



T.C.  
Uludağ Üniversitesi  
Fen Bilimleri Enstitüsü

**GÖZ KIRPMA TABANLI SÜRÜCÜ YORGUNLUK  
DEĞERLENDİRME SİSTEMİ TASARIMI**

**Soner Karagülmez**

**Yüksek Lisans Tezi**



**GÖZ KIRPMA TABANLI SÜRÜCÜ YORGUNLUK  
DEĞERLENDİRME SİSTEMİ TASARIMI**

**Soner Karagülmez**



T.C.  
ULUDAĞ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**GÖZ KIRPMA TABANLI SÜRÜCÜ YORGUNLUK DEĞERLENDİRME  
SİSTEMİ TASARIMI**

**Soner KARAGÜLMEZ**  
Doç. Dr. Ahmet Emir DİRİK  
(Danışman)

YÜKSEK LİSANS TEZİ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2017

## TEZ ONAYI

Soner Karagülmez tarafından hazırlanan "GÖZ KIRPMA TABANLI SÜRÜCÜ YORGUNLUK DEĞERLENDİRME SİSTEMİ TASARIMI" adlı tez çalışması aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Uludağ Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

**Danışman** : Doç. Dr. Ahmet Emir Dirik

**Başkan :** Doç. Dr. Ahmet Emir Dirik

**Üye:** Yrd. Doç. Dr. Ersen Yılmaz

Yrd. Doç. Dr. Cemal Hanilçi

İmza

Yukarıdaki sonucu onaylarım



**Prof. Dr. Ali BAYRAM**

**Enstitü Müdürü**

./././... (Tarih)

**U.Ü. Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;**

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
  - görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
  - başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
  - atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
  - kullanılan verilerde herhangi bir tahrifat yapmadığımı,
  - ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı
- beyan ederim.**

.././....

**İmza**

**Ad ve Soyadı**



## ÖZET

Yüksek Lisans Tezi

### GÖZ KIRPMA TABANLI SÜRÜCÜ YORGUNLUK DEĞERLENDİRME SİSTEMİ TASARIMI

**Soner KARAGÜLMEZ**

Uludağ Üniversitesi  
Fen Bilimleri Enstitüsü  
Elektrik-Elektronik Mühendisliği Anabilim Dalı

**Danışman:** Doç. Dr. Ahmet Emir DİRİK

Günümüzde birçok insan trafik kazalarında hayatlarını kaybetmektedir. Ölümcül trafik kazalarının en önemli nedeni olarak sürücülerin uykusuzluğu ve yorgunluğu gösterilmektedir. Bu çalışmada, araç sürücülerinin yorgun ya da dinç olduğuna karar veren yeni bir sistem sunulmaktadır. Önerilen sistem dört aşamadan oluşmaktadır. İlk aşamada sayısal bir kamera üzerinden göz bölgeleri tespit edilerek takip edilmektedir. İkinci aşamada takip edilen gözün açık ya da kapalı olup olmadığı kestirilmektedir. Bu aşama sonunda, sürücünün göz durumu ve zaman bilgisini veren bir göz durum vektörü oluşturulmaktadır. Üçüncü aşamada bu göz durum vektörü üzerinden 7 adet istatistiksel öznitelik çıkarılmaktadır. Son aşamada ise bu özniteliklerin kullanışlı olanları elenmekte ve elenen bu öznitelikler bir yapay sinir ağının değerlendirmesine sunularak sürücünün yorgun olup olmadığına karar verilmektedir. Deneysel sonuçlar, kişilerin yorgunluk derecelerinin 25 video ve 5 farklı kişi üzerinden %96'lık bir başarımla doğru olarak ölçülebildiğini göstermektedir.

**Anahtar Kelimeler:** Sürücü yorgunluk tespit sistemi, sürücü uykusuzluk tespit sistemi, göz kırpma analizi, göz durum analizi, Viola-Jones yöntemi

**2017, xi + 70 sayfa**

## ABSTRACT

MSc Thesis Thesis

EYE-BLINK BASED DRIVER FATIGUE ASSESSMENT SYSTEM DESIGN

**Soner KARAGÜLMEZ**

Uludağ University  
Graduate School of Natural and Applied Sciences  
Department of Electric-Electronic Engineering

**Supervisor:** Doç. Dr. Ahmet Emir DİRİK

Today, so many people die because of traffic accidents. It is shown that drowsiness and fatigue are the most important reasons of fatal traffic accidents. In this study a new system deciding whether a driver is fatigued or not is proposed. The proposed system consists of four stages. In the first stage, eye regions are found and tracked by a digital camera. In the second stage, it is predicted that the tracked eye is open or close. At the end of the stage, an eye state vector giving eye state and time information is created. In the third stage, seven statistical features are extracted from the eye-state vector. At the last stage, effective features are eliminated and presented as an evaluation of a neural network and it decides whether the driver is fatigued or not. Experimental results show that people's fatigue levels are correctly measured 96% accuracy on 25 videos and 5 different people.

**Key words:** Driver fatigue detection system, driver drowsiness detection system, eye-blink analysis, eye-state analysis, Viola-Jones method  
**2017, xi + 70 pages.**

## TEŐEKKÜR

Yakın zamanda kaybettiđim fakat desteđini sürekli hissettiđim anneme, süreç boyunca benden desteđini esirgemeyen aileme, desteđiyle beraber alıřmada deney kısımları için veri kümesini oluřturmada yardımcı olan eřim Pirnur Karagölmez, yeđenim Buse Kapucubař, iř arkadaşlarım Göke Kolay, Ömer Alın ve İbrahim Sara'a teőekkürlerimi sunarım.



Soner Karagölmez

.../.../....



## İÇİNDEKİLER

	Sayfa
ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
SİMGE VE KISALTMALAR DİZİNİ .....	vi
ŞEKİLLER DİZİNİ.....	viii
ÇİZELGELER DİZİNİ .....	x
1. GİRİŞ .....	1
2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI .....	4
2.1. KLT Nesne Takip Algoritması .....	4
2.2. Viola-Jones Algoritması ile Kaskad Sınıflandırıcı Eğitimi .....	7
2.3. Yapay Sinir Ağları .....	12
2.3.1. Yapay sinir ağı elemanları .....	13
2.3.2. Yapısına göre yapay sinir ağları.....	17
2.3.3. Öğrenme algoritmalarına göre yapay sinir ağları.....	18
2.3.4. LM algoritması ve LM algoritmasında kullanılan yöntemler .....	18
2.4. Leave-One-Out(Birini dışarda bırak) yöntemi ile çapraz doğrulama .....	30
3. MATERYAL VE YÖNTEM .....	31
3.1. Göz Bölgelerinin Bulunması ve Takip Edilmesi .....	33
3.2. Göz Açıklık – Kapalılığının (Göz Durumunun) Çıkarımı .....	34
3.3. İstatistiksel Özniteliklerin Çıkarılması.....	38
3.3.1. Göz kırpm sıklığı.....	38
3.3.2. Toplam göz kapalılık süresi .....	39
3.3.3. Gözün kapalı kaldığı sürelerin ortalaması.....	39
3.3.4. Gözün kapalı kaldığı sürelerin standart sapması.....	39
3.3.5. Toplam göz açıklık süresi .....	39
3.3.6. Gözün açık kaldığı sürelerin ortalaması.....	40
3.3.7. Gözün açık kaldığı sürelerin standart sapması.....	40
3.4. Yorgunluk Tespiti .....	40
4. BULGULAR .....	48
4.1. Göz Açıklık Kapalılık Bulguları .....	48
4.2. Sürücü Yorgunluk Bulguları .....	50

4.2.1. Deney 1 bulguları .....	50
4.2.2. Deney 2 bulguları .....	62
5. TARTIŞMA ve SONUÇ .....	64
KAYNAKLAR .....	66
ÖZGEÇMİŞ .....	70



## SİMGELER ve KISALTMALAR DİZİNİ

### Simgeler

$A$

$B$

$C$

$D_0$

$D_1$

$E$

$f(x)$

$F_1$

$F_2$

$F_3$

$F_4$

$F_5$

$F_6$

$F_7$

$G$

$h(x, f, p, \theta)$

$H$

$I$

$I_{int}$

$J$

$p$

$R$

$S(x,)$

$s$

$T(x)$

$t$

$u$

$V$

$W(x; p)$

$w_i$

$X_i$

$x$

$\tilde{x}$

$x'$

$\tilde{x}'$

$\bar{x}$

$y$

$\tilde{Z}$

$\tau_0(i)$

$\tau_1(i)$

### Açıklamalar

Aktivasyon fonksiyonu

Afin dönüşüm için keyfi dönüşüm matrisi

Kuvvetli sınıflandırıcı

Gözün toplam kapanma sayısı

Gözün toplam açık kalma sayısı

YSA hata fonksiyonu

Görüntü özneliği

Göz kırpma sıklığı özneliği

Toplam göz kapalılık süresi özneliği

Gözün kapalı kaldığı sürelerin ortalaması özneliği

Gözün kapalı kaldığı sürelerin standart sapması özneliği

Toplam göz açıklık süresi özneliği

Gözün açık kaldığı sürelerin ortalaması özneliği

Gözün açık kaldığı sürelerin standart sapması özneliği

Gradyan vektörü

Zayıf sınıflandırıcı

Hessien matrisi

Giriş imgesi

Integral imge

Jacobian matrisi

Çarpıtma katsayısı

Ortanormal dönüşüm matrisi

İmgenin kümülatif satır toplamı

Oransal benzerlik faktörü

Şablon imge

Öteleme dönüşüm vektörü

Polarlık

Toplam video süresi(sn)

Çarpıtılma kümesi

Yapay sinir ağı ağırlık değerleri

Yapay sinir ağı giriş verileri

İmge piksel konum vektörü(Kartezyen)

İmge piksel konum vektörü(Homojen)

Dönüşüme uğramış imge koordinat

vektörü(kartezyen)

Dönüşüme uğramış imge koord. vektörü(homojen)

Artırılmış imge koord. vektörü

Pozitif ve negatif sınıflandırıcı örnekleri

İzdüşüm dönüşüm matrisi

Göz kapalı kaldığı süreler

Gözün açık kaldığı süreler

$\omega_{t,i}$	Zayıf sınıflandırıcı ağırlık değeri
$e_i$	İmge doğru-yanlış sınıflandırılma değeri
$\alpha_t$	t sınıflandırıcısına uygulanan ağırlık değeri
$\alpha$	Yapay sinir ağı eğitim sabiti
$\mu$	LM algoritması skaler kombinasyon katsayısı
$\zeta$	İmge döndürme açısı
$\nabla$	I imgesinin gradyanı
$\theta$	Sınıflandırıcı eşik değeri
$\varepsilon_t$	İyi zayıf sınıflandırıcı
$\beta_t$	Adaboost algoritması ağırlık güncelleme katsayısı
$\delta$	

### Kısaltmalar

WHO  
TÜİK  
SVM  
YSA  
PCA  
LDA  
KLT  
MSE(OXH)  
TP  
TN  
FP  
FN  
LM  
NET  
PERCLOS

### Açıklama

Dünya sağlık örgütü  
Türkiye istatistik kurumu  
Destek vektör makineleri  
Yapay sinir ağları  
Temel bileşenler analizi  
Lineer diskriminant analizi  
Kanade-Lucas-Tomasi nesne takip algoritması  
Ortalama kare hatası  
Gerçekçi pozitif değeri  
Gerçekçi negatif değeri  
Yalancı pozitif değeri  
Yalancı negatif değeri  
Levenberg-Marquardt algoritması  
Yapay sinir ağı toplam fonksiyonu çıkışı  
Göz kapalılık yüzdesi

## ŞEKİLLER DİZİNİ

### Sayfa

Şekil 2.1.	2,3 ve 4 dikdörtgenli temel HAAR öznitelikler.....	7
Şekil 2.2.	Leinhart ve Maydt kenar(a), çizgi(b) ve merkez çevreli(c) öznitelikleri.....	8
Şekil 2.3.	D dikdörtgeninin pixel yoğunluk toplamları a, b, c, d noktalarının integral imgeleri kullanılarak hesaplanabilir. D bölgesinin pixel yoğunluk toplamları $I_{int}(d)+I_{int}(a)-(I_{int}(b)+I_{int}(c))$ olarak gösterilebilir. ....	9
Şekil 2.4.	Haar kaskad sınıflandırıcı gösterimi .....	11
Şekil 2.5.	Yapay sinir hücresi ile biyolojik sinir hücresinin karşılaştırmalı gösterimi .	12
Şekil 2.6.	Lineer doğrusal aktivasyon fonksiyonu gösterimi.....	14
Şekil 2.7.	Basamak aktivasyon fonksiyonu gösterimi .....	15
Şekil 2.8.	Parçalı-doğrusal aktivasyon fonksiyonu gösterimi.....	15
Şekil 2.9.	Sigmoid aktivasyon fonksiyonu gösterimi .....	16
Şekil 2.10.	Hiperbolik tanjant aktivasyon fonksiyonu gösterimi.....	16
Şekil 2.11.	2 adet giriş katmanı nöronu, 1 adet ara/gizli katman ve 3 adet ara/gizli katmanı nöronu, 1 adet çıkış katmanı nöronundan oluşan basit bir yapay sinir ağı .....	17
Şekil 2.12.	Basit bir lineer regresyon modeli gösterimi.....	19
Şekil 2.13.	Her bir YSA nöronunun parametreleri ve YSA modelinin geri kalanına karşılık gösterimi .....	26
Şekil 3.1.	Göz durum kestirimi akış diyagramı .....	31
Şekil 3.2.	Yorgunluk derecesi kestirimi akış diyagramı .....	32
Şekil 3.3.	Aynı imge için [16] (sol) ve [17] (sağ) referanslarındaki algoritmaların uygulanmasıyla elde edilen göz bölgesi sınıflandırma sonuçları .....	34
Şekil 3.4.	Göz bölgelerinin kırılması akış diyagramı.....	34
Şekil 3.5.	<i>Cascade Trainer</i> uygulamasında gösterilen pozitif örnekler.....	35
Şekil 3.6.	<i>Cascade Trainer</i> uygulamasında gösterilen negatif örnekler ve parametre değerleri .....	36
Şekil 3.7.	Zaman damgasıyla örnek göz durum vektörü.....	37
Şekil 3.8.	Eğitilen özel Haar kaskad sınıflandırıcının testi için gerçek zamanlı olarak yazılmış olan MATLAB uygulamasının görsel çıktısı .....	37
Şekil 3.9.	Temsili olarak göz durum vektörü üzerinde gözün kapandığı anların gösterimi .....	38
Şekil 3.10.	Yapay sinir ağı modeli parametrelerinin iyileştirilmesinde kullanılan sözde kod gösterimi. ....	41
Şekil 3.11.	Yorgunluk tespiti için kullanılan yapay sinir ağı modeli.....	42
Şekil 3.12.	Giriş ve hedef verilerin uygulama arayüzüne aktarılması .....	43
Şekil 3.13.	Yapay sinir ağı modelinin eğitim, test ve doğrulama verisi sayılarının uygulama üzerinden gösterimi.....	44
Şekil 3.14.	Uygulama üzerinde ne kadar gizli katman nöronu bulunacağını gösterimi ve temsili yapay sinir ağı .....	45
Şekil 3.15.	Eğitim algoritmasının seçimi ve ağın eğitilmesi.....	45

Şekil 3.16. Yapay sinir ağının opsiyonel testlerle değerlendirilmesinin yapılması .....	46
Şekil 3.17. Sonuçların MATLAB'e aktarılması ve MATLAB dosyası olarak çıktısının alınması .....	47
Şekil 4.1. Deneye katılan kişilerin doğru olarak değerlendirilen açık ve kapalı göz örnekleri .....	49
Şekil 4.2. Deneye katılan kişilerin hatalı olarak değerlendirilen açık ve kapalı göz örnekleri. 1. sütun gözün kapalı iken açık, 2. sütun gözün açık iken kapalı olarak değerlendirilen hatalı örneklerini temsil etmektedir .....	50



## ÇİZELGELER DİZİNİ

### Sayfa

Çizelge 2.1. Temel 2D uzamsal dönüşümler (SD: Serbestlik derecesi) .....	6
Çizelge 2.2. Yapay sinir hücresi ve biyolojik sinir hücresi karşılaştırması .....	12
Çizelge 3.1. Deneye katılanların fiziksel özellikleri .....	32
Çizelge 4.1. Göz açıklık kapalılık doğruluğu.....	48
Çizelge 4.2. İterasyon 1 için F1 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	51
Çizelge 4.3. İterasyon 1 için F2 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	52
Çizelge 4.4. İterasyon 1 için F3 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	52
Çizelge 4.5. İterasyon 1 için F4 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	53
Çizelge 4.6. İterasyon 1 için F4 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	53
Çizelge 4.7. İterasyon 1 için F6 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	54
Çizelge 4.8. İterasyon 1 için F7 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	54
Çizelge 4.9. İterasyon 2 için F1 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	55
Çizelge 4.10. İterasyon 2 için F2 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	56
Çizelge 4.11. İterasyon 2 için F3 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	56
Çizelge 4.12. İterasyon 2 için F4 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	57
Çizelge 4.13. İterasyon 2 için F5 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	57
Çizelge 4.14. İterasyon 2 için F7 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	58
Çizelge 4.15. İterasyon 3 için F1 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	59
Çizelge 4.16. İterasyon 3 için F2 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	59
Çizelge 4.17. İterasyon 3 için F4 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	60
Çizelge 4.18. İterasyon 3 için F5 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	60
Çizelge 4.19. İterasyon 3 için F7 özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları .....	61
Çizelge 4.20. Deney 1 hata matrisi .....	62

Çizelge 4.21. Deney 2 hata matrisi ..... 62





## 1. GİRİŞ

Trafik kazaları günümüzde insan hayatı için büyük tehdit unsuru oluşturmaktadır. Dünya Sağlık Örgütü'nün (WHO) istatistiksel verilerine göre trafik kazaları sonucu her yıl 1,3 milyon insan hayatını kaybetmekte ve 20 - 50 milyon arası insan yaralanmaktadır (Anonim 2016a). Amerika Otomobil Birliği (The American Automobile Association) verilerine göre tüm kazaların %7'si ve ölümcül trafik kazalarının %21'i uykulu sürücülerin karıştığı kazalar olarak gösterilmiştir. Ayrıca her 3 sürücüden biri hayatlarında bir kez direksiyon başında uyuyakalmış ve her 10 sürücüden biri de bir önceki yıl direksiyon başında uyuyakaldığını itiraf etmiştir (Anonim 2016b). TÜİK verilerine göre Türkiye'de ise 2015 yılında kayda geçen toplam 1.313.359 adet trafik kazası meydana gelmiştir. Bu kazaların 183.011 adedi ölümlü trafik kazaları olarak kayıtlara geçmiştir. Kazaların %89,3'ünün sürücü kusurlarından olduğu raporlanmıştır (Anonim 2016c). Bu ve bunun gibi dramatik veriler sürücü uyku/yorgunluk değerlendirme sisteminin insan hayatı açısından ne kadar önemli olabileceğini göstermektedir. Günümüzde bu sistemler birçok araştırmacı tarafından farklı yaklaşımlarla ele alınmıştır. Bunlar ölçüm yöntemlerine göre beş farklı kategoride değerlendirilebilir (Sahayadhas ve ark. 2012, Colic ve ark. 2014) :

- *Öznel teknikler:* Anketlere, anketi gerçekleştiren kişilere, uykunun elektro-fizyolojik ölçümlerine dayanır.
- *Araç tabanlı teknikler:* Direksiyon simidi hareketleri, gaz pedalı basınç ölçümleri, aracın yol çizgisine göre mesafesi gibi araç üzerinden alınan ölçümlere dayanır.
- *Davranışsal teknikler:* Sürücünün göz kapalılık ve göz kırpma miktarı, baş pozisyonu ve esnemesi gibi davranışsal özelliğini ele alır.
- *Fizyolojik teknikler:* Sürücünün kan basıncı, kalp atışı ve beyin dalgaları gibi fizyolojik ölçümlerine dayanır.
- *Hibrit Teknikler:* Sürücü uyku değerlendirme sisteminin daha yüksek doğrulukta olması için bahsi geçen diğer yöntemlerin birleşimlerinden oluşur.

Her tekniğin kendine göre belirli zayıf noktaları bulunmaktadır. Yol ve iklim koşullarının araç tabanlı teknikleri yanıltabildiği gibi öznel teknikler soru ve anketlere bağlı olduğundan yanıltıcı olabilir. Fizyolojik teknikler; genel olarak yüksek oranda doğruluk

sağlar fakat yorgunluk ölçümü için sürücünün, içerisinde sensör barındıran bir nesneyi giymesini ya da vücudunun çeşitli yerlerine yapıştırmasını gerektirebilir. Davranışsal teknikler ise ışık yoğunluğundan ve miktarından etkilenebilmektedir. Bu tezde doğruluk oranının yüksek olması (Bergasa ve ark. 2006, Sahayadhas ve ark. 2012) ve diğer tekniklere göre uygulanabilirliğinin daha kolay olması sebebiyle davranışsal sürücü yorgunluk değerlendirme sistemi önerilmektedir (Sahayadhas ve ark. 2012). Davranışsal sürücü uyku değerlendirme sistemleri göz kırpma miktarı (Akrouit ve Mahdi 2013, Murukesh ve Abhan 2015), göz açık-kapalı durumu (Golgiyaz ve ark. 2014), kafa pozisyonu ve esneme (Abtahi 2012), kaş ve kirpik (Hayırlı 2005) gibi yüz üzerinden alınan imge çeşidine göre kendi içlerinde farklılıklar teşkil eder. Kaydedilen imgeler üzerinde yapılan analizle birlikte karar aşamasında birçok yöntem kullanılmıştır. Bunlar SVM (Destek Vektör Makineleri) (Shuyan ve Gangtie 2009, Vural 2009), Yapay sinir ağları (Girit 2014), PCA (temel bileşeler analizi), LDA (lineer diskriminant analizi) (Golgiyaz 2013) ve bulanık mantık yöntemleridir.

