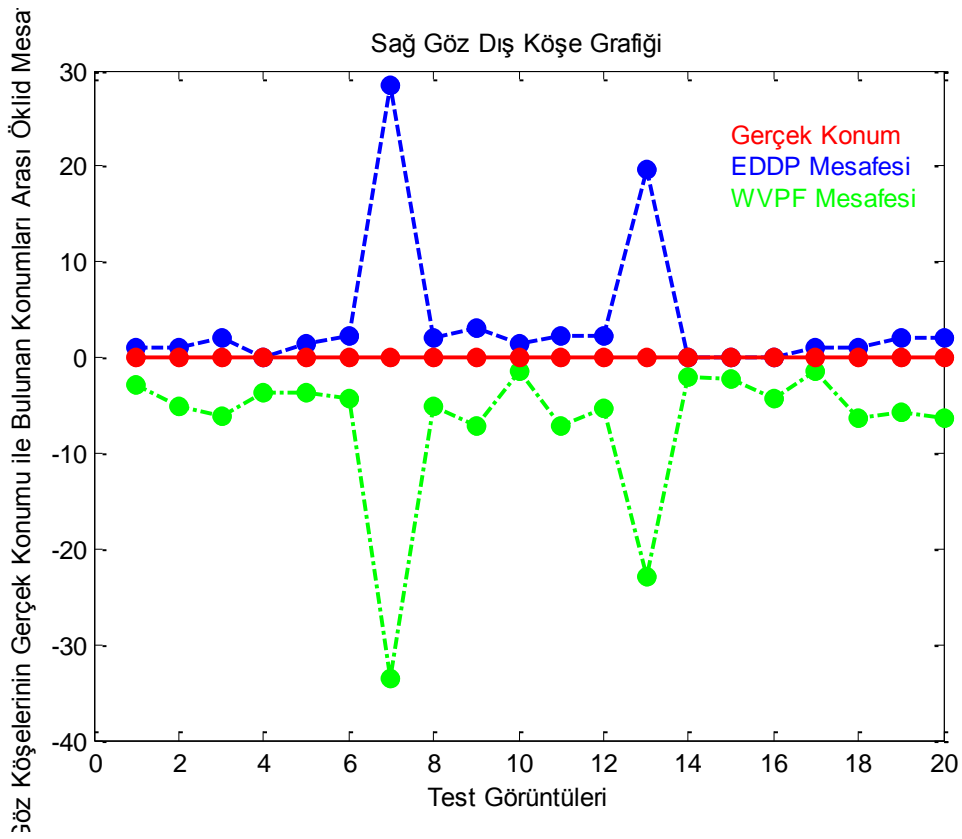
Şekil 4.14. Sol göz dış köşe grafiği [68]



Şekil 4.15. Sol göz iç köşe grafiği [68]

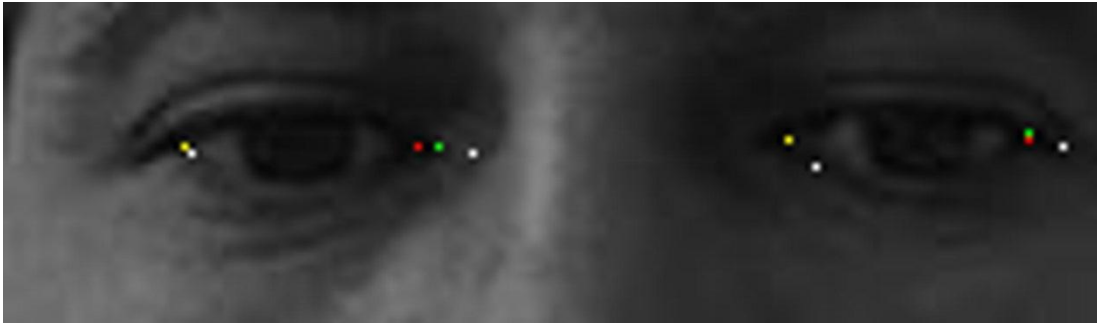


Şekil 4.16. Sağ göz iç köşe grafiği [68]



Şekil 4.17. Sağ göz dış köşe grafiği [68]

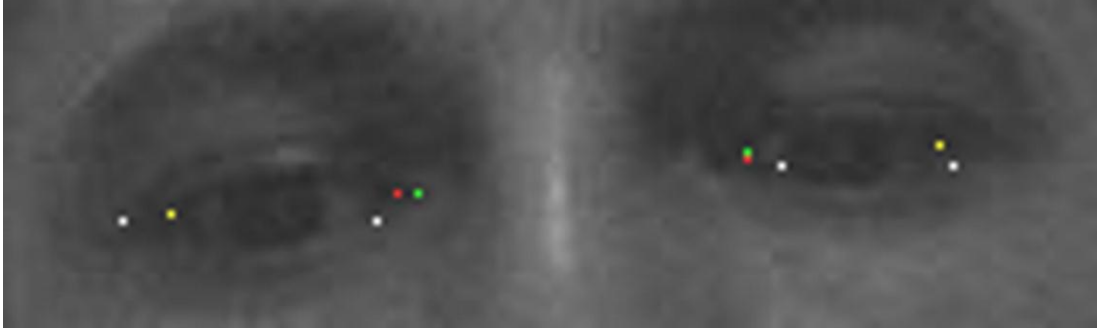
Şekil 4.14, Şekil 4.15, Şekil 4.16 ve Şekil 4.17’de verilen grafikler incelendiğinde, EDDP yöntemi ile bulunan göz köşelerinin WVPF yöntemi ile bulunan göz köşelerine nazaran, manuel olarak bulunan göz köşelerine daha yakın sonuçlar ürettiği açıkça görülmektedir. Bu açıdan tablolarda verilen değerler ile grafikler benzerlik göstermektedir. Fakat grafiklerdeki başarı oranı Tablo 4.6’da verilen oranlara göre daha düşüktür. Çünkü çalışmada her iki yöntemle bulunan köşeler manuel olarak işaretlenen köşe noktasından farklı bir piksel olsa da bazı durumlarda göz köşesini temsil ettiği düşünülmüş ve doğru kabul edilmiştir. Ancak grafiklerde göz köşe noktası koordinatlarının birebir örtüşmediği durumlarda manuel bulunan nokta ile çakışmamaktadır. Ayrıca bulunan Öklid uzaklığının yönünde grafiklerde verilememektedir. Bu sebeplerle grafik oluşturulurken kullanılan öklit uzaklığının göz köşesi olarak kabul edilebilecek noktaları iyi temsil edemediği düşünülmektedir. Bununla ilgili örnek görüntüler Şekil 4.18 ve 4.19’da verilmiştir. Görüntülerde manuel olarak belirlenen köşe noktaları kırmızı, EDDP yöntemi ile belirlenen köşe noktaları yeşil, WVPF yöntemi ile belirlenen köşe noktaları ise beyaz renk kullanılarak görüntüye basılmıştır [68].



Şekil 4.18. Göz köşelerinin bulunmasına yönelik örnek görüntü 1 [68]

Şekil 4.18’deki görüntü incelendiğinde, sol göz dış köşe belirlenirken her üç yöntem ile bulunan noktaların çakıştığı, sağ göz iç köşeyi ifade eden noktalardan manuel olarak bulunan ile EDDP yöntemi kullanılarak bulunan noktaların çakıştığı fakat WVPF yöntemi ile bulunan noktanın bunlardan farklı bir yeri ifade ettiği ve göz köşesi olarak kabul edilemeyeceği görülmektedir. Sol göz iç köşe ve sağ göz dış köşe noktalarında ise, her üçünün farklı noktaları ifade ettiği bunlardan manuel olarak belirlenen noktanın gerçek göz köşesi olduğu, EDDP yöntemi ile bulunan noktanın

da göz köşesi olarak kabul edilebileceği fakat WVPF yöntemi ile bulunan noktanın göz köşesi olarak kabul edilemeyeceği düşünülmektedir [68].



Şekil 4.19. Göz köşelerinin bulunmasına yönelik örnek görüntü 2 [68]

Şekil 4.19'daki görüntü incelendiğinde, sol dış ve sağ dış göz köşelerinde manuel olarak belirlenen göz köşesi ile EDDP yöntemi ile bulunan noktanın çakıştığı WVPF yöntemi ile bulunan noktanın farklı bir yer gösterdiği görülmektedir. Bunlardan sağ göz dış köşe için WVPF ile bulunan noktanın gerçek göz köşesine çok yakın olarak bulunmasına rağmen göz köşesini ifade edemeyeceği aşikârdır. Ancak sol göz iç köşe için EDDP yöntemi ile bulunan ve benzer bir öklit uzaklığına sahip olan nokta, göz köşesinin uzantısının içinde olması sebebiyle göz köşesini ifade edebileceği düşünülmüştür. Sağ göz iç köşe noktasının belirlenmesinde de EDDP yöntemi ile bulunan noktanın çok yakın olması sebebi ile göz köşesini ifade edebileceği görülmektedir [68].

4.2.2.3. Göz bölgesine ait piksel değerlerinin alınması

Görüntü içinde göz bölgesine ait en detaylı fakat en karmaşık bilgiler, göz bölgesi piksel değerlerinin içinde yer almaktadır. İris ve göz köşelerine ait bilgilerde, aslında bu değerler içinde gömülüdür. Fakat bu veri boyut olarak büyük bir veridir ve bu da verinin işlem süresini uzatan bir etkendir. Bu sebeple çalışmanın ilk aşamalarında her ne kadar iris ve göz köşe noktalarının bulunması üzerine yoğunlaşmış ise de, ilerleyen aşamalarda piksel değerlerinin daha kolay ulaşılabilir olması ve otomatik elde edilen değerlerin kullanıma daha uygun olması sebebiyle, çalışma tamamen göz bölgesine ait piksel değerleri üzerine yoğunlaşmıştır.

Göz bölgesi piksel değerlerine, çalışmanın ilk aşamasında geliştirilen YSA ile ulaşılabilmektedir. Bunun için ayrıca bir yöntem kullanma ihtiyacı yoktur. Ancak 48×160 boyutlarında elde edilen göz bölgesinde çok fazla veri vardır. Bu verinin, veri azaltma yöntemleri ile azaltılması gerekmektedir. Çalışmada veri azaltma yöntemi olarak matlab ortamının da 3. derece dalgacık dönüşümü ve "imresize" komutları kullanılmıştır. Bu veri azaltma yöntemleri arasında belirgin bir fark görülmemesi sebebi ile "imresize" komutunun kullanılmasına karar verilmiştir. Bu komut kullanılarak göz bölgesi verileri önce 6×20 boyutlarında bir matrise daha sonra matlab "cat" komutu kullanılarak 1×120 vektör haline getirilip YSA eğitiminde kullanılması planlanmıştır.

Ancak kullanıcının görüntü içinde göz ve kafa pozisyonlarını değiştirmeden yatay yada dikey ekseninde hareket ettiği durumda, göz bölgesinin piksel değerlerinde değişme olmamasına rağmen ekranda baktığı noktanın yatay yada dikey eksenindeki hareketle doğru orantılı olarak yer değiştireceği aşikardır. Görüntü içinde göz bölgesinin başlangıç koordinatlarının, bu bilgiyi içerdiği düşünülerek öznitelik olarak kullanılması gerektiğine karar verilmiştir.

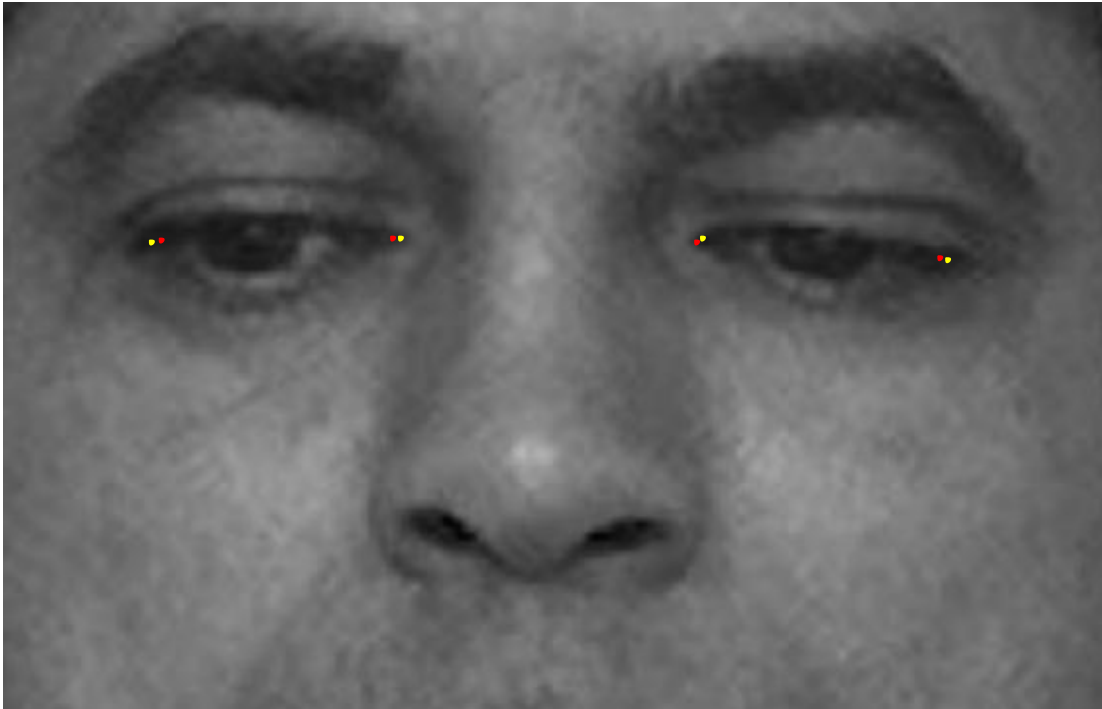
Böylece göz bölgesine ait veri sayısı göz bölgesinin başlangıç koordinatlarının ilavesi ile 1×122 adet olarak kullanılmıştır.

4.2.2.4. Göz bölgesine ait özniteliklerin çıkarılması

Göz bölgesine ait özniteliklerin, göz bölgesi içinde bulunan iris merkezi koordinatları, iris büyüklüğü ve göz köşe nokta koordinatları ile bu noktalar arasındaki ilişkiler olduğu düşünülerek, bu veriler öznitelik olarak belirlenmiştir.

Çalışma sürecinde özniteliklere, önce göz bölgesi başlangıç koordinatları daha sonra göz bölgesi piksel değerleri ilave edilmiştir. Bu aşamada çalışmaya piksel bazında devam edilmiş ve kullanıcı görüntüleri farklı bakış açılarında ve farklı ışık şiddetlerinde alınmıştır. Böylelikle çalışmada elde edilecek sonuçların daha geniş kapsamlı bir kullanım alanına sahip olması hedeflenmiştir. Ancak çalışmada yeterli sonuçlara ulaşılamamıştır. Özellikle farklı bakış açıları altında görüntü alma ağı

kestirim yapmasını zorlaştırmakta ve başarı oranını düşürmektedir. Ayrıca göz köşelerinin görüntü içinde bir nokta olarak ifade edilmeside başarı oranını büyük oranda düşürmektedir. Çünkü göz köşesi olarak aynı görüntüde farklı noktalar doğru olarak kabul edilebilmektedir. Şekil 4.20'de doğru kabul edilebilecek ancak birbirinden farklı koordinatlara sahip göz köşe örneği verilmiştir. Oysa kırmızı ve sarı renklerle ifade edilen göz köşe koordinatları arasında, sol göz dış köşe noktası için x ekseninde 6, y ekseninde 1 piksel, sol göz iç köşe noktası için x ekseninde 4, y ekseninde 0 piksel, sağ göz iç köşe noktası için x ekseninde 4, y ekseninde 2 piksel, sağ göz dış köşe noktası için x ekseninde 5, y ekseninde 1 piksel fark vardır. Bu fark kullanılan görüntü çözünürlüğüne göre farklılık göstermektedir. Literatürde göz köşelerinin bulunmasına yönelik çalışmalar olmasına rağmen Şekil 4.20'deki gibi durumlarda kırmızı renkle bulunan göz köşelerinin mi yoksa sarı renkle bulunan göz köşelerinin mi tercih edileceği noktasında bir kriter geliştirilmemiştir. Buda göz köşelerinin bakış açısının tespiti amacıyla kullanılmasında başarı oranını olumsuz etkilemektedir.