Türkiye’de bu konuda Sabancı Üniversitesinde Vural (2009) tarafından yapılan “Sürücü Yorgunluğunun Video Tabanlı Tespiti” başlıklı Doktora tezi, ODTÜ’de Girit (2014) tarafından yapılan “Görüntü İşleme Kullanılarak Yorgun Sürücü Tespiti” başlıklı Yüksek lisans tezi ve Fırat Üniversitesi’nde Golgiyaz (2013) tarafından yapılan “Gerçek Zamanlı Uykulu Sürüş Algılama Sistemi” başlıklı Yüksek lisans tezi önemli çalışmalar arasındadır. Vural (2009), yorgunluğun tespiti için çene yukarı kaldırma, dış kaş / iç kaş kaldırma gibi yüzle ilgili hareket kodlama sisteminin (FACS) bazı parçalarını ve SVM sınıflandırıcısını kullanmıştır. Bir Windows işletim sistemine sahip bilgisayar üzerinde kurulu benzetim oyunu kullanarak 4 denek üzerinden gece yarısı başlayan 3 saatlik görev sürüşleri ile deney düzeneğini hazırlamıştır. Girit (2014), yorgunluğun tespiti için göz bölgelerinin doğru bulunmasına ve histogram eşitleme yöntemi ile ışık duyarlılığının azaltılmasına çalışmış, elde ettiği göz bölgeleri imgelerini gri ölçekli hale getirmiş daha sonra uygun ölçüde tekrar boyutlandırarak yapay sinir ağları girişlerine aktarmıştır. Yapay sinir ağı çıkışları göz açık/yarı-açık/kapalı durumlarından birine karşılık rakamsal bir değere karşılık gelmiştir. Bunların 60 saniye için ortalama değeri ise kişinin yorgun olup olmadığını değerlendirmek için kullanılmıştır. Deney düzeneği bir bilgisayar benzetim ortamı, 4 denek ve gece yarısı 3 saati aşkın sürüş görevlerinden oluşmaktadır. Golgiyaz (2013) ise yüz tespiti ile birlikte yüz/kişi tanıma bilgisini çıkarmış, göz durumu

geçmişi ve maskelenmiş göz görüntüsü ile PERCLOS (göz kapalılık oranı) hesaplaması yaparak kişinin göz kırpma oranı ve bakış yönüne göre sürücünün ikaz edilip edilmeyeceğine karar vermiştir. Golgiyaz'ın (2013) sürücü tanıma ve sürücü özelliklerini kullanarak sistem parametrelerinin kişiden kişiye adaptif uygulama yaklaşımı literatürdeki diğer çalışmalara göre daha özeldir. Golgiyaz (2013), tezinde içerisinde 20 farklı kişi üzerinden 4 farklı durum için 80 adet video barındıran ZJU veri tabanını kullanmıştır (Anonim 2013).

Bu tezde ilk aşama olarak bir web kamerası aracılığı ile kaydedilen/aktarılan video görüntülerinden Shiqi'nin (2016) Haar kaskad sınıflandırıcısı kullanılarak göz bölgeleri bulunur ve KLT (Kanade-Lucas-Tomasi) öznitelik takip algoritması ile bu göz bölgeleri takip edilir. Daha sonra takip edilen bu göz bölgelerinde özel Haar kaskad sınıflandırıcı yardımıyla gözün açıklık ve kapalılığı bulunur. Üçüncü aşamada bu göz açıklık ve kapalılık bilgisi üzerinden istatistiksel olarak yedi adet öznitelik çıkarılır. Son aşamada bu özniteliklerin kullanışlı olanları bulunur ve yapay sinir ağı karar algoritmasının girişlerini oluşturur. Yapay sinir ağı çıkışı ise o örnek için yorgun olup olmadığını ifade eder. Önerilen sistemin eğitimi ve testi için 6 farklı denek üzerinden 3 aylık bir zaman periyodunda günün farklı zamanlarında çekilmiş toplam 66 adet yaklaşık 2 dk. süren video görüntüleri kullanılmıştır. Donanımsal deney düzeneği olarak bir adet web kamerası ve Windows işletim sistemli bir bilgisayar kullanılmıştır.

## 2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI

### 2.1. KLT (Kanade-Lucas-Tomasi) Nesne Takip Algoritması

Öznitelik tabanlı bir takip algoritması olan KLT’de amaç, takip edilecek uygun özniteliklerin seçilmesi (Lucas ve Kanade 1981, Tomasi ve Kanade 1991, Shi ve Tomasi 1994, Anonim 2017) ve bunların nasıl takip edileceğinin (Tomasi ve Kanade 1991, Shi ve Tomasi 1994, Anonim 2012, Anonim 2017) bulunmasıdır. Nesne takibi, şablon görüntünün 2D uzamsal dönüşümler sayesinde çerçeveler arasında hizalanmasına çalışılarak sağlanır.

İmgedeki pixel koordinatları; kartezyen koordinat sisteminde  $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$ , homojen koordinat sisteminde  $\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{w} \end{bmatrix}$  ve dönüşüme uğramış imge piksel koordinatları  $\mathbf{x}'$  olarak ifade edilmek üzere temel 2D uzamsal dönüşümler:

**Öteleme:** İmgenin kartezyen koordinat düzleminde  $\mathbf{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$  kadar yer değiştirmesidir.

$\mathbf{x}' = \mathbf{x} + \mathbf{t}$  ya da  $\bar{\mathbf{x}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$  artırılmış (augmented) vektör olmak üzere

$\bar{\mathbf{x}}' = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \bar{\mathbf{x}}$  olarak ifade edilir.

**Öklidyen:** 2D katı cisim hareketi olarak da bilinen bu dönüşüm, dönme ve öteleme hareketinin birleşiminden oluşmuştur.  $\xi$  dönme açısı,  $\mathbf{t}$  öteleme vektörü ve  $\mathbf{R} = \begin{bmatrix} \cos\xi & -\sin\xi \\ \sin\xi & \cos\xi \end{bmatrix}$  ortonormal dönüşüm matrisi (saat yönünde dönüş için) olmak üzere  $\mathbf{x}' =$

$\mathbf{R}\mathbf{x} + \mathbf{t}$  yada  $\bar{\mathbf{x}}$  artırılmış vektör olmak üzere  $\bar{\mathbf{x}}' = \begin{bmatrix} \cos\xi & -\sin\xi & t_x \\ \sin\xi & \cos\xi & t_y \\ 0 & 0 & 1 \end{bmatrix} \bar{\mathbf{x}}$  olarak ifade

edilir. Öklidyen dönüşümde, imge koordinatlarının Öklid uzaklıkları korunur.

**Benzerlik:** Oransal dönme adıyla da bilinen benzerlik dönüşümü, öteleme ve dönme hareketleri ile oransal değişimin birleşiminden oluşmuştur.  $s$  keyfi bir oransal faktör olmak üzere  $\mathbf{x}' = \mathbf{sR}\mathbf{x} + \mathbf{t}$  yada  $\bar{\mathbf{x}}$  artırılmış vektör olmak üzere  $\bar{\mathbf{x}}' =$

$\begin{bmatrix} a & -b & t_x \\ b & a & t_y \\ 0 & 0 & 1 \end{bmatrix} \bar{x}$  olarak ifade edilir.  $a$  ve  $b$  deęişkenleri  $s$  oransal faktörünün elemanlarıdır. Bu dönüşümde çizgiler arasındaki açıların deęişmemesi esastır.

**Afin:** Afin dönüşümler; öteleme, dönme, oransal deęişim ve kayma hareketlerinin birleşiminden oluşmuştur.  $B$  keyfi bir dönüşüm matrisi ve  $\bar{x}$  artırılmış vektör olmak üzere






$x' = B\bar{x}$  ya da  $\bar{x}' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ 0 & 0 & 1 \end{bmatrix} \bar{x}$  olarak ifade edilir. Afin dönüşümde paralel çizgiler, paralel olarak kalır.

**İzdüşüm:** Perspektif dönüşümü ya da homografi dönüşümü adı da verilen bu dönüşüm  $\tilde{Z}$  keyfi bir homojen  $3 \times 3$ 'lük dönüşüm matrisi olmak üzere  $\tilde{x}' = \tilde{Z}\tilde{x}$  olarak ifade edilir. Yalnız burada matrislerin homojen olduğuna dikkat edilmelidir ve  $x'$  deęerlerinin bulunması için  $\tilde{x}'$  vektörünün son elemanına bölünerek normalize edilmesi gerekmektedir (Bkz. Denklem 2.1).

$$\tilde{x}' = (\tilde{x}, \tilde{y}, \tilde{w}) = \tilde{w}(x, y, 1) = w\bar{x} \quad (2.1)$$

Temel 2D uzamsal dönüşümlerin parametrik ifadeleri ve Jacobien matris ifadeleri Çizelge 2.1'de gösterilmiştir (Szeliski 2010).

**Çizelge 2.1.** Temel 2D uzamsal dönüşümler (SD: Serbestlik derecesi)

Dönüşüm Çeşidi	Dönüşüm Matrisi	SD	Korunan bilgi	İkon	Parametreler (p)	Jacobien (J)
Öteleme	$[I   t]_{2 \times 3}$	2	yönelim		$(t_x, t_y)$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Öklidyen	$[R   t]_{2 \times 3}$	3	uzunluklar		$(t_x, t_y, \theta)$	$\begin{bmatrix} 1 & 0 & -s_\theta x - c_\theta y \\ 0 & 1 & c_\theta x - s_\theta y \end{bmatrix}$
Benzerlik	$[sR   t]_{2 \times 3}$	4	açılar		$(t_x, t_y, a, b)$	$\begin{bmatrix} 1 & 0 & x & -y \\ 0 & 1 & y & x \end{bmatrix}$
Afin	$[B]_{2 \times 3}$	6	paralellik, düz çizgiler ve açılar		$(t_x, t_y, a_{00}, a_{01}, a_{10}, a_{11})$	$\begin{bmatrix} 1 & 0 & x & y & 0 & 0 \\ 0 & 1 & 0 & 0 & x & y \end{bmatrix}$
İzdüşümsel	$[\tilde{Z}]_{3 \times 3}$	8	Düz çizgiler		$(z_{00}, z_{01}, \dots, z_{21})$	(Bkz. Szeliski, 2010)

$I(x)$ ,  $x$  imge koordinatlarını içeren sütun vektörü olmak üzere giriş imgesi;  $W(x; p)$ ,  $p$  parametre vektörü olmak üzere çarpıtılma kümesi;  $T(x)$  şablon görüntü;  $\nabla I$ ,  $I$  imgesinin gradyanı ve  $H^{-1}$  Hessian matrisinin tersi olmak üzere algoritma adımları:

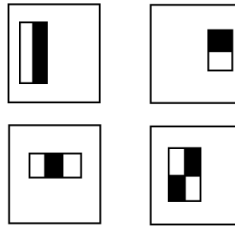
- $I$  imgesinin  $W(x; p)$  kadar çarpıtılması:  $I(W(x; p))$
- Şablon imgeden çarpıtılmış imgenin çıkarılması:  $[T(x) - I(W(x; p))]$
- Şablon imgenin gradyanının bulunması:  $\nabla I$
- Jacobian hesabı:  $\frac{\partial W}{\partial p}$  (Bkz. Çizelge 2.1)
- En dik iniş hesabı:  $\nabla I \frac{\partial W}{\partial p}$
- Ters Hessian hesabı:  $H^{-1}$
- En dik inişin hata ile çarpımı:  $\sum_x \left[ \nabla I \frac{\partial W}{\partial p} \right]^T [T(x) - I(W(x; p))]$
- Çarpıtılma değişimi miktarının hesaplanması:
 
$$\Delta p = H^{-1} \sum_x \left[ \nabla I \frac{\partial W}{\partial p} \right]^T [T(x) - I(W(x; p))]$$
- Parametre güncellemesi:  $p \rightarrow p + \Delta p$

şeklindedir. Çarpıtılma değışimi miktarı mevcut çarpıtılma değeriine eklenerek parametre güncellenir ve algoritma bir sonraki iterasyona hazırlanır. İterasyon, belirtilen bir eşik değerin altına düşene kadar tekrarlanır. Hessian matrisinin tersi iterasyon kararlılığını doğrudan etkilemektedir. Takip edilmesi uygun şablon pencereler (öznitelikler)  $T(x)$  Hessian matrisi tersinin özdeğerlerinin büyük değeri olduğu homojen ve dokusal yapıda olmayan pencerelerdir. Homojenlik değışimsizliği, dokusallık tek yönlü değışimi beraberinde getirdiğinden uzamsal dönüşümde kararsızlığa neden olabilir. Pencereler üzerindeki uzamsal dönüşüm için en uygun seçilecek noktalar da Hessian matrisinin yüksek özdeğerlere sahip olduğu çarpıtılmanın fazla olduğu noktalardır.

## 2.2. Viola-Jones Algoritması ile Kaskad Sınıflandırıcı Eğitimi

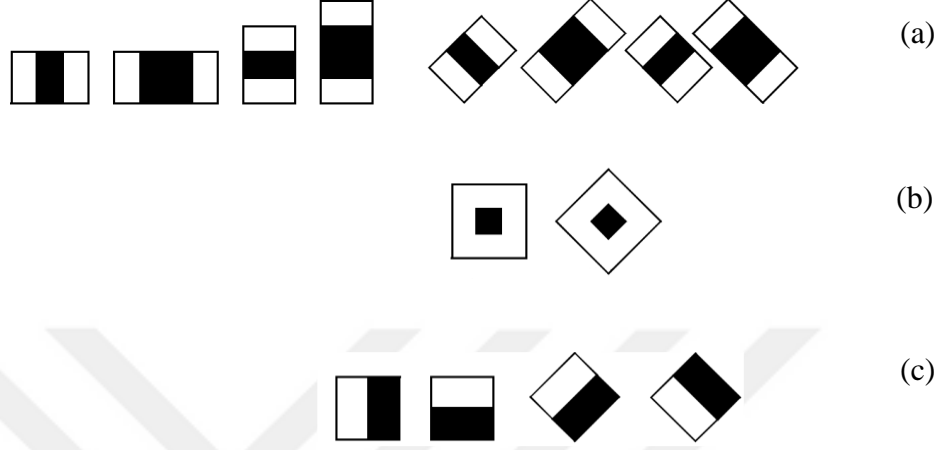
Viola ve Jones (2001), nesne bulma algoritmalarının hesap yükünü azaltan ve gerçek zamanlı olarak çalışabilme kabiliyetine getiren güçlü bir algoritma geliştirmişlerdir (Grauman 2008). Algoritma 4 ana bölümden oluşur. Bu bölümler Haar özniteliklerin seçimi, integral imgenin oluşturulması, uyarlanabilir yükseltme öğrenimi ve Haar sınıflandırıcı eğitimi bölümleridir. Aşağıdaki alt başlıklarda ayrıntılı olarak ele alınmıştır.

**Haar özniteliklerin seçimi:** Tüm insan yüzlerinin göz bölgelerinin, yanaklarının üst kısımlarından daha karanlık olması ya da burun bölgesinin, gözlerden daha aydınlık olması gibi ortak özellikler taşımasından yola çıkarak Viola ve Jones verilen bir imgeden yüzü tespit edebilmek için 4 temel öznitelik belirlemiştir. Bu öznitelikler içerisinde; beyaz dikdörtgenlerdeki pixel yoğunlukları toplamının, siyah dikdörtgenlerdeki pixel yoğunlukları toplamından fazla olduğu düşünülmüş, farklarının miktarı ise o bölgede aranan nesnenin olup olmadığı kararını vermek için kullanılmıştır (Bkz. Şekil 2.1).



Şekil 2.1. 2,3 ve 4 dikdörtgenli temel HAAR öznitelikler

Doğruluğu artırmak için bu öznitelikler Leinhart ve Maydt (2002) tarafından farklı dikdörtgenler ve dikdörtgenlerin dönel versiyonları eklenerek geliştirilmiştir (Bkz. Şekil 2.2).



**Şekil 2.2.** Leinhart ve Maydt kenar(a), çizgi(b) ve merkez çevreli(c) öznitelikleri

**Integral imgenin oluşturulması:** Haar özniteliklerin tüm imge üzerinde farklı büyüklük ve oranlarda uygulanması hesaplama yükünü artırır. Özniteliklerin hesaplama yükünü azaltmak için piksellerin birbirleri arasındaki referans toplam değerleri kullanılarak integral görüntü yaklaşımı geliştirilmiştir. Bu yöntemle göre; her pixelin imgenin sol üst noktasına kadar yoğunluk toplamı hesaplanarak, istenilen alanı ifade edecek şekilde fark denklemi çıkarılır (Bkz. Şekil 2.3).  $(x,y)$  lokasyonunda  $I(x,y)$  giriş görüntüsünün  $I_{int}(x,y)$  integral görüntüsü Denklem 2.2 ile ifade edilir ve toplam işlemleri tekrar düzenlenip yinelemeli işlemlerle hesap yükü azaltılabilir.

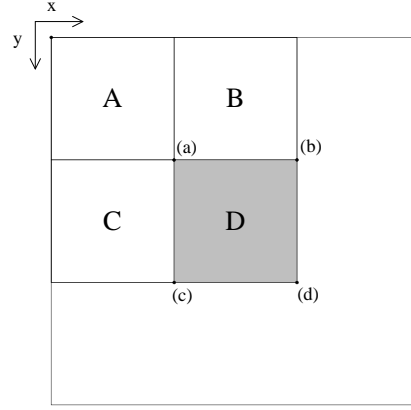
$$I_{int}(x,y) = \sum_{x' \leq x, y' \leq y} I(x',y') \quad (2.2)$$

$(x,y)$  lokasyonunda  $I(x,y)$  giriş görüntüsü,  $S(x,y)$  kümülatif satır toplamı ve  $I_{int}(x,y)$  integral görüntü ise bu yinelemeli işlemler Denklem 2.3 ve Denklem 2.4 ile ifade edilir.

$$S(x,y) = I(x,y) + S(x,y-1) \quad (2.3)$$

$$I_{int}(x,y) = I_{int}(x-1,y) + S(x,y) \quad (2.4)$$





**Şekil 2.3.** D dikdörtgeninin pixel yoğunluk toplamları a, b, c, d noktalarının integral imgeleri kullanılarak hesaplanabilir. D bölgesinin pixel yoğunluk toplamları  $I_{int}(d)+I_{int}(a)-(I_{int}(b)+I_{int}(c))$  olarak gösterilebilir

**Uyarlanabilir yükseltme (Adaboost):** Integral görüntü yönteminin işlemsel hız avantajı bu yöntemin çok fazla sayıda öznelik üzerinde gerçekleşmesi gerektiğinden nesne bulma algoritmalarının gerçek zamanlı olarak uygulanmasında yeterli olmamaktadır. Viola-Jones empirik olarak temelde  $24 \times 24$  'lük pixel çözünürlüğünün yüz tespiti gibi karmaşık hesap gerektiren bir problem için yeterince doyurucu sonuçlar verdiğini hesaplamıştır. Yalnız temel çözünürlükte bile olabilecek tüm boyut ve pozisyonlardaki özneliklerin kullanılması 160.000 kadar öznelik hesaplanması anlamına gelir. Bu da henüz efektif sayılamayacak kadar çok işlem yükünü beraberinde getirir. Uyarlanabilir yükseltme öğrenimi, zayıf sınıflandırıcıların ağırlıklı toplamlarıyla güçlü bir sınıflandırıcı oluşturmaya yarayan bir makine öğrenimi algoritması olup uygun görünmeyen özneliklerin elenmesi ile pozitif - negatif örnekleri en iyi ayıracak özneliklerin bulunması amaçlanır. Uyarlanabilir yükseltme öğreniminde her bir öznelik zayıf sınıflandırıcı ismiyle adlandırılmaktadır. Asıl sınıflandırıcı ise özneliklerin her birinin ağırlıklı toplamlarıyla doğrusal birleşiminden oluşur.  $I_w$   $24 \times 24$  'lük pixel alt pencereleri,  $f$  öznelik,  $u$  polarlık,  $\theta$   $I_w$  'in pozitif ya da negatif örnek olup olmadığına karar veren eşik değeri olmak üzere zayıf bir sınıflandırıcı Denklem 2.5 ile ifade edilir.

$$h(I_w, f, u, \theta) = \begin{cases} 1 & \text{eğer } uf(x) > u\theta \\ 0 & \text{değilse} \end{cases} \quad (2.5)$$

Uyarlanabilir yükseltme algoritması eğitim aşaması,  $(\mathbf{I}_{w1}, \dots, \mathbf{I}_{wn})$  örnek imgelerinin pozitif ve negatif örnekleri için  $(y_1, \dots, y_n) \in \{1, 0\}$  pozitif yada negatif sınıflandırma sonucu olmak üzere aşağıdaki adımlardan oluşur.

- Pozitif ve negatif örnek sayıları için ağırlıklara başlangıç değerlerinin atanması
- Sırasıyla  $t = 1, \dots, T$  aralığı için;

1) Ağırlıkların tüm ağırlıklar toplamına bölünerek normalize edilmesi  $\omega_{t,i} \leftarrow \frac{\omega_{t,i}}{\sum_{j=1}^n \omega_{t,j}}$

2) Ağırlıklı hata miktarına en küçük yapan parametrelerin bulunması (Bkz. Denklem 2.6)

$$\varepsilon_t = \min_{f, u, \theta} \sum_i \omega_i |h(\mathbf{I}_w, f, u, \theta) - y_i| \quad (2.6)$$

3)  $\varepsilon$  'u minimize eden değerler ( $f_t$ ,  $u_t$  ve  $\theta_t$ ) için  $h_t(x) = h(\mathbf{I}_{wt}, f_t, u_t, \theta_t)$  olan zayıf sınıflandırıcının tanımlanması

4)  $\mathbf{I}_{wt}$  'nin doğru ve yanlış sınıflandırıldığı durumlar için sırasıyla  $e_i = 0$ ,  $e_i = 1$  ve  $\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}$  için ağırlıkların güncellenmesi (Bkz. Denklem 2.7)

$$\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e_i} \quad (2.7)$$

Kuvvetli sınıflandırıcı ise  $\alpha_t = \log \frac{1}{\beta_t}$  olmak üzere Denklem 2.8 ile ifade edilir.

$$C(x) = \begin{cases} 1 & \text{eğer } \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{değilse} \end{cases} \quad (2.8)$$

Daha düşük eğitim süreleri ile daha yüksek doğruluk oranı elde etmek için literatürde uyarlanabilir yükseltme algoritması AsymBoost (Pham ve Cham 2007a,b), FloatBoost (Li ve ark. 2002), GentleBoost (Lienhart ve ark. 2003) ve RealBoost (Huang ve ark. 2007) (Wu ve ark. 2004) gibi birçok çalışmayla geliştirilmiştir. Bunun dışında Demirkır ve Sarkur (2006), Haar öznitelik seçimi için uyarlanabilir yükseltme algoritmasının yerine genetik algoritma kullanmışlar ve uyarlanabilir yükseltme algoritmasına göre daha az öznitelikle daha yüksek doğruluk oranı elde ettiklerini görmüşlerdir. Pham ve Cham

(2007a,b) ise istatistiksel yöntemler kullanarak uyarlanabilir yükseltme algoritmasına göre çok daha fazla özneteliği çok daha kısa sürelerde çok daha az bilgisayar hafızası kullanarak yapabildiklerini görmüşlerdir.

**Kaskad sınıflandırıcı:** Viola – Jones (2001) giriş imgesine bağlı olmaksızın tüm imge içerisinde yüz olabilecek alt pencereleri aramanın içerisinde yüz olmadığı kesin olan bölgelerin çıkarıldıktan sonra aranması işlemine göre çok daha zaman alacağını düşünmüştür. Bu nedenle yüz aramaya odaklanmak yerine yüz olmayan bölgelerin elenmesine odaklanmış böylece sınıflandırıcının değerlendirme süresini kayda değer ölçüde düşürmüştür (Bkz. Şekil 2.4). Haar kaskad sınıflandırıcının arkasında da bu fikir yatmaktadır. Haar kaskad sınıflandırıcı her biri kuvvetli sınıflandırıcı içeren basamaklardan (stage) oluşmaktadır. Her basamağın görevi, verilen alt-pencereler arasında yüz olmayacağı kesin olan bölgeleri tamamen ayırarak bir sonraki basamağa hesap edilecek daha az imge parçası bırakmaktır. Böylelikle son basamaklara kalan alt pencerelerin, yüz bulundurma olasılığı yükselecektir.

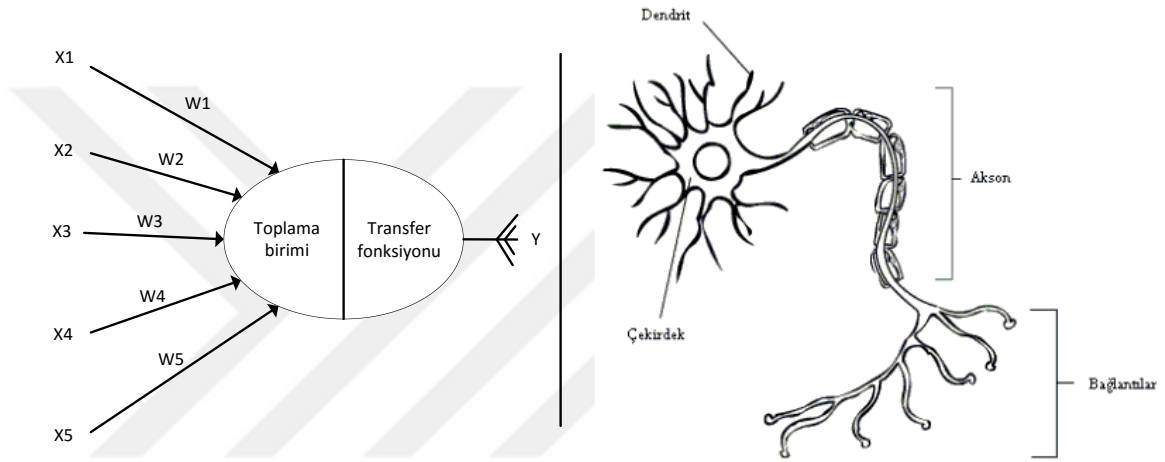


**Şekil 2.4.** Haar kaskad sınıflandırıcı gösterimi

Haar kaskad sınıflandırıcısı yüz ve yüzün içerisinde bulunan nesnelere sınıflandırılması ile birlikte literatürde farklı nesnelere bulunması için de kullanılmıştır. Başer ve Altun (2016), Haar kaskad sınıflandırıcılar kullanarak trafikte bulunan araçları; kamyon, otobüs, sedan, mikrobüs, minivan ve SUV olarak maksimum %92 oranında kategorize etmeyi başarmışlardır. Başer ve Altun (2016), trafikte bulunan araçların sınıflandırılması için OpenCV kütüphanesi ve Python programlama dili yardımıyla içerisinde 9850 araç imgesi içeren BIT araç veri kümesi üzerinde çalışmıştır. Nasrollahi ve ark. (2013), Haar öznetelikler ile biyometrik tanıma amaçlı kulak, iris ve el damarları üzerinde çalışmıştır. Zang ve ark. (2014) trafikte yayaların tespitinin iyileştirilmesi için Haar öznetelikler kullanmışlardır.

### 2.3. Yapay Sinir Ağları

Yapay Sinir ağları insan beyninin yapısal durumu göz önüne alınarak beyin nöronlarının modellenmesi fikri üzerine kurulu bir öğrenim yöntemidir (Stergiou ve Siganos 2017). İnsan beyninde bulunan sinir hücreleri birbirleri ile sinapslar aracılığı ile iletişim kurup işlediği bilgiyi aksonlar aracılığı ile diğer hücelere gönderirler. Yapay sinir ağlarında da benzer şekilde sinir hücreleri dışarıdan gelen bilgileri bir toplama fonksiyonu ile toplar ve aktivasyon fonksiyonundan geçirerek çıktı üretip diğer hücelere aktarır (Öztemel 2012).



**Şekil 2.5.** Yapay sinir hücresi ile biyolojik sinir hücresinin karşılaştırmalı gösterimi

Şekil 2.5'de ve Çizelge 2.2'de yapay sinir hücresi ve biyolojik sinir hücresinin karşılaştırması yapılmıştır.  $X_1 \dots X_5$  giriş değerlerini  $W_1 \dots W_5$  ilgili ağırlıkları temsil etmektedir.  $Y$  değeri bir sonraki hücreye ya da hücelere aktarılacak çıkış değerini göstermektedir.

**Çizelge 2.2.** Yapay sinir hücresi ve biyolojik sinir hücresi karşılaştırması

Gerçek Sinir Hücresi	Yapay Sinir Hücresi
Akson	Çıktı
Dendrit	Toplama Fonksiyonu
Çekirdek	Aktivasyon/Transfer Fonksiyonu
Sinaps	Ağırlıklar

Yapay Sinir Ağları, matematiksel modelin belirlenemediği karmaşık problemlerin çözümü için kullanılmaktadır. Bir matematiksel modele ya da fonksiyona bağlı olmadığından dolayı kara-kutu modellemesi de denilmektedir. Belirli bir algoritmanın izlenmesiyle klasik problem çözme yaklaşımının uzağında olup beyin nöronlarının birbirlerine paralel olarak ortak çalışması mantığına dayanır (Yalçın 2017). Yapay sinir ağları giriş, ara/gizli ve çıkış katmanlarından oluşmaktadır ve her katmanda ise yapay sinir ağı nöronları bulunmaktadır. Bu nöronlar kendilerine gelen toplam bilgiyi transfer fonksiyonları aracılığı ile bir değere adresler ve kendinden önce veya sonra gelen nörona iletir. Bu şekilde bilgi son olarak çıkış katmanına iletilir ve çıkış katmanının aldığı değer, yapay sinir ağı modelinin çıkış değerini ifade eder.

### 2.3.1. Yapay sinir ağı elemanları

Basit bir yapay sinir ağı modeli aşağıdaki elemanlardan oluşur.

**Girdiler:** Yapay sinir ağlarına dışarıdan gelen bilgileri temsil eder.

**Ağırlıklar:** Girdilerin o hücre için etki büyüklüğünü ifade eden sayısal değerdir. Yapay sinir ağları eğitimi başlamadan önce ağırlıklara rastgele değerler atanır. Eğitim adımlarında ise bu değerler eğitim tamamen bitene kadar kendini günceller.

**Toplama(Birleştirme) fonksiyonu:** Sırayla girdilerin ağırlıklarla çarpılıp çıkan sonucun toplanması ya da çarpılması gibi bir fonksiyon ile hücre için net girdinin hesaplanmasını sağlar. Temel bazı toplama fonksiyonları şu şekilde tanımlanabilir:

**Toplam:** Sırayla girdiler ağırlıklarla çarpılıp çıkan değerler toplanır.

$NET = \sum_{i=1}^N X_i \cdot W_i$  olarak ifade edilir. Yapay sinir ağlarında en sık kullanılan toplama fonksiyonudur.

**Çarpım:** Sırayla girdiler ağırlıklarla çarpılıp çıkan değerler birbirleri ile çarpılır.  $NET = \prod_{i=1}^N X_i \cdot W_i$  olarak ifade edilir.

**Maksimum:** Sırayla girdiler ağırlıklarla çarpılıp çıkan değerler birbirleri ile karşılaştırılır. Maksimum değer, net girdi olarak kabul edilir.  $NET = \max(X_i \cdot W_i)$  olarak ifade edilir.

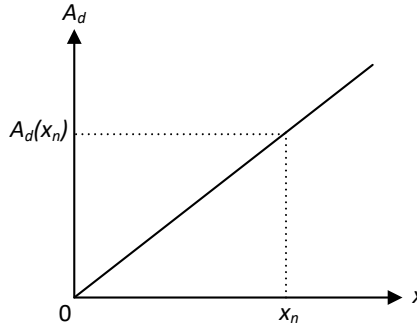
*Minimum:* Sırayla girdiler ağırlıklarla çarpılıp çıkan değerler birbirleri ile karşılaştırılır. Minimum değer, net girdi olarak kabul edilir.  $NET = \min(X_i \cdot W_i)$  olarak ifade edilir.

*Kümülatif Toplam:* Daha önce hücreye gelen toplam girdi ile yeni hesaplanan net girdi değeri toplanır.  $NET_E$ , önceki toplam girdiyi ifade etmek üzere  $NET = NET_E + \sum_{i=1}^N X_i \cdot W_i$  olarak ifade edilir.

Bir yapay sinir ağında toplama fonksiyonlarına ek olarak bir nöronun  $NET$  toplamına etki eden *bias* değeri olması da mümkündür. *Bias*, matematiksel olarak her nöronun  $NET$  değerine katkısının sabit bir sayı ile artırılması ya da azaltılması anlamını taşır. Bu artırılma ya da azaltma işlemi aktivasyon fonksiyonu sonucunu daha anlamlı kılma adına yapılmaktadır.

**Aktivasyon fonksiyonu:** Net girdinin hangi çıktıya karşılık geleceğini adresler. Temel aktivasyon fonksiyonları:

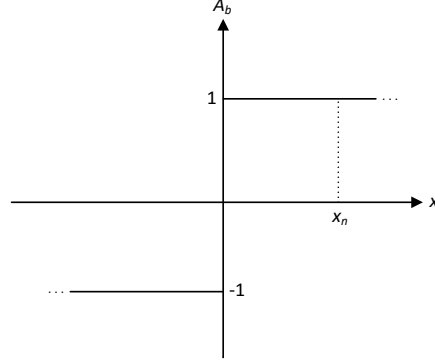
*Lineer doğrusal:* Toplama fonksiyonundan aktivasyon fonksiyonuna iletilen değer herhangi bir katına karşılık gelmektedir.  $x_n$  aktivasyon fonksiyonu girişi olmak üzere  $A_d$  lineer doğrusal aktivasyon fonksiyonu,  $k \in \mathfrak{R}$  için  $A_d(x_n) = kx_n$  olarak ifade edilebilir (Bkz Şekil 2.6).



**Şekil 2.6.** Lineer doğrusal aktivasyon fonksiyonu gösterimi

*Basamak:* Basamak aktivasyon fonksiyonu, toplama fonksiyonundan aktivasyon fonksiyonuna iletilen değer belirli bir eşik değerin üzerine çıktığında ya da altına düştüğünde çıktısını değiştirir (Bkz. Şekil 2.7).  $x_n$  aktivasyon fonksiyonu girişi olmak üzere  $A_b$  basamak aktivasyon fonksiyonu  $A_b(x_n) = \begin{cases} 1 & x_n \geq 0 \\ -1 & x_n < 0 \end{cases}$  olarak ifade edilebilir.

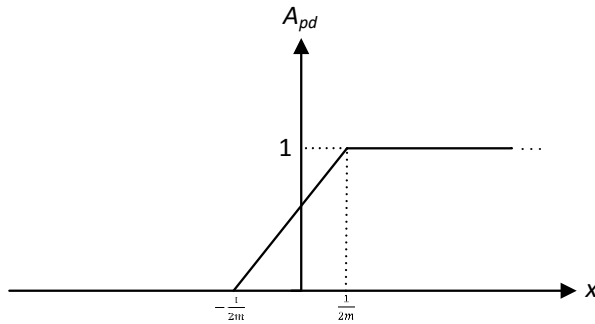
Fonksiyon, istenilen aktivasyon fonksiyonu çıkış değerleri için(-1, 0 ya da 1) özelleştirilebilir.



**Şekil 2.7.** Basamak aktivasyon fonksiyonu gösterimi

*Parçalı - doğrusal:* Parçalı-doğrusal aktivasyon fonksiyonu, toplama fonksiyonundan aktivasyon fonksiyonuna iletilen değerin belirli bir aralığı için doğrusal, diğer aralıkları için sabit bir çıktı üretir. Bu sabit değere, aktivasyon fonksiyonunun doyum değerleri de denmektedir.  $x_n$ , aktivasyon fonksiyonu girişi ve  $m$ , parçalı doğrusal fonksiyonun eğim miktarı için  $A_{pd}$  parçalı - doğrusal aktivasyon fonksiyonu Denklem 2.9 ile ifade edilebilir. Şekil 2.8’de parçalı - doğrusal aktivasyon fonksiyonu gösterimi yer almaktadır.

$$A_{pd}(x_n) = \begin{cases} 1 & x_n \geq \frac{1}{2m} \\ mx_n + \frac{1}{2} & -\frac{1}{2m} < x_n < \frac{1}{2m} \\ 0 & x_n \leq -\frac{1}{2m} \end{cases} \quad (2.9)$$

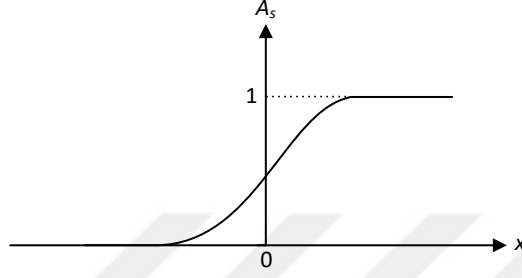


**Şekil 2.8.** Parçalı-doğrusal aktivasyon fonksiyonu gösterimi

*Sigmoid:* Sigmoid aktivasyon fonksiyonu; doğrusal yapıda olmaması, sürekli ve türevi alınabilir olması ve çıkış değerlerinin 0 ve 1 arasında olması nedeniyle diğer aktivasyon

fonksiyonlarından farklıdır ve yapay sinir ağı uygulamalarında sıklıkla tercih edilmektedir.  $x_n$ , aktivasyon fonksiyonu girişi için  $A_s$  sigmoid aktivasyon fonksiyonu Denklem 2.10 ile ifade edilebilir. Şekil 2.9’da sigmoid aktivasyon fonksiyonu gösterimi yer almaktadır.

$$A_s(x_n) = \frac{1}{1+e^{-x_n}} \quad (2.10)$$

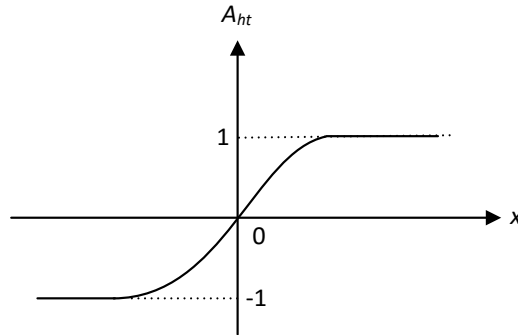


**Şekil 2.9.** Sigmoid aktivasyon fonksiyonu gösterimi

*Hiperbolik tanjant:* Hiperbolik tanjant aktivasyon fonksiyonu, sigmoid aktivasyon fonksiyonuna benzer şekilde doğrusal değildir, sürekli ve türevi alınabilir. Sigmoid aktivasyon fonksiyonundan farklı olarak çıkış değerleri -1 ve 1 arasındadır.  $x_n$ , aktivasyon fonksiyonu girişi için  $A_{ht}$  hiperbolik tanjant aktivasyon fonksiyonu Denklem 2.11 ile ifade edilebilir.

$$A_{ht}(x_n) = \tanh(x_n) = \frac{e^{x_n} - e^{-x_n}}{e^{x_n} + e^{-x_n}} \quad (2.11)$$

Şekil 2.10’da hiperbolik tanjant aktivasyon fonksiyonu gösterimi yer almaktadır.



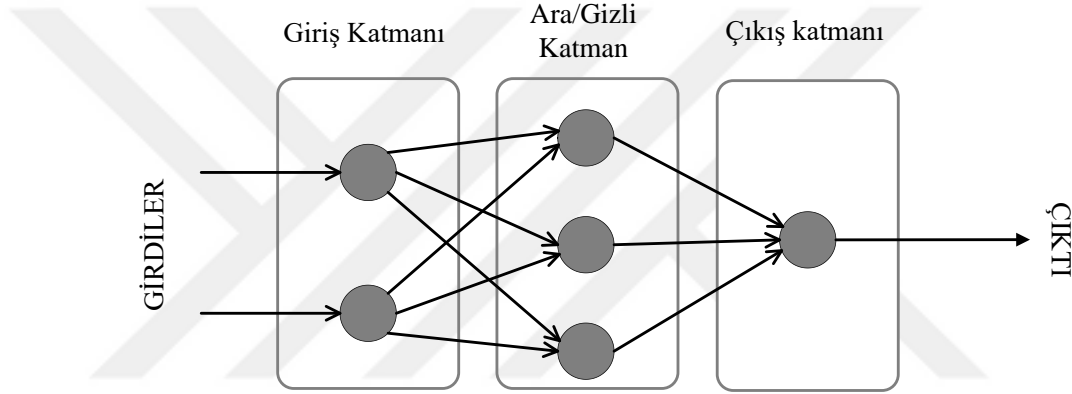
**Şekil 2.10.** Hiperbolik tanjant aktivasyon fonksiyonu gösterimi



**Çıktılar:** Aktivasyon fonksiyonu sonucunda elde edilen bilginin kendisidir.

Bir yapay sinir ağı modelini diğer yapay sinir ağı modellerinden ayıran ve o modeli karakterize eden özellikleri; ağın topolojisi (giriş katmanı, ara katman/katmanları, çıkış katmanı sayıları, ileri besleme, geri besleme olması vb.), kullanılan aktivasyon ve toplama fonksiyonları ve öğrenme şeklidir (Eğitim fonksiyonu).

Basit bir yapay sinir ağı Şekil 2.11'deki gibi ifade edilebilir.



**Şekil 2.11.** 2 adet giriş katmanı nöronu, 1 adet ara/gizli katman ve 3 adet ara/gizli katmanı nöronu, 1 adet çıkış katmanı nöronundan oluşan basit bir yapay sinir ağı

Oluşturulacak yapay sinir ağı modelinin karmaşıklığına göre istendiği durumda, başarıyı artırmak için gizli katman sayısı ve/veya nöronu artırılabilir. Fakat her yapay sinir ağı modeli için bu başarıyı artırıcı bir etki göstermeyebilir.

YSA, yapısına ve öğrenme algoritmalarına göre 2 farklı şekilde sınıflandırılabilir.

### 2.3.2. Yapısına göre yapay sinir ağları

**İleri beslemeli:** Bilgiler sırasıyla giriş katmanı, ara katman ve çıkış katmanından geçerek bir katman sadece kendinden sonraki katmana veri aktarımı yapar.

**Geri beslemeli:** Bilgiler sadece kendinden sonra gelen nöron katmanına girdi olarak verilmez. Bir nörona gelen bilgi kendi katmanından başka bir nöron ya da farklı

katmanlardaki nöronlara aktarılabilir. Bu, doğası gereği daha karmaşık ve hesap yükünü artıran bir model oluşturulmasını beraberinde getirir.

### 2.3.3. Öğrenme algoritmalarına göre yapay sinir ağları

**Danışmanlı:** Öğretim esnasında ağa verilen giriş değerleri ile birlikte olması arzu edilen çıkış değerleri de verilir. Ağ, girişlere karşı istenen çıkışı verebilmek için ağırlıklarını günceller. Ağın çıktıları ile olması gereken çıkış değerleri arasındaki farka göre hata hesaplanır ve bu hataya göre ağırlıklar yeniden belirlenir. Kabul edilecek hata miktarının altına düşene kadar ağırlıklar güncellenmeye devam eder.

**Danışmansız:** Öğretim esnasında ağa sadece giriş değerleri verilir. Ağ, bu giriş değerlerine göre kendi kurallarını oluşturarak benzer sonuçlar yakaladığı örnekleri sınıflandırır.

**Destekleyicili:** Öğretim sırasında yapılan her iterasyonda o iterasyonun olumlu/olumsuz çıkarımını bir sonraki öğrenme adımında değerlendirerek öğrenme sırasında sonuç çıkarmaya devam eder.

Bu tezde; yapay sinir ağı eğitim algoritması olarak danışmanlı, ileri beslemeli ve geri yayımlı bir eğitim algoritması olan Levenberg-Marquardt(LM) algoritması seçilmiştir(Marquardt 1963) (Levenberg 1944).

### 2.3.4. LM algoritması ve LM algoritmasında kullanılan yöntemler

LM lineer olmayan en küçük kareler problemlerini çözmek için standart bir tekniktir. Arzu edilen değerler ve fonksiyon çıkışı arasındaki toplam hata karelerini minimuma indirmek için iteratif olarak parametreleri günceller. Parametre güncellemesi en dik iniş yöntemi ile Gauss-Newton yöntemlerinin birleşiminden oluşur (Yu ve Wilamowski 2011).

**En dik iniş yöntemi:** Gradyan inişi ya da hata geri yayılım algoritması da denilen bu yöntem, hata uzayında minimum değerini bulması için hata fonksiyonunun 1. dereceden kısmi türev ifadelerini kullanır. Hata fonksiyonunun gradyanı da toplam hata fonksiyonunun ağırlıklara göre 1. derece kısmi türevlerine eşittir (Bkz. Denklem 2.12).

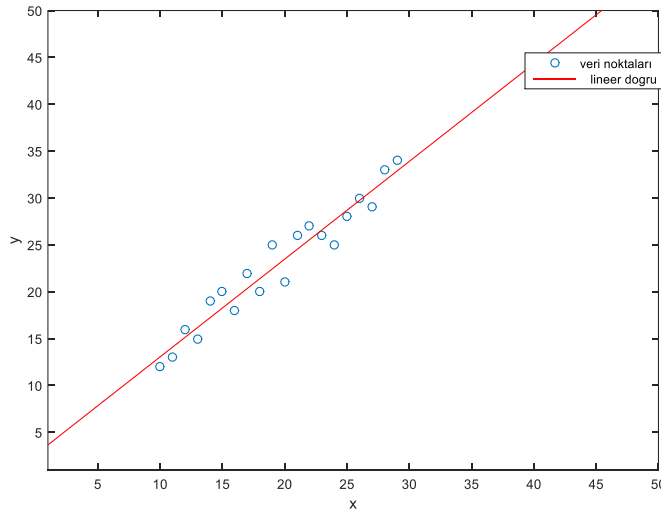
$$\mathbf{G} = \nabla E = \left[ \frac{\partial E}{\partial w_1} \frac{\partial E}{\partial w_2} \frac{\partial E}{\partial w_3} \dots \frac{\partial E}{\partial w_n} \right]^T \quad (2.12)$$

En dik iniş yöntemine göre parametre güncelleme işlemi,  $\alpha$  eğitim sabiti,  $\mathbf{w}_k$  iterasyonun bulunduğu adımdaki ağırlıklar vektörü,  $\mathbf{w}_{k+1}$  bir sonraki iterasyondaki ağırlıklar vektörü olmak üzere Denklem 2.13’de ifade edilmiştir.

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha \mathbf{G}_k \quad (2.13)$$

$\alpha$  eğitim sabitini büyük tutmak öğrenme hızını artırabileceği gibi fonksiyonun minimum noktasının kaçırılmasına da sebep olabilir.  $\alpha$  eğitim sabitini küçük tutmak ise öğrenme hızını yavaşlatacağı gibi monotonik olmayan fonksiyonlarda yerel minimum’un bulunmasını böylece fonksiyonun gerçekte en düşük değerinin hiçbir zaman bulunamamasına da neden olabilir. Bundan dolayı monotonik olmayan fonksiyonlarda birden fazla  $\alpha$  eğitim sabiti kullanılarak iterasyona başlanır.

Basit bir örnekle en dik iniş yöntemi açıklanmak istenirse bir giriş ve bir çıkışlı bir sistem için lineer regresyon problemi ele alınabilir. Şekil 2.12’de x giriş değerlerine karşılık alınan y çıkış değerleri ve bu problemin temsil edilmesi istenilen lineer regresyon modeli gösterilmiştir.