Şekil 4.20. Aynı görüntüde göz köşelerinin farklı bulunması

Bu sebeplerle çalışma sınırlandırılmıştır. Bu sınırlandırma üç aşamada yapılmıştır. İlk olarak bakış açısı ve ışık şiddeti gibi ortam değişkenlerini bertaraf etmek

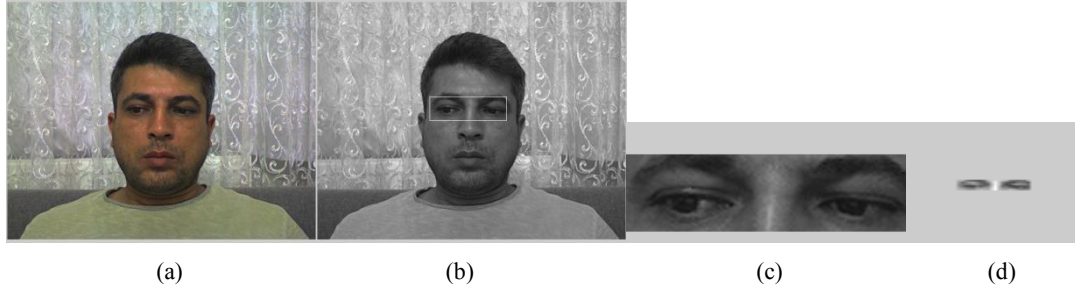
amacıyla, görüntüler benzer ışık şiddeti ve benzer bakış açılarında sınırlandırılarak alınmıştır. Daha sonra bölge temelli sonuçların yeterli olacağı düşünülerek ekran 24×24 piksel boyutlarında bölgelere ayrılmıştır. Son aşamada, öznitelik olarak belirlenmiş olan göz köşelerinin göz bölgesi şeklinde ifade edilmesinin daha doğru olacağı ve bunun da göz bölgesi içinde yer alıyor olması sebebiyle, göz bölgesi piksel değerleri ve göz bölgesi başlangıç koordinatlarının eğitim veri seti hazırlanmasında yeterli olacağı kararlaştırılmıştır. Böylece bölge temelli çalışma esnasında, her bir görüntü için göz bölgesi piksel değerlerini ve göz bölgesi başlangıç koordinatlarını içeren 1×122 boyutlarına sahip öznitelikler kullanılmıştır.

4.2.3. Koordinat tespiti ve imlecin taşınması

İkinci aşamada bilgisayar ekranına uygun olarak hazırlanan, eğitim ve test bölgelerini içeren görüntü arka plan olarak kullanılmış ve her bir bölgeye ait ilk aşamada geliştirilen YSA'nın göz bölgesini doğru olarak bulduğu 20 adet görüntü alınmıştır. Bu aşamada kullanılan görüntüler bir kullanıcıya aittir. Bu görüntüler kullanıcı belirlenmiş bölgelere bakarken sabitleme (fixation) anlarında kendisi tarafından alınmıştır. Gözün, sıçrama (saccade) anlarında görüntü alınmamıştır. Çünkü sıçrama anında gözün her ne kadar belirli iki nokta arasında hareket ettiği bilirse de nereye bakıldığı kestirilemez. Sıçrama anlarında alınacak görüntülerin koordinat tespitinde kullanılamaması sebebiyle gözün sıçraması anında görüntü alınmamıştır [34].

Eğitim setinin hazırlanmasında, önce görüntülerden göz bölgeleri elde edilmiş ve bu görüntüler, 1×120 boyutlarında vektör matrisi haline getirilerek kullanılmıştır. Bununla birlikte göz bölgelerinin başlangıç koordinatlarının da elde edilen veriye ilave edilmesiyle her bir görüntü için toplam 1×122 boyutlarına sahip veri elde edilmiştir. Test görüntülerinden elde edilen veriler de aynı aşamalardan geçirilerek test veri seti hazırlanmıştır [34].

Eğitim setinin hazırlanmasında kullanılan kullanıcı görüntüleri ve bu görüntülerden elde edilmiş göz bölgeleri ile azaltılmış veriye ait görüntüler Şekil 4.21'de verilmiştir.



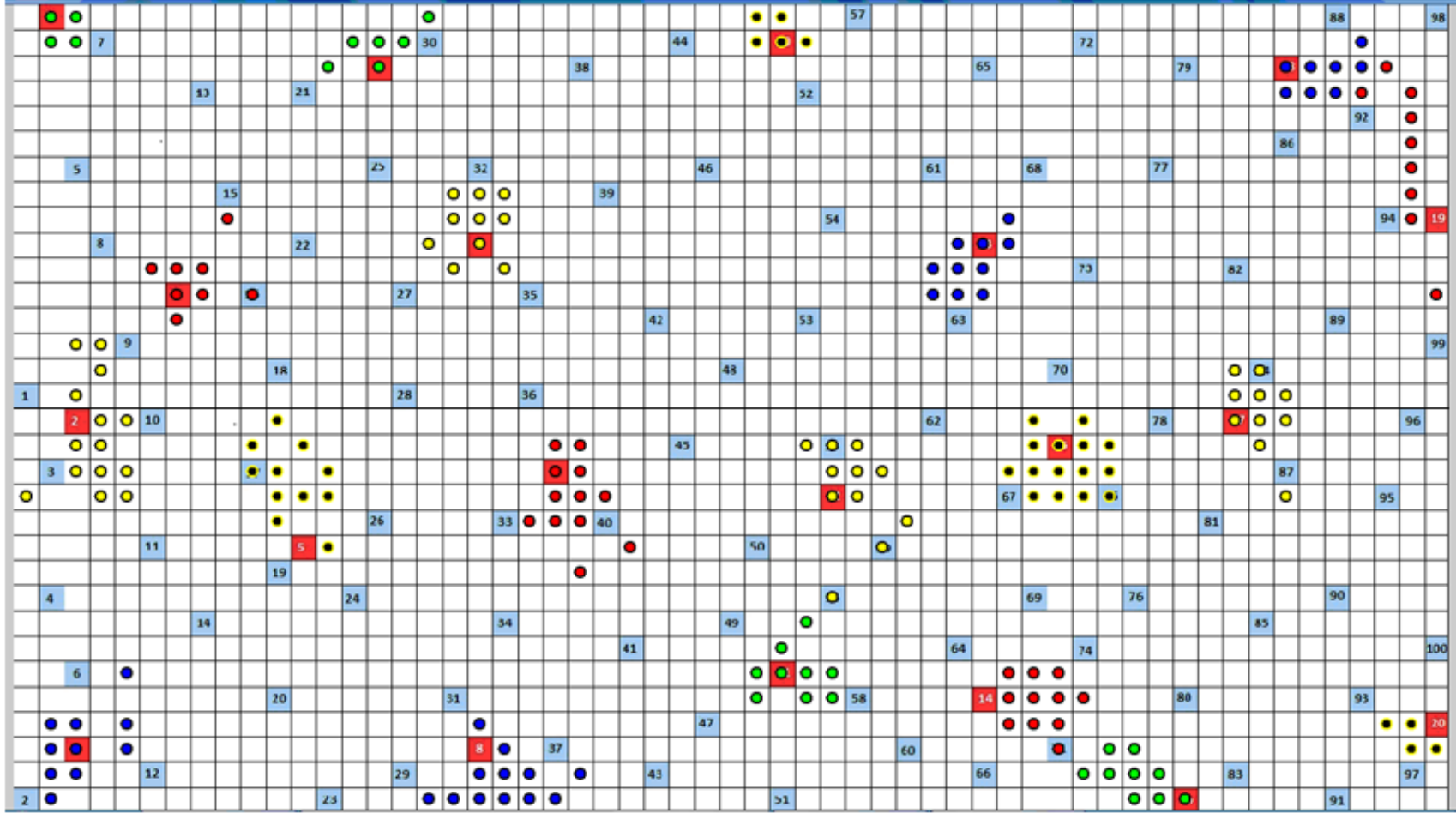
Şekil 4.21. (a) Eğitim setinde kullanılmak üzere alınmış bir görüntü (b) göz bölgesi bulunmuş görüntü (c) göz bölgesi (d) matlab ortamında boyutları küçültülmüş göz bölgesi[34]

Eğitimde YSA oluşturulurken ileri beslemeli geri yayımlı ağ modeli kullanılmıştır. Oluşturulan YSA 2 ara, 1 çıkış katmanından oluşmaktadır. 1. ara katmanda 90 adet, 2. ara katmanda 9 adet, çıkış katmanında 2 adet nöron kullanılmıştır (EK12). Ağ parametreleri, ağın tekrar tekrar eğitilip test edilmesi sonucu elde edilen test sonuçlarının başarıım yüzdesine göre belirlenmiştir. Hazırlanan ağa her bir giriş için 122 değerden oluşan öznitelik vektörü uygulanmıştır. Çıkışta girişin ifade ettiği bölge x (1-57) ve y (1-32) koordinatları olarak belirlenmiştir. Giriş ve çıkış değerleri eğitim ve test esnasında 0-1 aralığına ölçeklenerek kullanılmıştır. Koordinat tespiti için geliştirilen yapay sinir ağının özellikleri Tablo 4.7'de verilmiştir.

Tablo 4.7. Koordinat tespiti için kullanılan yapay sinir ağının parametreleri

Giriş seti	2000 görüntü verisi giriş vektörleri oluşturulurken kullanılmıştır.
Ağ modeli	İleri beslemeli geri yayımlı
Katman sayısı	2 ara katman 1 çıkış katmanı
Nöron sayısı	1. Ara katman 90 adet, 2. Ara katman 9 adet, Çıkış katmanı 2 adet
Aktivasyon fonksiyonu	Ara katmanlarda ve çıkış katmanında tanjant sigmoid
Öğrenme algoritması	trainscg
Öğrenme katsayısı	0.6
Momentum katsayısı	0.2

YSA eğitildikten sonra önceden hazırlanmış test seti kullanılarak test edilmiş ve her bir test girişi için çıkışta bakılan bölgelerin tahmini koordinatları tespit edilmiştir. YSA kullanılarak elde edilen bu bölge koordinatları ile test görüntülerinin çekiminde kullanılan test bölgeleri Şekil 4.22'de verilmiştir.



Şekil 4.22. Koordinat tespiti için eğitilen ağı test verilerine uygulanması ile elde edilen sonuçlar (a) kırmızı renkli kare bölgeler test bölgeleridir (b) test bölgelerinin etrafında daire şeklinde farklı renklerle gösterilen bölgeler ise YSA tarafından tahmin edilen bölgelerdir [34]

Her test bölgesi için 20'şer adet, toplamda 400 adet test sonucu elde edilmiştir. Test sonuçlarının ekran görüntüsü üzerine basıldığı görüntüde, kırmızı bölgeler test bölgelerini ifade etmektedir. Bu görüntüde test sonuçları ise, her test bölgesi için mümkün olduğunca farklı renklerde daireler şeklinde görüntü üzerine temsil ettiği bölgenin merkezine basılmıştır.

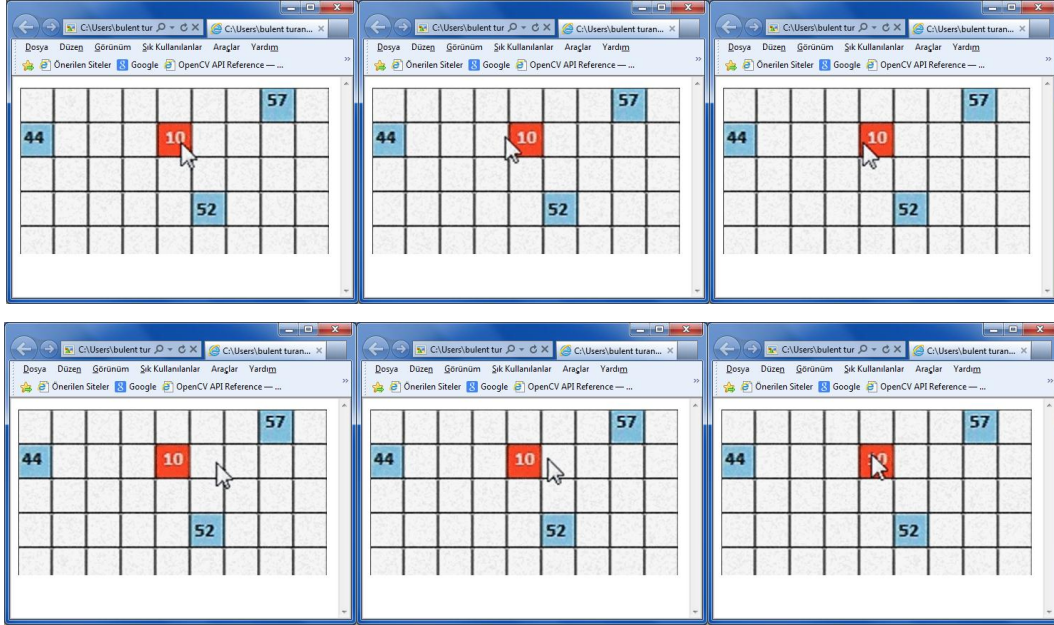
20 adet test bölgesinin gerçek koordinatları ve bu 20 test bölgesine bakarken çekilen 400 adet test görüntüsüne ait verilerin geliştirilen YSA ile işlenmesi sonucu elde edilen bölge koordinatları, ayrıca her test bölgesi için ayrı ayrı hesaplanan mutlak hata ortalaması (MAE) Tablo 4.8'de verilmiştir.

Tablo 4.8 incelendiğinde 6. ve 10. Test bölgesine ait mutlak hata ortalamalarını 1 bölgenin (1×24 piksel) altında, 19. Bölgeye ait mutlak hata ortalamasının ise 3 bölgenin (3×24 piksel) üstünde olduğu ve diğer bölgelere ait mutlak hata ortalamalarının bu değerlerin arasında olduğu görülmektedir. Ayrıca Tablo 4.7'ye göre test bölgeleri için hesaplanan mutlak hata ortalamalarının ekranın belirli bölgeleri (Ekranın üstü, sağ, solu, altı, ortası, kenarları gib) için farklılıklar göstermediği görülmektedir.

Tablo 4.8. YSA ile işlenen test görüntüsü verilerinin sonuçları

TEST BÖLGELERİ	GERÇEK BÖLGE KOORDİNATI		BULUNAN BÖLGE KOORDİNATLARI																				MAE
			01. Test Görüntüsü	02. Test Görüntüsü	03. Test Görüntüsü	04. Test Görüntüsü	05. Test Görüntüsü	06. Test Görüntüsü	07. Test Görüntüsü	08. Test Görüntüsü	09. Test Görüntüsü	10. Test Görüntüsü	11. Test Görüntüsü	12. Test Görüntüsü	13. Test Görüntüsü	14. Test Görüntüsü	15. Test Görüntüsü	16. Test Görüntüsü	17. Test Görüntüsü	18. Test Görüntüsü	19. Test Görüntüsü	20. Test Görüntüsü	
1. Test Bölgesi	X	2	3	3	2	3	3	3	3	2	2	2	2	2	2	2	3	2	3	2	2	1.0743	
	Y	1	2	2	2	2	2	1	2	2	2	2	1	2	2	2	2	1	2	2	2		
2. Test Bölgesi	X	3	3	4	4	4	3	4	3	3	1	4	4	4	4	4	5	5	5	5	3	4	2.2242
	Y	17	16	15	14	14	14	20	19	18	20	18	18	19	18	18	19	20	17	19	19	17	
3. Test Bölgesi	X	3	5	3	2	2	3	3	3	2	3	3	5	2	3	5	3	3	3	3	3	2	1.1660
	Y	30	30	30	29	29	31	31	31	30	29	29	29	32	30	27	31	29	30	31	30	31	
4. Test Bölgesi	X	7	8	10	9	8	8	9	8	7	8	7	7	7	7	7	6	7	7	7	8	1.1934	
	Y	12	12	12	9	12	12	9	11	12	11	11	12	11	12	12	11	11	11	13	11		
5. Test Bölgesi	X	12	11	12	10	12	11	11	13	12	12	12	13	11	13	10	11	11	11	12	11	11	2.7710
	Y	22	20	20	19	20	20	20	22	20	18	20	20	19	19	18	21	20	19	18	19	17	
6. Test Bölgesi	X	15	16	15	15	13	15	15	15	15	15	17	15	14	15	15	15	15	15	16	16	15	0.7743
	Y	3	2	2	3	3	3	3	3	3	3	1	2	2	2	3	3	2	2	2	2	3	
7. Test Bölgesi	X	19	18	18	19	19	19	20	19	19	19	19	17	19	19	19	20	20	19	18	18	19	1.3272
	Y	10	9	11	9	9	9	8	8	8	9	8	10	10	10	9	11	9	8	8	9	10	
8. Test Bölgesi	X	19	21	21	19	21	18	20	19	18	21	20	20	21	20	20	23	19	22	21	17	23	2.3208
	Y	30	31	32	31	32	32	32	32	32	32	31	31	32	31	30	31	29	32	31	32	31	
9. Test Bölgesi	X	22	25	24	23	23	23	22	23	22	21	23	23	22	23	22	22	23	23	23	23	22	1.6987
	Y	19	22	20	20	20	18	21	20	20	21	21	23	21	18	18	20	18	19	18	19	19	
10. Test Bölgesi	X	31	31	31	30	31	30	31	30	32	30	31	30	32	31	31	30	31	31	31	30	30	0.7536
	Y	2	2	1	1	2	2	1	1	2	1	2	2	2	2	2	2	2	2	1	1	1	
11. Test Bölgesi	X	21	32	32	32	32	31	32	32	31	33	31	33	31	30	32	32	32	33	30	32	30	1.2683
	Y	27	27	27	25	27	26	25	27	27	27	26	28	27	28	28	27	27	27	28	28	27	
12. Test Bölgesi	X	33	32	34	34	33	34	33	33	33	33	33	33	33	32	33	34	35	33	35	36	34	1.6793
	Y	20	18	20	20	18	18	19	20	20	18	20	19	19	18	18	18	19	24	22	21	19	
13. Test Bölgesi	X	39	39	39	37	39	38	38	38	38	37	39	39	38	39	39	38	40	39	40	40	39	1.2686
	Y	10	12	11	12	11	12	10	11	11	11	11	11	11	11	10	9	10	10	9	10	10	
14. Test Bölgesi	X	39	41	42	41	40	40	41	41	42	40	41	40	42	42	42	41	42	40	41	43	41	2.3701
	Y	28	27	29	27	28	27	29	28	28	27	27	27	29	28	30	29	27	29	29	28	29	
15. Test Bölgesi	X	42	43	44	41	44	44	42	43	43	42	42	41	43	42	41	42	42	41	40	42	41	1.4422
	Y	18	20	18	18	19	20	19	19	17	18	19	20	18	18	19	20	19	19	19	19	17	
16. Test Bölgesi	X	47	46	45	47	44	45	47	45	46	46	46	45	46	43	45	45	43	44	46	46	44	2.0392
	Y	32	32	30	32	30	31	32	30	31	32	32	31	31	31	31	32	31	31	32	31	31	
17. Test Bölgesi	X	49	51	50	51	50	50	50	51	49	50	50	51	50	50	50	50	49	49	50	49	49	1.5725
	Y	17	20	17	17	15	16	16	16	16	16	17	16	16	16	15	17	16	16	15	18	17	
18. Test Bölgesi	X	51	52	52	52	52	51	53	51	53	54	54	53	55	55	54	54	52	55	52	52	52	2.0528
	Y	3	3	4	4	4	3	3	4	4	3	3	3	3	3	3	2	4	3	3	3	3	
19. Test Bölgesi	X	57	56	56	56	56	56	56	56	57	56	56	56	56	55	56	56	56	54	56	55	55	3.0750
	Y	9	8	6	7	8	9	9	7	12	7	9	8	5	3	5	7	4	4	9	3	3	
20. Test Bölgesi	X	57	56	56	55	56	56	56	56	56	56	56	56	56	56	57	56	56	56	56	55	56	1.2243
	Y	29	29	29	29	29	29	29	30	30	30	30	29	30	29	30	29	30	29	29	29	29	

Test sonucunda her bir bölge için elde edilen 20 farklı sonuç ile Matlab “pointerlocation” komutu kullanılarak imleç kontrolü gerçekleştirilmiştir. 10 numaralı test bölgesi görüntülerinden elde edilen sonuçlar ile gerçekleştirilen imleç kontrolü Şekil 4.23’de verilmiştir.



Şekil 4.23. 10 numaralı test bölgesine ait imleç kontrol örnekleri

4.2.4. Farklı kullanıcılar ile çalışmanın tekrarı

Çalışmanın farklı kullanıcılar ile tekrarlanabilmesi için farklı kullanıcı görüntüleri ile hazırlanmış veri setine ihtiyaç vardır. Ancak bu amaçla geliştirilmiş bir veri seti olmaması sebebiyle araştırmacı farklı kullanıcılardan görüntüler toplayarak 11 denek için ayrı birer veri seti oluşturmuştur. Araştırmanın farklı kullanıcılar ile tekrar edilebilmesi amacıyla her bir kullanıcı için benzer şekilde veri setlerinin hazırlanması gerekmektedir. Veri setlerinin mümkün olduğunca az hatayla hazırlanabilmesi amacıyla bir arayüz programı geliştirilmiştir. Arayüz programı, hem denekleri yönlendirmekte hemde görüntü alma, kaydetme gibi işlemleri basitleştirmektedir. Ayrıca deneklerin kendi bilgisayarları ile çalışmalarını, dolayısı ile de zamandan ve mekândan bağımsız çalışabilmelerine olanak sağlamaktadır. Ancak deneklerin kullandığı bilgisayarların 15,6" ekran büyüklüğüne sahip olması ve 1366×768 ekran çözünürlüğünde çalışılması gerekmektedir.

Çalışmada bahsedilen özelliklerde bilgisayara sahip, gönüllü 12 öğrenci denek olarak belirlenmiştir. Bu deneklerden 8 tanesi göz bölgesinin bulunması amacıyla kullanılan YSA'nın geliştirilmesi aşamasında görev almışlardır. Daha önceki aşamalarda kullanıcının göz bölgesinin bulunması amacıyla geliştirilmiş olan YSA modeli deneklerden alınan görüntüler kullanılarak deneklerin göz bölgelerini bulması için güncellenmiştir. Bu aşamada her denek için ayrı bir YSA geliştirilmemiştir. İlk geliştirilen YSA, kullanıcı görüntüleri ile veri seti genişletilerek güncellenmiştir. Göz bölgesinin bulunması aşamasında, belirlenen YSA parametreleri bu aşamada değiştirilmeden kullanılmıştır (EK12). Yani tek bir YSA araştırmacının ve deneklerin göz bölgelerini bulabilmektedir. Geliştirilen YSA, çalışmaya sonradan katılmak isteyen denekler için güncellemeye gerek kalmadan onlarında göz bölgelerini bulabilmektedir. Bu aşama tamamlandıktan ve deneklerin göz bölgelerinin YSA ile gerçek zamanlı bulunabildiği test edildikten sonra ikinci aşamaya geçilmiştir. Göz bölgesini bulan YSA'nın güncellenmesi için deneklerden alınan görüntü örnekleri Şekil 4.24'de verilmiştir. Göz bölgelerinin her ortamda bulunabilmesi amacıyla, bu aşamada görüntü alınırken ışık şiddeti, bakış açısı gibi değişkenler önemsenmemiştir.



Şekil 4.24. Göz bölgelerinin bulunması için deneklerden alınan görüntü örnekleri

Çalışmada denekler ile yapılan ikinci aşama, deneklerin test bölgelerine bakarken görüntülerinin alınması aşamasıdır. Bu aşama için, öncelikle hazırlanan arayüz programı denek bilgisayarlarına yüklenmiş ve sorunsuz çalışıp çalışmadığı kontrol edilmiştir. Arayüzün yüklenmesi aşamasında, ayrıca deneklere programın nasıl çalıştığı ve görüntü alma aşamasında kendilerinin neler yapmaları gerektiği anlatılmış ve örnek görüntüler alınmıştır. Göz bölgesinin bulunması için geçek zamanlı kullanılan YSA, araştırmacının ayrık zamanlı görüntüleri ile yapılan test aşamasında olduğu gibi %100 doğru sonuca ulaşamamaktadır. Bu sorun göz bölgesinin yanlış bulunduğu görüntüler ve görüntü verileri kayıttan önce denekler tarafından silinmesi sağlanarak giderilmiştir. Arayüz programı görüntüyü ve bulunan göz bölgesi piksel değerlerini kaydedecek şekilde tasarlanmıştır. Böylece veriler görüntü kullanmadan otomatik olarak kaydedilip kullanılmaktadır. Kaydedilen görüntüler ise sadece kontrol amacıyla değerlendirilmektedir.

Çalışmanın bu aşamasında 1 denek çalışmadan ayrılmış 4 denek ise çalışmaya sonradan dâhil olmuştur. 8 erkek ve 3 kız olmak üzere toplam 11 denek ile

çalışmanın bu aşaması tamamlanmıştır. Bu aşamada deneklerden alınan görüntü örnekleri Şekil 4.25’de verilmiştir.



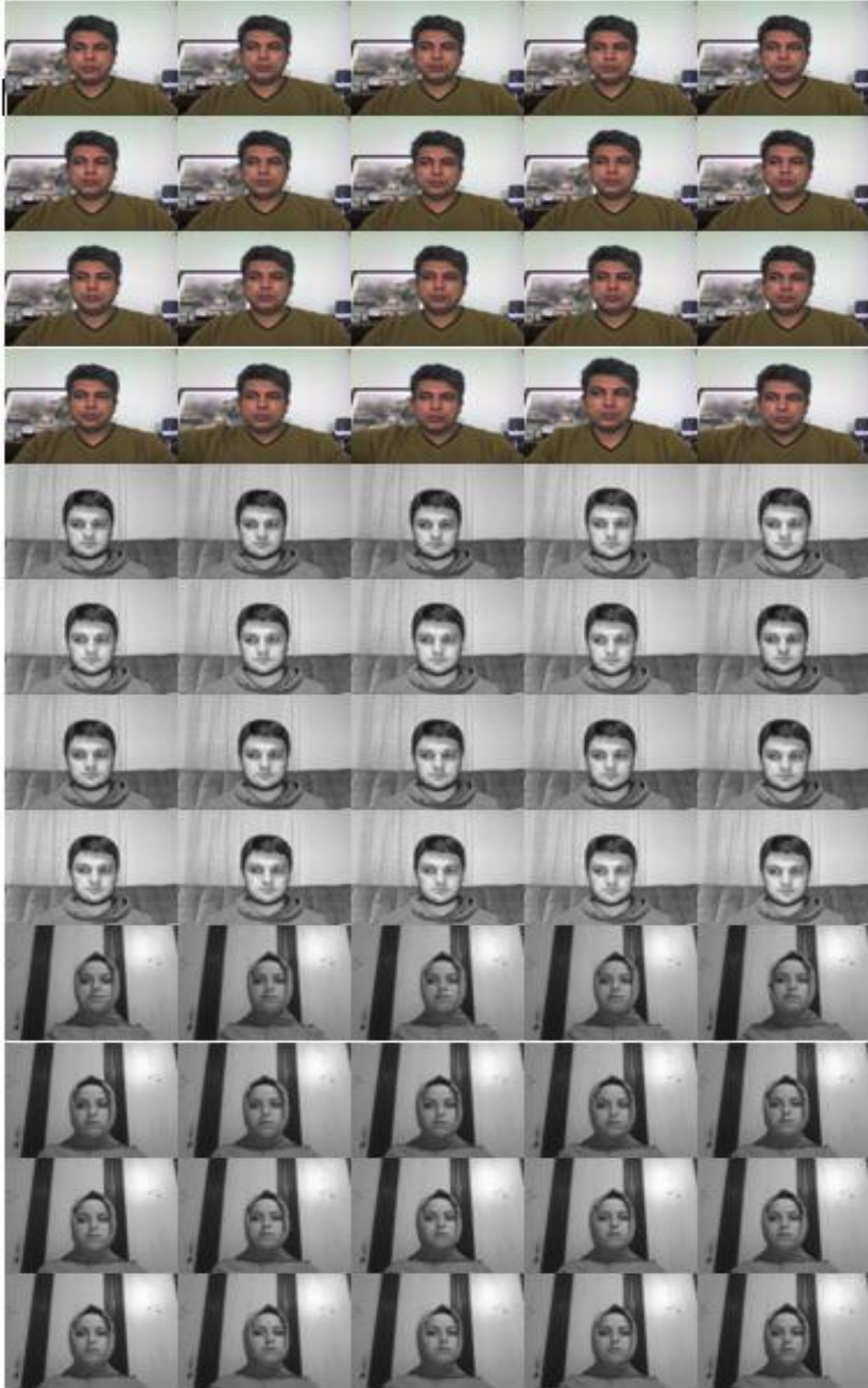
Şekil 4.25. Tüm denekler 75 nolu eğitim bölgesine bakarken alınan birer görüntü

Deneklerin görüntü alma işlemi tamamlandıktan sonra, YSA eğitime ve test aşamalarına geçilmiştir. Deneklerin eğitim ve test aşamalarında kullanılan YSA’lar EK12’de verilmiştir. Her denek için eğitim ve test süreçleri ayrı ayrı tekrarlanmıştır.

Çalışma boyunca kullanılan görüntü örnekleri Şekil 4.26 ve Şekil 4.27’de verilmiştir. Bu görüntüler kullanılarak geliştirilen YSA’ların uygulandığı test görüntülerinden elde edilen sonuçlar Şekil 4.28 ve Şekil 4.29’da verilmiştir.

Deneklerden elde edilen sonuçlar incelendiğinde, mutlak hata oranlarının farklılıklar gösterdiği, en düşük mutlak hata ortalamasının arařtırmacının elde ettiđi mutlak hata ortalamasına yaklařtıđı görölmektedir. Denek görüntüleri incelendiğinde, elde edilen hata ortalamaları arasında bu řekilde fark olmasının ve deđerlerin arařtırmacının elde ettiđi deđerlerden yüksek olmasının nedenlerinin, görüntülerin alınması esnasında deneklerin kafa pozisyonlarının deđiřmesi olduđu kanısına varılmıřtır. řekil 4.26 ve řekil 4.27’de arařtırmacı ve deneklere ait en iyi ve en kötü sonuçların elde edildiđi 5-85 nolu eđitim bölgesine bakarken çekilmiř görüntüler verilmiřtir. Bu görüntüler incelendiğinde, arařtırmacı ve ilk deneđin görüntüler alınırken bakıř ađısının deđiřmediđi, ikinci deneđin ise bakıř ađısının deđiřtiđi ađıkça görölmektedir. řekil 4.28 ve řekil 4.29’da alıřmanın farklı kullanıcılar için tekrarlanması sonucu deneklerden elde edilen en iyi ve en kötü sonuçlar verilmiřtir.