**Şekil 2.12.** Basit bir lineer regresyon modeli gösterimi

Burada lineer regresyon modelini temsil eden doğru;  $m$  eğim,  $b$  öteleme miktarı olmak üzere  $y = mx + b$  ile ifade edilebilir. Temsil edilen doğrunun ise gerçek değerlere ne kadar yaklaştığını ifade etmek için *Ortalama Kare Hatası* (OKH) ile performans ölçümü yapılmaktadır. OKH'da her bir gerçek çıkış değerinden hesaplanan çıkış değeri çıkarılarak karesi alınır ve toplam çıkış sayısına bölünür. OKH değerlerinin bulunması için kullanılan fonksiyon literatürde hata fonksiyonu ya da maliyet fonksiyonu olarak da adlandırılmaktadır.  $y_G(i)$  gerçek çıkış değeri ve  $y_T(i)$  hesaplanan çıkış değeri olmak üzere OKH fonksiyonunun genel gösterimi Denklem 2.14 ile ifade edilmiştir. Ele alınan lineer regresyon problemi için ise Denklem 2.15 ile ifade edilmiştir.

$$OKH = \frac{1}{N} \sum_{i=1}^N (y_G(i) - y_T(i))^2 \quad (2.14)$$

$$E = \frac{1}{N} \sum_{i=1}^N (y_G(i) - (mx_i + b))^2 \quad (2.15)$$

En dik iniş yöntemine göre Denklem 2.3.7'deki hata fonksiyonunu minimum yapacak  $m$  ve  $b$  değerlerinin bulunması amaçlanmaktadır. Bunun için keyfi bir  $m$  ve  $b$  değeri ile iterasyona başlanır ve her adımda bu değerler Denklem 2.13'ün düzenlenmesi ile Denklem 2.16 ve Denklem 2.17'deki şekliyle güncellenir.

$$m_{x+1} = m_x - \alpha \frac{\partial E}{\partial m} \quad (2.16)$$

$$b_{x+1} = b_x - \alpha \frac{\partial E}{\partial b} \quad (2.17)$$

Kısmi türev ifadeleri ise Denklem 2.18 ve Denklem 2.19 ile hesaplanmaktadır.

$$\frac{\partial E}{\partial m} = \frac{2}{N} \sum_{i=1}^N -x_i (y_G(i) - (mx_i + b)) \quad (2.18)$$

$$\frac{\partial E}{\partial b} = \frac{2}{N} \sum_{i=1}^N -(y_G(i) - (mx_i + b)) \quad (2.19)$$

İterasyon, parametre değişiminin çok az olduğunu ifade eden belirli bir eşik değerinin altına düştüğünde durur.

En dik iniş yönteminde genellikle fonksiyonun minimum noktasının atlanmaması için küçük adımlar ( $\alpha$  eğitim sabiti) tercih edilir bu da eğitim hızının yavaş olmasını beraberinde getirmektedir. Gauss-Newton yönteminin ise hata fonksiyonunun 2. dereceden türevini ihtiva etmesi fonksiyonun nerelerde yön değiştirdiği bilgisini de içerdiğinden uygun eğitim sabitinin bulunmasını ve çok hızlı bir şekilde eğitimin tamamlanmasını sağlayabilir. Gauss-Newton yönteminin temelini aşağıda açıklanmış olan Newton yöntemi oluşturmuştur. Newton yönteminde yer alan 2. dereceden türev ifadelerinde Hessien matrisi hesabının yapılması yerine Gauss-Newton yönteminde Jacobien matrisi hesabının yapılması Gauss-Newton yönteminin hesaplanmasında hız avantajı sağlamıştır.

**Newton yöntemi:** Newton yönteminde tüm gradyan bileşenlerinin her birinin, tüm ağırlıkların birer fonksiyonu olduğu ve ağırlıkların lineer olarak bağımsız olduğu varsayılmaktadır. Her bir gradyan bileşeninin içerisinde tüm ağırlıkları içeren bu fonksiyonu, 1. derece yaklaşımla Taylor serilerine açılırsa Denklem 2.20 elde edilir.

$$\left\{ \begin{array}{l} G_1 \approx G_{1,0} + \frac{\partial G_1}{\partial w_1} \Delta w_1 + \frac{\partial G_1}{\partial w_2} \Delta w_2 + \dots + \frac{\partial G_1}{\partial w_N} \Delta w_N \\ G_2 \approx G_{2,0} + \frac{\partial G_2}{\partial w_1} \Delta w_1 + \frac{\partial G_2}{\partial w_2} \Delta w_2 + \dots + \frac{\partial G_2}{\partial w_N} \Delta w_N \\ \dots \\ G_N \approx G_{N,0} + \frac{\partial G_N}{\partial w_1} \Delta w_1 + \frac{\partial G_N}{\partial w_2} \Delta w_2 + \dots + \frac{\partial G_N}{\partial w_N} \Delta w_N \end{array} \right. \quad (2.20)$$

Eşitlik 2.3.12 'deki gradyan değerlerinin  $\frac{\partial E}{\partial w}$  cinsinden yazılmasıyla 2. dereceden parçalı türev ifadeleri elde edilir (Bkz. Denklem 2.21).

$$\left\{ \begin{array}{l} G_1 \approx G_{1,0} + \frac{\partial^2 E}{\partial w_1^2} \Delta w_1 + \frac{\partial^2 E}{\partial w_1 \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_1 \partial w_N} \Delta w_N \\ G_2 \approx G_{2,0} + \frac{\partial^2 E}{\partial w_2 \partial w_1} \Delta w_1 + \frac{\partial^2 E}{\partial w_2^2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_2 \partial w_N} \Delta w_N \\ \dots \\ G_N \approx G_{N,0} + \frac{\partial^2 E}{\partial w_N \partial w_1} \Delta w_1 + \frac{\partial^2 E}{\partial w_N \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_N^2} \Delta w_N \end{array} \right. \quad (2.21)$$

$E$  hata fonksiyonunun minimum değer noktasında tüm gradyan bileşenlerinin 0'a eşit olması gerektiğinden Denklem 2.21'deki gradyan bileşenlerinin 0'a eşitlenmesi ile Denklem 2.22 elde edilir.

$$\left\{ \begin{array}{l} -G_{1,0} \approx \frac{\partial^2 E}{\partial w_1^2} \Delta w_1 + \frac{\partial^2 E}{\partial w_1 \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_1 \partial w_N} \Delta w_N \\ -G_{2,0} \approx \frac{\partial^2 E}{\partial w_2 \partial w_1} \Delta w_1 + \frac{\partial^2 E}{\partial w_2^2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_2 \partial w_N} \Delta w_N \\ \dots \\ -G_{N,0} \approx \frac{\partial^2 E}{\partial w_N \partial w_1} \Delta w_1 + \frac{\partial^2 E}{\partial w_N \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_N^2} \Delta w_N \end{array} \right. \quad (2.22)$$

Denklem 2.22 ile N bilinmeyenli N adet denklem elde edilmiş olunur. İfadenin matris formunda yazılmasıyla Denklem 2.23 elde edilir.

$$\begin{bmatrix} -G_1 \\ -G_2 \\ \dots \\ -G_N \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_1 w_2} & \dots & \frac{\partial^2 E}{\partial w_1 \partial w_N} \\ \frac{\partial^2 E}{\partial w_2 w_1} & \frac{\partial^2 E}{\partial w_2^2} & \dots & \frac{\partial^2 E}{\partial w_2 \partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 E}{\partial w_N \partial w_1} & \frac{\partial^2 E}{\partial w_N \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_N^2} \end{bmatrix} \times \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \\ \dots \\ \Delta w_N \end{bmatrix} \quad (2.23)$$

Burada 2. dereceden kısmi türevlerini içeren matris ifadesinin Hessian matris olduğu görülmektedir. Denklemde matris formundaki bileşenler daha da basitleştirilerek

Denklem 2.24'deki şekilde yazılabilir. Buradan ağırlık değişimi ifadesi yalnız bırakılarak Denklem 2.25 elde edilmiş olur.

$$-G = H\Delta w \quad (2.24)$$

$$\Delta w = -H^{-1}G \quad (2.25)$$

Bundan dolayı ağırlık güncellemesi her bir ağırlık için Denklem 2.26 ile ifade edilir.

$$w_{k+1} = w_k - H_k^{-1}G_k \quad (2.26)$$

Denklem 2.13 ile Denklem 2.26 karşılaştırıldığında en dik iniş yönteminde  $\alpha$  eğitim sabitinin yerini Newton yönteminde Hessien matrisinin tersinin aldığı görülmektedir. Hessien matrisi içerisinde  $E$  hata fonksiyonunun 2. türevlerini içerdiğinden fonksiyonun eğrildiği ve yön değiştirdiği yerler hakkında bilgi vermektedir. Bu da eğitim adım büyüklüğünün Newton yönteminde daha düzgün seçilmesini ve en dik iniş yöntemine göre çok daha hızlı bir şekilde sonuca yaklaşılmasını sağlamaktadır.

**Gauss-Newton yöntemi:** Hessien matrisi içerisinde hata fonksiyonunun 2. derece türev hesaplarını içerdiğinden Newton yönteminde bazı hesaplamalar çok karmaşık kalabilmektedir. Bundan dolayı Hessien matrisi yerine Gauss-Newton yönteminde Jacobien matrisi kullanılmaktadır. Denklem 2.27'de örnek bir Jacobien matrisi gösterilmiştir. Burada  $N$  indisi, yapay sinir ağı girişlerini oluşturan örnek kümesini;  $M$  indisi, yapay sinir ağı çıkış nöron sayısını ve  $P$  indisi, yapay sinir ağı ağırlıklarını ifade etmektedir.

$$J = \begin{bmatrix} \frac{\partial E_{1,1}}{\partial w_1} & \frac{\partial E_{1,1}}{\partial w_2} & \frac{\partial E_{1,1}}{\partial w_p} \\ \frac{\partial E_{1,2}}{\partial w_1} & \frac{\partial E_{1,2}}{\partial w_2} & \frac{\partial E_{1,2}}{\partial w_p} \\ \vdots & \vdots & \vdots \\ \frac{\partial E_{1,M}}{\partial w_1} & \frac{\partial E_{1,M}}{\partial w_2} & \frac{\partial E_{1,M}}{\partial w_p} \\ \vdots & \vdots & \vdots \\ \frac{\partial E_{N,1}}{\partial w_1} & \frac{\partial E_{N,1}}{\partial w_2} & \frac{\partial E_{N,1}}{\partial w_p} \\ \frac{\partial E_{N,2}}{\partial w_1} & \frac{\partial E_{N,2}}{\partial w_2} & \frac{\partial E_{N,2}}{\partial w_p} \\ \vdots & \vdots & \vdots \\ \frac{\partial E_{N,M}}{\partial w_1} & \frac{\partial E_{N,M}}{\partial w_2} & \frac{\partial E_{N,M}}{\partial w_p} \end{bmatrix} \quad (2.27)$$

Denklem 2.14’de genelleştirilmiş hata fonksiyonunu çıkış değerlerimizin birden fazla olabileceğini göz önünde bulundurarak ve Denklem 2.12’deki gradyan hesabına eklendiğinde her bir gradyan bileşeni için Denklem 2.28 elde edilmektedir ( $\frac{1}{N}$  değeri yerine türevi alındığında sabit değerlerin birbirlerini götürmesi için  $\frac{1}{2}$  eklenmiştir.).

$$G_i = \frac{\partial E}{\partial w_i} = \frac{\partial \left( \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^M E_{p,m}^2 \right)}{\partial w_i} = \sum_{n=1}^N \sum_{m=1}^M \left( \frac{\partial E_{p,m}}{\partial w_i} E_{p,m} \right) \quad (2.28)$$

Denklem 2.27 ve 2.28 birleştirildiğinde ise Denklem 2.29 elde edilmektedir. Burada  $\mathbf{G}$  gradyan vektörünü,  $\mathbf{J}$  jacobien matrisini ve  $\mathbf{E}$  hata vektörünü ifade etmektedir.

$$\mathbf{G} = \mathbf{J}\mathbf{E} \quad (2.29)$$

Hata fonksiyonunun Denklem 2.23’de yer alan Hessien matrisi içerisine eklenmesi ile Denklem 2.30 elde edilmektedir.

$$h_{i,j} = \frac{\partial^2 E}{\partial w_i \partial w_j} = \frac{\partial^2 \left( \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^M E_{n,m}^2 \right)}{\partial w_i \partial w_j} = \sum_{n=1}^N \sum_{m=1}^M \left( \frac{\partial E_{n,m}}{\partial w_i} \frac{\partial E_{n,m}}{\partial w_j} \right) + K_{i,j} \quad (2.30)$$

$K_{i,j} = \sum_{n=1}^N \sum_{m=1}^M \left( \frac{\partial^2 E_{n,m}}{\partial w_i \partial w_j} E_{n,m} \right)$  değeri Gauss-Newton yöntemindeki varsayıma göre sıfırdır (Hagan ve Menhaj 1994) ve bu durumda Hessien matrisi Jacobien matrisi



cinsinden  $H \approx J^T J$  şeklinde yazılabilir. Bu bilgi ile Denklem 2.26 ve Denklem 2.29 birleştirilerek Gauss-Newton algoritmasındaki parametre değişim ifadesi Denklem 2.31 ile ifade edilir.

$$\mathbf{w}_{k+1} = \mathbf{w}_k - (J_k^T J_k)^{-1} J_k E_k \quad (2.31)$$

Gauss-Newton algoritmasının Newton algoritmasına göre 2. dereceden kısmi türev ifadelerinin hesaplanmamasından dolayı daha hızlı olduğu görülmektedir fakat karşılaşılabilecek problem Newton algoritmasıyla benzerdir. Burada, Newton algoritmasındaki Hessien matrisinin tersi olamayabileceği gibi  $J^T J$  ifadesinin tersinin bulunamayabileceği de görülmektedir. Bundan dolayı optimizasyonun bir sonuca yakınsamayabileceği olasılığı da mevcuttur.

**Levenberg-Marquardt yöntemi:** Levenberg-Marquardt yönteminde, Gauss-Newton algoritmasındaki  $J^T J$  ifadesinin sürekli tersi alınabilir duruma getirilmesi için köşegen matris ifadelerinin değerlerini sıfırdan büyük olabilecek bir düzenleme yapılmıştır (Bkz. Denklem 2.32).

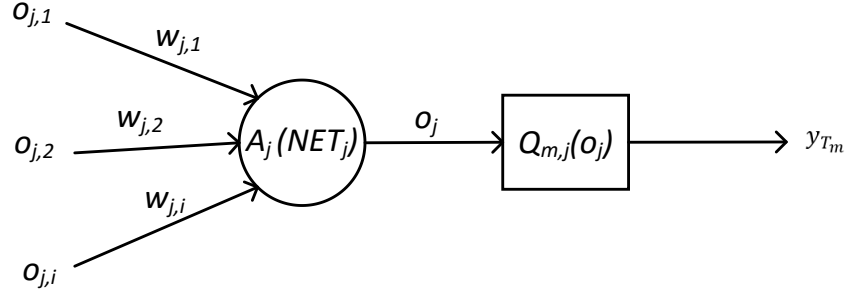
$$H \approx J^T J + \mu I \quad (2.32)$$

Böylelikle Gauss-Newton yönteminde karşılaşılabilecek yakınsamama problemi bu yöntemde giderilmiştir. Eğitim esnasında  $\mu$  skaler kombinasyon katsayısı değerinin çok küçük olması Denklem 2.33'den de görüleceği üzere yöntemi, Gauss-Newton yöntemine benzetmektedir.  $\mu$  değerinin çok büyük olması ise  $\alpha = 1 / \mu$  olmak üzere yöntemi, en dik iniş yöntemine benzetmektedir.

$$\mathbf{w}_{k+1} = \mathbf{w}_k - (J_k^T J_k + \mu I)^{-1} J_k E_k \quad (2.33)$$

Levenberg Marquardt yönteminin uygulanmasında Denklem 2.33'den de görülebileceği üzere Jacobien matris hesaplarının yapılması ve eğitim adımlarının tasarlanması önem arz etmektedir. Uygulama ileri yönde hesaplama adımları ve geri yayılım adımları olarak iki aşamada gerçekleşmektedir. Jacobien matris hesabının yapılması ve hata fonksiyonlarının ağırlıklara göre kısmi türevlerinin hesaplanabilmesi için her bir nöron üzerinde YSA modelinin geri kalanına göre kısmi türevlerde zincir kuralı uygulanmaktadır (Bkz. Şekil 2.13 ve Denklem 2.34). Burada  $o_{j,i}$  ifadesi j. Nöronun giriş

değerlerini;  $w_{j,i}$  ilgili girişlere karşı ağırlık değerlerini;  $A_j$ , aktivasyon fonksiyonunu; NET, toplam fonksiyonunu; Q, j. nöronun çıkışı ile yapay sinir ağının çıkışı arasındaki lineer olmayan ilişkiyi gösteren fonksiyonu temsil etmektedir.



**Şekil 2.13.** Her bir YSA nöronunun parametreleri ve YSA modelinin geri kalanına karşılık gösterimi

$$\frac{\partial E_{n,m}}{\partial w_{j,i}} = \frac{\partial (y_{G_{n,m}} - y_{T_{n,m}})}{\partial w_{j,i}} = -\frac{\partial y_{T_{n,m}}}{\partial w_{j,i}} = -\frac{\partial y_{T_{n,m}}}{\partial o_j} \frac{\partial o_j}{\partial NET_j} \frac{\partial NET_j}{\partial w_{j,i}} \quad (2.34)$$

Denklem 2.34'ün aktivasyon fonksiyonu eğimi olan  $\frac{\partial o_j}{\partial NET_j}$  parametresinin  $s_j$  ifadesi şeklinde düzenlenmesiyle ve toplam fonksiyonunun ağırlıklara göre kısmi türevi  $o_{j,i}$  giriş değerlerini verdiği bilgisini kullanarak Denklem 2.35 elde edilir.

$$\frac{\partial E_{n,m}}{\partial w_{j,i}} = -Q'_{m,j} s_j o_{j,i} \quad (2.35)$$

Denklem 2.35'de görülmektedir ki Jacobien matrisini oluşturan hata fonksiyonlarının ağırlıklara göre kısmi türevleri; her bir nöron çıkışı ile yapay sinir ağı çıkışı arasındaki lineer olmayan fonksiyonun türevi ( $Q'_{m,j}$ ), her bir nöronda kullanılan aktivasyon fonksiyonunun eğimi ( $s_j$ ) ve her bir nörona giriş olarak gelen verilerin ( $o_{j,i}$ ) çarpımından oluşmaktadır.  $Q'_{m,j} s_j$  ifadesi literatürde  $\delta_{m,j}$  olarak geri yayılım parametresi şeklinde adlandırılmaktadır. Ayrıca çıkış katmanındaki nöronların çıkışı aynı zamanda yapay sinir ağı çıkışlarını ifade ettiğinden çıkış katmanında  $Q'_{m,j} = 1$  ve  $\delta_{m,j} = s_j$  olduğu görülebilmektedir. Bu bilgi ileri yönde hesaplama adımlarından geri yayılım adımlarına geçerken kullanılacaktır.

3 katmanlı bir yapay sinir ağı için Levenberg Marquardt algoritmasının uygulama adımları aşağıdaki şekilde örneklendirilebilir.

*İleri yönde hesaplama:*

Adım 1. İlk katmanın tüm nöronları için  $NET$  toplam fonksiyonu çıkışının (Denklem 2.36), aktivasyon fonksiyonu çıkış değerlerinin (Denklem 2.37) ve aktivasyon fonksiyonu eğim değerlerinin (Denklem 2.38) hesaplanması (Üst indis “1” ilk katmanı,  $n_i$  örnek veri kümesi için toplam giriş sayısını,  $o_i$  nörona gelen giriş değerlerini,  $w_{j,i}^1$  giriş değerlerinin ağırlıklarını,  $w_{j,0}^1$  bias değerini temsil etmektedir.)

$$NET_j^1 = \sum_{i=1}^{n_i} o_i w_{j,i}^1 + w_{j,0}^1 \quad (2.36)$$

$$o_j^1 = A_j^1(NET_j^1) \quad (2.37)$$

$$s_j^1 = \frac{\partial A_j^1}{\partial NET_j^1} \quad (2.38)$$

Adım 2. İlk katmanın çıkış değerlerini 2. katmanın girişleri olarak kullanarak benzer işlemlerin yapılması (Üst indis “2” 2. katmanı,  $n_l$  indisi ilk katmandaki toplam nöron sayısını ifade etmektedir.)

$$NET_j^2 = \sum_{i=1}^{n_l} y_i^1 w_{j,i}^2 + w_{j,0}^2 \quad (2.39)$$

$$o_j^2 = A_j^2(NET_j^2) \quad (2.40)$$

$$s_j^2 = \frac{\partial A_j^2}{\partial NET_j^2} \quad (2.41)$$

2. katmandaki tüm nöronlar için Denklem 2.39,  $NET$  toplam fonksiyonu çıkışının; Denklem 2.40, aktivasyon fonksiyonu çıkışının ve Denklem 2.41, aktivasyon fonksiyonu eğim değerlerinin hesaplanmasını ifade etmektedir.

Adım 3. 2. katmanın çıkış değerlerini 3. katmanın girişleri olarak kullanarak benzer işlemlerin yapılması (Üst indis “3” 3. katmanı  $n_2$  indisi ikinci katmandaki toplam nöron sayısını ifade etmektedir.)

$$NET_j^3 = \sum_{i=1}^{n_2} y_i^2 w_{j,i}^3 + w_{j,0}^3 \quad (2.42)$$

$$o_j^3 = A_j^3(NET_j^3) \quad (2.43)$$

$$s_j^3 = \frac{\partial A_j^3}{\partial NET_j^3} \quad (2.44)$$

3. katmandaki tüm nöronlar için Denklem 2.42,  $NET$  toplam fonksiyonu çıkışının; Denklem 2.43, aktivasyon fonksiyonu çıkışının ve Denklem 2.44, aktivasyon fonksiyonu eğim değerlerinin hesaplanmasını ifade etmektedir.

İleri yönde hesaplama adımlarından sonra eğitilmek istenen ilk veri kümesi için tüm çıkış nöronlarında(3. katman nöronları) toplam hata miktarı  $y_{G_m} - y_{T_m}$  ile hesaplanır. Çıkış aktivasyon fonksiyonunun eğimi olarak başlangıç  $\delta$  parametreleri  $\delta_{j,j}^3 = s_j^3 = \frac{\partial A_j^3}{\partial NET_j^3}$  ile hesaplanır.