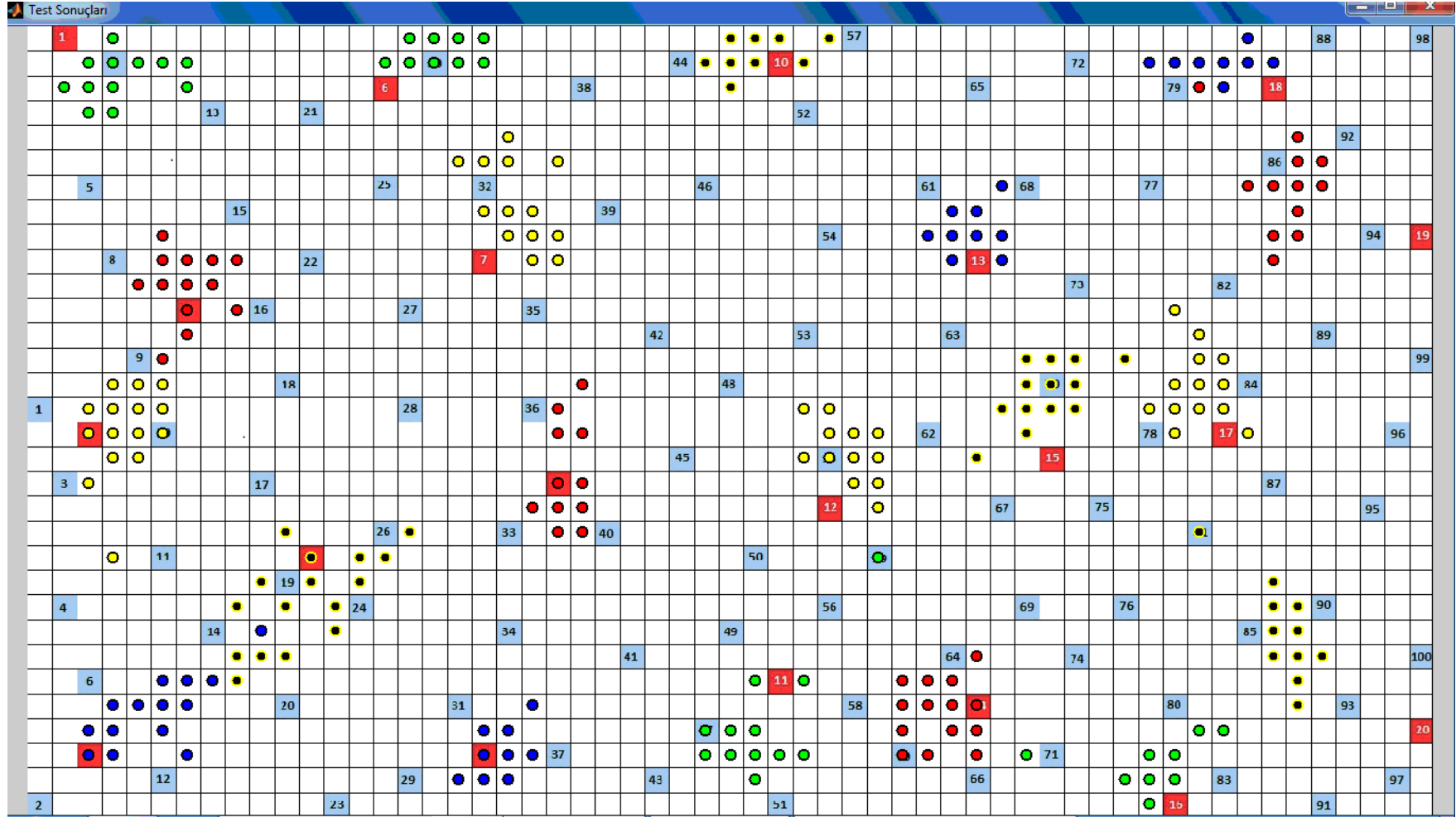
20 adet test bölgesinin gerek koordinatları ve bu 20 test bölgesine bakarken çekilen 5. ve 7. deneđin test görüntülerine ait verilerin geliřtirilen YSA ile iřlenmesi sonucu elde edilen bölge koordinatları, ayrıca her test bölgesi için ayrı ayrı hesaplanan mutlak hata ortalaması (MAE) Tablo 4.9 ve Tablo 4.10’da verilmiřtir.



Şekil 4.26. Araştırmacı ile en iyi ve en kötü sonuçların elde edildiği 5. ve 7. deneklerin 5 nolu eğitim bölgesine bakarken elde edilmiş görüntüleri



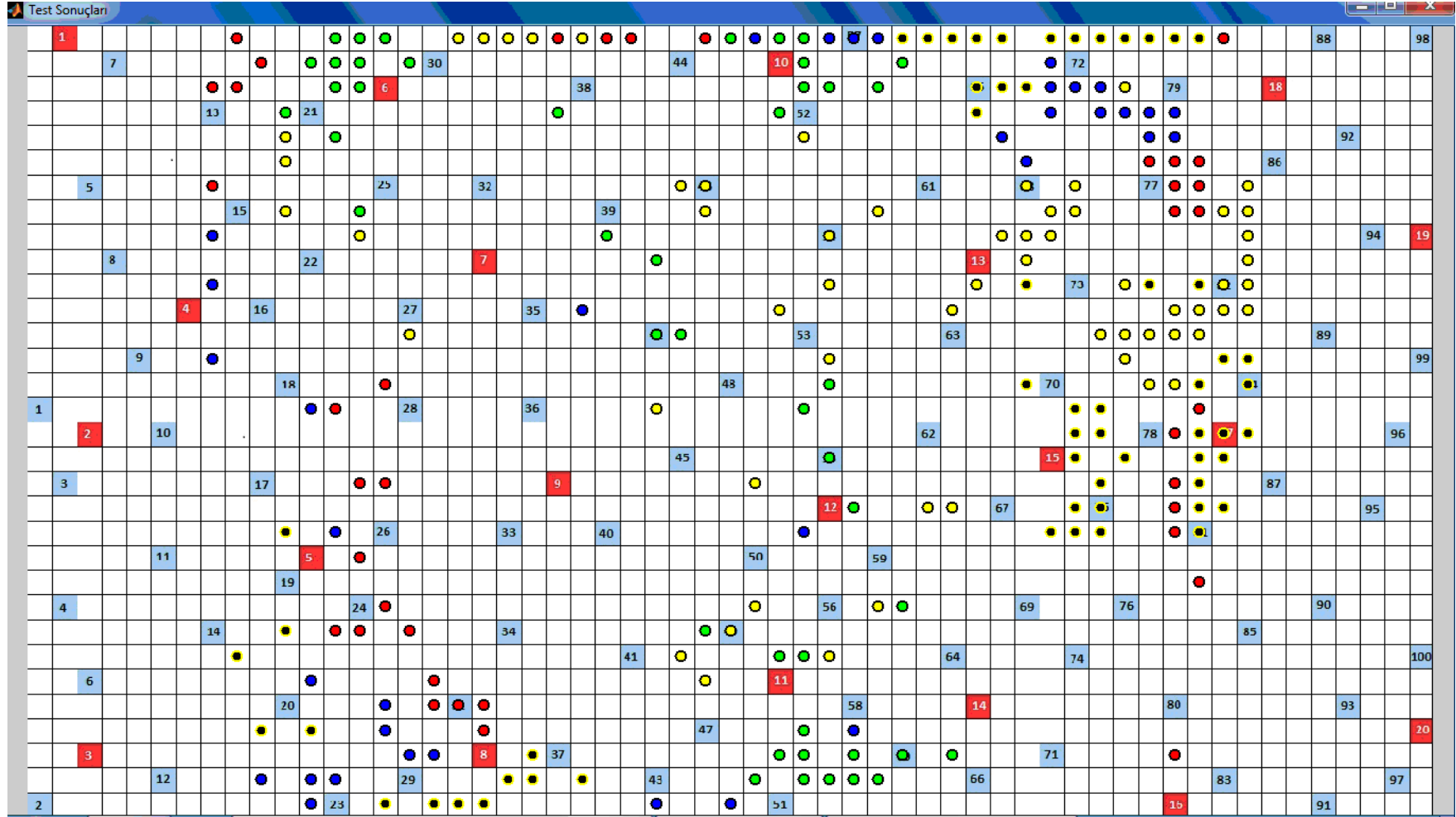
Şekil 4.27. Araştırmacı ile en iyi ve en kötü sonuçların elde edildiği 5. ve 7. deneklerin 85 nolu eğitim bölgesine bakarken elde edilmiş görüntüleri



Şekil 4.28. Deneklerden elde edilen en iyi sonuçlar

Tablo 4.9. YSA ile işlenen 5. deneğe ait test görüntüsü verilerinin sonuçları

TEST BÖLGELERİ	GERÇEK BÖLGE KOORDİNATI	BULUNAN BÖLGE KOORDİNATLARI																				MAE	
		01. Test Görüntüsü	02. Test Görüntüsü	03. Test Görüntüsü	04. Test Görüntüsü	05. Test Görüntüsü	06. Test Görüntüsü	07. Test Görüntüsü	08. Test Görüntüsü	09. Test Görüntüsü	10. Test Görüntüsü	11. Test Görüntüsü	12. Test Görüntüsü	13. Test Görüntüsü	14. Test Görüntüsü	15. Test Görüntüsü	16. Test Görüntüsü	17. Test Görüntüsü	18. Test Görüntüsü	19. Test Görüntüsü	20. Test Görüntüsü		
1. Test Bölgesi	X	2	4	4	3	2	3	2	3	3	4	7	4	3	4	7	3	4	6	3	4	5	2.5983
	Y	1	1	4	3	3	4	3	3	2	2	3	3	2	1	2	2	2	2	2	1	2	
2. Test Bölgesi	X	3	5	5	5	4	5	5	4	4	6	3	5	6	5	4	4	3	4	5	6	3	2.1234
	Y	17	16	17	17	17	17	16	15	17	16	17	15	15	18	22	16	16	18	17	17	19	
3. Test Bölgesi	X	3	4	7	7	5	4	3	7	7	6	6	3	6	7	4	8	10	4	5	4	3	3.0791
	Y	30	29	28	27	28	30	30	30	28	28	29	29	27	28	30	27	25	29	28	28	30	
4. Test Bölgesi	X	7	7	7	6	7	6	6	7	6	9	7	8	6	9	6	5	7	8	6	7	7	1.7751
	Y	12	11	10	11	11	9	11	12	9	12	11	11	10	10	9	11	11	10	14	13	11	
5. Test Bölgesi	X	12	13	11	12	9	16	13	14	11	10	11	16	9	11	12	14	9	15	10	12	13	2.8135
	Y	22	24	26	22	24	21	25	22	21	23	24	21	27	24	23	23	26	22	26	23	24	
6. Test Bölgesi	X	15	16	15	18	18	18	17	18	17	18	18	18	17	17	15	18	16	18	19	18	19	2.8760
	Y	3	2	2	6	2	2	2	2	2	2	2	2	1	1	2	1	1	2	1	2	2	
7. Test Bölgesi	X	19	21	21	20	20	20	19	20	20	19	21	18	21	21	21	20	22	21	22	21	22	2.8734
	Y	10	9	8	9	8	6	8	8	5	6	9	6	9	10	8	8	10	9	9	9	6	
8. Test Bölgesi	X	19	20	20	20	19	19	20	20	20	19	21	19	21	21	18	19	20	21	20	19	20	1.2071
	Y	30	31	31	31	30	31	30	30	30	30	30	30	30	28	31	29	30	28	29	30	29	
9. Test Bölgesi	X	22	23	22	23	22	23	22	23	23	23	21	22	22	22	22	22	22	22	22	22	23	1.6037
	Y	19	20	21	21	21	20	20	15	21	19	20	20	20	17	16	19	20	19	20	17	17	
10. Test Bölgesi	X	31	30	30	29	30	29	32	30	31	29	29	29	29	29	31	29	33	29	28	30	31	1.6679
	Y	2	2	2	1	2	2	2	2	1	2	2	2	3	1	1	2	1	2	2	1	1	
11. Test Bölgesi	X	21	29	31	30	32	30	30	31	32	29	30	28	35	30	29	30	30	30	30	30	28	2.9659
	Y	27	29	30	29	30	31	30	30	27	30	29	30	22	29	30	29	29	27	30	29	29	
12. Test Bölgesi	X	33	35	34	35	34	32	33	34	32	34	33	33	33	35	33	33	35	33	34	34	35	2.4660
	Y	20	17	17	19	19	18	18	19	16	18	18	17	17	18	16	18	19	17	19	19	20	
13. Test Bölgesi	X	39	40	37	38	37	38	38	38	39	38	37	39	38	38	39	40	40	39	38	38	39	1.8647
	Y	10	9	9	9	9	10	10	8	8	8	9	9	8	9	8	7	10	8	8	8	8	
14. Test Bölgesi	X	39	39	37	38	37	37	39	39	37	39	38	36	38	38	37	36	39	36	36	38	38	1.8651
	Y	28	28	28	28	27	28	29	28	27	26	29	29	27	29	30	28	30	27	30	29	29	
15. Test Bölgesi	X	42	41	39	45	43	43	42	41	42	42	41	42	40	42	42	42	43	42	42	43	41	3.1204
	Y	18	16	18	14	16	15	16	15	14	14	14	15	16	16	16	14	14	15	15	14	17	
16. Test Bölgesi	X	47	47	46	46	47	45	46	47	45	46	49	41	47	45	46	47	47	47	46	46	48	1.9554
	Y	32	31	31	31	30	31	31	31	31	30	29	30	31	31	31	31	30	31	32	31	29	
17. Test Bölgesi	X	49	47	48	48	46	47	47	47	49	47	49	48	48	48	50	49	47	48	49	48	49	2.3986
	Y	17	12	15	16	16	16	16	17	15	15	16	16	13	16	17	16	16	13	15	14	14	
18. Test Bölgesi	X	51	49	51	50	50	46	49	49	48	47	49	49	49	49	50	49	51	49	49	49	50	2.2258
	Y	3	3	2	2	2	2	3	2	2	2	2	2	2	2	2	3	2	3	2	2	1	
19. Test Bölgesi	X	57	51	52	51	52	51	52	52	52	53	52	52	52	52	50	52	51	51	51	48	53	5.9054
	Y	9	10	8	10	8	9	9	7	8	7	6	9	8	5	7	7	7	7	7	3	6	
20. Test Bölgesi	X	57	53	51	52	52	52	52	52	52	52	52	52	52	52	52	51	52	51	48	51	51	6.7802
	Y	29	26	24	24	28	25	26	25	27	25	25	25	24	26	26	25	25	26	21	23	24	