*Geri yönde yayılım:*

Adım 1.  $\delta$  parametresinin 3. katmanın girişlerinden 2. katmanın çıkışlarına yayılımı(Bkz. Denklem 2.45)(k indisi 2. katmanın nöronlarını, j indisi 3. katmanın nöronlarını ifade etmektedir.)

$$\delta_{j,k}^2 = w_{j,k}^3 \delta_{j,j}^3 \quad (2.45)$$

Adım 2.  $\delta$  parametresinin 2. katmanın çıkışlarından 2. katmanın girişlerine aktarılması(Bkz. Denklem 2.46) (k indisi 2. katmanın nöronlarını, j indisi 3. katmanın nöronlarını ifade etmektedir.)

$$\delta_{j,k}^2 = \delta_{j,k}^2 s_k^2 \quad (2.46)$$

Adım 3.  $\delta$  parametresinin 2. katmanın girişlerinden ilk katmanın çıkışlarına aktarılması(Bkz. Denklem 2.47) (k indisi ilk katmanın nöronlarını, j indisi 3. katmanın nöronlarını ifade etmektedir.)

$$\delta_{j,k}^1 = \sum_{i=1}^{n_2} w_{j,i}^2 \delta_{j,i}^2 \quad (2.47)$$

Adım 4.  $\delta$  parametresinin ilk katmanın çıkışlarından ilk katmanın girişlerine aktarılması(Bkz. Denklem 2.48) (k indisi ilk katmanın nöronlarını, j indisi 3. katmanın nöronlarını ifade etmektedir.)

$$\delta_{j,k}^1 = \delta_{j,k}^1 s_k^1 \quad (2.48)$$

Geri yayılım işlemi, tüm çıkış nöronları için tekrarlanır. Böylelikle ilk örnek veri kümesi için toplam eleman sayıları  $(n_1 \times n_2) + (n_2 \times n_3)$  olan  $\delta$  ve  $\mathbf{o}$  dizileri oluşturulur ve Jacobien matrisinin ilk örnek veri kümesi için satır ve sütunları hesaplanır( $n_1$ , 1. katman nöron sayısı;  $n_2$ , 2. katman nöron sayısı;  $n_3$ , 3. katman nöron sayısı). Daha sonra, yapılan ileri yönde hesaplama ve geri yayılım adımları tüm örnek veri kümesi için sırayla tekrarlanır ve tüm Jacobien matrisi oluşturulur. Oluşturulan Jacobien matrisi Denklem 2.33'de yerine konularak bir sonraki ağırlık değerleri( $w_{k+1}$ ) hesaplanır(ilk ağırlık değerleri rastgele verilir) ve ağırlıklar güncellenmiş olur. Eğer ağırlıkların güncellenmesi sonucunda elde edilen toplam hata miktarı güncellenmeden önceki toplam hata miktarından fazla ise ağırlık değerleri bir öncekine geri döndürülür ve  $\mu$  skaler kombinasyon katsayısının değeri sabit bir katsayı ile çarpılıp artırılarak Denklem 2.33 ile tekrar bir sonraki ağırlık değerleri( $w_{k+1}$ ) hesaplanır. Eğer yeni güncellenen ağırlıklar ile toplam hata miktarı azalır ise yapılan işlemler geçerli kabul edilir ve  $\mu$  skaler kombinasyon katsayısının değeri sabit bir katsayı ile çarpılıp azaltılarak bir sonraki ağırlık güncellemesi için aynı işlemler tekrarlanır. Hata miktarı, istenilen hata miktarından az olana kadar ağırlık güncelleme işlemleri bu şekilde devam etmektedir.

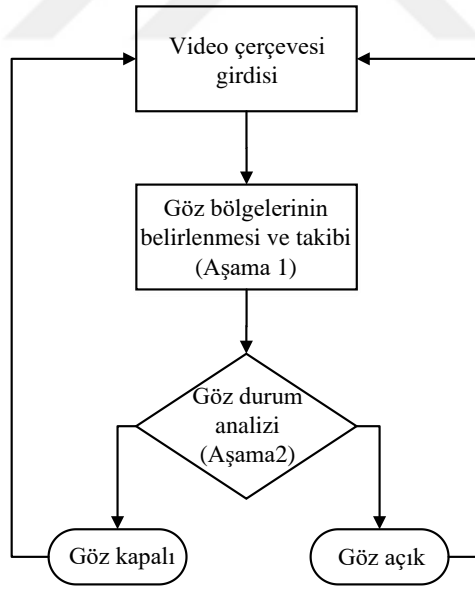
#### **2.4. Birini Dışarıda Bırak yöntemi ile çapraz doğrulama**

Birini dışarıda bırak çapraz doğrulaması (Leave-One-Out-Cross-Validation & LOOCV) örnek veri kümesinin az olduğu durumlarda kullanılan ve her defasında bir örneğin dışarıda tutularak diğer örneklerle eğitilen modelin, dışarıda tutulan örnekle test edildiği bir çapraz doğrulama yöntemidir. Test sonuçlarının ortalaması alınarak başarımlar ölçülür. Örnek veri kümesinde bulunan veri adedi kadar birbirleriyle eşlenik özellikte modeller üretilmesi nedeniyle eğitim süreci zahmetli ve zaman alıcıdır.

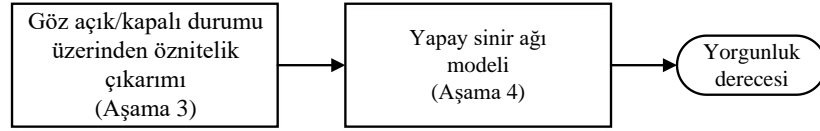
Tezde kullanılan örnek sayısının az olması nedeniyle ve diğer yöntemlere göre yüksek başarımlar sağladığından LOOCV yöntemi kullanılmıştır. Bu yöntemle sadece yapay sinir ağı sonuçları incelenmemiş ayrıca yapay sinir ağı karakteristiğini oluşturan gizli katman nöron sayıları ve giriş nöron sayısı (giriş için kullanılacak öznelik sayısına eşdeğerdir) parametrelerinin iyileştirilmesi çalışması yapılmıştır.

### 3. MATERYAL VE YÖNTEM

Bu bölümde uykulu sürücü tespit sisteminin aşamaları ayrıntılı olarak ele alınacaktır. Bölüm 3.1’de bir web kamerası aracılığı ile kaydedilen/aktarılan video görüntülerinden Shiqi’nin (2016) Haar kaskad sınıflandırıcısı kullanılarak göz bölgelerinin bulunması ve KLT (Kanade-Lucas-Tomasi) öznitelik takip algoritması ile bu göz bölgelerinin takip edilmesi anlatılmaktadır. Göz bölgelerini takip etmemizdeki amaç Haar kaskad sınıflandırıcısını her bir video çerçevesine uygulamaktan kaçınarak hesaplama yükünü azaltmaktır. Bu bize ayrıca göz bölgelerini daha hassas doğrulukla bulmamızı sağlamaktadır. Bölüm 3.2’de takip edilen bu göz bölgeleri üzerinde özel Haar kaskad sınıflandırıcısı kullanılarak gözün açıklık ve kapalılığının bulunması anlatılmaktadır (Bkz. Şekil 3.1). Bölüm 3.3’de bu göz açıklık ve kapalılık bilgisi üzerinden istatistiksel olarak yedi adet özneliğin çıkarılması aşamaları anlatılmaktadır. Bölüm 3.4’de ise bu özneliklerin kullanışlı olanlarının elenmesi ve elenen bu öznelikler ile yapay sinir ağı modellenmesi anlatılmaktadır. Yapay sinir ağı modelinin çıktısı kişinin yorgun olup olmadığı bilgisini vermektedir (Bkz. Şekil 3.2).



Şekil 3.1. Göz durum kestirimi akış diyagramı



**Şekil 3.2.** Yorgunluk derecesi kestirimi akış diyagramı

Sürücü uyku değerlendirme sisteminin donanımsal ekipmanı Intel i7 2.40 GHz işlemcili 16 GB RAM'e sahip bir bilgisayar ve bu bilgisayar üzerine gömülü bir web kamerasından oluşmaktadır. Deneyler kapsamında çekilen videolar ev/ofis ortamında 3 aylık bir zaman periyodunda çekilmiştir. Her bir video yaklaşık 2 dakikadan oluşmaktadır.

Eğitilen Haar kaskad sınıflandırıcının doğruluk testi için 6 farklı denek üzerinden 1837 adet açık ve 1322 adet kapalı göz bölgesi imgesi kullanılmıştır. Sistemin karar algoritmasının tasarımı ve testi için 6 farklı denek üzerinden 2 farklı deney yapılmıştır. Deneylerdeki kişilerin fiziksel özellikleri Çizelge 3.1.'de belirtilmiştir.

**Çizelge 3.1.** Deneye katılanların fiziksel özellikleri

Kişi	Cinsiyet	Yaş	Yüz renk tonu
# 1	Kadın	16	Kumral-Sarışın
# 2	Kadın	40	Kumral-Esmer
# 3	Erkek	33	Esmer
# 4	Kadın	24	Kumral
# 5	Erkek	48	Esmer
# 6	Erkek	28	Esmer

2 deneyin kapsamı aşağıdaki gibidir:

**Deney 1:** 5 farklı kişi üzerinde 25 adet örnek (16 dinç, 9 yorgun) kullanılarak *birini dışarda bırak çapraz doğrulama (leave-one-out-cross-validation)* yöntemi (Bkz. Bölüm 2.4.) ile en yüksek başarımın sağlandığı yapay sinir ağı oluşturulmuştur. Bu yöntemle yapay sinir ağı modeli için en uygun 1. ve 2. gizli katman nöron adetleri bulunmuş ve yapay sinir ağı girişlerini oluşturan özniteliklerin kullanışlı olanları elenmiştir. Özniteliklerin elenmesi için her iterasyonda bir adet öznitelik dışarda bırakılmış ve başarımın en yüksek olduğu öznitelikler bir sonraki basamağa aktarılmıştır. Bu şekilde



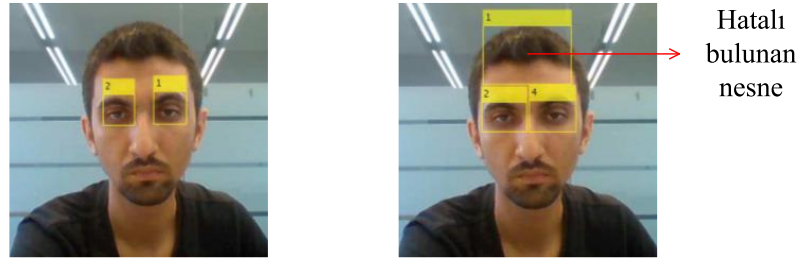
çıkarılan 7 adet öznelik sayısı 4'e düşürülmüş ve yapay sinir ağı modeli için negatif etki yaratan öznelikler elenmiştir. Mevcut iterasyon içerisinde 1. ve 2. gizli katman nöron adetlerinin sırasıyla 7-17 arasında değiştirilmesi de eklenerek yapay sinir ağının eğitilmesi ile en iyi başarımın sağlandığı nöron sayıları her iki katmanda 16 olarak bulunmuştur.

**Deney 2:** Deney 1'de en yüksek başarımı veren yapay sinir ağı kullanılarak yazar üzerinden alınan 41 adet örnek (21 dinç, 20 yorgun) üzerinde *birini dışarda bırak çapraz doğrulama* (*leave-one-out-cross-validation*) yöntemi uygulanmıştır.

Her iki deney için yapay sinir ağı transfer fonksiyonu olarak her katmanda logistic-sigmoid transfer fonksiyonu kullanılmıştır. Yapay sinir ağının çıkış katmanı iki nöronlu olup değerleri 0 ve 1 arasında çıkış olasılıklarını yansıtmaktadır. Bu değerlerin birbirlerine göre büyüklükleri kişinin yorgun olup olmadığına karar verilmesini sağlar. Yapay sinir ağı modellenmesi için *birini dışarda bırak* yönteminin her adımında kullanılan 1'er örnek dışında test ve doğrulama örneği kullanılmamıştır.

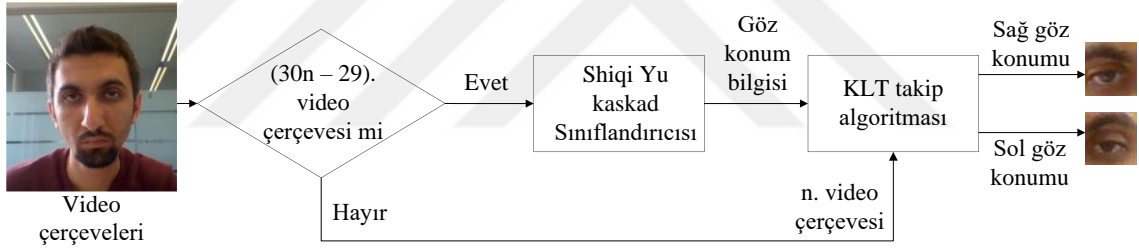
### **3.1. Göz Bölgelerinin Bulunması ve Takip Edilmesi**

Göz bölgelerinin belirlenmesi ve imge üzerinden kırılması için web kamerası üzerinden alınan  $640 \times 480$  video çerçeveleri üzerinde çalışılmıştır. Video çerçeveleri üzerinde göz bölgelerinin bulunması için Shiqi'nin (2016) eğitmiş olduğu Haar kaskad sınıflandırıcısı ile Castrillón vd.'nin (2007) kaskad sınıflandırıcısı denenmiştir. Castrillón vd.'nin kaskad sınıflandırıcısı bazı çerçeveler için kaş, sakal ve saçları da göz bölgesi olarak tespit ettiği için bunun yerine daha başarılı sonuç verdiği için Shiqi'nin kaskad sınıflandırıcısı kullanılmıştır (Bkz. Şekil 3.3). Shiqi; FERET, BioID ve VALID veri tabanlarını kullanarak ve iki gözü ayrı olarak 6000 pozitif örneğin üzerinde örnek kullanarak eğitim veri tabanını oluşturmuştur.



**Şekil 3.3.** Aynı imge için Shi Qi'nin (2016) (sol) ve Castrillón vd.'nin (2007) (sağ) referanslarındaki algoritmaların uygulanmasıyla elde edilen göz bölgesi sınıflandırma sonuçları

Tezde hareketli çerçeveler arasındaki geçişte göz bölgelerinin bulunma doğruluğunu artırmak ve hesaplama yükünü azaltmak için kaskad sınıflandırıcının ardından KLT öznelik takip algoritması uygulanmıştır. Video çerçeve sayısı  $n \in \mathbb{N}^+$  olmak üzere her  $30n - 29$  çerçevede bir kaskad sınıflandırıcının bulduğu göz bölgeleri sonraki çerçevelerde öznelik takibi için kullanılacak şablon imgeyi tanımlar (Bkz. Şekil 3.4).



**Şekil 3.4.** Göz bölgelerinin kırılması akış diyagramı

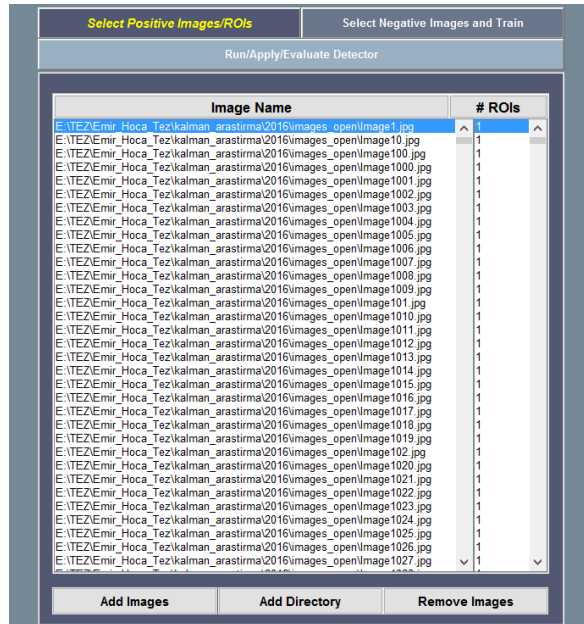
### 3.2. Göz Açıklık – Kapalılığının (Göz Durumunun) Çıkarımı

Göz bölgelerinin kırılmasından sonra elde edilen imgeler üzerinde göz durumu analizi yapılabilmesi için Haar kaskad sınıflandırıcı eğitimi yapılmıştır. Bu Haar kaskad sınıflandırıcı bize gözün açık / kapalı bilgisini vermektedir. Haar kaskad sınıflandırıcı tezde göz bölgelerinin belirlenmesi için (Göz bölgelerinin kırılması) ve göz açık-kapalı durumunu sınıflandırmak için (Göz durumu analizi) 2 kez kullanılmıştır. Göz bölgelerinin bulunması için kullanılan sınıflandırıcı Shi Qi Yu kaskad sınıflandırıcısı olup göz açık-kapalı durumunu sınıflandırmak için veri seti oluşturularak özel Haar kaskad sınıflandırıcı yaratılmıştır. Özel sınıflandırıcı, belirlenen bu göz bölgeleri üzerinden gözün açık durumunu sınıflandırmıştır. Özel sınıflandırıcı eğitiminde 3 kaskad etap, 2795

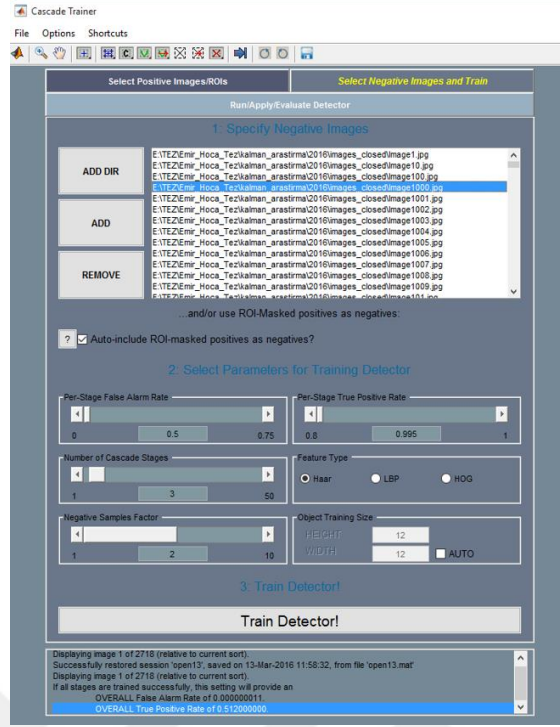
pozitif örnek ve 4367 negatif örnek kullanılmıştır. Hata uyarı oranı(FAR - False Alarm Rate) 0,023 ve doğru pozitif(TP True-Positive) oranı 0,80 olarak seçilmiştir. Pozitif örnekler açık gözlerden ve 2 esmer, 1 kumral erkek ile 1 kumral, 1 esmer, 1 sarışın kadın olmak üzere 6 farklı kişi üzerinden toplanmıştır. Negatif örnekler, kapalı gözlerden ve göz olmayan imgelerden oluşmaktadır. Özel kaskad sınıflandırıcı sadece içerisinde açık ya da kapalı göz bulunduran imge parçalarına uygulandığından belirtilen parametreler ile % 98 TP ve % 87 TN ortalama doğruluk oranı elde edilmiştir. Sezgisel yaklaşımla bir göz kırpma süresinin kameranın bir çerçeve yakalaması için geçen süreden (30 çerçeve/sn(fps) ) kısa olamayacağı düşünülerek açık çerçeveler arasında gelen tek kapalı çerçeve ile kapalı çerçeveler arasında gelen tek açık çerçeveyi düzelterek % 0.014 daha fazla doğruluk oranı elde edilmiştir(Bkz. Denklem 3.1).

$$x_i = \begin{cases} 1 & x_{i-1} = 1, x_{i+1} = 1 \\ 0 & x_{i-1} = 0, x_{i+1} = 0 \end{cases} \quad (3.1)$$

Özel sınıflandırıcı eğitiminde MATLAB yazılımı içerisinde eklenen *Cascade Trainer* uygulaması kullanılmıştır. Tüm parametreler ve örnek uzayı bu uygulama üzerine eklenerek sınıflandırıcı eğitilmiştir. Şekil 3.5’de uygulama üzerinde gösterilen pozitif imgeler, Şekil 3.6’da uygulama üzerinde gösterilen negatif imgeler ve eğitim parametreleri gösterilmektedir.



Şekil 3.5. Cascade Trainer uygulamasında gösterilen pozitif örnekler

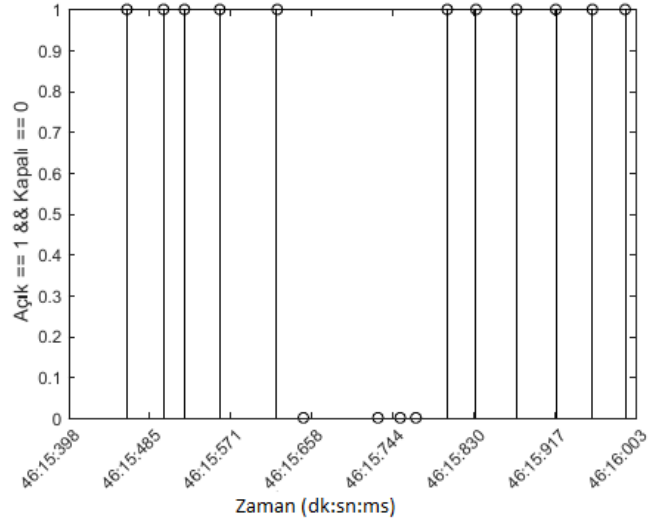


**Şekil 3.6.** *Cascade Trainer* uygulamasında gösterilen negatif örnekler ve parametre değerleri

MATLAB ortamında *Cascade Trainer* uygulaması kullanılarak eğitilen özel kaskad sınıflandırıcı yine MATLAB ortamında kullanılması mümkün .xml uzantılı dosya çıktısı vermektedir. Elde edilen .xml uzantılı dosya, MATLAB dosyası içerisinde kullanılacak sınıflandırıcıyı belirtecek fonksiyona giriş niteliği taşır.

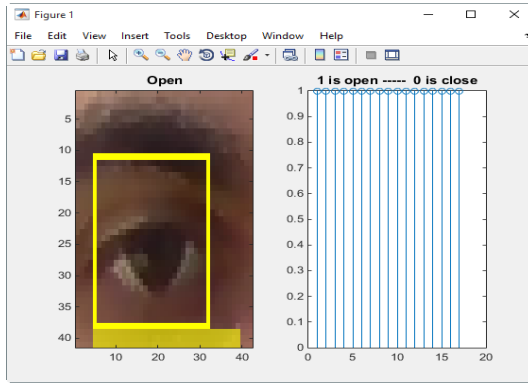
```
opendetector = vision.CascadeObjectDetector('open13.xml'); %Ornek MATLAB kodu
```

Göz açık durumu ve dışında kalan örnekler için gözün kapalı olduğu düşünülerek zamana karşılık 1(Açık) - 0(Kapalı) göz durum vektörü oluşturulmuştur. Sağ ve sol gözün aynı anda açılıp kapanacağı varsayımıyla sınıflandırıcı, bir çerçeve için sadece sağ göze uygulanmaktadır. Şekil 3.7’de örnek bir göz durum vektörü gösterilmiştir. Ardı ardına gelen ‘Açık’ çerçeveler gözün açık olduğu, ardı ardına gelen ‘Kapalı’ çerçeveler gözün kapalı olduğu anları temsil etmektedir. Ardı ardına gelen açık çerçeveler sonrasında gelen kapalı çerçeve, gözün kapandığı anı temsil etmektedir. Bu bilgiler sayesinde 0,03 saniye yaklaşıklıkla gözün kapandığı an ve açıldığı an yakalanabilir.