Şekil 4.29. Deneklerden elde edilen en kötü sonuçlar

Tablo 4.10. YSA ile işlenen 7. deneye ait test görüntüsü verilerinin sonuçları

TEST BÖLGELERİ	GERÇEK BÖLGE KOORDİNATI	BULUNAN BÖLGE KOORDİNATLARI																				MAE	
		01. Test Görüntüsü	02. Test Görüntüsü	03. Test Görüntüsü	04. Test Görüntüsü	05. Test Görüntüsü	06. Test Görüntüsü	07. Test Görüntüsü	08. Test Görüntüsü	09. Test Görüntüsü	10. Test Görüntüsü	11. Test Görüntüsü	12. Test Görüntüsü	13. Test Görüntüsü	14. Test Görüntüsü	15. Test Görüntüsü	16. Test Görüntüsü	17. Test Görüntüsü	18. Test Görüntüsü	19. Test Görüntüsü	20. Test Görüntüsü		
1. Test Bölgesi	X	2	16	32	35	32	36	32	14	15	15	14	13	13	13	15	14	31	13	33	13	14	18.6781
	Y	1	2	3	3	2	2	2	1	1	1	1	1	1	1	1	1	4	2	3	2	2	
2. Test Bölgesi	X	3	30	27	29	28	30	39	38	41	40	41	27	28	28	27	42	36	38	37	14	16	29.9682
	Y	17	19	26	25	27	24	11	12	9	9	7	7	7	8	7	4	24	20	20	9	13	
3. Test Bölgesi	X	3	29	26	8	8	8	13	12	14	14	14	12	14	15	15	16	17	12	10	13	12	14.1599
	Y	30	32	32	9	11	14	21	16	25	25	22	27	25	28	29	30	30	32	31	31	31	
4. Test Bölgesi	X	7	15	13	14	15	15	14	13	14	16	18	14	19	17	17	19	9	8	8	9	8	13.1651
	Y	12	15	16	19	24	19	22	25	25	25	28	25	28	28	27	29	1	3	3	3	7	
5. Test Bölgesi	X	12	11	11	9	10	12	39	39	40	39	15	41	19	17	18	42	21	21	20	21	23	16.7798
	Y	22	21	25	26	29	29	4	3	3	3	32	3	32	32	32	3	31	30	31	31	31	
6. Test Bölgesi	X	15	28	31	35	36	32	13	12	13	14	14	13	11	13	14	13	24	26	27	26	22	11.4102
	Y	3	25	26	24	24	26	2	2	2	3	3	2	4	3	8	5	9	10	13	13	4	
7. Test Bölgesi	X	19	43	26	45	47	45	45	33	35	32	35	33	33	31	33	43	42	41	41	42	42	20.3471
	Y	10	8	16	4	4	4	3	26	24	5	8	11	9	12	14	7	8	11	10	9	9	
8. Test Bölgesi	X	19	34	23	48	31	31	32	31	31	32	37	35	34	36	36	36	32	34	34	35	33	31.0197
	Y	30	29	12	11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
9. Test Bölgesi	X	22	22	24	21	23	25	25	11	10	10	10	10	28	49	49	45	45	44	45	38	37	22.8567
	Y	19	1	1	1	1	1	1	5	2	2	2	2	1	1	1	1	1	1	1	1	1	
10. Test Bölgesi	X	31	36	36	37	38	39	40	45	44	43	43	42	44	46	47	47	47	47	48	48	48	12.2978
	Y	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
11. Test Bölgesi	X	21	31	31	32	31	32	31	31	31	32	31	30	31	30	30	31	30	31	30	30	29	26.0125
	Y	27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
12. Test Bölgesi	X	33	11	11	19	18	11	19	20	11	18	20	21	21	23	20	20	50	50	50	49	50	23.0909
	Y	20	8	6	1	1	6	1	1	5	1	1	1	1	1	1	1	9	7	8	8	7	
13. Test Bölgesi	X	39	44	44	43	42	42	42	42	42	42	42	43	44	30	30	30	32	30	41	40	42	9.0056
	Y	10	3	3	3	3	2	3	2	2	2	2	3	3	1	1	1	21	1	6	5	4	
14. Test Bölgesi	X	39	43	48	47	48	48	47	48	47	47	47	47	47	47	48	48	48	47	48	47	48	12.0291
	Y	28	18	20	21	20	19	19	18	20	20	19	19	20	17	17	16	19	30	18	20	23	
15. Test Bölgesi	X	42	43	41	41	41	43	43	44	44	43	44	42	43	44	44	43	44	43	44	44	45	2.7361
	Y	18	16	15	11	15	16	17	16	16	17	16	21	18	17	19	20	21	21	20	17	18	
16. Test Bölgesi	X	47	34	33	32	33	33	32	32	31	31	32	30	34	34	34	34	35	33	36	34	38	15.3942
	Y	32	20	18	16	15	18	29	30	30	30	31	31	30	30	31	31	31	31	30	31	30	
17. Test Bölgesi	X	49	50	49	50	50	50	49	48	48	47	47	48	46	45	47	44	47	45	45	45	46	5.3788
	Y	17	12	11	11	10	10	12	12	13	15	13	11	15	13	12	13	12	14	14	11	13	
18. Test Bölgesi	X	51	45	45	45	45	46	44	44	45	45	45	45	46	47	47	46	47	47	47	47	47	5.3974
	Y	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	5	4	5	4	4	5	
19. Test Bölgesi	X	57	45	46	47	47	46	47	47	47	47	47	48	48	48	48	48	48	47	47	48	48	9.7945
	Y	9	4	4	4	4	5	4	5	4	4	5	8	7	8	8	8	8	6	7	7	7	
20. Test Bölgesi	X	57	50	50	50	49	46	48	48	48	49	48	48	48	48	48	48	48	48	49	49	49	15.3238
	Y	29	17	15	14	14	11	15	11	15	17	17	20	21	15	18	17	20	19	14	20	18	

BÖLÜM 5. SONUÇLAR VE ÖNERİLER

5.1. Sonuçlar

5.1.1. Göz bölgesinin bulunmasına yönelik sonuçlar

Çalışmada göz bölgesi özniteliklerinin elde edilebilmesi amacıyla görüntü içinde göz bölgesinin bulunması için bir YSA modeli geliştirilmiştir. Bu YSA modelinde önce, araştırmacının kendi görüntülerini kullanılarak YSA eğitimi gerçekleştirilmiş ve geliştirilen YSA, 20 adet test görüntüsüne uygulandığında %100 başarıya ulaşılmıştır.

Eğitilen YSA, özniteliklerin belirlenmesi ve koordinat tespiti aşamalarında (ayrık zamanlı ve gerçek zamanlı) başarıyla kullanılmıştır. YSA bu aşamalarda sadece araştırmacı görüntülerinde göz bölgesini bulması amacıyla kullanılmıştır.

Ayrıca çalışmanın farklı kullanıcılar ile tekrarlanması aşamasında geliştirilen ağ modeli, eğitim veri seti genişletilerek göz bölgesinin denek görüntülerinde de bulunabilmesi için yeniden eğitilmiş ve kullanılmıştır. Denek görüntülerinde de başarıyla göz bölgesinin bulunduğu görülmüştür. Hatta eğitim esnasında çalışmaya sonradan katılmaları sebebiyle görüntüleri kullanılmayan 4 deneye ait görüntülerde de göz bölgesi başarıyla tespit edilmiştir. Böylece bu 4 denek için de test sonuçları elde edilmiştir.

5.1.2. Özniteliklerin çıkarılmasına yönelik sonuçlar

İrisi bulmak amacıyla göz bölgesi bulunan görüntüye CHT uygulanmış fakat düşük bir başarı oranı elde edilmiştir. Başarının artırılabilmesi için görüntü içinde irisin aranacağı bölgenin daraltılması amacıyla sol ve sağ iris bölgeleri için ayrı ayrı yapay sinir ağları geliştirilmiş ve daha sonra tekrar CHT uygulanmıştır.

Çalışmanın bir sonraki aşamasında göz köşelerinin bulunması amaçlanmıştır. Ağırlıklı Varyans Bağdaştırma Fonksiyonu (WVPF) kullanılmış fakat istenen başarı oranına ulaşılamamıştır. Bu sebeple farklı bir yöntem kullanılmıştır. Kullanılan ikinci yöntemde de, önce göz köşesini içeren bölge belirlenerek farklı ön işlemlerden geçirilmiş ve bu bölge içindeki en düşük değerlikli pikselin yeri belirlenerek göz köşeleri bulunmaya çalışılmıştır. Bu yöntemin kullanılması ile başarı oranı arttırılmıştır.

Ayrıca bu aşamada görüntü içinde kullanıcıya ait göz bölgesi piksel değerleri öznitelik olarak alınmıştır. Bunun için, görüntü içinde göz bölgesi bulunduktan sonra matlab ortamında 6×20 boyutlarına düşürülmüş ve 1×120 vektör haline getirilmiştir.

Görüntü içinde göz bölgesi başlangıç koordinatlarının da bakılan bölge bilgisini taşıdığı düşünülmektedir. Çünkü kullanıcının bakış açısı değişmeden x ya da y ekseninde hareket etmesi sonucu ekranda baktığı noktada aynı eksende kayacaktır. Aynı zamanda web kamerasının aldığı görüntü içinde kullanıcı göz bölgesi aynı ekseninde yer değiştirecektir. Bu sebeple görüntü içinde göz bölgesi başlangıç koordinatları belirlenerek öznitelik olarak kullanılmıştır.

5.1.3. Koordinat tespiti ve imlecin taşınmasına yönelik sonuçlar

Koordinat tespiti için ileri beslemeli geri yayımlı (feedforward backpropagation) ağ yapısı kullanılarak, bilgisayar ekranında bakılan bölge koordinatları tespit edilmeye çalışılmıştır. Geliştirilen YSA'dan elde edilen koordinat bölgesi tahminleriyle gerçek koordinat bölgeleri Şekil 4.22'de ve Tablo 4.8'de verilmiştir. Şekil 4.13 ve Tablo 4.8 incelendiğinde YSA ile elde edilen bölge koordinatlarının, gerçek bölge koordinatlarına oldukça yakın olduğu görülmektedir.

Tespit edilen bölge koordinatlarının doğruluğu değerlendirilirken literatürde de genellikle kullanılan *RMSE* (Hata Kareleri Ortalamasının Karekökü), *MAPE* (Mutlak Hata Oranları Ortalaması) [76,77] ve *MAE* (Mutlak Hata Ortalaması) [78]

performans ölçümlerine bakılmıştır. Ölçüm denklemleri iki boyutlu veriler için düzenlenerek çalışmada kullanılmıştır. Bu denklemler aşağıda verilmiştir.

$$RMSE = \sqrt{\frac{\sum (Y_x - Y'_x)^2 + (Y_y - Y'_y)^2}{n}} \quad (5.1)$$

$$MAPE = \frac{1}{n} \sum \frac{\sqrt{(Y_x - Y'_x)^2 + (Y_y - Y'_y)^2}}{\sqrt{Y_x^2 + Y_y^2}} \times 100 \quad (5.2)$$

$$MAE = \frac{1}{n} \sum \sqrt{(Y_x - Y'_x)^2 + (Y_y - Y'_y)^2} \quad (5.3)$$

Denklem 5.1, 5.2 ve 5.3'de;

n = Tahmin sayısı,

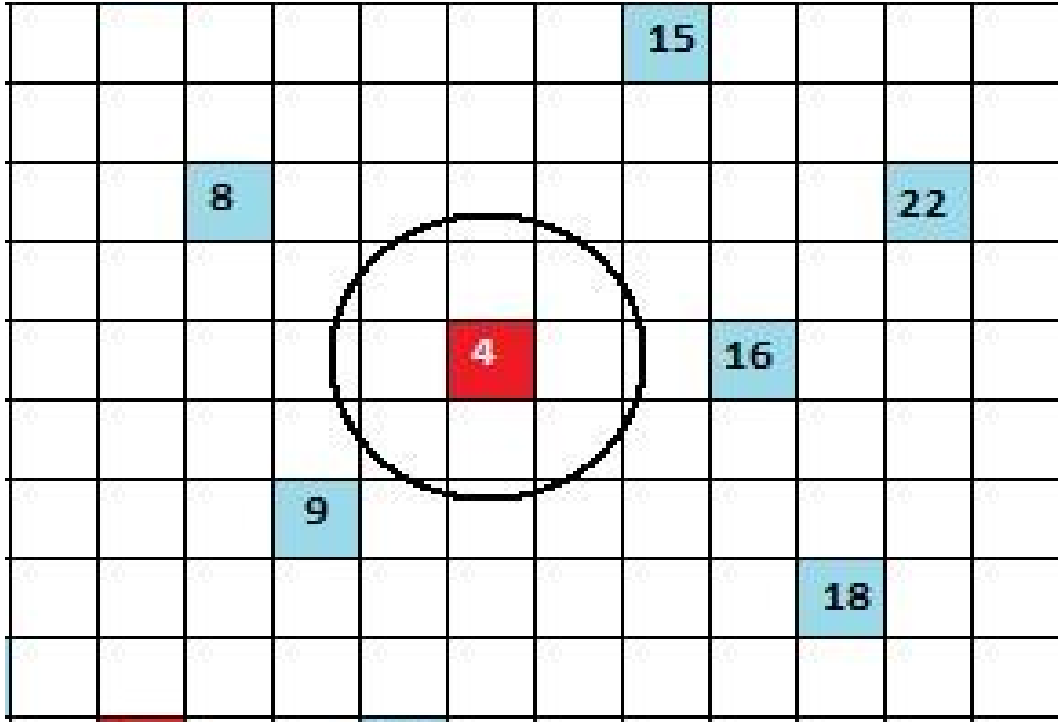
Y_x = Gerçek x koordinatı,

Y'_x = Tahmin edilen x koordinatı,

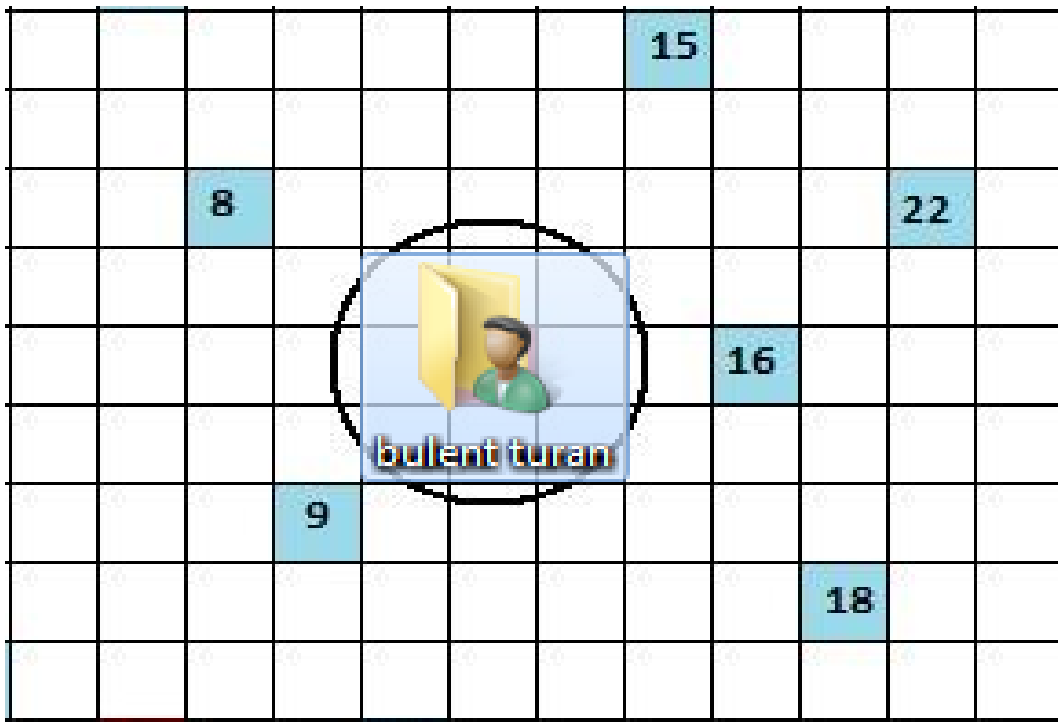
Y_y = Gerçek y koordinatı,

Y'_y = Tahmin edilen y koordinatıdır [34].