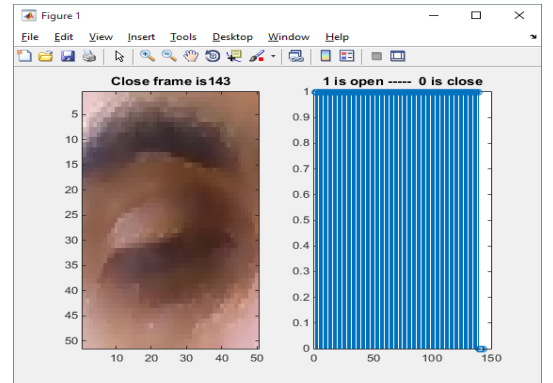


**Şekil 3.7.** Zaman damgasıyla örnek göz durum vektörü

Şekil 3.8, göz bölgelerinden oluşan bir video içerisindeki her bir çerçeve ve onlara ait açık-kapalı durumunun gerçek zamanlı olarak izlenebildiği MATLAB arayüzünden görüntüler içermektedir. Her bir görüntü, kırılan göz bölgeleri ve açık – kapalı durumunu yansıtan göz durum vektörü alt çizimlerini içerir. Göz durum vektöründeki 1 gözün açık; 0 gözün kapalı olduğunu ifade eder. Gözün açık durumunun sınıflandırıldığını göstermek için kırılan göz bölgesi üzerine dikdörtgen bir açıklama penceresi çizilmiştir.



(a)



(b)

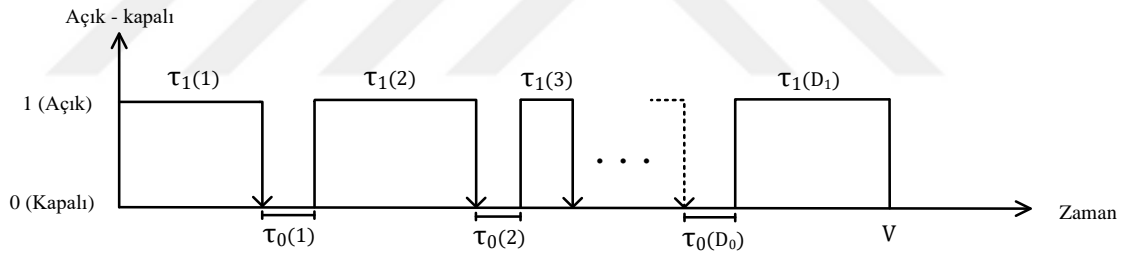
**Şekil 3.8.** Özel Haar kaskad sınıflandırıcının testi için yazılmış olan MATLAB uygulamasının (a) göz açıkken ve (b) göz kapalı iken görsel çıktısı

### 3.3. İstatistiksel Özniteliklerin Çıkarılması

Zaman etiketli göz durum vektörü üzerinden yorgunluk derecesini belirlememize yarayacak öznitelikler çıkarabiliriz. Bu öznitelikler göz kırpma frekansı, gözün kapalı kalma sürelerinin toplamı, gözün tek seferde ortalama kapalı kalma süresi, göz kapalılık sürelerinin standart sapması, gözün açık kalma sürelerinin toplamı, gözün tek seferde ortalama açık kalma süresi, gözün açık kalma sürelerinin standart sapması ve kapalılık yüzdesi şeklinde sıralanabilir. Göz durum vektöründeki düşen ve yükselen kenarların zaman bilgisini çıkararak basitçe bu öznitelikler hesaplanabilir. Bölüm 3.3'ün alt bölümlerinde sırasıyla bu özniteliklerin nasıl çıkarıldığı açıklanmıştır.

#### 3.3.1. Göz kırpma sıklığı

Şekil 3.9, temsili olarak yaklaşık 2 dakika boyunca kaydedilmiş bir video üzerinden çıkarılmış göz durum vektörü olmak üzere, göz kırpma sıklığı özneliği bu vektörün bir dakikada düşen kenarlarının sayısı kadardır.



**Şekil 3.9.** Temsili olarak göz durum vektörü üzerinde gözün kapandığı anların gösterimi

$D_0$  düşen kenarların toplam sayısı,  $V$  saniye cinsinden toplam video süresi olmak üzere göz kırpma sıklığı Denklem 3.2 ile ifade edilir.

$$F_1 = \frac{60 \cdot D_0}{V} \quad (3.2)$$

### 3.3.2. Toplam göz kapalılık süresi

$\tau_0(i)$  göz kapama süreleri olmak üzere toplam göz kapalılık süresi her bir göz kırpmada gözün kapalı kaldığı sürelerin toplamının dakikadaki karşılığıdır. Denklem 3.3 ile ifade edilir.

$$F_2 = \frac{60 \cdot \sum_{i=1}^{D_0} \tau_0(i)}{V} \quad (3.3)$$

### 3.3.3. Gözün kapalı kaldığı sürelerin ortalaması

Gözün kapalı kaldığı sürelerin ortalaması, gözün tek kapanışta ortalama ne kadar süre kapalı kaldığı bilgisini vermektedir. Denklem 3.4 ile ifade edilir.

$$F_3 = \frac{1}{D_0} \sum_{i=1}^{D_0} \tau_0(i) \quad (3.4)$$

### 3.3.4. Gözün kapalı kaldığı sürelerin standart sapması

Standart sapma, ortalama değerden ne kadar uzaklaşan veri ya da veriler olduğunun ifade edilmesi için kullanılan istatistiksel bir ölçü birimidir. Göz kapama sürelerinin standart sapması Denklem 3.5 ile ifade edilir.

$$F_4 = \sqrt{\frac{1}{D_0} \sum_{i=1}^{D_0} (\tau_0(i) - F_3)^2} \quad (3.5)$$

### 3.3.5. Toplam göz açıklık süresi

Toplam göz açıklık süresi, video boyunca gözün ne kadar süre açık kaldığının dakikadaki karşılığıdır.  $\tau_1(i)$  gözün açık kaldığı süreler olmak üzere Denklem 3.6 ile ifade edilir.  $D_1$  gözün açık olarak videonun bittiği durumda  $D_0 + 1$ 'e, gözün kapalı olarak videonun bittiği durumlarda  $D_0$ 'a eşittir.

$$F_5 = \frac{60 \cdot \sum_{i=1}^{D_1} \tau_1(i)}{V} \quad (3.6)$$

### 3.3.6. Gözün açık kaldığı sürelerin ortalaması

Video boyunca gözün açık kaldığı sürelerin ortalamasıdır. Denklem 3.7 ile ifade edilebilir.

$$F_6 = \frac{1}{D_1} \sum_{i=1}^{D_1} \tau_1(i) \quad (3.7)$$

### 3.3.7. Gözün açık kaldığı sürelerin standart sapması

Gözün açık kaldığı sürelerin ortalamasıyla olan istatistiksel ilişkiyi vermektedir. Denklem 3.8 ile ifade edilir.

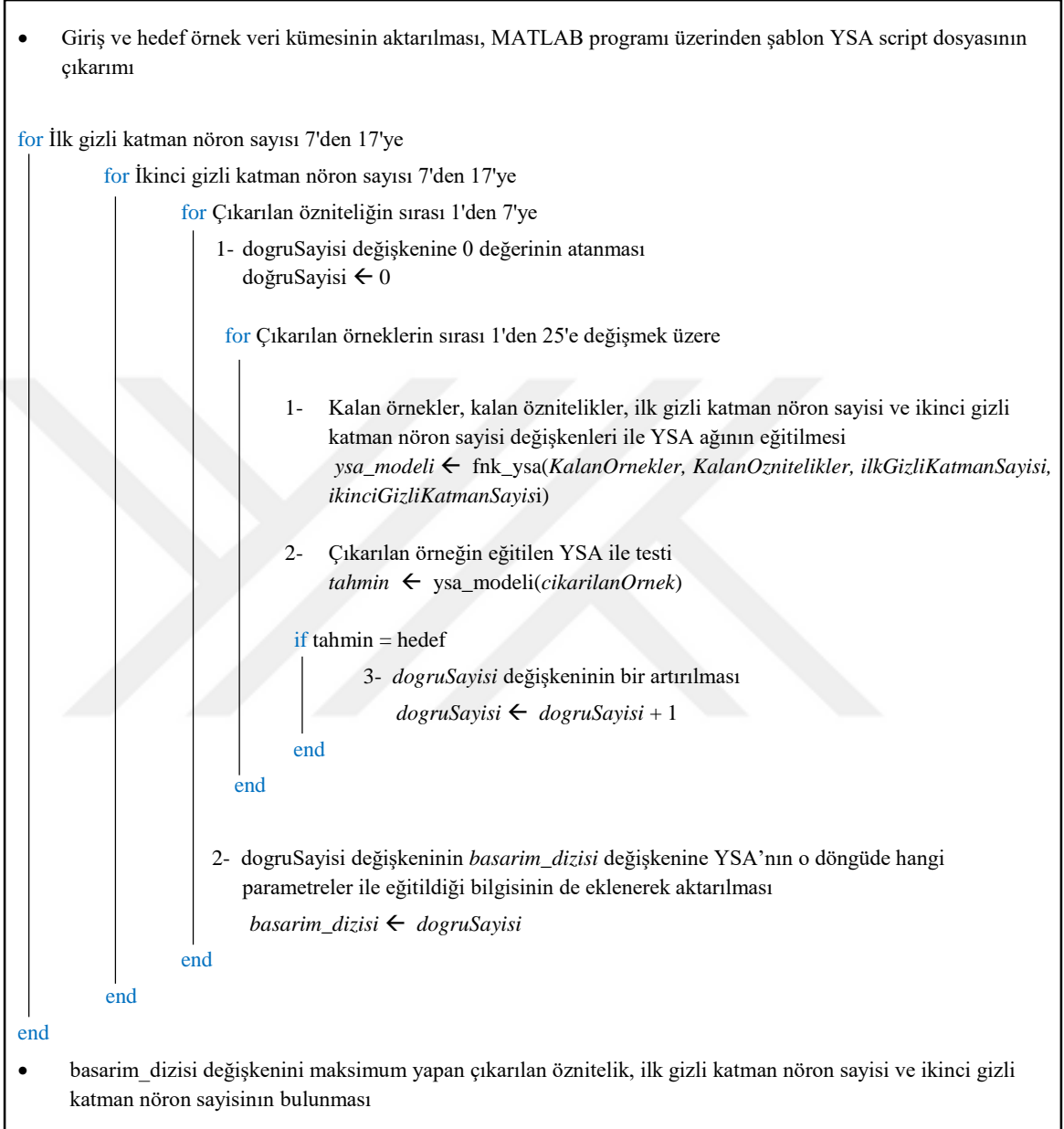
$$F_7 = \sqrt{\frac{1}{D_1} \sum_{i=1}^{D_1} (\tau_1(i) - F_6)^2} \quad (3.8)$$

## 3.4. Yorgunluk Tespiti

Göz durum vektörü üzerinden çıkarılan öznitelikler üzerinde, özniteliklerin birbirleri arasında direkt bir matematiksel modelinin kurulamaması nedeniyle yorgunluk tespitinde karar algoritması olarak danışmanlı yapay sinir ağı modellenmesi uygun bulunmuştur. Danışmanlı yapay sinir ağı modeli için toplam 25 adet 2 dakikalık video bölümleri çekilmiş ve tüm örnekler eğitim aşamasında değerlendirilmiştir. Çekilen video kliplerinden çıkarılan 7 adet özniteliğe, yapay sinir ağı modelinin en iyi sonucu vermesi için her defasında birer adet örnek ve öznitelik dışarıda bırakılarak *birini dışarıda bırak* yöntemi uygulanmıştır. Birini dışarıda bırak yöntemi ile ardı ardına iterasyonlarla öznitelik sayısı 4 adede düşürülmüş, sadece yüksek başarımla sağlanan öznitelikler yapay sinir ağı giriş nöronuna uygulanmıştır. Oluşturulacak yapay sinir ağı modelinin başarımlarını artırmak için yapay sinir ağının optimum birinci ve ikinci gizli katman nöronlarının sayısı birini dışarıda bırak yöntemi ile elde edilen başarımların yüzdeleri kullanılarak empirik(deneme yanılmaya dayalı) olarak belirlenmiştir. Şekil 3.10'da yapay



sinir ağı karakteristiğini yansıtan 1. ve 2. gizli katman nöron sayıları ve yüksek başarılı özniteliklerin bulunmasına dair sözde-kod gösterilmektedir.



**Şekil 3.10.** Yapay sinir ağı modeli parametrelerinin iyileştirilmesinde kullanılan sözde-kod gösterimi

Şekil 3.10'daki sözde-kodu yansıtan algoritma ile öznitelik setinden çıkarıldığında başarımlarını en çok artıran öznitelik tespit edilmiş ve YSA bu öznitelik kullanılmadan tekrar eğitilip test edilmiştir. Ardı ardına yapılan iterasyonlar

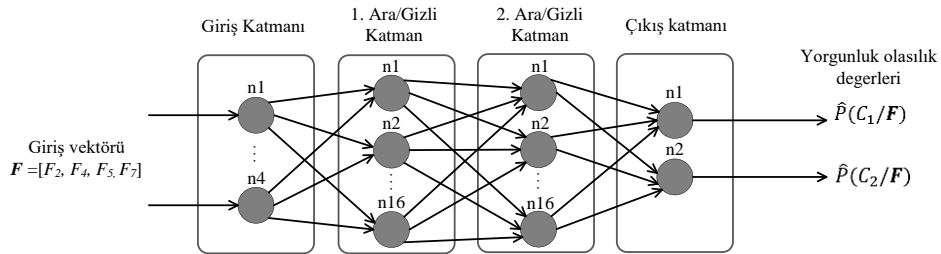
göstermektedir ki yapay sinir ağı maksimum başarımı sadece 4 adet öznelik kullanılarak elde edilebilmektedir. Bu öznelikler; gözün kapalı kaldığı sürelerin standart sapması  $F_4$ , gözün kapalı kaldığı sürelerin toplamı  $F_2$ , gözün açık kaldığı sürelerin standart sapması  $F_7$  ve gözün açık kaldığı sürelerin toplamı  $F_5$ 'tir. Buna ek olarak bulunan maksimum başarımla, yapay sinir ağı 1. ve 2. gizli katman sayıları 16 iken sağlanmıştır. Yapay sinir ağı modellenirken her bir iterasyonda elenen öznelikler ve elenmesini sağlayan başarımla kriterleri:

İterasyon 1: En yüksek başarımla  $F_6$  özneliği çıkarıldığında ve 16 farklı 1. ve 2. gizli katman sayısı durumu için 25 örneğin 20'sinin doğru olarak tespit edildiği durumda sağlanmıştır.

İterasyon 2: Kalan 6 öznelik arasından en yüksek başarımla  $F_3$  özneliği çıkarıldığında ve 9 farklı 1. ve 2. gizli katman sayısı durumu için 25 örneğin 21'inin doğru olarak tespit edildiği durumda sağlanmıştır.

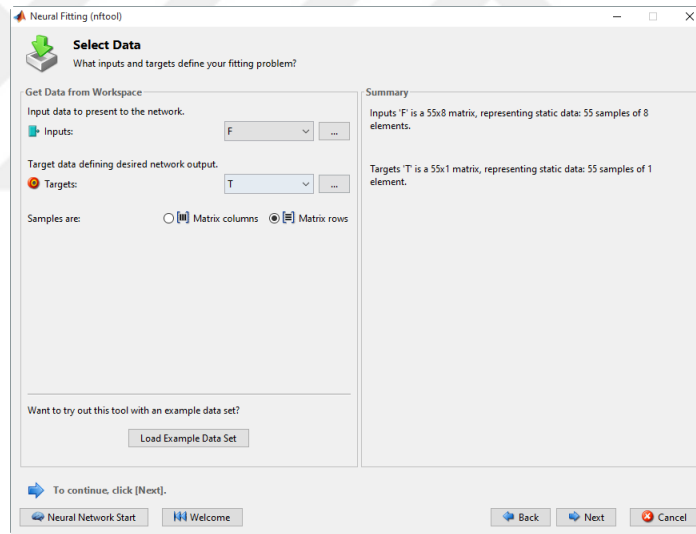
İterasyon 3: Kalan 5 öznelik arasından en yüksek başarımla  $F_1$  özneliği çıkarıldığında ve sadece 1. ve 2. gizli katman sayısının 16 adet olduğu durumda, 25 örneğin 24'ünün doğru olarak tespit edildiği durumda sağlanmıştır.

İterasyon 3'den sonra yapılan iterasyon neticesinde başarımla artırıcı bir sonuç elde edilememiştir. Bundan dolayı iterasyonlarda elenen öznelikler sonucunda kalan öznelikler ve gizli katmanlardaki nöron sayıları 16 adet olduğu durum için yapay sinir ağı modeli ele alınmıştır. Yapay sinir ağı çıkışında 2 adet nöron bulunmakta ve bu nöronlar yorgunluk-ayıklık olasılık değerleri bilgisini vermektedir (Bkz. Şekil 3.11).



**Şekil 3.11.** Yorgunluk tespiti için kullanılan yapay sinir ağı modeli

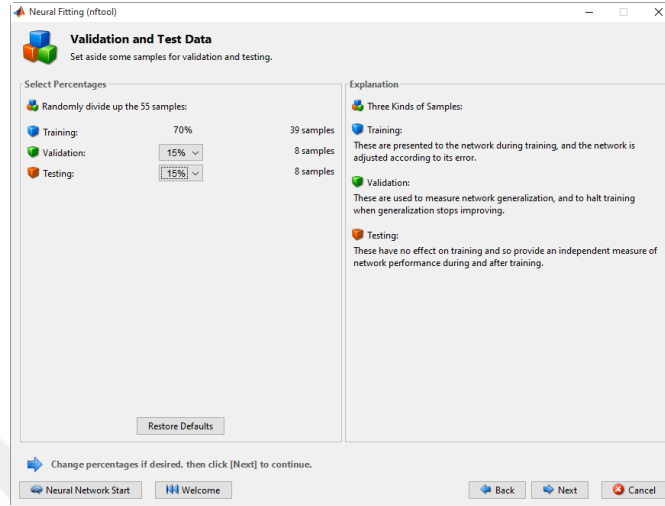
Bu olasılık deęerleri karřılařtırılarak, kiřinin yorgun olup olmadıęı kararı verilmektedir. Eęitim algoritması olarak daha az bir eęitim suresi gerektirdięinden ve eęiteceęimiz ok fazla sayıda giriř verilerimiz olmadıęından geri beslemeli bir eęitim algoritması olan Levenberg-Marquardt (LM) algoritması seilmiřtir (Quesada 2016). Eęitim iin kullanılan hedef ıkıř deęerleri, video blm ekilen kiřiye sorularak subjektif olarak alınmiřtır. Bařlangı aęırlıkları iin bilgisayar ortamında rastgele deęerler alınmiřtır. Aktivasyon fonksiyonları her katmanda logistic-sigmoid olarak seilmiřtir. Danıřmanlı yapay sinir aęı modellemesinde ilk olarak MATLAB yazılımının Neural Network Toolbox eklentisi ve Neural Fitting arayz uygulaması kullanılmıřtır. Bu eklenti ve arayz, yapay sinir aęı modelinin oluřturulması iin kullanılan fonksiyonun řablon dosyasını sunmaktadır. řekil 3.12’de giriř verileri ve hedef verilerin uygulamaya aktarılması gsterilmektedir. Uygulamanın MATLAB zerinden aılması iin komut penceresi zerine *nftool* yazılması yeterlidir.



**řekil 3.12.** Giriř ve hedef verilerin uygulama arayzne aktarılması

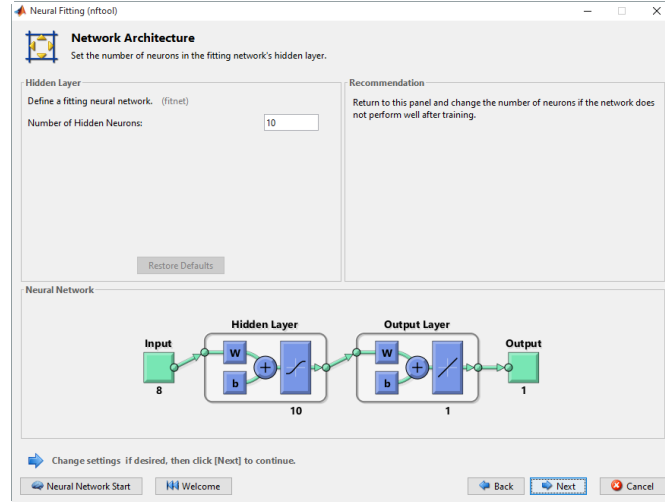
Kullanılan giriř ve hedef verileri MATLAB Workspace’inde nceden dzenlenmiř ve 2 farklı deęiřken olarak bulunmaktadır. řekil 3.13’de eęitim, test ve doęrulama verilerinin uygulama zerinden gsterimi bulunmaktadır. Her ne kadar tm veri kmesinin eęitim iin kullanıldıęı belirtilmiř olsa da bu kısımda varsayılan olarak uygulama zerinde seilen rnekler arasında %70’lik bir blmnn eęitim, %15’lik bir blmnn test ve geri kalan %15’lik bir blmnn doęrulama verisi olarak kullanılacaęı belirtilmiřtir.

Bunun nedeni arayüz üzerinde tüm veri kümesinin eğitim için kullanılacağı belirtilemiyor olmasıdır. Daha sonra elde edilen MATLAB script dosyası üzerinden bu kısma ait parametreler değiştirilecektir.



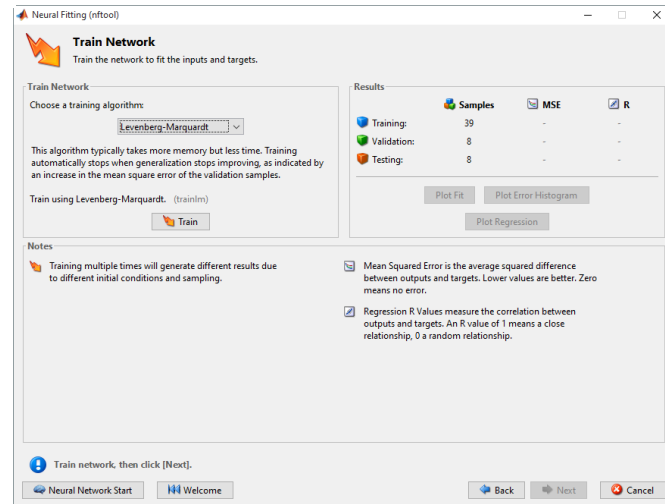
**Şekil 3.13.** Yapay sinir ağı modelinin eğitim, test ve doğrulama verisi sayılarının uygulama üzerinden gösterimi

Şekil 3.14’de gizli katmanda ne kadar nöron olacağını uygulamada gösterimi ve buna karşılık tezde kullanılan yapay sinir ağının temsili gösterimi bulunmaktadır. Bir önceki aşamada veri kümesi gösterimine benzer şekilde arayüz üzerinden 2. gizli katmanın nöron sayısı gösterilememektedir. Daha sonra elde edilecek olan *Advanced script* dosyası üzerinden 1. ve 2. gizli katmanların nöron sayıları gösterilecektir.



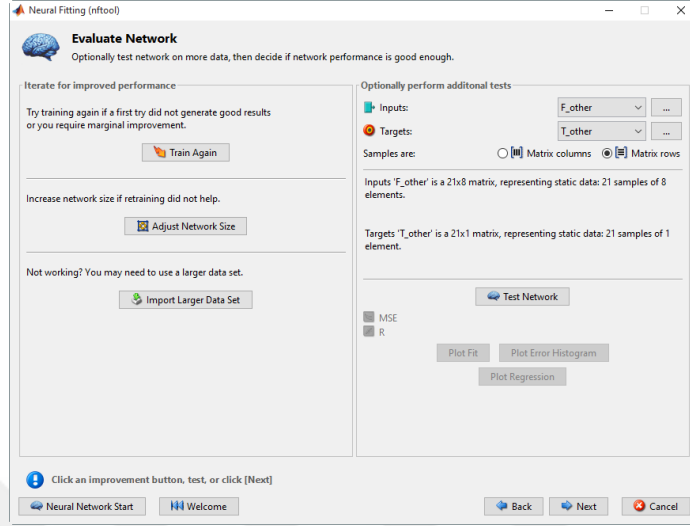
**Şekil 3.14.** Uygulama üzerinde ne kadar gizli katman nöronu bulunacağını gösterimi ve temsili yapay sinir ağı

Şekil 3.15’de eğitim algoritmasının seçimi ve Train butonu ile eğitimin başlatılması gösterilmiştir. Burada başarımlar *Results* kısmı altında MSE(Ortalama kare hatası) ve R(Pearson korelasyon katsayısı) değerleri gösterilerek belirtilmiştir. Henüz istediğimiz parametre girişlerini gösteremediğimiz için burada yer alan başarımlarını dikkate almadan bir sonraki basamağa geçilmiştir.



**Şekil 3.15.** Eğitim algoritmasının seçimi ve ağı eğitilmesi

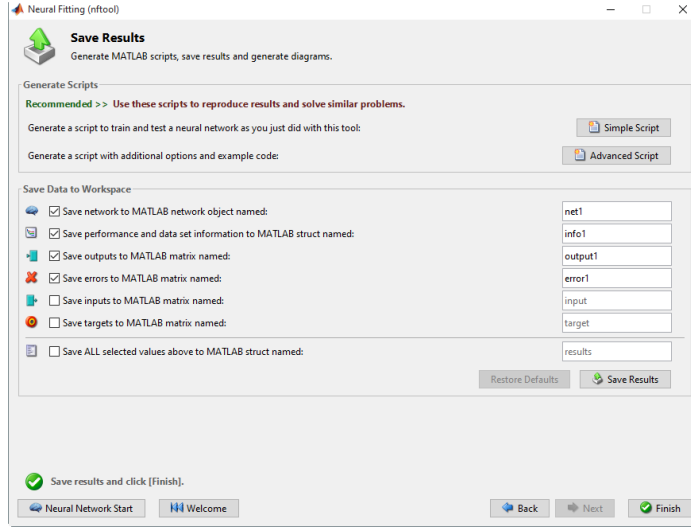
Şekil 3.16’da farklı test veri kümeleri kullanarak opsiyonel testlerin gösterildiği uygulama bölümü yer almaktadır.



**Şekil 3.16.** Yapay sinir ağının opsiyonel testlerle değerlendirilmesinin yapılması

Şekil 3.17’de sonuçların MATLAB değişkeni olarak nasıl kaydedileceği ve MATLAB dosyaları olarak gösterilmesi seçeneklerinin yer aldığı uygulama arayüzü bulunmaktadır. Advanced script butonunu kullanarak elde edilen dosya üzerinde ilgili yerler değiştirilerek ağ parametreleri tekrar düzenlenmiştir. Düzenlenen parametreler:

- Yapay sinir ağının 2 katmanlı olarak gösterilmesi ve katmanlardaki nöron sayısı
- Doğrulama ve test verisinin tüm veri kümesine oranla varsayılan %15 değerinden %0 değerine düşürülmesi.
- Her katmandaki aktivasyon fonksiyonlarının logistic sigmoid olarak ifade edilmesi



**Şekil 3.17.** Sonuçların MATLAB'e aktarılması ve MATLAB dosyası olarak çıktısının alınması

Düzenlenen parametreler ile birlikte elde edilen *Advanced Script* dosyası, döngü içerisinde değişken verilere uygun olması açısından MATLAB fonksiyonu haline getirilmiştir.

## 4. BULGULAR

Bu bölümde yapılan testler ve uygulanan deneyler ayrıntılı olarak ele alınmıştır. Bölüm 4.1’de göz açıklık ve kapalılığının bulunması için eğitilen Haar kaskad sınıflandırıcısının başarımından bahsedilmektedir. Bölüm 4.2’de ise sistem başarım kriteri olarak eğitilen yapay sinir ağı başarımı ve diğer çalışmalarla kıyaslaması yapılmaktadır.

### 4.1. Göz Açıklık Kapalılık Bulguları

1837 adet açık ve 1322 adet kapalı göz imgesi kullanılarak deneye katılan kişiler üzerinden Çizelge 4.1’de gösterilen doğruluk oranları elde edilmiştir. Burada TP(doğru pozitif) gözün açık olup sınıflandırmanın açık sonuçlandığı durumları, FN(hatalı negatif) gözün açık olup sınıflandırmanın kapalı olarak sonuçlandığı durumları, TN(doğru negatif) gözün kapalı olup sınıflandırmanın kapalı olarak sonuçlandığı durumları, FP(hatalı pozitif) gözün kapalı olup sınıflandırmanın açık olarak sonuçlandığı durumları temsil eder. Değerler yüzdelik olarak verilmiştir.

#### Çizelge 4.1. Göz açıklık kapalılık doğruluğu

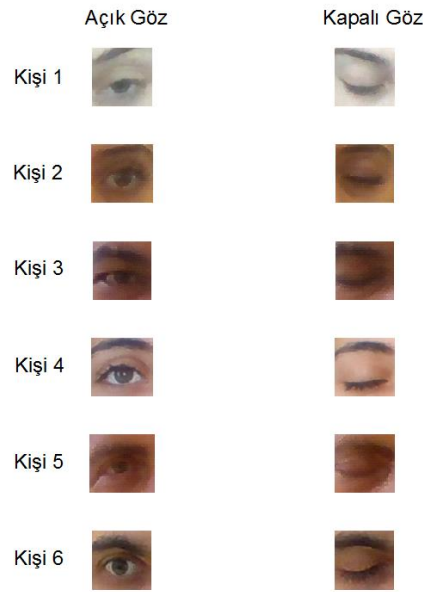
(Göz açık – pozitif , Göz kapalı - negatif)

Denek	TP	TN	FP	FN
Kişi 1	0,99	0,69	0,31	0,01
Kişi 2	1	1	0	0
Kişi 3	0,99	1	0	0,01
Kişi 4	1	0,70	0,30	0
Kişi 5	0,99	1	0	0,01
Kişi 6	0,96	0,99	0,01	0,04
Toplam	0,98	0,83	0,17	0,02













Haar kaskad sınıflandırıcı eğitiminde kullanılan parametre seçimi ve gösterilen negatif örnekler Çizelge 4.1’deki göz açıklık - kapalılık doğruluğunu direkt olarak etkilemektedir. Kullanılan kaskad bölümlerin sayısını artırmak ve bu parametreyi destekleyecek şekilde negatif örnek sayısını artırmak doğruluk değerlerini yükseltecektir. Ayrıca göz bölgeleri küçük olan deneklerde göz bölgeleri büyük olanlara göre doğruluk değerinin daha düşük olarak bulunduğu gözlemlenmiştir. Genel olarak açık gözlerin açık olarak tespit edilmesi, kapalı gözlerin kapalı olarak bulunmasına göre daha yüksek



doğrulukta bulunmuştur. Gözlerin açık anından kapalı anına veya kapalı anından açık anına geçişlerdeki yarı-açık göz durumlarının açık olarak değerlendirilmesi ve sınıflandırıcı eğitiminde bu şekilde kullanılması FP değerlerinde artışa neden olmuştur. Girit'in (2014) göz sınıflandırıcısına benzer şekilde 3 durumlu (göz açık – göz yarı açık – göz kapalı) bir sınıflandırıcı kullanımı, göz durumlarını ifade etmek için daha doğru bir sonuç üretebileceği düşünülmüştür. Fakat burada gözün yarı açık durumunun bulunması başarıyı %31,9 olarak ölçüldüğünden dolayı geçiş anlarını doğru olarak ifade etmekte problem yaşanacağı düşünülmüştür (Girit 2014). Yorgunluk tespiti için gözün durumları arasındaki geçiş anları karar algoritmamızın girişlerini oluşturduğundan bize göz yarı açıklık durumu yorgunluk kararını vermek için ek bir başarıyı sağlamayacağı düşünülmüş ve göz açık-kapalı olarak 2 sınıf kullanılmıştır. Deneye katılan kişilerin örnek değerlendirme sonuçları, doğru değerlendirilenler olarak Şekil 4.1'de ve yanlış değerlendirilenler olarak Şekil 4.2'de belirtilmiştir.



**Şekil 4.1.** Deneye katılan kişilerin doğru olarak değerlendirilen açık ve kapalı göz örnekleri

	Kapalı Göz	Açık Göz
Kişi 1		
Kişi 2		
Kişi 3		
Kişi 4		
Kişi 5		
Kişi 6		

**Şekil 4.2.** Deneye katılan kişilerin hatalı olarak değerlendirilen açık ve kapalı göz örnekleri. 1. sütun gözün kapalı iken açık, 2. sütun gözün açık iken kapalı olarak değerlendirilen hatalı örneklerini temsil etmektedir (Gri bölgeler o kişi ve durum için örnek veri bulunamadığı anlamını taşır)

## 4.2. Sürücü Yorgunluk Bulguları

Bu bölümde 6 farklı kişi üzerinden yapılan 2 adet deneyin yorgunluk değerlendirme sonuçları sunulmakta ve diğer çalışmalarla kıyaslanmaktadır.

### 4.2.1. Deney 1

Deney 1, 5 farklı kişi üzerinden alınan 9 adet yorgun ve 16 adet dinç örnek olmak üzere 25 adet video örneği üzerinde yapılan yapay sinir ağı parametre iyileştirme çalışmalarını kapsamaktadır. *Birini dışarıda bırak* yöntemiyle gerçekleştirilen iterasyonlar sonunda göz açıklık ve kapalılık bilgilerinden çıkarılan 7 adet öznitelik sayısı 4'e düşürülmüştür. Her bir iterasyonda alınan sonuçlar:

*İterasyon 1:* Her defasında 7 adet özniteliğin birini dışarıda bırakarak, 7-17 gizli katman nöron adetleri arasında yapay sinir ağı eğitimlerinden oluşmaktadır. Çizelge 4.2’de *F1* özniteliğinin dışarıda bırakılması ile 25 adet örneğin her birinin sırasıyla dışarıda bırakılması ve toplamda dışarıda bırakılan örneklerin ne kadarının doğru tespit edildiğine dair sonuçlar yer almaktadır. Satırlar 1. gizli katman nöron adedini, sütunlar 2. gizli katman nöron adedini temsil etmektedir. Örnek bir çizelge okuma şekli:

*F1* özniteliğinin çıkarıldığı durumda, 1. gizli katman 9 adet ve 2. gizli katman 11 adet seçildiğinde eğitilen yapay sinir ağı kullanılarak sırasıyla test edilen 25 örneğin 15’inin doğru olarak tespit edildiği görülmüştür.

**Çizelge 4.2.** İterasyon 1 için *F1* özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları

2.GK nöron adedi \ 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	13	15	18	14	15	12	11	13	17	14	16
8	14	19	17	14	13	17	17	13	16	18	17
9	17	14	17	17	<b>15</b>	18	15	17	17	13	18
10	17	19	13	15	16	12	14	16	13	13	15
11	16	14	15	15	16	13	17	15	16	16	19
12	11	12	18	14	16	14	16	15	14	13	18
13	15	18	14	10	14	18	14	19	14	17	18
14	16	16	15	13	16	15	17	13	17	17	14
15	15	15	15	16	15	17	15	14	14	17	14
16	15	16	15	13	14	14	16	15	15	15	15
17	13	14	17	11	14	17	15	17	13	17	11

Çizelge 4.3’de  $F2$  özniteliğinin çıkarıldığı durum için başarımların sonuçları yer almaktadır.

**Çizelge 4.3.** İterasyon 1 için  $F2$  özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımların sonuçları

2.GK nöron adedi 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	13	14	18	16	18	18	14	14	16	14	16
8	17	15	19	16	17	14	15	17	12	19	12
9	17	15	16	16	16	17	16	16	18	14	14
10	18	15	14	14	16	18	16	17	16	16	13
11	14	15	13	15	14	11	15	18	15	16	16
12	19	17	11	14	17	16	14	19	13	14	15
13	17	13	16	15	17	14	18	19	15	17	18
14	17	14	14	14	12	17	11	14	15	14	16
15	18	13	12	18	17	19	17	14	14	17	15
16	16	14	14	14	15	12	16	18	15	14	12
17	16	15	15	15	15	14	15	14	19	15	17

Çizelge 4.4’de  $F3$  özniteliğinin çıkarıldığı durum için başarımların sonuçları yer almaktadır.

**Çizelge 4.4.** İterasyon 1 için  $F3$  özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımların sonuçları

2.GK nöron adedi 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	20	16	15	19	17	18	16	17	17	16	14
8	16	15	14	16	18	15	16	14	14	17	16
9	16	16	16	15	16	15	17	16	14	15	17
10	17	17	16	14	17	14	14	20	18	15	17
11	18	17	16	16	18	18	13	16	15	18	17
12	16	18	17	15	15	17	18	19	21	17	19
13	16	17	17	19	17	18	19	16	17	15	18
14	15	16	16	18	16	16	15	15	18	18	16
15	18	15	15	15	17	16	15	13	19	11	15
16	15	13	18	17	19	15	15	14	13	14	16
17	17	15	16	19	18	17	15	19	17	16	17

Çizelge 4.5’de  $F4$  özniteliğinin çıkarıldığı durum için başarımların sonuçları yer almaktadır.

**Çizelge 4.5.** İterasyon 1 için  $F4$  özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımların sonuçları

2.GK nöron adedi 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	14	15	14	13	14	14	13	14	14	14	17
8	14	14	15	13	14	13	14	16	11	12	16
9	15	14	14	12	15	14	14	15	15	14	16
10	14	12	15	14	13	15	14	13	13	16	12
11	16	20	17	15	15	13	14	13	15	16	13
12	14	11	13	17	14	14	12	14	15	13	14
13	11	13	14	13	17	13	15	14	15	15	14
14	12	13	10	14	13	16	13	15	16	13	13
15	15	12	16	12	14	13	14	15	13	14	14
16	13	16	12	14	13	17	13	12	16	12	14
17	15	14	15	15	14	13	13	12	11	13	13

Çizelge 4.6’da  $F5$  özniteliğinin çıkarıldığı durum için başarımların sonuçları yer almaktadır.

**Çizelge 4.6.** İterasyon 1 için  $F5$  özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımların sonuçları

2.GK nöron adedi 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	14	13	15	13	17	17	14	15	16	15	17
8	20	18	18	17	14	16	16	14	13	13	14
9	15	17	15	15	13	17	15	13	18	17	12
10	15	14	15	12	16	15	16	16	14	16	17
11	16	15	16	16	13	14	17	14	14	15	17
12	17	12	14	15	16	13	14	16	14	13	16
13	13	14	16	13	17	15	19	18	14	16	16
14	17	17	15	14	16	16	14	14	14	16	13
15	15	17	15	17	15	17	18	16	15	16	18
16	16	16	16	15	14	16	17	16	14	13	14
17	14	14	20	18	13	17	15	17	15	17	17

Çizelge 4.7’de  $F6$  özneliğinin çıkarıldığı durum için başarımları yer almaktadır.

**Çizelge 4.7.** İterasyon 1 için  $F6$  özneliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları

2.GK nöron adedi 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	17	19	14	17	16	15	19	16	17	20	18
8	16	17	17	18	15	15	17	20	17	17	18
9	13	16	18	14	16	17	19	20	20	16	17
10	17	19	17	19	20	16	19	18	18	17	18
11	16	18	15	16	19	20	17	16	16	19	19
12	14	15	16	18	18	18	15	16	19	18	17
13	20	17	17	17	18	18	18	19	17	13	18
14	17	13	15	16	18	20	15	20	18	20	18
15	16	18	14	20	20	20	20	17	17	20	19
16	16	17	16	16	18	20	20	17	14	17	17
17	19	18	18	16	18	17	19	17	17	17	15

Çizelge 4.8’de  $F7$  özneliğinin çıkarıldığı durum için başarımları yer almaktadır.

**Çizelge 4.8.** İterasyon 1 için  $F7$  özneliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları

2.GK nöron adedi 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	14	12	16	13	13	17	15	16	14	12	10
8	14	16	15	18	14	12	15	15	14	15	12
9	14	17	14	14	15	15	16	18	15	15	14
10	14	12	12	14	15	15	14	11	15	17	15
11	13	17	11	14	14	15	16	14	12	13	13
12	14	13	14	16	13	12	15	17	14	18	13
13	19	16	15	14	15	17	17	11	17	14	15
14	16	14	12	15	17	13	16	16	18	14	16
15	15	17	14	15	17	14	15	13	15	16	16
16	14	17	14	15	17	14	15	17	15	14	11
17	17	15	15	15	15	16	15	14	13	16	15

Çizelge 4.2, Çizelge 4.8 ve aralarında kalan çizelgeler incelendiğinde Çizelge 4.7’de 17 farklı 1. ve 2. katman nöron sayılarıyla oluşturulan yapay sinir ağları için 25 örneğin 20 adedinin doğru olarak bulunduğu görülmektedir. Bu diğer çizelgelerdeki doğru bulunan örnek sayısından fazladır. Bundan dolayı 1. iterasyonda elde edilen maksimum başarıım  $F6$  özniteliği çıkarıldığında  $\frac{20}{25} = \% 80$  olarak ölçülebilmektedir.

İterasyon 2: İterasyon 2, tüm özniteliklerden  $F6$  özniteliğinin çıkarıldığı durum için İterasyon 1’in tekrar edilmesini kapsamaktadır. Bu kapsamda, her defasında 6 adet özniteliğin birini dışarıda bırakarak, 7-17 gizli katman nöron adetleri arasında yapay sinir ağı eğitimlerinden oluşmaktadır. Çizelge 4.9’da  $F1$  özniteliğinin dışarıda bırakılması ile 25 adet örneğin her birinin sırasıyla dışarıda bırakılması ve toplamda dışarıda bırakılan örneklerin ne kadarının doğru tespit edildiğine dair sonuçlar yer almaktadır.

**Çizelge 4.9.** İterasyon 2 için  $F1$  özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarıım sonuçları

2.GK nöron adedi \ 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	17	15	14	15	16	20	18	16	19	13	15
8	14	17	18	14	20	15	18	18	14	16	16
9	15	14	19	17	15	15	15	20	17	20	17
10	17	17	16	15	16	17	17	16	18	16	15
11	14	19	14	18	17	19	19	19	18	20	16
12	14	14	13	16	15	17	18	17	17	15	16
13	17	18	22	19	16	12	17	16	16	19	16
14	17	19	15	16	16	14	15	16	17	18	17
15	14	17	16	18	19	14	15	17	18	15	15
16	17	19	19	19	19	17	15	15	16	18	17
17	18	16	16	14	18	18	18	19	19	20	19

Çizelge 4.10'da  $F2$  özneliğinin çıkarıldığı durum için başarımlar sonuçları yer almaktadır.

**Çizelge 4.10.** İterasyon 2 için  $F2$  özneliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımlar sonuçları

2.GK nöron adedi \ 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	17	17	18	16	18	17	16	14	16	18	17
8	17	18	14	17	19	16	17	17	19	18	19
9	18	16	14	19	17	14	20	19	18	18	18
10	20	18	17	17	18	18	16	16	18	18	14
11	18	16	19	15	17	16	14	16	21	14	17
12	16	19	20	15	18	14	17	21	18	18	20
13	20	17	16	17	17	18	21	15	17	15	18
14	16	17	16	19	15	16	19	16	15	15	18
15	19	18	17	16	13	15	17	19	18	14	15
16	19	19	17	18	19	20	18	18	16	16	17
17	16	15	15	18	15	15	16	16	18	15	17

Çizelge 4.11'de  $F3$  özneliğinin çıkarıldığı durum için başarımlar sonuçları yer almaktadır.

**Çizelge 4.11.** İterasyon 2 için  $F3$  özneliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımlar sonuçları

2.GK nöron adedi \ 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	19	20	18	17	16	20	18	18	17	15	17
8	16	17	15	17	15	20	19	19	17	14	16
9	18	19	18	20	21	20	18	18	21	20	20
10	17	17	18	21	17	19	17	18	17	18	18
11	14	19	19	18	19	17	18	17	19	21	19
12	17	15	18	17	20	19	18	20	17	20	18
13	18	17	17	17	19	15	19	17	17	20	18
14	20	14	21	19	16	17	20	21	19	20	15
15	15	16	18	19	19	19	19	18	16	21	19
16	19	16	21	19	18	18	21	20	20	20	18
17	21	21	18	17	18	17	16	19	20	18	18



Çizelge 4.12’de  $F4$  özneliliğinin çıkarıldığı durum için başarımlar sonuçları yer almaktadır.

**Çizelge 4.12.** İterasyon 2 için  $F4$  özneliliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımlar sonuçları

2.GK nöron adedi \ 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	15	14	14	15	12	16	14	16	15	18	15
8	13	13	13	15	15	16	15	13	18	18	16
9	15	16	15	14	17	14	17	19	14	15	13
10	14	17	16	18	15	15	11	15	17	16	15
11	14	14	17	18	18	17	17	19	17	14	15
12	18	16	16	13	16	15	15	16	12	14	17
13	18	13	13	15	15	14	15	17	14	15	15
14	15	15	15	15	16	15	15	14	16	17	12
15	15	15	16	17	17	15	14	16	16	16	14
16	17	12	18	17	16	17	15	17	16	15	15
17	12	16	16	15	16	15	14	16	16	14	16

Çizelge 4.13’de  $F5$  özneliliğinin çıkarıldığı durum için başarımlar sonuçları yer almaktadır.

**Çizelge 4.13.** İterasyon 2 için  $F5$  özneliliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımlar sonuçları

2.GK nöron adedi \ 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	16	14	15	17	20	18	18	17	18	16	17
8	15	19	20	16	16	17	17	15	17	17	15
9	15	18	18	20	15	17	15	14	17	19	19
10	15	19	18	18	19	21	16	18	21	15	16
11	15	18	15	15	19	15	16	18	16	17	17
12	16	15	13	17	20	18	20	18	20	20	17
13	17	19	20	19	20	16	14	18	19	16	16
14	20	18	17	21	15	14	13	18	18	17	14
15	19	19	21	17	14	17	17	17	14	16	18
16	18	16	19	15	18	19	16	16	18	18	19
17	17	14	16	17	20	14	17	19	20	13	18

Çizelge 4.14’de  $F7$  özniteliğinin çıkarıldığı durum için başarımların sonuçları yer almaktadır.