Geliştirilen YSA'ya test verilerinin uygulanması sonucu elde edilen değerler hangi bölge içinde yer alıyorsa bu bölge koordinatları tespit edilen bölge koordinatları olarak kullanılmıştır. Gerçek bölge koordinatları ile tespit edilen bölge koordinatları arasındaki uzaklık, hata olarak hesaplanmış ve bu hata değerlerine göre performans ölçümleri yapıldığında, $RMSE=2.0069$, $MAPE=0.0735$, $MAE=1.6648$ bölge (1.6648×24 piksel) olarak bulunmuştur. Performans ölçüm değerlerinin küçük olması gerçek bölge koordinatları ile YSA tarafından tespit edilen bölge koordinatları arasındaki sapmaların küçük olduğunu göstermektedir. MAE değerine göre, herhangi bir bölge için bulunması beklenen değerlerin ortalaması ve standart simge büyüklüğü ile karşılaştırması Şekil 5.1'de verilmiştir.



(a)



(b)

Şekil 5.1. (a) 4 numaralı test bölgesi için elde edilmesi beklenen sonuçların ortalaması (b) sonuçların Standart bir simge büyüklüğü ile karşılaştırılması

Hata ortalaması bilgisayar ekranındaki standart simgenin büyüklüğü ile kıyaslandığında, geliştirilen YSA ile elde edilen sonuçların tıklama fonksiyonları da

geliştirildiği takdirde bir dosyanın seçilmesinde ve açılmasında kullanılabileceği görülmektedir.

Çalışmada elde edilen sonuçların doğrulanması amacıyla çapraz doğrulama yapılmıştır. Çapraz doğrulama yapmak için eğitim seti içinde bulunan ve rasgele seçilen 20 bölgeye ait giriş verileri, test için kullanılan 20 bölgeye ait giriş verileri ile değiştirilmiştir. Yeni eğitim ve test verileri ile elde edilen sonuçlar kullanılarak performans ölçümleri tekrarlanmış $RMSE=2.6453$, $MAPE=0.0715$, $MAE=2.0070$ bölge (2.0070×24 piksel) olarak bulunmuştur. Çapraz doğrulama sonucunda performans ölçümlerinden birinde düşme (MAPE) meydana gelmiştir. Diğer 2 performans ölçümünde (RMSE ve MAE) ise yükselme olduğu görülmüştür. Ancak değişim miktarları incelendiğinde değişimin çok büyük olmadığı söylenebilir.

Test görüntülerinden elde edilen sonuçlar ile Matlab “pointerlocation” komutu kullanılarak imleç kontrolü gerçekleştirilmiştir. 10 numaralı test bölgesi görüntülerinden elde edilen sonuçlar ile gerçekleştirilen imleç kontrolüne yönelik görüntüler Şekil 4.23’de verilmiştir.

Çalışmada daha iyi sonuçlara ulaşılamamasının sebeplerinden biri olarak sabitleme anlarında dahi gözdeki hareketliliğin devam etmesi olduğu düşünülmektedir. Bu küçük hareketliliğin kullanıcıya ait görüntülerin kaydı esnasında tespit edilemeyeceği ve belli oranda hatalı kayıt yapılacağı düşünülmektedir. Böyle bir durumun sonuçları olumsuz etkileyeceği ise kesindir.

5.1.4. Arayüzün kullanılmasına yönelik sonuçlar

Çalışmada farklı kullanıcıların eğitim ve test görüntülerinin alınması için hazırlanan arayüz kullanılmıştır. Böylece deneklerin tek bilgisayara bağımlı çalışma zorunluluğu giderilmiş, kullanım ve kayıt işlemleri otomatikleştirilerek görüntü alma süreleri kısaltılmıştır.

5.1.5. Farklı kullanıcılar ile koordinat tespitine yönelik sonuçlar

Aynı ağıın deneklerden alınan görüntüler ile eğitim ve test aşamalarının tekrarlanması sonucunda ise elde edilen performans ölçüm değerleri Tablo 5.1’de verilmiştir.

Tablo 5.1. Deneklerden elde edilen performans ölçümleri

Denek No	RMSE	MAPE	MAE
1	5.5245	0.28168	4.1566
2	6.0787	0.39425	5.2363
3	5.527	0.27387	4.6644
4	5.65	0.25946	5.0021
5	3.2202	0.1357	2.7083
6	7.7756	0.38961	7.0399
7	18.3218	0.929	15.7423
8	6.9441	0.35913	6.1379
9	7.823	0.299	6.5481
10	7.5598	0.50591	6.2501
11	5.7171	0.21133	4.9078

5.2. Öneriler

Elde edilen sonuçlar doğrultusunda, bakışlar ile imleç kontrolüne yönelik gerçekleştirilen bu çalışmanın geliştirilerek, gerçek zamanlı ve özellikle büyük ekranlı (14" üzeri TV, PC gibi ekranlar) uygulamalarda rahatlıkla kullanılabileceği söylenebilir. Böylelikle büyük ekranlı uygulamalarda dokunmatik ekran maliyeti ve kullanım zorluğunun oluşturacağı olumsuzluk ortadan kaldırılabilir. Ayrıca küçük ekranlarda (4,7" altı Telefon ekranları) kullanım için hassasiyeti artırılabilirse, dokunmatik ekranlarda parmak boyutlarının büyüklüğünden dolayı yaşanan probleme de bir çözüm olabilir.

Ekranla bakılan nokta koordinatları ile kafa pozisyonları bilgisini birlikte ihtiva eden standartlaştırılmış (ışık şiddeti, bakış açısı gibi veriler açısından) veri seti

hazırlanabilir. Böylelikle, konu üzerine çalışacak araştırmacıların sonuçlarının tekrarlanabilirliği ve geçerliliği ile birlikte, araştırma sayısının da artmasına destek sağlanabilir.

Çalışmada göz bölgesinin bulunmasına yönelik kullanılan YSA eğitim setinin genişletilmesi (etnik farklılıklarında dikkate alınması ile tüm morfolojik farklılıkları ifade edecek bir denek gurubu kullanılarak) ile kullanıcıdan bağımsız hale getirilebilir. Böylece farklı kullanıcıların kullanımı için tasarlanan arayüz geliştirilerek kullanıcıların kendi Eye Gaze sistemlerini eğitmeleri ve kullanmaları sağlanabilir.

Çalışmada kullanılan yöntem, çalışma esnasında referans noktasına yada kalibrasyon işlemine ihtiyaç duymamaktadır. Bu özelliği, Opengazer sisteme göre avantajını ifade eder. Ancak her kullanıcı için eğitim sürecinin tekrarlanma gereği ise yöntemin dezavantajıdır. Çalışma farklı kullanıcılar için yapılacak çalışmalar birleştirilerek ya da çok kullanıcıya yeni bir çalışma yapılarak kullanıcıdan bağımsız hale getirilebilir. Böylece her kullanıcı için ayrı ayrı eğitime şeklinde ortaya çıkan dezavantaj bertaraf edilebilir.

Çalışmada 15.6" ekran boyutu ve 1366×768 ekran çözünürlüğü kullanılmıştır. Diğer ekran boyutları ve çözünürlüklerine yönelik uygunlaştırma çalışmaları yapılarak, çalışma ekran boyutu ve ekran çözünürlüğünden bağımsızlaştırılabilir.

Yakın bir gelecekte Eye Gaze sistemlerde ekranda bakılan noktanın tespiti amacıyla göz takipçisi yerine web kamerası görüntüleri kullanılabilir. Bu da Eye Gaze sistemlerinin kullanımlarını basitleştirir ve maliyetlerini olumlu yönde etkiler. Böylece kullanımlarının yaygınlaşmasına ve kullanım alanlarının genişlemesine de katkı sağlar.

Göz bölgesinden alınan piksel değerlerinin yanında göz bölgesinden elde edilecek özniteliklerinde (daha iyi yöntemler ile görüntü işleyerek elde edilecek iris merkezi ve göz köşe koordinatları vb.) giriş verilerine ilave edilmesinin sonuçları olumlu yönde etkileyeceği öngörülebilir.

Koordinat tespiti amacıyla geliştirilen YSA eğitimde kullanılan toplam bölge sayısının artırılması ile de sonuçların iyileştirilmesi düşünülebilir.

Deneklerden elde edilen sonuçlar doğrultusunda ise, göz bölgesi verileri ile kafa pozisyonu verilerinin birlikte kullanılmasının, elde edilen sonuçlardaki hata oranlarını azaltmasının yanında programın kullanımını da kolaylaştıracağı öngörülebilir.

KAYNAKLAR

- [1] ACARTÜRK, C., ÇAĞILTAY K., İnsan Bilgisayar Etkileşimi ve ODTÜ'de Yürütülen Çalışmalar. Akademik Bilişim 2006 Bildiriler Kitapçığı, 2006.
- [2] KARAY, F., ALEMZADEH, M., SALEH, SA., ARAB, MO., Human-Computer Interaction: Overview on State of the Art. International Journal On Smart Sensing And Intelligent Systems, Vol. 1, No. 1, March 2008.
- [3] ExtremeTech, Canesta says “Virtual Keyboard” is reality. <http://www.extremetech.com/article2/0,1558,539778,00.asp>, Erişim Tarihi: 21.07.2014.
- [4] Dark, dijital kalem (i-Pen) yeni nesil turkce destekli pc, mac, iphone, ipad uyumlu dijital cizim kalemi ve ajanda. <http://www.dark.com.tr/cevrebirimleri/dijital-kalem/dark-i-pen-yeni-nesil-turkce-destekli-pc-mac-iphone-ipad-uyumlu-dijital-cizim-kalemi-ve-ajanda.html>, Erişim Tarihi: 21.07.2014.
- [5] GÜNDÜZ, H.C., Göz Hareketlerinin Takibi ve Kaydedilmesi, 2005 <http://ileriseviye.org/arasayfa.php?inode=eye-tracking.html>, Erişim Tarihi: 22.07.2014.
- [6] ÖZÇELİK, E., KURŞUN, E., ÇAĞILTAY, K., Göz Hareketlerini İzleme Yöntemiyle Üniversite Web Sayfalarının İncelenmesi. Akademik Bilişim 2006 Bildiriler Kitapçığı, Denizli, 9-11 Şubat, 2006.
- [7] İNCE, O., GÖKTÜRK, M., Güvenlik Sistemi İzleyici Personelinin Görsel Tarama Davranışının Analizi. Akademik Bilişim konferansı, pp:699-704 Urfa Türkiye, Şubat 2009.
- [8] JACOB, RJK., AND KARN, KS., Eye tracking in Human-computer interaction and usability research: Ready to deliver the promises. in the Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research, ed. by J. Hyona, R. Radach, and H. Deubel, Amsterdam, Elsevier Science, pp 573-605, 2003.

- [9] POOLE, A., AND BALL, LJ., Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects. In Ghaoui, C. (ed.) *Encyclopedia of Human Computer Interaction*. Hershey, PA: Idea Group. 211-219, 2005.
- [10] Eye Gaze System. <http://parvathyk.blogspot.com.tr/p/embedded-systems.html>, Erişim Tarihi: 22.07.2014.
- [11] Moving Ahead With Eye Power. http://spinoff.nasa.gov/spinoff2002/hm_1.html, Erişim Tarihi: 22.07.2014.
- [12] Samsung Galaxy S4 to Introduce Eye Scroll. <http://www.blog.windowpasswordsrecovery.com/samsung-galaxy-s4-to-introduce-eye-scroll/>, Erişim Tarihi: 25.07.2014.
- [13] <http://www.teknokistan.com/samsung-galaxy-s4-ozellikleri/>. Erişim Tarihi: 25.07.2014.
- [14] AYDEMİR , A., UNERİ, A., Nistagmusta gozun hizli fazli hareketlerinin tespit ve analizi (VNG) detection and analysis of quick phase eye movements in nystagmus (VNG). Signal Processing and Communications Applications, IEEE 14th, pp.1-4, Antalya, 2006.
- [15] BODADE, R., TALBAR, S., Dynamic iris localisation: a novel approach suitable for fake iris detection. International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM) Vol.2, pp.163-173, 2010.
- [16] HUSAM, A., LAHRASH, E., NORDİN, MJ., An enhanced segmentation approach for iris detection. European Journal of Scientific Research Vol.59 No.2, pp.179-190, 2011 <http://www.eurojournals.com/ejsr.htm>, Erişim Tarihi: 25.07.2014.
- [17] GRECO, J., KALLENBORN, D., NECHYBA, MC., Statistical pattern recognition of the iris. In 17th annual Florida Conference on the Recent Advances in Robotics (FCRAR), 2004, http://www.mil.ufl.edu/publications/fcrar04/fcrar2004_iris.pdf, Erişim Tarihi: 25.07.2014.
- [18] DAUGMAN, J., New methods in iris recognition. IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, Vol.37, No.5, pp.1167-1175, 2007.
- [19] ERDOGMUS, N., DUGELAY, JL., Automatic extraction of facial interest points based on 2D and 3D data. SPIE 2011, Electronic Imaging Conference on 3D Image Processing (3DIP) and Applications, Vol 7864, San Francisco, California, January 23-27, 2011.

- [20] BATISTA, P.J., Locating facial features using an anthropometric face model for determining the gaze of faces in image sequences. Lecture Notes in Computer Science, Volume 4633/2007, pp.839-853, 2007.
- [21] KUO, P., HANNAH, J., An improved eye feature extraction algorithm based on deformable templates. In Proceedings of International Conference on Image Processing, IEEE, volume II, pp.1206-1209, 2005.
- [22] MAJUMDER, A., BEHERA, L., SUBRAMANIAN VK., Automatic and robust detection of facial features in frontal face images. UKSim 13th International Conference on Modelling and Simulation, pp.331-336, 2011.
- [23] DIAMANTOPOULOS, G., Novel eye feature extraction and tracking for non-visual eye-movement applications. A Thesis Submitted Department Of Electronic, Electrical And Computer Engineering To The University Of Birmingham For The Degree Of Doctor Of Philosophy, 2010.
- [24] DEMİR, İ., Bilgisayarlı görmeye giriş. e-bergi, ODTÜ bilgisayar topluluğu elektronik dergisi, <http://e-bergi.com/y/Bilgisayarli-Gormeye-Giris> Erişim Tarihi: 10.02.2015.
- [25] Bilgisayar görüşü ve imge işleme. Haziran, 2001, <http://robot.cmpe.boun.edu.tr/593/gorusimge.pdf>, Erişim Tarihi: 10.02.2015.
- [26] GONZALEZ, RC., WOODS, RE., Digital Image Processing, Prentice Hall 2nd Edition. New Jersey, 2002.
- [27] ARSLAN, E., Hücresel Sinir Ağı Sistemleri Kullanarak Hareketli Nesnelerin Görüntü İşleme Uygulamaları. İstanbul Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Doktora Tezi, İstanbul, 2011.
- [28] KURTULMUŞ, F., Olgunlaşmamış Şeftali Meyvesini Doğal Bahçe Koşullarında Alınmış Görüntülerde Görüntü İşleme Teknikleri Ve Yapay Sınıflandırıcılarla Saptayarak Sayan Algoritmaların Geliştirilmesi. Doktora Tezi, Uludağ Üniversitesi Fen Bilimleri Enstitüsü, Tarım Makinaları Anabilim Dalı, Bursa, 2012.
- [29] AKAR, F., Şablon eşleme yöntemi ile plaka tanıma ve değerlendirme sistemi. Doktora Tezi, AÜ Fen Bilimleri Enstitüsü, Matematik Anabilim Dalı, Erzurum, 2009.
- [30] PRATT, WK., Digital Image Processing, Fourth Edition, Wiley-Interscience. New Jersey, 2007.