**Çizelge 4.14.** İterasyon 2 için  $F7$  özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımlarının sonuçları

2.GK nöron adedi \ 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	21	14	17	12	15	16	16	16	14	12	15
8	16	17	16	16	16	15	12	14	14	12	12
9	14	15	13	15	18	15	16	13	16	14	16
10	17	15	16	15	17	12	16	14	10	13	13
11	13	14	14	14	13	14	16	15	13	13	13
12	6	13	13	14	16	15	14	16	14	13	20
13	15	16	15	15	15	16	13	16	16	16	15
14	13	16	12	15	18	13	15	13	15	15	16
15	13	15	16	15	15	15	17	15	14	15	19
16	15	16	13	14	14	18	18	13	15	14	17
17	17	18	16	15	14	13	14	16	16	16	15

Çizelge 4.9, Çizelge 4.14 ve aralarında kalan çizelgeler incelendiğinde Çizelge 4.11’de 11 farklı 1. ve 2. katman nöron sayılarıyla oluşturulan yapay sinir ağları için 25 örneğin 21 adedinin doğru olarak bulunduğu görülmektedir. Bu diğer çizelgelerdeki doğru bulunan örnek sayısından fazladır. Bundan dolayı 2. iterasyonda elde edilen maksimum başarımların  $F3$  özniteliği çıkarıldığında  $\frac{21}{25} = \% 84$  olarak ölçülebilmektedir.

İterasyon 3: İterasyon 3, tüm özniteliklerden  $F6$  ve  $F3$  özniteliğinin çıkarıldığı durum için İterasyon 1’in tekrar edilmesini kapsamaktadır. Bu kapsamda, her defasında 5 adet özniteliğin birini dışarıda bırakarak, 7-17 gizli katman nöron adetleri arasında yapay sinir ağı eğitimlerinden oluşmaktadır. Çizelge 4.15’de  $F1$  özniteliğinin dışarıda bırakılması ile 25 adet örneğin her birinin sırasıyla dışarıda bırakılması ve toplamda dışarıda bırakılan örneklerin ne kadarının doğru tespit edildiğine dair sonuçlar yer almaktadır.

**Çizelge 4.15.** İterasyon 3 için  $F1$  özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları sonuçları

2.GK nöron adedi 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	16	18	16	18	19	22	18	18	19	18	16
8	22	20	18	22	21	16	17	20	19	21	19
9	20	16	19	19	19	18	19	19	19	19	19
10	16	18	20	19	17	20	16	18	20	20	18
11	20	18	18	20	18	17	22	23	21	18	17
12	22	21	19	22	20	20	18	19	20	17	18
13	19	20	19	22	20	18	20	17	19	19	19
14	21	19	20	20	20	20	21	20	21	20	19
15	18	16	15	18	22	20	19	15	22	20	21
16	20	19	19	22	19	21	18	20	21	24	22
17	20	20	18	20	19	18	18	21	19	19	20

Çizelge 4.16’da  $F2$  özniteliğinin çıkarıldığı durum için başarımları sonuçları yer almaktadır.

**Çizelge 4.16.** İterasyon 3 için  $F2$  özniteliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımları sonuçları

2.GK nöron adedi 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	21	17	19	21	18	20	16	17	20	17	20
8	13	19	18	20	16	19	15	19	18	15	20
9	18	18	16	23	17	16	17	14	20	19	18
10	18	19	20	19	17	19	18	18	18	19	16
11	18	20	19	21	19	17	19	22	15	21	23
12	18	20	20	16	18	17	19	17	18	20	19
13	19	18	17	19	18	21	18	18	19	18	19
14	19	22	20	18	18	17	20	16	19	18	18
15	17	19	17	18	19	19	21	18	19	19	21
16	18	21	18	21	18	19	18	21	18	21	19
17	21	18	18	15	18	20	18	19	20	19	20

Çizelge 4.17’de  $F4$  özneliğinin çıkarıldığı durum için başarımlar sonuçları yer almaktadır.

**Çizelge 4.17.** İterasyon 3 için  $F4$  özneliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımlar sonuçları

2.GK nöron adedi \ 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	14	17	18	16	17	19	17	18	15	20	18
8	16	15	15	16	18	17	14	17	14	16	16
9	18	17	16	19	16	19	18	21	16	19	15
10	17	17	17	18	17	15	16	15	18	17	14
11	13	18	17	16	18	18	17	18	16	16	16
12	17	16	15	16	17	16	17	16	15	18	19
13	15	16	16	18	17	16	17	21	17	17	15
14	19	15	11	16	15	16	16	14	18	19	19
15	14	14	19	18	19	16	16	17	16	18	16
16	15	17	18	15	15	17	14	18	14	15	15
17	14	18	18	20	14	16	16	16	17	15	15

Çizelge 4.18’de  $F5$  özneliğinin çıkarıldığı durum için başarımlar sonuçları yer almaktadır.

**Çizelge 4.18.** İterasyon 3 için  $F5$  özneliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımlar sonuçları

2.GK nöron adedi \ 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	15	15	18	22	19	22	18	20	17	18	17
8	19	20	17	18	18	19	16	20	18	19	19
9	18	17	20	19	18	19	17	18	23	20	17
10	15	19	17	16	17	21	17	19	20	17	20
11	20	17	19	15	17	18	18	17	18	17	19
12	16	15	20	20	19	19	19	19	18	21	20
13	18	20	20	18	17	16	17	20	19	20	22
14	20	18	17	19	20	16	20	20	19	20	18
15	17	16	18	16	17	20	20	17	18	22	18
16	20	19	18	18	19	21	20	19	18	18	22
17	20	19	21	21	17	20	18	18	17	19	21

Çizelge 4.19’da  $F7$  özneliğinin çıkarıldığı durum için başarımlar sonuçları yer almaktadır.

**Çizelge 4.19.** İterasyon 3 için  $F7$  özneliğinin dışarıda bırakılması ile elde edilen yapay sinir ağı başarımlar sonuçları

2.GK nöron adedi \ 1.GK nöron adedi	7	8	9	10	11	12	13	14	15	16	17
7	16	15	14	17	15	15	13	17	15	14	12
8	13	20	15	13	16	15	17	13	14	12	15
9	14	16	13	14	11	13	12	13	15	15	14
10	16	18	13	12	15	15	18	14	16	14	13
11	16	13	15	16	15	12	16	18	15	16	14
12	13	17	13	13	11	13	16	14	14	16	13
13	14	16	17	14	16	18	16	15	17	17	16
14	17	16	14	16	16	14	14	16	14	12	12
15	14	16	16	15	15	18	16	13	16	12	16
16	14	17	12	17	14	14	16	16	17	13	14
17	13	16	15	15	14	16	15	13	15	17	17

Çizelge 4.15, Çizelge 4.19 ve aralarında kalan çizelgeler incelendiğinde Çizelge 4.15’de 11 farklı 1. ve 2. katman nöron sayılarıyla oluşturulan yapay sinir ağları için 25 örneğin 22 adedinin doğru olarak bulunduğu görülmüştür. Ayrıca 1. Gizli katman nöron sayısı 11 ve 2. Gizli katman nöron sayısı 14 olduğu durum için 23 örneğin doğru olarak bulunduğu, 1. ve 2. Gizli katman nöron sayısının 16 olduğu durum için 24 örneğin doğru olarak bulunduğu görülmüştür. Bundan dolayı 3. iterasyonda elde edilen maksimum başarımlar  $F1$  özneliği çıkarıldığında  $\frac{24}{25} = \% 96$  olarak ölçülebilmektedir.

Bir sonraki iterasyon için bulunan doğru örnek sayısında bir artış gözlemlenmemiştir. Böylece yapay sinir ağının sadece 4 adet öznelikle daha yüksek başarımlar sağlanacak şekilde temsil edilebildiği görülmüştür. Bu öznelikler  $F2, F4, F5, F7$  öznelikleridir.

Çizelge 4.20, maksimum başarımlar durumu için deney 1 hata matrisini ifade etmektedir.

**Çizelge 4.20.** Deney 1 hata matrisi

	Yorgun	Dinç
Yorgun	9	0
Dinç	1	15

Çizelge 4.20'deki hata matrisine göre Deney 1 için TP(True-Pozitive) değeri  $\frac{9}{9} = 1$ , FN(False-Negative) değeri  $\frac{0}{9} = 0$ , FP(False-Pozitive) değeri  $\frac{1}{16} = 0,06$ , TN(True-Negative) değeri  $\frac{15}{16} = 0,93$  olarak ölçülmüştür. TP değeri kişinin yorgun olduğu durumda yorgun tespit edilen örnek oranını, FN değeri kişinin yorgun olduğu durumda dinç tespit edilen örnek oranını, FP değeri kişinin dinç olduğu durumda yorgun tespit edilen örnek oranını, TN değeri kişinin dinç olduğu durumda dinç tespit edilen örnek oranını vermektedir.

#### 4.2.2. Deney 2

Deney 2, yazar üzerinden alınan 21 adet dinç ve 20 adet yorgun toplam 41 adet video örneği üzerinde yapılan yapay sinir ağı değerlendirme sonuçlarını kapsamaktadır. Deney 1'de modellenmiş ve iyileştirilmiş olan yapay sinir ağı kullanılarak *Birini dışarıda bırak* yöntemiyle ve her defasında birer adet örneğin dışarıda bırakılması ile gerçekleştirilen iterasyonlar sonunda 41 örneğin 19'unun doğru olarak tespit edildiği görülmüştür. Bu sonuç toplam 6 kişi üzerinden alınan ve 66 adet videodan oluşan veri kümesinin problem çözümünde genelleştirme için yeterli olmadığı anlamını taşımaktadır. Çözümün genelleştirilebilmesi için daha fazla denek ve örnek kullanılması gereklidir.

Çizelge 4.21, deney 2 hata matrisini göstermektedir.

**Çizelge 4.21.** Deney 2 hata matrisi

	Yorgun	Dinç
Yorgun	2	18
Dinç	4	17

Çizelge 4.21'deki hata matrisine göre Deney 1 için TP(True-Pozitive) değeri  $\frac{2}{20} = 0,1$ , FN(False-Negative) değeri  $\frac{18}{20} = 0,9$ , FP(False-Pozitive) değeri  $\frac{4}{21} = 0,19$ , TN(True-Negative) değeri  $\frac{17}{21} = 0,81$  olarak ölçülmüştür.



## 5. TARTIŞMA ve SONUÇ

Bu tezde olası trafik kazalarını önlemek için davranışsal bir uykulu / yorgun sürücü tespit sistemi önerilmiştir. Deneklerin göz kırpmaları üzerinden çıkarılan istatistiksel öznitelikler kullanılarak kişinin yorgunluk derecesi bir yapay sinir ağı ile modellenmiştir. Önerilen sistemin performansı 5 kişi ve 25 adet video örneği üzerinden %96 olarak ölçülmüştür. Bu oran literatürdeki benzer çalışmalarla da uyumludur (Vural 2009) (Girit 2014) (Golgiyaz 2013). Sistem, bu 5 kişiden farklı olarak yazar üzerinden alınan örneklerle değerlendirildiğinde %46 oranında bir başarıyı sergilemektedir. Bu sonuç; toplam 5 kişiden alınan 25'er adet örnek üzerinden eğitilen yapay sinir ağının, problem çözümünün geliştirilmesi için yeterli olmadığı sonucunu taşımaktadır. Denek sayısı ve üzerlerinden alınan örnekler artırılarak sistem, geliştirme için uygun hale getirilebilir.

Sistem, bu şekilde sadece bir firmanın kritik işlerde çalışan belirli bir çalışan grubuna uygulanabilir. Böylece o çalışan grubu için eğitilen sistemin, çalışanların yorgun olması durumunda firmaya ya da kişiye uyarı vermesi sağlanabilir. Örneğin havaalanlarındaki kule görevlileri veya güvenlik çalışanları böyle bir sistemle çalışma esnasında kontrol edilebilir.

Çalışanların gözlüklü olması, ten renklerinin göz açıklık kapalılığının bulunmasında kullanılan eğitim veri kümesinin içerdiğinden çok farklı olması ya da sistemin ölçüm yaptığı ortamda yeterli aydınlatmanın olmaması göz açıklık kapalılığının tespit edilmesini zorlaştırabilir. Bu durum, göz açıklık kapalılık üzerinden çıkarılan özniteliklerin yanlış ölçülmesine sebep olabilir ve yapay sinir ağının yanlış karar vermesine yol açabilir. Bu durumların önüne geçmek için; göz açıklık kapalılığının bulunması amacıyla eğitilen Haar kaskad sınıflandırıcının örnek veri kümesi, farklı ten renkleri ve gözlük takılması durumları da eklenerek artırılabilir.

Sistemin performansı, sistemin örnek veri kümesine bir gece görüş kamerasından alınan görüntülerin eklenmesi ile artırılabilir. Yapay sinir ağı, eklenen bu gece görüş kamerası görüntülerinden elde edilen öznitelikler ile eğitilerek gün ışığı olmadan da değerlendirme yapabilecek duruma getirilebilir. Ayrıca kişinin yorgunluk değerlendirmesi yapılırken



günün saati, nabız durumu, esneme durumu gibi özellikler eklenerek hibrit bir sistemle performans artışı gözlemlenebilir.



## KAYNAKLAR

- Abtahi, S. 2012.** Driver Drowsiness Monitoring Based On Yawning Detection. *Master thesis*, University of Ottawa, Electrical and Computer Engineering, Canada.
- Akrout, B., Mahdi, W. 2013.** A blinking Measurement Method for Driver Drowsiness Detection: Advanced in Intelligent Systems and Computing, Springer International Publishing, doi: 10.1007/978-3-319-00969-8\_64.
- Anonim, 2012.** KLT Tracking. University of Central Florida, <http://crcv.ucf.edu/courses/CAP5415/Fall2012/Lecture-10-KLT.pdf>-(Erişim tarihi: 18.02.2017).
- Anonim, 2013.** ZJU eyeblink database. Computer science and technology of Zhejiang University, [www.cs.zju.edu.cn/~gpan/database/db\\_blink.html](http://www.cs.zju.edu.cn/~gpan/database/db_blink.html)-(Erişim tarihi: 2.7.2017)
- Anonim, 2016a.** Acute sleep deprivation and risk of motor vehicle crash involvement. AA Foundation for Traffic Safety, <https://www.aaafoundation.org/acutesleep-deprivation-and-crash-risk>-(Erişim tarihi: 14.02.2017).
- Anonim, 2016b.** Global status report on road safety 2015. WHO, [http://www.who.int/gho/road\\_safety/en](http://www.who.int/gho/road_safety/en)-(Erişim tarihi: 14.02.2017).
- Anonim, 2016c.** Karayolu trafik kaza istatistikleri 2015. Türkiye İstatistik Kurumu, <http://www.tuik.gov.tr/PreHaberBultenleri.do?id=21611>-(Erişim tarihi: 14.02.2017).
- Anonim, 2017.** Computer Vision Lecture 23. Carnegie Mellon The Robotics Institute, <http://www.cs.cmu.edu/~16385/lectures/Lecture23.pdf>-(Erişim tarihi: 14.02.2017).
- Başer, E., Altun, Y. 2016.** Detection and classification of vehicles in traffic by using haar cascade classifier. Proceedings of 58th ISERD International Conference, 23 -24 December 2016, Prague, Czech Republic.
- Bergasa, L. M., Nuevo, J., Sotelo, M. A., Barea, R., Lopez, M. E. 2006.** Real-Time System for Monitoring Driver Vigilance. *IEEE transactions on intelligent transportation systems*, 7: 63-77.
- Castrillón, M., Dénis, O., Guerra, C., Hernández, M. 2007.** ENCARA2: Real-time detection of multiple faces at different resolutions in video streams. *In Journal of Visual Communication and Image Representation*, 130-140.
- Colic, A., Marques, O., Furht, B. 2014.** Driver Drowsiness Detection Systems and Solutions. SpringerBriefs in Computer Science, ISSN: 2191-5768.

- Demirkır, C., Sankur, B. 2006.** Object Detection Using Haar Feature Selection Optimization. IEEE 14th Signal Processing and Communications Applications, DOI: 10.1109/SIU.2006.1659787, 17-19 April 2006
- Girit, A. 2014.** Drowsy Driver Detection Using Image Processing, *Yüksek Lisans Tezi*, ODTÜ, Fen bilimleri enstitüsü, Ankara.
- Golgiyaz, S. 2013.** Gerçek Zamanlı Uykulu Sürüş Algılama Sistemi, *Yüksek Lisans Tezi*, Fırat Üniversitesi, Fen bilimleri enstitüsü, Elazığ.
- Golgiyaz, S., Kocamaz, A. F., Okumuş, F. 2014.** Video Tabanlı Uykulu Sürücü Algılama Sistemi, Eleco 2014 Elektrik – Elektronik – Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu, 27-29 Kasım 2014, Bursa.
- Grauman, K., 2008.** Object Detection using Haar-like Features. Department of Computer Science University of Texas at Austin, [http://www.cs.utexas.edu/~grauman/courses/spring2008/slides/Faces\\_demo.pdf](http://www.cs.utexas.edu/~grauman/courses/spring2008/slides/Faces_demo.pdf) (Erişim tarihi 18.02.2017).
- Hagan, M. T., Menhaj, M. 1994.** Training Feedforward Networks with the Marquardt Algorithm. *IEEE Transactions on Neural Networks*, 5(6):989-993
- Hayırlı, S. 2005.** Uykulu Sürücü Belirleme Sistemi, *Yüksek Lisans Tezi*, Hacettepe Üniversitesi, Fen bilimleri enstitüsü, Ankara.
- Huang, C., Ai, H., Li, Y., Lao, S. 2007.** High-performance rotation invariant multiview face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*(TPAMI), April 2007.
- Levenberg, K. A. 1944.** A Method for the Solution of Certain Non-Linear Problems in Least Squares, *Quarterly of Applied Mathematics*, 2:164-168.
- Li, S. Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H., Shum, H. 2002.** Statistical learning of multi-view face detection. European Conference on Computer Vision(ECCV), 28-31 May 2002, Copenhagen, Denmark.
- Lienhart, R., Kuranov, A., Pisarevsky, V. 2003.** Empirical analysis of detection cascades of boosted classifiers for rapid object detection. 25th DAGM-Symposium, 10-12 September 2003, Magdeburg, Germany.
- Lienhart, R., Maydt, J. 2002.** An Extended Set of Haar-like Features for Rapid Object Detection. International Conference on Image Processing, Santa Clara, USA.

- Lucas, B. D., Kanade, T. 1981.** An Iterative Image Registration Technique with an Application to Stereo Vision. *International Joint Conference on Artificial Intelligence*, 674-679.
- Marquardt, D. W. 1963.** An Algorithm for Least-Squares Estimation of Nonlinear Parameters, *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431-441.
- Murukesh, C., Abhan, P. P. 2015.** Drowsiness Detection for Drivers Using Computer Vision 2015. *WSEAS transactions on Information Science and Applications*, E-ISSN: 2224-3402, Volume 12.
- Nasrollahi, K., Moeslund, T. B., Rashidi, M. 2013.** Haar-like Rectangular Features for Biometric Recognition. In *International Conference on Biometrics IEEE*, 04-07 June 2013, Madrid, Spain.
- Öztemel, E. 2012.** Yapay sinir ağları. Papatya yayıncılık, Türkiye, 44 s.
- Pham, M. T., Cham, T. J. 2007a.** Fast training and selection of Haar features using statistics in boosting-based face detection. *IEEE International Conference on Computer Vision (ICCV)*, 14-21 October 2007, Rio de Janeiro, Brazil.
- Pham, M. T., Cham, T. J. 2007b.** Online learning asymmetric boosted classifiers for object detection. *Computer Vision and Pattern Recognition IEEE Conference on 2007 (CVPR)*, 17-22 June, 2007, Minneapolis, USA.
- Quesada, A. 2016.** Memory and speed comparison, 5 algorithms to train a neural network.  
[https://www.neuraldesigner.com/blog/5\\_algorithms\\_to\\_train\\_a\\_neural\\_network](https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network)-(Erişim tarihi: 07.07.2017).
- Rezaei, M., Klette, R. 2011.** 3D Cascade of Classifiers for Open and Closed Eye Detection in Driver Distraction monitoring. *14th International Conference on Computer Analysis of Images and Patterns (CAIP)*, 29-31 August 2011, Seville, Spain.
- Sahayadhas, A., Sundaraj, K., Murugappan, M. 2012.** Detecting Driver Drowsiness Based on Sensors: A Review. *Sensors*, 16937-16953; doi:10.3390/s121216937.
- Shi, J., Tomasi, C. 1994.** Good Features to Track. *9th IEEE Conference on Computer Vision and Pattern Recognition*, DOI: 10.1109/CVPR.1994.323794.
- Shiqi, Y. 2016.** Eye Detection, <http://yushiqi.cn/research/eyedetection>-(Erişim tarihi: 15.04.2016)

- Shuyan, H., Gangtie, Z. 2009.** Driver drowsiness detection with eyelid related parameters by Support Vector Machine. *Expert Systems with Applications*, 36, 7651–7658, doi:10.1016/j.eswa.2008.09.030.
- Stergiou, C., Siganos, D. 2017.** Introduction to Neural Networks. Imperial College, department of computing, [https://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/cs11/report.html](https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html)-(Erişim tarihi: 10.01.2017)
- Szeliski, 2010.** Computer Vision: Algorithms and Applications. Springer, 957 pp.
- Tomasi, C., Kanade, T. 1991.** Detection and Tracking of Point Features. Carnegie Mellon University Technical Report, CMU-CS-91-132.
- Viola, P., Jones, M. J. 2001.** Rapid Object Detection using a Boosted Cascade of Simple Features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, Hawaii, USA.
- Vural, E. 2009.** Video Based Detection of Driver Fatigue, *Doktora tezi*, Sabancı Üniversitesi, Mühendislik ve Doğa Bilimleri, İstanbul.
- Wu, B., Ai, H., Huang, C., Lao, S. 2004.** Fast rotation invariant multi-view face detection based on real adaboost. 6th IEEE International Conference on Automatic Face and Gesture Recognition (FGR), 17-19 May 2004, Seoul, South Korea.
- Yalçın, H. 2017.** Yapay sinir ağları. İTÜ, [http://web.itu.edu.tr/hulyayalcin/Signal\\_Processing\\_Books/neural-net-giris.pdf](http://web.itu.edu.tr/hulyayalcin/Signal_Processing_Books/neural-net-giris.pdf)-(Erişim tarihi: 11.01.2017).
- Yu, H., Wilamowski, B. M. 2011.** Levenberg-Marquardt training: Industrial Electronics Handbook, Ed.: CRC Press, pp: 1-15
- Zang, S., Bauckhage, C., Cremers, A. B. 2014.** Informed Haar-like Features Improve Pedestrian Detection. Computer Vision and Pattern Recognition (CVPR), 23-28 June 2014, Ohio, USA.

## ÖZGEÇMİŞ

Adı Soyadı : Soner Karagülmez  
Doğum Yeri ve Tarihi : Kdz Ereğli, 1988  
Yabancı Dili : İngilizce

### Eğitim Durumu (Kurum ve Yıl)

Lise : Kdz. Ereğli Anadolu Lisesi, 2006  
Lisans : Uludağ Üniversitesi, 2010

Çalıştığı Kurum/Kurumlar ve Yıl : FİGES, 2012-2015; ERDEMİR 2015- ...  
İletişim (e-posta) : sonerkaragulmez1@gmail.com