- [31] ŐENGÜR, A., TÜRKOĐLU, İ., İNCE, MC., İki Boyutlu Entropi İle Görüntü EŐikleme Uygulamaları. IEEE 13. Sinyal İŐleme ve İletiŐim Uygulamaları Kurultayı, 131 (2), Kayseri, Türkiye, Mayıs,2005, <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1567674>, EriŐim Tarihi: 02.04.2015.
- [32] NAGI, J., AHMED, SK., NAGİ, F., A MATLAB based Face Recognition System using Image Processing and Neural Networks. 4th International Colloquium on Signal Processing and its Applications, Kuala Lumpur, Malaysia, March 7-9, 2008.
- [33] LIN, CS., HO, CW., CHEN, WC., CHIU, CC., YEH, MS., Powered Wheelchair Controlled by Eye-Tracking System. Optica Applicata, Vol XXXVI, No. 2-3, 2006.
- [34] TURAN, B., ESKİKURT, Hİ., CAN, MS., Estimated of coordinates of user's looked point on laptop screen by ANN. Signal Processing and Communications Applications Conference (SIU), 2014 22nd, pp. 108-111, 23-25 April 2014.
- [35] GOSEN, T., KOTZYBA, M., STOBER, S., NURNBERGER, A., Voice-Controlled Search User Interfaces for Young Users. 7th annual Symposium on Human-Computer Interaction and Information Retrieval, 2013.
- [36] YAZDANI, N., KHAZAB, F., FİTZGİBBON, S., LUERSSSEN, M., POWERS, D., CLARK, CR., Towards a brain-controlled Wheelchair Prototype. BCS '10 Proceedings of the 24th BCS Interaction Specialist Group Conference, pp. 453-457, 2010.
- [37] GRAIMANN, B., ALLISON, B., PFURTSCHELLER, G., Brain-Computer Interfaces: A Gentle Introduction. Brain-Computer Interfaces, The Frontiers Collection, pp 1-27, 2010.
- [38] İBRAHİM, O., ELGENDY, H., ELSHAFEE, AM., Speed Detection Camera System using Image Processing Techniques on Video Streams. International Journal of Computer and Electrical Engineering, Vol. 3, No. 6, December 2011.
- [39] YÜCEL, M., ERTUĐRUL, U., PARLAK, E.B., KÖSE, H., İNCE, G., Bateri Çalma Oyununda İnsan-Robot EtkileŐimi. Signal Processing and Communications Applications Conference (SIU), 2014 22nd, pp. 919-922, 23-25 April 2014.
- [40] KARADAL, Y., KIVRAK, H., KÖSE, H., Engelli Çocuklar için Kinect tabanlı EtkileŐimli Müzik Uygulaması. Signal Processing and Communications Applications Conference (SIU), 2014 22nd, pp. 734-737, 23-25 April 2014.

- [41] NEL, E., WILLIAMS, O., CIPOLLA, R., Opengazer: open-source gaze tracker for ordinary webcams. The Opengazer project is supported by Samsung and the Gatsby Foundation and by the European Commission in the context of the AEGIS project, 2012. <http://www.inference.phy.cam.ac.uk/opengazer/>, Eriřim Tarihi: 02.04.2015.
- [42] İŐLER, Y. S., ARTIKLAR, M., Bir tek noron kullanılarak resimler ierisinde goz kısmının bulunması. KSU Fen ve Mühendislik Dergisi, vol.11, no.1, pp.59-63, 2008.
- [43] TURKAN, M., PARDAS, M., ETİN, AE., Edge projections for eye localization. Optical Engineering 47, 047007, vol.47, no.4, 2008.
- [44] XİA, HY., YAN, GP., A novel method for eye corner detection based on weighted variance projection function. Image and Signal Processing, 2009. CISP'09. 2nd International Congress on , vol., no., pp.1-4, 2009.
- [45] ERDOGMUS, N., DUGELAY, JL., An efficient ırıs and eye corners extraction method. In Proceedings of SSPR/SPR, pp.549-558, 2010.
- [46] SİROHEY, SA., ROSENFELD, A., Eye detection in a face image using linear and nonlinear filters. Pattern Recognition, vol.34, pp.1367-1391, 2001.
- [47] HAYKİN, S., Neural networks; a comprehensive foundation, Second edition. Prentice hall, 1999.
- [48] DOĐAN, M., Yapay sinir ađları temelli tıbbi teřhis sistemi. Yüksek lisans tezi, Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliđi Anabilim Dalı, Ađustos, 2003.
- [49] AYMAZ, O., Yapay sinir ađları | avantaj ve dezavantajları. <http://www.elektrikport.com/teknik-kutuphane/yapay-sinir-aglari-avantaj-ve-dezavantajlari/15007#ad-image-0>, Eriřim Tarihi: 16.02.2015.
- [50] FAUSET, L., Fundamentals of neural networks; architectures, algorithms and applications. 1994.
- [51] ÖZTEMEL, E., Yapay Sinir Ađları. Papatya yayınevi, İstanbul, Ekim, 2006.
- [52] Introduction & Fundamentals. <http://uhaweb.hartford.edu/ilumokanw/Chap1student.ppt>, Eriřim Tarihi: 15.02.2015.
- [53] YARAR, A., ONÜÇYILDIZ, M., Yapay Sinir Ađları İle Beyřehir Gölü Su Seviyesi Deđişimlerinin Belirlenmesi. II. Ulusal Su Mühendisliđi Sempozyumu, 21-24 Eylül 2005, İzmir, Türkiye, 2005.

- [54] ŞAHİN, İ., AKAYA, A., Anndes: Bir Yapay Sinir Ağı Tasarım Ve Eğitim Yazılımı. 5. International Computer & Instructional Technologies Symposium, 22-24 September 2011, Fırat University, Elazığ, Turkey, 2011.
- [55] YURTOĞLU, H., Yapay Sinir Ağları Metodolojisi İle Öngörü Modellemesi: Bazı Makroekonomik Değişkenler İçin Türkiye Örneği. Uzmanlık Tesi, Ekonomik Modeller Ve Stratejik Araştırmalar Genel Müdürlüğü, Yayın No: DPT: 2683, Şubat 2005, Türkiye, 2005.
- [56] ANDERSON, D., MCNEİL, G., Artificial Neural Networks Technology. Kaman Science Corporation, New York, USA, 1992.
- [57] UÇAN, ON., OSMAN, O., ALBORA, AM., Görüntü işleme teknikleri ve Mühendislik uygulamaları. Nobel yayın dağıtım, Ankara, 2006
- [58] MINSKY, ML., PAPERT, SA., Perceptrons, Cambridge, MA: MIT Pres, 1969.
- [59] CİNSDİKİCİ, M., Yapay sinir ağları. <http://ube.ege.edu.tr/~cinsdiki/UBI521/Chapter-1/cinsdikici-neural-net-giris.pdf>, Erişim Tarihi: 10.02.2015.
- [60] KAASTRA, I., BOYD, M., Designing a Neural Network for Forecasting Financial and Economic Time Series. Neurocomputing, (10):215-236, 1996.
- [61] AKEL, V., KARACAMEYDAN, F., Yatırım Fonları Net Varlık Değerlerinin Yapay Sinir Ağları Yöntemiyle Tahmin Edilmesi. AÜ, Sosyal Bilimler Dergisi, pp. 87-105, 2012.
- [62] AKBİLGİÇ, O., Hibrit Radyal Tabanlı Fonksiyon Ağları İle Değişken Seçimi Ve Tahminleme: Menkul Kıymet Yatırım Kararlarına İlişkin Bir Uygulama. Doktora tezi, İÜ, Sosyal Bilimler Enstitüsü, İşletme Ana Bilim Dalı, İstanbul, Türkiye, 2011.
- [63] ORHAN, U., Makine öğrenmesi. <http://bmb.cu.edu.tr/uorhan/DersNotu/Ders06.pdf>, Erişim Tarihi: 12.02.2015
- [64] HULIANE, MS., CICERO, AS., FLAVIUS, LG., Self – Organizing map based strategy for heterogenous teaming. InTech, DOI: 10.5772/52776 November 21, 2012. <http://www.intechopen.com/books/howtoreference/applications-of-self-organizing-maps/a-self-organizing-map-based-strategy-for-heterogeneous-teaming>, Erişim Tarihi: 12.02.2015.

- [65] LIPPMANN, RP., An introduction to computing with neural nets. IEEE ASSP Mag., vol 4, April, 1987.
- [66] SEZGİN, N., TEKİN, R., ÇALIŞKAN, A., Örüntü tanımda Hopfield ağının kullanılması. Batman Üniversitesi, Yaşam Bilimleri Dergisi, Cilt 1, Sayı 2, Batman, Türkiye, 2012.
<http://www.yasambilimleridergisi.com/makale/pdf/1356118552.pdf>,
Erişim Tarihi: 15.02.2015.
- [67] YOUNG, D., A hough transform function for detecting circles, optimised to allow search over radii.
http://www.mathworks.com/matlabcentral/fileexchange/26978-hough-transform-for-circles/content/circle_hough.m, Erişim Tarihi: 20.03.2012.
- [68] TURAN, B., ESKİKURT, Hİ., Iris and eye corner detection by processing internal webcam images. Turkish Journal of Electrical Engineering & Computer Sciences. E-ISSN: 1303-6203, DOI: 10.3906/elk-1401-96, file:///C:/Users/B%C3%BClentTuran/Downloads/ELK-1401-96_manuscript_2%20(1).pdf, Erişim Tarihi: 09.07.2015.
- [69] RIZON, M., AND OTHERS, Object detection using circular hough transform. American Journal of Applied Sciences, vol.12, no.2, pp.1606-1609, 2005.
- [70] ZHOU, ZH., GENG, X., Projection functions for eye detection. Pattern Recognition vol.37 pp.1049–1056, 2004.
- [71] FENG, GC., YUEN, PC., Variance projection function and its application to eye detection for human face recognition. Pattern Recognition Letters, vol.19, pp.899–906, 1998.
- [72] ASPDEN, TJ., ADLER, J., DAVIS, SS., SKAUGRUD, Q., ILLUM, L., Chitosan as a nasal delivery system: evaluation of the effect of chitosan on mucociliary clearance rate in the frog palate model. International Journal of Pharmaceutics, vol.122, Issues 1-2, pp.69-78, 1995.
- [73] ZHOU, W., BOVIK, AC., A universal image quality index. Signal Processing Letters, IEEE, vol.9, Issue 3, pp.81-84, 2002.
- [74] ABRAMOFF, MD., NIEMEIJER, M., The automatic detection of the optic disc location in retinal images using optic disc location regression. Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE, pp.4432-4435, 2006.
- [75] KWOK-WAI, W., KIN-MAN, L., WAN-CHİ, S., A robust scheme for live detection of human faces in color images. Signal Processing: Image Communication, vol 18, Issue 2, pp. 103-114, 2003.

- [76] CHO, V., A comparison of Three Different Approaches to Tourist Arrival Forecasting, *Tourism Management*, 24 (3): 323-330, 2003.
- [77] CEYLAN, H., AVAN, M., Türkiye'deki İş Kazalarının Yapay Sinir Ağları ile 2025 Yılına Kadar Tahmini. *International Journal of Engineering Research and Development*, Vol.4, No.1, January 2012.
- [78] WILLMOTT, C.J., MATSUURA, K., Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, Vol. 30: 79–82, December 2005.

EKLER

EK1: Arařtırmacıya ve Deneklere ait Görüntüler

Çalıřmada göz bölgesinin bulunması ve koordinat tespiti için kullanılan dâhili webcam görüntüleri, YSA eđitiminde kullanılan dođru ve yanlış göz bölgesi görüntüleri CD'de verilmiřtir.

EK2: Arařtırmacıya ve Deneklere ait veriler

Çalıřmada göz bölgesinin bulunması ve koordinat tespiti için YSA eđitiminde kullanılan veriler Excel dosyaları haline getirilerek CD'de verilmiřtir

EK3: Göz Bölgesinin Bulunması Amacıyla Geliřtirilen YSA Testinde Kullanılan Test Görüntüleri



EK5: Göz Bölgesinin Bulunması Amacıyla Geliştirilen YSA'nın ayırık zamanlı test görüntüleri ile test edilmesi için hazırlanan Matlab yazılımı

```
%% ağ test ediliyor
load net100
for ii=2:20
    if ii==1
        test1=imread('1test.jpg');
    elseif ii==2
        test1=imread('2test.jpg');
    elseif ii==3
        test1=imread('3test.jpg');
    elseif ii==4
        test1=imread('4test.jpg');
    elseif ii==5
        test1=imread('5test.jpg');
    elseif ii==6
        test1=imread('6test.jpg');
    elseif ii==7
        test1=imread('7test.jpg');
    elseif ii==8
        test1=imread('8test.jpg');
    elseif ii==9
        test1=imread('9test.jpg');
    elseif ii==10
        test1=imread('10test.jpg');
    elseif ii==11
        test1=imread('11test.jpg');
    elseif ii==12
        test1=imread('12test.jpg');
    elseif ii==13
        test1=imread('13test.jpg');
    elseif ii==14
        test1=imread('14test.jpg');
    elseif ii==15
        test1=imread('15test.jpg');
    elseif ii==16
        test1=imread('16test.jpg');
    elseif ii==17
        test1=imread('17test.jpg');
    elseif ii==18
        test1=imread('18test.jpg');
    elseif ii==19
        test1=imread('19test.jpg');
    elseif ii==20
        test1=imread('20test.jpg');
    end
    test11=rgb2gray(test1);aaa=size(test1);
    if aaa(1,1)<480
        test11=imresize(test1,2);
    end
    y=-1;a=80;b=150;figure,imshow(test11);
    while y < .9
        if a < 350
            if b < 290
                test1=test11;
                tst(1:48,1:160)=test1(a+1:a+48,b+1:b+160);    X=double(tst);
                [cA] = dwt2(X,'db1');XX=cA/2;[cA] = dwt2(XX,'db1');XXX=cA/2;
                [cA] = dwt2(XXX,'db1');cA=uint8(cA);cA=uint8(cA);
                v1=cA(1,:);v2=cA(2,:);v3=cA(3,:);v4=cA(4,:);v5=cA(5,:);v6=cA(6,:);
                vtst=cat(2,v1,v2,v3,v4,v5,v6);    Test=double(vtst');
                y = sim(net, Test);
            else
                y=1;
            end
        end
    end
end
```

```

end
a=a+4;
else
    a=80;    b=b+4;
end
test1(a,b:b+159)=255;    test1(a+48,b:b+159)=255;
test1(a:a+48,b)=255;    test1(a:a+48,b+159)=255;
imshow(test1);pause (0.1)
end
y = sim(net, Test)
tst=uint8(tst);
gb(1:48,1:160)=test1(a+1:a+48,b+1:b+160);
if ii==1
    save ('gb1', 'gb');
elseif ii==2
    save ('gb2', 'gb');
elseif ii==3
    save ('gb3', 'gb');
elseif ii==4
    save ('gb4', 'gb');
elseif ii==5
    save ('gb5', 'gb');
elseif ii==6
    save ('gb6', 'gb');
elseif ii==7
    save ('gb7', 'gb');
elseif ii==8
    save ('gb8', 'gb');
elseif ii==9
    save ('gb9', 'gb');
elseif ii==10
    save ('gb10', 'gb');
elseif ii==11
    save ('gb11', 'gb');
elseif ii==12
    save ('gb12', 'gb');
elseif ii==13
    save ('gb13', 'gb');
elseif ii==14
    save ('gb14', 'gb');
elseif ii==15
    save ('gb15', 'gb');
elseif ii==16
    save ('gb16', 'gb');
elseif ii==17
    save ('gb17', 'gb');
elseif ii==18
    save ('gb18', 'gb');
elseif ii==19
    save ('gb19', 'gb');
elseif ii==20
    save ('gb20', 'gb');
end
test1(a,b:b+159)=255;test1(a+48,b:b+159)=255;
test1(a:a+48,b)=255;test1(a:a+48,b+159)=255;
figure,imshow(test1);
end

```


EK6: Göz Bölgesinin Bulunması Amacıyla Geliştirilen YSA'nın real time kullanılması için hazırlanan Matlab yazılımı

```
clear all; clc; load net100; %load sadece_dwt_ysa12
tst=ones(24,80); atla=0; fotono=0; p=1; v=1; z=0;
obj = videoinput('winvideo', 1, 'YUY2_320x240');
stop(obj); pause(20); set(obj,'ReturnedColorSpace','grayscale');
set(obj,'FramesPerTrigger',Inf); start(obj); pause(20);
video.FrameGrabInterval=1;
while p==1;
    if v==1
        z=0; v=v+1; y=-1; a=50; b=90;
        test3 = getsnapshot(obj); test2 = padarray(test3,[0
320],'symmetric','pre');
        test1(:,1:320)=test2(:,1:320);
        test11=test1;
        while y < .9
            if a < 170
                if b < 190
                    tst(1:24,1:80)=test11(a+1:a+24,b+1:b+80);
                    X=double(tst); [cA] = dwt2(X,'dbl'); XX=cA/2; [cA] = dwt2(XX,'dbl');
                    cA=uint8(cA);
                    v1=cA(1,:); v2=cA(2,:); v3=cA(3,:); v4=cA(4,:); v5=cA(5,:);
                    v6=cA(6,:);
                    vtst=cat(2,v1,v2,v3,v4,v5,v6); Test=double(vtst'); y = sim(net, Test);
                    else
                        y=1;
                    end
                    a=a+4;
                else
                    a=50; b=b+4;
                end
                test1(a,b:b+79)=255; test1(a+24,b:b+79)=255;
            test1(a:a+24,b)=255; test1(a:a+24,b+79)=255; imshow(test1);pause(0.05);
            end
            else
                test3 = getsnapshot(obj); test2 = padarray(test3,[0
320],'symmetric','pre');
                test1(:,1:320)=test2(:,1:320); test11=test1;
                tst(1:24,1:80)=test11(a+1:a+24,b+1:b+80);
                X=double(tst); [cA] = dwt2(X,'dbl'); XX=cA/2; [cA] = dwt2(XX,'dbl');
                cA=uint8(cA);
                v1=cA(1,:); v2=cA(2,:); v3=cA(3,:); v4=cA(4,:); v5=cA(5,:);
                v6=cA(6,:);
                vtst=cat(2,v1,v2,v3,v4,v5,v6); Test=double(vtst'); y = sim(net, Test);
                q=1; t=1;
                while y < .9
                    if q<10
                        if t < q+1
                            if rem(q,2) == 1
                                u=-4;
                            else
                                u=+4;
                            end
                            a=a+u;
                        tst(1:24,1:80)=test1(a+1:a+24,b+1:b+80);
                        X=double(tst); [cA] = dwt2(X,'dbl'); XX=cA/2; [cA] = dwt2(XX,'dbl');
                        cA=uint8(cA);
                        v1=cA(1,:);v2=cA(2,:);v3=cA(3,:);v4=cA(4,:);v5=cA(5,:);v6=cA(6,:);
                        vtst=cat(2,v1,v2,v3,v4,v5,v6); Test=double(vtst'); y = sim(net, Test);
                        t=t+1;
                    elseif t < (2*q)+1
                        if rem(q,2) == 1
                            u=-4;

```

```

        else
            u+=4;
        end
        b=b+u;
tst(1:24,1:80)=test1(a+1:a+24,b+1:b+80);
X=double(tst); [cA] = dwt2(X,'db1'); XX=cA/2; [cA]=dwt2(XX,'db1');
cA=uint8(cA);
v1=cA(1,:);    v2=cA(2,:);    v3=cA(3,:);    v4=cA(4,:);    v5=cA(5,:);
v6=cA(6,:);
vtst=cat(2,v1,v2,v3,v4,v5,v6);    Test=double(vtst');    y = sim(net, Test);
t=t+1;
        else
            t=1; q=q+1;
        end
        else
            y=1; atla=1;
        end
        test1(a,b:b+79)=255; test1(a+24,b:b+79)=255;
test1(a:a+24,b)=255; test1(a:a+24,b+79)=255; imshow(test1);pause (0.05);
    end
end
y = sim(net, Test);
    if y < .9
        z=z+1;
    end
tst=uint8(tst);
test1(a,b:b+79)=255;    test1(a+24,b:b+79)=255;    test1(a:a+24,b)=255;
test1(a:a+24,b+79)=255;
fotono=fotono+1; pause(.0000001); imshow(test1); pause (0.05);
    if a < 50 || a > 170 || b < 90 || b > 190 || z==2
        v=1;
    end
end
end

```

EK7: Çalışmada Dâhili Webcam ile Görüntü almak Amacıyla Bilgisayar Arka Planı Olarak Oluşturulan Görüntü

Paint ortamında oluşturulan görüntü CD’de verilmiştir.

EK8: Koordinat Tespitinin Yapılması Amacıyla Geliştirilen YSA Testinde Kullanılan Test Görüntüleri




```
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 ...
19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 ...
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 ...
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 ...
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 ...
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 ...
28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 ...
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 ...
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ...
17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 ...
31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 ...
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 ...
26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26];
```

```
T(1,:)=T(1,:)./57;
T(2,:)=T(2,:)./32;
```

```
net = newff(P,T,[90 9] ,{'tansig' 'tansig' 'tansig' 'tansig'}, 'trainscg',
'learngdm');
net.trainparam.epochs = 2000000; %epoc sayısını giriyoruz
net.trainparam.lr = .6; %öğrenme katsayısını giriyoruz.
net.trainparam.mc = .3; %momentum katsayısını giriyoruz
net.trainparam.goal = 2.75*(10^(-4)); %hedef için kabul edilebilir hata
net.divideFcn = '';
net.trainParam.max_fail = 5;
net_koord = train(net,P,T); %eğitim 200000 epocr5 devam ediyor

Y = (sim(net_koord,P));
Y(1,:)=Y(1,:).*57;
Y(2,:)=Y(2,:).*32;
save('sadece_dgb6x20_ysa15','net_koord'); %ağ yapısı son hali ile
kaydediliyor
```

EK 10: Koordinat Tespiti Yapması Amacıyla Geliştirilen YSA'nın ayrık zamanlı test görüntüleri ile test edilmesi için hazırlanan Matlab yazılımı

```
clear all;clc
load sadece_dgb6x20_ysa9
load kontrol_icin_c_mtrs
load test_dgb6x20_2000
%%
for iii=1:400
test= (test_dgb6x20_2000(iii,:));
    test_120(1:1,1:120)=test(1:1,1:120);
    testt_120=test_120./255;
    test_2(1:1,1:2)=test(1:1,121:122);
    testt_2=test_2./1144;
testt=(vertcat(testt_120',testt_2'))';
Test=(testt)';
    y = (sim(net_koord, Test));
    %y(1,:)=y(1,:);
    %y(2,:)=y(2,:);
    b=y';
    c(iii,1)=round(b(1,1)*57);
    c(iii,2)=round(b(1,2)*32);
end

%% x y koordinatları ekran görüntüsü üzerine basılıyor
c1=((c*24)-12);
koordinat_goruntu=imread('matlab_koord.jpg');
figure,imshow(koordinat_goruntu)

hold on
asil_x1 = c1(20:20,3);bulnn_x1 = c1(1:20,1);asil_y1 =
c1(20:20,4);bulnn_y1 = c1(1:20,2);
asil_x2 = c1(40:40,3);bulnn_x2 = c1(21:40,1);asil_y2 =
c1(40:40,4);bulnn_y2 = c1(21:40,2);
asil_x3 = c1(60:60,3);bulnn_x3 = c1(41:60,1);asil_y3 =
c1(60:60,4);bulnn_y3 = c1(41:60,2);
asil_x4 = c1(80:80,3);bulnn_x4 = c1(61:80,1);asil_y4 =
c1(80:80,4);bulnn_y4 = c1(61:80,2);
asil_x5 = c1(100:100,3);bulnn_x5 = c1(81:100,1);asil_y5 =
c1(100:100,4);bulnn_y5 = c1(81:100,2);
asil_x6 = c1(120:120,3);bulnn_x6 = c1(101:120,1);asil_y6 =
c1(120:120,4);bulnn_y6 = c1(101:120,2);
asil_x7 = c1(140:140,3);bulnn_x7 = c1(121:140,1);asil_y7 =
c1(140:140,4);bulnn_y7 = c1(121:140,2);
asil_x8 = c1(160:160,3);bulnn_x8 = c1(141:160,1);asil_y8 =
c1(160:160,4);bulnn_y8 = c1(141:160,2);
asil_x9 = c1(180:180,3);bulnn_x9 = c1(161:180,1);asil_y9 =
c1(180:180,4);bulnn_y9 = c1(161:180,2);
asil_x10 = c1(200:200,3);bulnn_x10 = c1(181:200,1);asil_y10 =
c1(200:200,4);bulnn_y10 = c1(181:200,2);
asil_x11 =c1(220:220,3);bulnn_x11 = c1(201:220,1);asil_y11 =
c1(220:220,4);bulnn_y11 = c1(201:220,2);
asil_x12 = c1(240:240,3);bulnn_x12 = c1(221:240,1);asil_y12 =
c1(240:240,4);bulnn_y12 = c1(221:240,2);
asil_x13 = c1(260:260,3);bulnn_x13 = c1(241:260,1);asil_y13 =
c1(260:260,4);bulnn_y13 = c1(241:260,2);
```



```

asil_x14 = c1(280:280,3);bulnn_x14 = c1(261:280,1);asil_y14 =
c1(280:280,4);bulnn_y14 = c1(261:280,2);
asil_x15 = c1(300:300,3);bulnn_x15 = c1(281:300,1);asil_y15 =
c1(300:300,4);bulnn_y15 = c1(281:300,2);
asil_x16 = c1(320:320,3);bulnn_x16 = c1(301:320,1);asil_y16 =
c1(320:320,4);bulnn_y16 = c1(301:320,2);
asil_x17 = c1(340:340,3);bulnn_x17 = c1(321:340,1);asil_y17 =
c1(340:340,4);bulnn_y17 = c1(321:340,2);
asil_x18 = c1(360:360,3);bulnn_x18 = c1(341:360,1);asil_y18 =
c1(360:360,4);bulnn_y18 = c1(341:360,2);
asil_x19 = c1(380:380,3);bulnn_x19 = c1(361:380,1);asil_y19 =
c1(380:380,4);bulnn_y19 = c1(361:380,2);
asil_x20 = c1(400:400,3);bulnn_x20 = c1(381:400,1);asil_y20 =
c1(400:400,4);bulnn_y20 = c1(381:400,2);

```

```

    plot(bulnn_x1, bulnn_y1,
'o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'g', 'Marke
rSize', 8);
    plot(bulnn_x2, bulnn_y2,
'o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'y', 'Marke
rSize', 8);
    plot(bulnn_x3, bulnn_y3,
'o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b', 'Marke
rSize', 8);
    plot(bulnn_x4, bulnn_y4,
'o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'r', 'Marke
rSize', 8);
    plot(bulnn_x5, bulnn_y5,
'o', 'LineWidth', 2, 'MarkerEdgeColor', 'y', 'MarkerFaceColor', 'k', 'Marke
rSize', 8);
    plot(bulnn_x6, bulnn_y6,
'o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'g', 'Marke
rSize', 8);
    plot(bulnn_x7, bulnn_y7,
'o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'y', 'Marke
rSize', 8);
    plot(bulnn_x8, bulnn_y8,
'o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b', 'Marke
rSize', 8);
    plot(bulnn_x9, bulnn_y9,
'o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'r', 'Marke
rSize', 8);

plot(bulnn_x10,bulnn_y10,'o', 'LineWidth', 2, 'MarkerEdgeColor', 'y', 'Ma
rkerFaceColor', 'k', 'MarkerSize', 8);

plot(bulnn_x11,bulnn_y11,'o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'Ma
rkerFaceColor', 'g', 'MarkerSize', 8);

plot(bulnn_x12,bulnn_y12,'o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'Ma
rkerFaceColor', 'y', 'MarkerSize', 8);

plot(bulnn_x13,bulnn_y13,'o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'Ma
rkerFaceColor', 'b', 'MarkerSize', 8);

plot(bulnn_x14,bulnn_y14,'o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'Ma
rkerFaceColor', 'r', 'MarkerSize', 8);

plot(bulnn_x15,bulnn_y15,'o', 'LineWidth', 2, 'MarkerEdgeColor', 'y', 'Ma
rkerFaceColor', 'k', 'MarkerSize', 8);

```

```
plot(bulnn_x16,bulnn_y16,'o','LineWidth',2,'MarkerEdgeColor','k','Ma
rkerFaceColor','g','MarkerSize',8);

plot(bulnn_x17,bulnn_y17,'o','LineWidth',2,'MarkerEdgeColor','k','Ma
rkerFaceColor','y','MarkerSize',8);

plot(bulnn_x18,bulnn_y18,'o','LineWidth',2,'MarkerEdgeColor','k','Ma
rkerFaceColor','b','MarkerSize',8);

plot(bulnn_x19,bulnn_y19,'o','LineWidth',2,'MarkerEdgeColor','k','Ma
rkerFaceColor','r','MarkerSize',8);

plot(bulnn_x20,bulnn_y20,'o','LineWidth',2,'MarkerEdgeColor','y','Ma
rkerFaceColor','k','MarkerSize',8);
hold off
screen_size = get(0, 'ScreenSize');
f1 = figure(1);
set(f1, 'Position', [0 0 screen_size(3) screen_size(4)
],'menubar','none' );
set(gca, 'Visible', 'Off', 'Position',[0 0 1 1])
```

EK11: Deneklerin Görüntü Alması İçin Geliştirilen Arayüz

Matlab ortamında geliştirilen arayüz programı bileşenleri ile birlikte CD'de verilmiştir.

EK12: Çalışmada Eğitilen YSA'lar

Çalışmada göz bölgesinin bulunması ve ekranda bakılan nokta koordinatlarının tespit edilmesi amacıyla eğitilip kullanılan yapay sinir ağları ve testlerin yapılabilmesi için hazırlanan Matlab yazılımları CD'de verilmiştir.

EK13: Çalışmada göz bölgesinin ayrık/gerçek zamanlı bulunması aşamasında hazırlanan gif ve videolar

Bu aşamada hazırlanan gif ve videolar CD'de verilmiştir.

ÖZGEÇMİŞ

Bülent TURAN, 24.10.1971 de Manisa' da doğdu. İlk, orta ve lise eğitimini Manisa'da tamamladı. 1989 yılında başladığı GÜTEF Elektrik Eğitimi Bölümünü 1994 yılında bitirdi ve aynı yıl Milli Eğitim Bakanlığına bağlı Teknik okullarda Teknik Öğretmen olarak göreve başladı. 2006 yılında Gaziosmanpaşa Üniversitesi Zile Meslek Yüksekokulu Mekatronik Programında öğretim görevlisi oldu. 2009 yılında Sakarya Üniversitesi, Elektronik-Bilgisayar Eğitimi Bölümünde doktora eğitimine başladı. Evli ve 2 çocuk sahibidir.